TOPOLOGY AND PRINTING ORIENTATION OPTIMIZATION OF ORTHOTROPIC MATERIAL FOR ADDITIVE MANUFACTURING

AHMED ELSAYED MOTER

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

GRADUATE PROGRAM IN MECHANICAL ENGINEERING
YORK UNIVERSITY
TORONTO, ONTARIO

AUGUST 2021

Abstract

Mechanical properties of 3d printed polymers have witnessed remarkable improvements as a result of optimization in topology and raster orientation. This study proposes and evaluates a strategy to deal with stiffness maximization of structures in elastic region subjected to global volume and compliance constraints. This work also presents a unique approach to dealing with local constraints, specifically stresses along and perpendicular to the fibers of the printed structures (principal material axes). A modified SIMP (modified solid isotropic material with penalization) method is used with the method of moving asymptotes (MMA) as the optimizer. Each element has fictitious density and angle as the main design variables. To reduce the number of stress constraints and thus the computational cost, a new clustering strategy is employed in which the highest stresses in principal material coordinates are grouped separately into two clusters using an adjusted P-norm. A detailed description of the formulation and sensitivity analysis are discussed. 2D structures are then analyzed in the numerical examples section. The method can also be used for 3D structures, as the formulation derivation is general. Results show that this method is effective for producing manufacturable 3D printed structures that avoid stress concentrations.

Acknowledgements

I would like to express my endless gratitude to my supervisor, Prof. Aleksander Czekanski. This work would not have been possible without his invaluable and continuous support. I would also like to express my infinite gratitude to my lab colleague Mohamed Abdulhamid, who made Canada feel like home when I first arrived. Mohamed offered his academic support to me throughout my master's journey, making the process much easier for me. I would also like to thank my supervisory committee chair Prof. Zhu and my external examiner Prof. Pantazopoulou for their time, and willingness to help.

I am grateful to my lovely family for their help and support during my career, from kindergarten all the way to grad school. I will be thankful to them for the rest of my life. I am grateful to all my IDEA-lab colleagues for the fun we had together here in Canada, especially Ahmed, Daphene, Carlos, Minha, and Tejas. My eternal thanks go to my lovely cousin Mervat Motier and her husband, Hael, for their continuous support while I have been here in Canada and for making it feel like home. Not to forget my Egyptian friends, Mohamed Ayman, Yosif, Omar Elsayed, Ismail Elkomy, Ali Al-alawadi, Mohamed Mousa, Ahmed Zaher, Ahmed Safwat, Marah, Maryam Abid, Eva, Angelina, and Nehal for being there for me whenever I needed them.

Table of Contents

Abstract ii

Acknowledgements iii

Table of Contents iv

List of Tables vii

List of Figures viii

Notations x

Acronyms xiii

1 INTRODUCTION 1

1.1 Background 1

1.2 Justification of the study 3

1.3 Research objectives 4

1.4 Research approach 5

1.5 Thesis outline 6

2 LITERATURE REVIEW 7

2.1 Additive manufacturing 7

2.2 Topology optimization 9
2.2.1 Topology optimization methods 10
2.3 Topology optimization of orthotropic materials 15
2.4 Optimization methods 17
2.5 Stress-constrained problems 19

3 NUMERICAL IMPLEMENTATION 21
3.1 Geometry definition 21
3.2 Finite element model 22
3.2.1 Material model and penalization 27
3.2.2 Sensitivity analysis 28
3.3 Optimization 30
3.4 Filtering 32
3.5 Ansys verification model 34
3.6 Conclusions 35

4 COMPLIANCE AND MASS MINIMIZATION 36
4.1 Problem formulation 36
4.2 Gradient calculation 38
4.3 Numerical examples 40
4.3.1 Verification of results 40
4.3.2 Case studies 44
4.4 Conclusions 54

5 DIRECTION-DEPENDANT STRESS CONSTRAINTS 55
5.1 Problem formulation 55
5.2 Stress measure 57
5.2.1 Failure criteria 57

5.2.2 Stress measure 58
5.3 Objective function and gradients calculation 61
5.4 Modelling limitations 65
5.5 Numerical examples 67
5.5.1 Case study 1: Verification of the method 67
5.5.2 Case study 2: Effect of number of stress evaluation points on final topology 70
5.5.3 Case study 3: Adding one more stress cluster in each direction . 73
5.5.4 Case study 4: Effect of P-values on the stress results 75
5.6 Conclusions 79

6 CONCLUSIONS AND FUTURE WORK 80
6.1 Statement of the problem 80
6.2 Objectives 81
6.3 General conclusions 82
6.3.1 Adding the fiber angle as a design variable to the TO problem . 82
6.3.2 Mass savings and stiffness increase improvements 82
6.3.3 Direction-dependant stress constraints 82
6.3.4 General comments on code performance 83
6.4 Thesis main contributions 83
6.5 Future work 84

Bibliography 85

Appendix A MATLAB Code for Stress Constrained TO of Orthotropic Material 93

List of Tables

4.1 Mechanical properties of ABS and EG 40

5.1 Stress clustering using stress level technique. 60

5.2 Stress clustering using stress averaging technique. 61

5.3 Comparing different stress clustering techniques. 74

5.4 Effect of P-value on the efficiency of the results of stress-constrained problem with $X = 60$ kPa and $Y = 25$ kPa. 75

List of Figures

1.1 Comparison of anisotropy introduced in 3D printed parts by different AM techniques [8]. 3

2.1 Structural optimization categories [14] 9

3.1 Anti-clockwise numbering of the quad element. 23
3.2 Effect of penalization value on the discreteness of the solution. 28
3.3 Effect of filtering results of *L*-bracket structure using a sensitivity filter. 34

4.1 Cantilever beam fully fixed from the left and showing the design space dimensions and load. 41
4.2 Compliance comparison from a study done on Ansys with 20 degrees increment and on our algorithm 42
4.3 Structures generated from our code and from Ansys at the optimal angle 44
4.4 Case study 1, compliance minimization of L-bracket structure with volume fraction = 0.25 45
4.5 Graphs of case study 1 46
4.6 Case study 2. Mass minimization of MBB beam with compliance limit = 0.1082 Nm 48
4.7 Graphs of case study 2 49

4.8 Cantilever beam half fixed from the left and showing the design space dimensions and load. 51
4.9 Case study 3. Compliance minimization of half fixed rectangular structure with volume fraction = 0.3. 52
4.10 Case study 3. Mass minimization of half fixed rectangular structure with maximum compliance = 0.1082. 53

5.1 Difference between adding a load at one point and distributing the load. 66
5.2 Cantilever beam fully fixed from the left and a distributed load on six consecutive nodes in the bottom-right corner. 67
5.3 Optimized topology and raster angle. 69
5.4 Stresses along the fibers 69
5.5 Stresses perpendicular to the fibers 69
5.6 Final topology and raster angle. 71
5.7 Stresses along the fibers 71
5.8 Stresses perpendicular to the fibers 72
5.9 Convergence scheme of objective function 72
5.10 Results of L-bracket stress-constrained topology optimization when using two stress clusters in each of the material coordinate directions (final topology, fibers orientation, and convergence scheme) 74
5.11 Results of L-bracket stress-constrained topology optimization when using two stress clusters in each of the material coordinate directions (stress values) 75
5.12 Effect of P-value on topology and raster angle. 76
5.13 Effect of P-value on stresses along the fibers 77
5.14 Effect of P-value on stresses perpendicular to the fibers 78

Notations

$\bar{C}$	Augmented compliance function
a	Billinear energy function
Ω_{max}	Cluster domain that holds the maximum stress values
C	Compliance
D	Constitutive matrix
u	Displacements parallel to x axis
v	Displacements parallel to y axis
E_{ijkl}	Elastic stiffness tensor
A	Element area
ρ	Element density
k_e	Element stiffness matrix
θ	Fiber angle
$\tilde{x}_e$	Filtered density
$r_{\min}$	Filter radius
$\mathbf{F}$	Force vector
ψ	General objective function
σ^{PN}	Global stress measure
λ	Lagrange multiplier

$\phi(x)$	Level set function
$\mathbf{L}$	Linear load function
N_{is}	Linear shape functions
M	Mass
x^C	Maximum allowed compressive stress along fibers
Y^C	Maximum allowed compressive stress perpendicular to fibers
X^T	Maximum allowed tensile stress along fibers
Y^T	Maximum allowed tensile stress perpendicular to fibers
μ	Material density
Ω^{mat}	Material domain
τ_{23}	Maximum allowed shear stress
τ_{23}	Maximum allowed shear stress
σ_a	Penalized stress evaluation point tensor
v_{12}	Poisson ratio
Ω	Reference domain
G_{12}	Shear modulus

K	Stiffness matrix
$\mathbf{B}$	Strain–displacement matrix
T_2	Strain transformation matrix from x-y to principal material axes
$\hat{\sigma}_a$	Stress evaluation point tensor
T_1	Stress transformation matrix from x-y to principal material axes
$\mathbf{E}'$	Transformed constitutive matrix
h	Thickness of the finite element
$(\sigma_1^C)_{ult}$	Ultimate compressive stress along fibers
$(\sigma_2^C)_{ult}$	Ultimate compressive stress perpendicular to fibers
$(\sigma_2^T)_{ult}$	Ultimate tensile stress perpendicular to fibers
$(\sigma_1^T)_{ult}$	Ultimate tensile stress along fibers
V_f	Volume fraction
H_{ei}	Weight factor
E_1	Young's modulus along first principal axis
E_2	Young's modulus along second principal axis

Acronyms

2D	Two dimensional
3D	Three dimensional
ABS	Acrylonitrile butadiene styrene
AM	Additive manufacturing
CAD	Computer aided design
EG	Epoxy glass
ESO	Evolutionary structural optimization
FEA	Finite element analysis
FFF	Fused filament fabrication
MBB	Messerschmitt–Bölkow–Blohm
MMA	Method of moving asymptotes
SIMP	Solid isotropic material with penalization
SLP	Sequential linear programming
SQP	Sequential quadratic programming
TO	Topology optimization

Chapter 1

INTRODUCTION

Summary: In this chapter, we define the research question, identify the research objectives, and justify the approach used to reach the intended objectives. Finally, we provide a summary of the layout of the thesis.

1.1 Background

Topology optimization (TO) is a computational optimization tool that is used to determine the optimum material distribution in a predefined design domain. This method has led to the production of numerous high-performing and innovative designs, attracting the attention of applied mathematicians and designers. In 1960, Schmidt [1] announced that there was the potential to use finite element analysis together with TO methods in structural design. Later, in 1988, Bendsoe [2] published the first work on TO. According to Google Scholar, researchers have since published more than 34,000 articles in this field, demonstrating just how effective TO has been in structural optimization in the automotive, aerospace, machinery, bioengineering, and additive manufacturing (AM) sectors.

AM is the process of manufacturing a broad array of three-dimensional (3D) model data structures and complex geometries. The process involves the printing of successive layers of materials on top of one another. Charles Hull developed this technology in 1986 through a stereolithography (SLA) process, which was followed by further developments, such as the fusion of powder beds, fused filament fabrication (FFF), inkjet printing, and the production of contours (CC). 3D printers have evolved significantly over the last few years. They are now capable of transforming manufacturing and logistical procedures with a variety of techniques, materials, and equipment.

AM is widely used in various industries, including machining, prototyping, and biomechanical engineering. The use of 3D printing has been particularly slow and limited in the construction industry, despite its many advantages, such as reduced waste, freedom of design, and automation [3]. New applications are developing continuously as new materials and AM methods are being developed. The expiry of earlier patents is one of the main factors driving the increased accessibility of this technology, which is now enabling manufacturers the opportunity to develop new 3D printers. In recent years, 3D printers have been cutting costs and extending their applications in colleges, homes, libraries, and laboratories. At first, in view of their fast and cost-efficient prototyping capability, 3D printing was extensively used by designers to create efficient and appealing prototypes. 3D printing has allowed business owners to design user-customized products in small quantities, as printing depends only on the CAD model and does not require a specific mold each time a product is altered. A notable example of 3D printed products is the Winsun house built in China. These completely 3D printed houses cost only $4,800 USD per unit [4]. In addition, in the medical field, 3D printing has been used in the production of implants based on CT-imaged tissue replicas [5].

1.2 Justification of the study

FFF uses a continuous filament of a thermoplastic polymer to 3D print layers of a material. The filament is heated to a semi-liquid state at the nozzle before being extruded on the platform or on top of previously printed layers. The thermoplasticity of the polymer filament is critical for this method because it allows the filaments to fuse together during printing and then solidify at room temperature after printing. The main processing parameters that affect the mechanical properties of printed parts are layer thickness, filament width and orientation, and air gap. One of the major industrial problems currently facing manufacturers [6, 7] is raster angle, as it is the main reason why anisotropy is introduced into a material. Figure 1.1 [8] shows how a major difference in tensile and compressive strength can result from FFF printing.

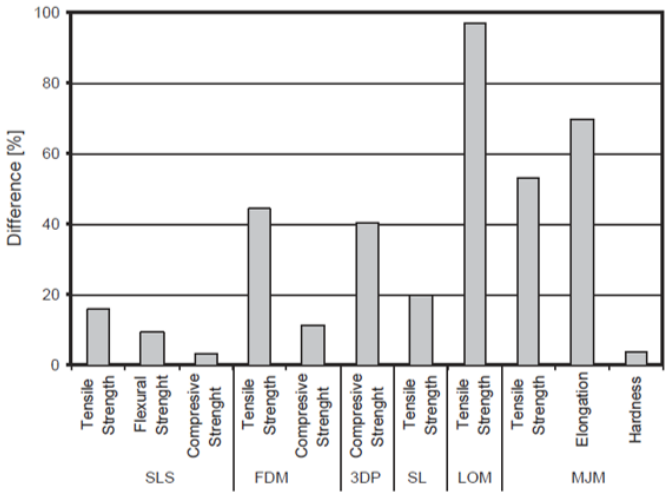


Figure 1.1: Comparison of anisotropy introduced in 3D printed parts by different AM techniques [8].

In addition to FFF, other 3d printing processes such as stereo-lithography, or composites with fibres such as fibre glass can also generate anisotropic components that need raster angle determination. Furthermore, TO is an important step that must be completed prior to 3D printing the parts in order to maximise the material savings associated with mass manufacturing [9, 10]. However, each raster angle configuration will have a distinctive topology. Each TO design will yield a unique fibre orientation [9, 10, 11, 12]. Moreover, none of the currently available commercial tools can solve these challenges. Hence, my research question can now be defined as follows: How can the structural topology and raster angle optimization problems be coupled into a single problem that is subject to local and global constraints?

1.3 Research objectives

This research aims to code the topology and raster angle optimization problems for different objective functions that are subject to different local and global constraints. The specific objectives of this work are to:

1. Derive the isoparametric Q4 element constitutive matrix, with the raster angle **theta** defined as a design variable.

2. Develop a robust and generic in-house code that solves the compliance minimization problem, subject to volume constraints, where the produced topologies are a good start in the conceptual design process for AM.

3. Solve mass minimization problems, subject to compliance constraints.

4. Derive of stress constraints for AM polymers, both along the fibers and perpendicular to the fibers (bonding strength).

5. Solve compliance minimization problems, subject to stress constraints, to reduce the formation of stress concentration regions in the final topology.

1.4 Research approach

The aforementioned objectives are achieved by developing an in-house code from scratch. The first step in the numerical procedure is devoted to geometry importing. The software is able to process any 2D design domain drawn on any CAD tool. The geometry will be drawn then imported to Gmsh, an external open-source meshing software, via Matlab to calculate the MSH file. This file contains the coordinates of each node and the connectivity matrix. The second step is defining the problem constants as the material model, loads, boundary conditions, objective function, design variables, and constraints. The third step is to solve for the displacements of each node in the continuum using the method found in [13] for isoparametric elements; this step entails a sensitivity analysis in which the optimization gradients are calculated using the direct method found in [14]. The fourth step is the optimization step, whereby the method of moving asymptotes [15] is used, as it has proven to be effective in handling structural problems [16]. The optimization process is an iterative process where convergence criteria are set on the change on the design variables between two consecutive iterations so that the change does not exceed a certain predefined limit. Each iteration has its own design variables and a constitutive matrix, which is a factor of the initial one depending on the penalization value. The main challenge is to add the stress constraint to the problem, which will be done by using the maximum stress failure criteria [17]. Then, the stress clustering method found in [18] will be used to calculate the stresses in the continuum for each iteration.

1.5 Thesis outline

The rest of this thesis is structured as follows: **Chapter 2** discusses the relevant literature on topology optimization, additive manufacturing, and optimization methods, as well as stress-constrained problems and methods. The methods and numerical procedure structure are presented in **Chapter 3**, including the finite element model, the optimization method, the filtering method, and the Ansys model. **Chapter 4** discusses the compliance and mass minimization problem, starting with the problem formulation, finite element equations, and gradients and closing with a discussion of verification and case studies. Stress-constrained problems are discussed in **Chapter 5**, it begins with problem formulation, then defines the finite element equations and gradients, and concludes with results and applications. Finally, **Chapter 6** presents the conclusions, identifies the original contributions, and suggests future work.

Chapter 2

LITERATURE REVIEW

Summary: This chapter is devoted to discussing relevant literature. It is divided into 5 sections: (i) "Additive manufacturing", presents the new updates in the field of AM and its usefulness to the manufacturing industry. (ii) "Topology optimization", demonstrates TO optimization techniques used in literature. (iii) "Topology and orientation optimization", discusses the work done to modify the TO methods to be used in optimizing both the topology and fiber angles. (iv)"Optimization methods", provides brief summary of optimization methods that can be used in structural optimization and the usefulness of each method. (v)"Stress-constrained problems", provides summary of what has been done in the field of stress constrained TO.

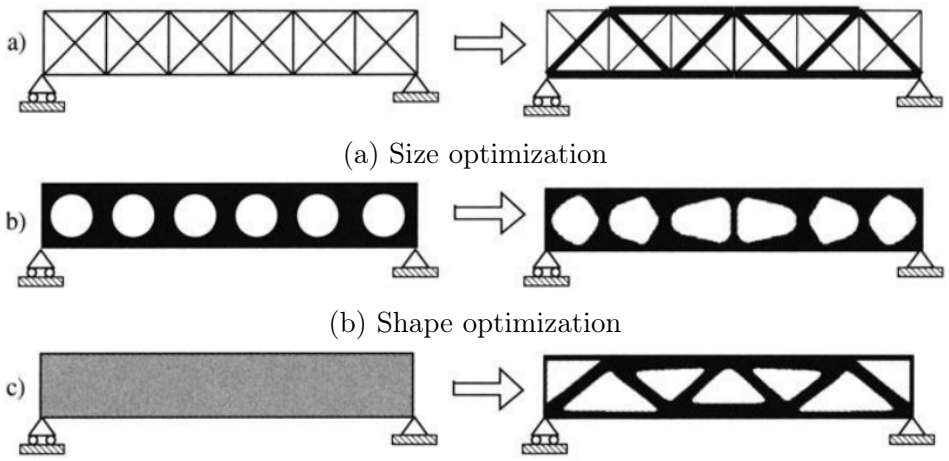
2.1 Additive manufacturing

AM creates structures by adding a material layer by layer on the 3D printer bed to form a component with the desired geometry In contrast to traditional (subtractive) manufacturing methods, such as machining or CNC, AM requires less raw material and minimizes building time and tooling costs. Furthermore, AM can be used to create parts with highly complex geometry that would be difficult to create using traditional

manufacturing techniques. The added sophistication and versatility of AM technology has piqued the interest of many industries. In 1987, the first industrial AM machine was produced. It was a stereolithography machine that used a laser to solidify liquid polymer and shape the part [19]. Consequently, several other methods were commercialized after that. Over the last two decades, the demand for AM technologies has expanded considerably. According to the 2016 Wohlers Report [20], the AM market surpassed 5.1$ billion USD in 2015, a 1$ billion USD rise for the second year in a row. According to this report, more and more businesses are embracing AM technology: in 2015, 62 vendors sold industrial-grade AM systems, up from 49 in 2014, which was more than double the number in 2011. The value of 3D printed parts is expected to increase at a 15 percent compound annual growth rate (CAGR), from $12 billion in 2020 to $51 billion in 2030 [21] . Government institutions such as the National Science Foundation and the U.S. Department of Defense have contributed greatly to AM research; aerospace and automobile companies have also invested extensively in AM to boost the efficiency of the components produced. The latest trend demonstrates that the AM technology of the future will continue to expand. The present research employs fused filament fabrication (FFF), a popular AM technique. It is expected that the carbon fiber filled (CFF) polymer feedstock will drive FFF technology forward in the industrial sector [9, 22]. As FFF produces structures in the form of melted fibers, the orientation of these fibers plays a major role in the mechanical characteristics of the final product. Optimizing the print orientation provides a rare ability to increase the rigidity of the drawn geometry, and can also aid TO to conserve mass. In this work, raster angle is added as a design variable in the TO problems of mass, compliance, and stress constraints.

2.2 Topology optimization

Structural optimization is accomplished using three conventional techniques: size optimization, shape optimization, and topology optimization (TO) [23]. Size optimization focuses on optimizing the dimensions of the structure, that is, thickness and the cross-sectional area of the elements. Shape optimization focuses on optimizing the shape contours according to the partial differential equation defining the domain without changing the "connectivity" of the shape. Finally, TO finds the optimal distribution of material in a predefined design domain.



(a) Size optimization

(b) Shape optimization

(c) Topology optimization

Figure 2.1: Structural optimization categories [14]

TO was first introduced in 1988 by Bendsoe [2] using the homogenization method. TO is privileged in having the broadest applicability of the three optimization techniques, as the user chooses the optimal distribution of material that fits in a certain design domain, subject to certain boundary conditions. It was initially developed to be an efficient substitute for the conventional iterative approaches used in industry to help produce preliminary designs. Numerous studies ([24, 25, 26, 27]) have shown that TO

produces novel and efficient designs.

2.2.1 Topology optimization methods

This section addresses the main TO methods used to solve the discretized TO formulation. These are homogenization, solid isotropic material with penalization (SIMP; the method used here), level sets, and evolutionary structural optimization (ESO). The development and drawbacks of each method are discussed in turn.

Homogenization

Briefly, the homogenization approach helps to calculate the efficacious properties of a substance with an endless number of small gaps, so that the optimal distribution is calculated within a given domain. For example, if a composite material with a unit square cell is to be distributed in a domain Ω that belongs to $\mathbb{R}^2$, then the homogenization equation will be written in the following form:

$$E_{ijkl} = E_{ijkl}(\alpha, \theta) \tag{2.1}$$

where $\boldsymbol{\alpha}$ is the pointwise unit cell hole size, and $\boldsymbol{\theta}$ is the pointwise varying angle of rotation of the cell. Material density is defined as $\boldsymbol{\mu} = (1 - \boldsymbol{\alpha})$ For the compliance minimization problem subject to a volume constraint, the homogenization optimization problem is written as

$$\begin{aligned} &\underset{\boldsymbol{\mu},\boldsymbol{\theta}}{\text{minimize}} \quad L(\boldsymbol{u}) \\ &\text{so that } a(\boldsymbol{u}, \boldsymbol{v}) = L(\boldsymbol{v}) \text{ for all } \boldsymbol{v} \in U \\ &\int_{\Omega} \boldsymbol{\mu} \mathrm{d}x \leqslant \mathrm{Vol}, \quad 0 \leqslant \mu \leqslant 1 \end{aligned} \tag{2.2}$$

where a is the bilinear energy function; $\boldsymbol{L}$ is the linear load function; $\boldsymbol{u}$ and $\boldsymbol{v}$ are the displacements in the x and y directions, respectively; and $\boldsymbol{U}$ is the space of displacement fields [2].

In observing the mathematical equations of the homogenization method, there appears to be a sizing problem; however, it is actually more of a shape optimization method, and that is what gave rise to TO. This method was famous for 1 year before the density approaches were developed and used as a substitute, as they were easier to implement and produced more valid results [28]. Later on, in 2010, the homogenization method showed its ability to handle composite structure problems by using the concurrent approach, which couples micro- and macro-structural optimization [10].

Solid isotropic material with penalization (SIMP)

Soon after the homogenization approach was introduced, Bendsoe [29] and others [30, 31] suggested the SIMP, or power law, approach. It was first designed to reduce the complexity of the homogenization approach and improve the convergence approach; this was a simple but artificial approach. In 1999, [32] introduced a physical rationale for the method. The relationship between the density design vector and the material property is determined by the power law as follows:

$$E\left(\rho_i\right) = g\left(\rho_i\right) E_0 = \rho_i^p E_0, \quad g\left(\rho_i\right) = \rho_i^p \tag{2.3}$$

where p is the penalization parameter and E_0 is Young's modulus of the solid material. These equations show that the main role of penalization is to help generate discrete answers for the density values of each element. For the SIMP method, the compliance

minimization problem subject to volume constraints is written as

$$\begin{aligned} \min_{\rho,\theta} \quad & C(\rho) = U(\rho)^T F \\ \text{s.t.} \quad & \tfrac{v(\rho)}{V_0} = f \\ & K(\rho)U(\rho) = F \\ & \rho_{\min} \leq \rho_e \leq \rho_{\max} \end{aligned} \tag{2.4}$$

where $x \in R^n$ is a vector representing the density of each element, n is the number of elements, v_i is the volume of material in an element, $\boldsymbol{U}$ is the displacement vector, $\boldsymbol{F}$ is the load vector, and f is the volume fraction of the whole design domain determined by the user.

A new penalization formulation was introduced in 2001 by [33]. Referred to as the rational approximation of material properties (RAMP), it has the form

$$E\left(\rho_i\right) = \frac{\rho_i}{1 + q\left(1 - \rho_i\right)} E_0, \quad g\left(\rho_i\right) = \frac{\rho_i}{1 + q\left(1 - \rho_i\right)}, \tag{2.5}$$

where q is the penalization parameter. RAMP led to a nonconcavity alleviation in the original SIMP system of interpolation [33] and thereby to a convergence to 0–1 solutions. However, realistically, this aspect does not appear to play a major role [28]. The key distinction between SIMP and RAMP is that SIMP is nonzero for $\rho_i = 0$, which can affect the convergence characteristics and reduce problems with false low-density modes in time-variant problems [34].

The correct choice of penalization factors p or q is uniquely determined and relies heavily on the physical problem solved. The SIMP and RAMP criteria must be selected to ensure appropriate deterrence for the minimization of compliance. The compliance

minimization problem is convex for $p = 1$. Thus, [28] recommended using $p = 1$ and gradually increasing it until better designs are produced. However, the efficacy of this ongoing strategy cannot generally be assured.

Finally, Allaire et al. [35, 35] introduced an explicit penalization approach instead of the implicit one mentioned earlier. The method involved multiplying the objective function by

$$\alpha \int_{\Omega} \rho(1-\rho) dV \tag{2.6}$$

or adding it as a constraint, where α is a weighting factor. However, the method was not widely used because it turned out to be difficult to determine a weighting factor or a constraint that makes the solution converge.

Evolutionary structural optimization (ESO)

Introduced by Xie et al. [36], the ESO technique is straightforward. The model domain is first broken down into elements. Second, finite element analysis (FEA) is used to measure the von Mises stress distribution in the architectural domain, as von Mises stress is most often used to estimate the stress distribution of isotropic domains. A stress threshold is determined by multiplying a user-defined rejection ratio (RR) by the design domain's maximal stress, and any variable with stress less than the threshold value is removed from the design domain. Using the same RR, the FEA and element removal are repeated until the steady state is reached. After that, an evolutionary ratio (ER) is applied to the RR, which is used to maximize the value of the RR, and the previous analysis and element exclusion procedure is replicated before the final optimal criterion is achieved. A common convergence criterion is that all stress values within the design domain are within 25% of the maximum value. Despite its simplicity, the technique has been proved to be reliable and has been used to solve a variety of structural issues.

One disadvantage of the ESO approach is that it only removes elements during iteration. It is conceivable that after certain elements are removed, their adjacent elements are subjected to high pressure, requiring the previously discarded elements to reappear, which the ESO forbids.

The bidirectional evolutionary structural optimization (BESO) approach, suggested by Querin et al. [37], is an evolution of the ESO method. The BESO algorithm searches in both directions, allowing not only for the removal of low-stress material but also for the addition of additional material to the high-stress field. This approach conducts a more rigorous search in the architectural environment, making it more possible to prevent the use of a local minimum solution. The ESO and BESO approaches are straightforward to execute, but stress-based schemes ignore the content property's directionality. As a result, these methods should not be used for the latest optimization of AM orthotropic materials [9].

Level set

The level set approach was first introduced by Osher et al. [38]. In this method, the design boundary is determined through the level set function $\phi(x)$, and the domain is defined as follows:

$$\rho = \begin{cases} 0 : \forall \mathbf{x} \in \Omega : \phi < 0 \\ 1 : \forall \mathbf{x} \in \Omega : \phi \geq 0 \end{cases} \tag{2.7}$$

Level set solution update methods are numerous, with the most prominent solution being the Hamilton–Jacobi equation [28]. The update scheme was

$$\begin{aligned} &\tfrac{\partial \phi}{\partial t} = -V\mathbf{n}\cdot\nabla\phi \\ &\text{ or with } \mathbf{n} = (\nabla\phi)/(|\nabla\phi|) : \\ &\tfrac{\partial \phi}{\partial t} + V|\nabla\phi| = 0 \end{aligned} \tag{2.8}$$

where t is a pseudo-time for the design evolution and V is a speed function. The speed function V (x) at x : $\phi(x) = 0$ represents the sensitivity of moving the interface in the normal direction $\mathbf{n}$ with respect to a merit function, scaled by the spatial gradient of the level set function.

Updating the level set function via the Hamilton–Jacobi equation does not allow for the formation of new holes since the level set function evolves only along interfaces that exist in the initial design. In addition, to avoid rapidly oscillatory level set field and interface geometries, additional regularization techniques must be implemented. Since the Hamilton–Jacobi equation (2.8) has a strictly convective basic form, stabilization schemes must be used to solve it numerically [28].

2.3 Topology optimization of orthotropic materials

Apart from the aforementioned TO methods used in 2.2.1, the optimization of problems involving the fiber angles as a design resulted in the evolution of modified versions of these methods and even completely new approaches. These included concurrent TO using homogenization [10], the lamination parameters method [39], and the modified SIMP approach [9]. It is noteworthy that no work used the level sets or ESO approach to solve these kind of problems for the reasons mentioned in 2.2.1. This section briefly

discusses the methodology of each of the proposed methods, and explains which of the methods will be used in this work and why.

The homogenization process was recently reused since it shows its strength in the analysis of isotropic material properties, as it is capable of handling problems with multi-scale optimization. This ability is shown in [40], where optimization of micro-structures is analyzed and coded in Matlab. Furthermore, Gao et al. [10] were able to create a simultaneous approach that combines problems related to macro- and micro-structure optimization. However, it is not reasonable to use this approach when evaluating fiber angle orientation, since micro-structures have the same topology at the end of the solution and have no sign of fiber angle orientation. Densities of the micro- and macro-structures represent the design variables of this process, and the solution algorithm is so complicated that it is very difficult to add the fiber angle as a new design variable. The lamination parameters approach has been used to tackle multi-layered composites by Peeters et al. [39]; however, it did not include adding the fiber angle as a new design variable. In addition, Peeters et al. mentioned that the solution process is suspected of major simplifications and assumptions, leading to the results being nonmanufacturable. Finally, the modified SIMP approach was introduced by Hai et al. [11]. Their contribution was mainly in adding the $\boldsymbol{\theta}$ of each element as a design variable representing the fiber angle in this area to the formulation 2.4. It therefore becomes

$$\begin{aligned} &\text{Minimize: } c(\rho,\theta) = \mathrm{U}(\rho,\theta)^{\mathrm{T}}\mathrm{F} \\ &\text{Subject to : } \begin{cases} \frac{v(p)}{V_0} = f \\ \mathbf{K}(\rho,\theta)\mathbf{U}(\rho,\theta) = \mathbf{F} \\ \rho_{\min} \leq \rho_e \leq 1 \\ -2\pi \leq \theta_{\mathrm{e}} \leq 2\pi \end{cases} \end{aligned} \tag{2.9}$$

This approach will be used here, as its efficiency has been demonstrated in the works of [9, 12, 41], especially when using the method of moving asymptotes (MMA) [42] as the optimizer. The work here will not only focus on mass- and compliance-constrained problems but will also cover the stress-constrained problems, which is considered a state-of-the-art contribution in the field of optimization of orthotropic material.

2.4 Optimization methods

Choosing the optimization method is a crucial step in solving structural optimization problems, as the method chosen determines whether or not feasible solutions will be reached and the time required to find these solutions. Optimization methods are categorized into non-gradient-based (NGB) and gradient-based (GB) methods. In [43], Sigmund argued that NGB methods are inefficient in solving problems involving as large a number of design variables as that used in TO. His main arguments were as follows:
(1) Although NGB uses global search algorithms, it rarely converges to globally optimal solutions except for some exceedingly coarse discretizations that do not show clear representation of the geometry physics.
(2) NGB needs to make 10^{299} function evaluations to solve a problem involving 1000 elements, which will obviously take a long time to solve. This research involves two design variables per element, so it will take 10^{299} function evaluations for only 500 elements.

Hence, those techniques may be useful if supercomputers are used to solve TO problems. On the other hand, GB methods are more efficient for solving structural optimization problems and have been used extensively in the literature [2, 9, 11, 23, 44]. Well-known GB methods include sequential linear programming (SLP), sequential quadratic programming (SQP), optimality criteria, and MMA. SLP [45, 46] solves optimization

problems with nonlinear objective functions and constraints, by linearizing both of them around an initial guess point using a Taylor series expansion. After that, the problem can be solved using linear methods such as the simplex method. The obtained solution is then used as the linearization point of the next iteration and the process repeats. The central idea of SQP is that it models the problem at the current iteration by a quadratic subproblem and uses the solution of this subproblem to calculate the next iteration solution. It can be considered the application of Newton's method to the Karush–Kuhn–Tucker conditions [47]. The main issue with this method is finding the Hessian approximation of the Lagrangian function, as numerous approximations exist and their effectiveness differs depending on the problem. SLP and SQP work well overall for nonlinear problems, but MMA works very well especially for structural optimization and converges faster than SLP and SQP algorithms [48].

The optimality criteria method [29] is used in problems involving one constraint and has proved to be effective in [49], where only the volume fraction constraint was imposed. However, in this work, stress constraints will be imposed in addition to constraints on the volume fraction. Another problem for the optimality criteria method is made apparent by looking at its formulation:

$$
x_e^{\text{new}} = \begin{cases} \max\left(x_{\min}, x_e - m\right) \\ \quad \text{if} \quad x_e B_e^{\eta} \leq \max\left(x_{\min}, x_e - m\right), \\ x_e B_e^{\eta} \\ \quad \text{if } \max\left(x_{\min}, x_e - m\right) < x_e B_e^{\eta} < \min\left(1, x_e + m\right), \\ \min\left(1, x_e + m\right) \\ \quad \text{if } \min\left(1, x_e + m\right) \leq x_e B_e^{\eta}, \end{cases} \tag{2.10}
$$

$$B_e = \frac{-\frac{\partial c}{\partial x_e}}{\lambda \frac{\partial V}{\partial x_e}} \tag{2.11}$$

where m (move) is a positive move-limit, η is a numerical damping coefficient, and λ is a Lagrangian multiplier. The term B_e will always be undefined if fiber angles are the design variables of the problem because the derivative of the fiber angles $\boldsymbol{\theta}$ with respect to the volume fraction is always zero.

Eventually, MMA became one of the most widely used TO algorithms [12, 50, 51, 52]. MMA generates subproblems that approximate the original problem. The subproblems are chosen to be distinct and convex, ensuring that they are always optimal and simple to solve [42]. The work here mainly depends on using MMA, as its efficiency in solving TO problems has been demonstrated [53]. The method will be discussed in detail in the methods section.

2.5 Stress-constrained problems

Stress-constrained problems are a famous type of problem in the literature [18, 54, 55, 56, 57]. However, the researchers who have investigated stress-constrained problems have discussed them from the point of view of isotropic materials. In this work, light will be shed on a stress-constrained material with anisotropic properties.

Material failure theories predict when material failure will occur. The von Mises yield criterion, for example, is often used to estimate metal yielding in isotropic material. It is also critical to consider these failure conditions during optimization in order to achieve configured designs that do not fail under the applied loads. Composite failure methods vary at the fiber/matrix level. Fiber fracture, fiber buckling, fiber fracturing, matrix cracking, and radial cracks are all failure modes at the fiber level. Laminate defects

include transverse cracks in planes parallel to or perpendicular to the threads, as well as delamination between the laminate layers. Delamination can happen due to failure of bonding between the fibers. Hence, stress limits perpendicular to the fibers can be set as the bonding strength between fibers. Failure theories for composites include maximum stress, maximum strain, Tsai–Hill, and tensor polynomial failure criterion [17, 58].

Stress-constrained TO problems confront two main problems: computational efficiency and the singularity problem. To begin with, using GB optimization, TO with several local (stress) constraints becomes computationally very costly. TO has traditionally been used to solve problems with a few constraints compared to the large number of design variables, for instance, compliance minimization subject to a volume constraint, a problem easily solved using an adjoint method [29]. In the case of stress constraints, however, one does not know a priori in which region the stress is critical, and stress constraints are extended at any point in the architectural domain. As a result, the number of constraints is of the same order as the number of implementation variables. Singularity occurs when the globally optimal solution is a degenerate subspace of dimension less than the original feasible space of the problem. For example, when solving a truss problem, some members' areas become zeros, despite the fact that these members are essential and cannot be removed. Similarly, when dealing with a continuum, an element with a low density can still have a strain, resulting in a high stress value in this area when it should be zero [59, 60, 61, 62, 63].

These challenges are related to FEA. Hence, it will occur whether the material is isotropic or anisotropic. Several of the methods introduced to address these challenges will be discussed in more detail.

Chapter 3

NUMERICAL IMPLEMENTATION

Summary: This chapter discusses the main pillars of the code that will be used to solve the optimization problems in the following chapters.

3.1 Geometry definition

Geometry importing is the first step in the analysis. Hence, it should be imported in a smooth and correct manner for the rest of the analysis to be correct. For the sake of robustness, the code can accept any STEP file imported from an external CAD tool.

After that, the geometry is imported into an external meshing software called Gmsh via Matlab. The MSH file that is extracted from Matlab contains the node coordinates and connectivity matrix. Gmsh was chosen after searching for meshing methods on Matlab because it is more convenient to use an external mesher rather than writing a specific code for meshing the objects, which would be an extremely hard task to code. Matlab has a PDE toolbox that can be used for meshing objects. Nonetheless, that toolbox restricts the mesh to tetrahedrons as it reads STL files ("Standard Triangular Language"). Meanwhile, I am using Q4 elements. According to [64], Gmsh is a 3D finite

element grid generator with a build-in CAD engine and post-processor. Its design goal is to provide a fast, light, and user-friendly meshing tool with parametric input and advanced visualization capabilities. It is built around four modules: geometry, mesh, solver, and post-processing. Each module can be controlled either interactively using the GUI or the scripting language. Thus, Gmsh is the most appropriate software for this work because it is:

1. Fast: On a standard computer, Gmsh can run instantaneously. Its mesh time is also lower than the average mesh time compared to the standards of the finite element community. For instance, in 2008, Gmsh was able to generate about 1 million tetrahedrons in less than a minute and could visualize such a mesh together with associated post-processing datasets at interactive speeds.

2. Memory-efficient: The memory footprint of the application is minimal, and the source code is small enough that a single developer can handle it. Installing or running the software does not depend on any non-widely available third-party software package.

3. User-friendly: The GUI is designed in such a way that a new user can learn how to create simple meshes in a matter of minutes. In addition, as an open-source code it is robust, portable, scriptable, extensible, and thoroughly documented.

3.2 Finite element model

TO is a finite element based method. Hence, before starting optimization, the process displacements of each node in the mesh should be calculated from the global state equation:

$$\mathbf{KU} = \mathbf{F} \tag{3.1}$$

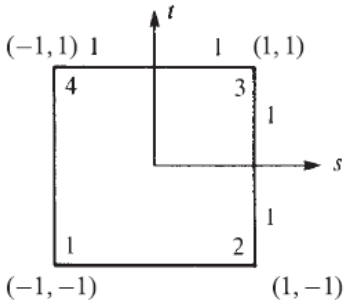


Figure 3.1: Anti-clockwise numbering of the quad element.

The force vector **F** is predefined; however, the stiffness matrix **K** needs to be derived for the isoparametric (Q4) element. The first step in the derivation is to define the local element coordinates (s–t) and its relation with the global coordinates (x–y). Figure 3.1 shows the local coordinates and the anti-clockwise node numbering of a general quad element. Global coordinates can be related to the natural coordinates by the following equations [13]:

$$\begin{aligned} x &= a_1 + a_2 s + a_3 t + a_4 st \\ y &= a_5 + a_6 s + a_7 t + a_8 st \end{aligned} \tag{3.2}$$

Solving for a_i, we get:

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} \tag{3.3}$$

where N_{is} are linear shape functions that describe how the element will move when it is subjected to an external force. u & v are the displacements parallel to the x and y coordinates, respectively.

$$\begin{aligned} N_1 &= \tfrac{(1-s)(1-t)}{4} \quad N_2 = \tfrac{(1+s)(1-t)}{4} \\ N_3 &= \tfrac{(1+s)(1+t)}{4} \quad N_4 = \tfrac{(1-s)(1+t)}{4} \end{aligned} \tag{3.4}$$

Strains can be related to displacements through the **B** matrix, which is defined as follows:

$$[B(s,t)] = \frac{1}{|[J]|}\begin{bmatrix}[B_1] & [B_2] & [B_3] & [B_4]\end{bmatrix} \tag{3.5}$$

$$[B_i] = \begin{bmatrix} a\left(N_{i,s}\right) - b\left(N_{i,t}\right) & 0 \\ 0 & c\left(N_{i,t}\right) - d\left(N_{i,s}\right) \\ c\left(N_{i,t}\right) - d\left(N_{i,s}\right) & a\left(N_{i,s}\right) - b\left(N_{i,t}\right) \end{bmatrix} \tag{3.6}$$

where $i = 1, 2, 3, 4$ and other constants are defined as follows:

$$\begin{aligned} a &= \tfrac{1}{4}\left[y_1(s-1) + y_2(-1-s) + y_3(1+s) + y_4(1-s)\right] \\ b &= \tfrac{1}{4}\left[y_1(t-1) + y_2(1-t) + y_3(1+t) + y_4(-1-t)\right] \\ c &= \tfrac{1}{4}\left[x_1(t-1) + x_2(1-t) + x_3(1+t) + x_4(-1-t)\right] \\ d &= \tfrac{1}{4}\left[x_1(s-1) + x_2(-1-s) + x_3(1+s) + x_4(1-s)\right] \end{aligned} \tag{3.7}$$

$$|[J]| = \frac{1}{8}\left\{X_c\right\}^T \begin{bmatrix} 0 & 1-t & t-s & s-1 \\ t-1 & 0 & s+1 & -s-1 \\ s-t & -s-1 & 0 & t+1 \\ 1-s & s+1 & -t-1 & 0 \end{bmatrix} \left\{Y_c\right\} \tag{3.8}$$

$$\left\{X_c\right\}^T = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix} \tag{3.9}$$

$$\{Y_c\} = \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{Bmatrix} \tag{3.10}$$

Finally, the stiffness matrix can be defined as

$$[k] = \iint_A [B]^T [D][B] h dx dy \tag{3.11}$$

where h is a constant thickness usually set to unity. The integrations can be solved using the Gaussian points method, where $dxdy$ is the elemental area [13, 65]. After integration, the stiffness matrix of a 2D finite element with unity thickness can be written as follows [65]:

$$\mathbf{k} = \mathbf{B}^T \mathbf{D} \mathbf{B} \tag{3.12}$$

where $\mathbf{B}$ is the strain–displacement matrix, and $\mathbf{D}$ is the constitutive matrix in the elastic region defined as follows for an orthotropic material:

$$\mathbf{D} = \begin{bmatrix} \frac{E_1}{1-\nu_{12}} & \frac{E_2\,\nu_{12}}{1-\nu_{12}^2} & 0 \\ \frac{E_2\,\nu_{12}}{1-\nu_{12}^2} & \frac{E_2}{1-\nu_{12}} & 0 \\ 0 & 0 & G_{12} \end{bmatrix} \tag{3.13}$$

The print direction angle is added as a design variable as follows: [17]:

$$\boldsymbol{\sigma} = \mathbf{D}' \boldsymbol{\epsilon}, \tag{3.14}$$

$$\mathbf{D}'(\theta_e) = \mathbf{T_1}(\theta_e)^{-1} \mathbf{D} \mathbf{T_2}(\theta_e), \tag{3.15}$$

$$\mathbf{T_1} = \begin{bmatrix} c^2 & s^2 & 2cs \\ s^2 & c^2 & -2cs \\ -cs & cs & c^2 - s^2 \end{bmatrix}, \tag{3.16}$$

$$\mathbf{T_2} = \begin{bmatrix} c^2 & s^2 & cs \\ s^2 & c^2 & -cs \\ -2cs & 2cs & c^2 - s^2 \end{bmatrix} \tag{3.17}$$

where c and s represent $cos(\theta)$ and $sin(\theta)$, respectively. According to the classical laminate theory:

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \mathbf{T_1}^{-1}\mathbf{D}\mathbf{T_2} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{bmatrix}, \tag{3.18}$$

$$\mathbf{k_e}(\theta_e) = \mathbf{B}^T\mathbf{D}'(\theta_e)\mathbf{B} \tag{3.19}$$

After calculating the stiffness matrix, the global state equation written in (3.1) can be redefined as

$$\mathbf{K}(\boldsymbol{\rho}, \boldsymbol{\theta})\mathbf{U}(\boldsymbol{\rho}, \boldsymbol{\theta}) = \mathbf{F} \tag{3.20}$$

Where $\mathbf{K}(\boldsymbol{\rho}, \boldsymbol{\theta})$ is the global penalized stiffness matrix consisting of contributions from the stiffness matrix of each element:

$$\mathbf{K}(\boldsymbol{\rho}, \boldsymbol{\theta}) = \sum_{\mathbf{e}=1}^{\mathrm{N}} (\rho_{\mathrm{e}})^p \, \mathbf{k_e}\,(\theta_{\mathbf{e}}) \tag{3.21}$$

The new state equation is used to calculate the state variables of each element $\mathbf{U}(\boldsymbol{\rho}, \boldsymbol{\theta})$ which will be used in the objective functions of each problem from the problems defined in the following chapters.

3.2.1 Material model and penalization

For the SIMP approach, the material model is defined as

$$E_{ijkl}(\rho) = 1_{\Omega_{mat}} E^0_{ijkl}, \quad 1_{\Omega_{mat}} = \begin{cases} 1 & if \in \Omega_{mat} \\ 0 & if \in \Omega/\Omega_{mat} \end{cases} \tag{3.22}$$

$$\int 1_{\Omega_{mat}} d\Omega = Vol(\Omega^{mat}) \leq V \tag{3.23}$$

where E_{ijkl} is the elastic stiffness tensor, Ω^{mat} is the material domain, Ω is the reference domain, and V is the limit volume (volume fraction). This problem has integer variables. The common way to solve it in the literature is to convert the integer variables to continuous variables and then penalize the solution to ensure the discreteness of the solution domain [14]. These continuous variables are represented in the form of a function that is interpreted as the fictitious density of the material, which is used to scale Young's modulus tensor to the stiffness required. So the final shape of the equation becomes

$$E_{ijkl}(\rho) = \rho(x) E^0_{ijkl}, \quad p > 1 \tag{3.24}$$

$$\int \rho(x) d\Omega \leq V, 0 \leq \rho(x) \leq 1, x \in \Omega \tag{3.25}$$

where the introduced function ρ is the density function over the entire domain, which interpolates the stiffness tensor to:

$$E_{ijkl}(\rho = 0) = 0, E_{ijkl}(\rho = 1) = 1 \tag{3.26}$$

The figures below show the effect of the penalization factor on the discreteness of the solution:

(a) p = 1.5 (b) p = 3 (c) p = 5 (d) Chart

Figure 3.2: Effect of penalization value on the discreteness of the solution.

These figures demonstrate that increasing the penalization parameter makes the solution more discrete; however, increasing it too much may cause the solution to diverge from the local minimum [39].

3.2.2 Sensitivity analysis

After the material model is complete, a sensitivity analysis must be carried out to track the behaviour of the variables of interest. For example, if compliance is minimized, subject to volume constraint. The sensitivity of the compliance and the volume with respect to the design variable, which is the fictitious density, must be calculated [49]. Two famous approaches for static response analysis are the direct and adjoint methods. Consider finding the sensitivities/derivatives of the following function:

$$g(u, x) \leq 0 \tag{3.27}$$

where u is the displacement vector obtained by solving $Ku = f$, and the $x \in R$ is the design variable. One design variable is used for simplicity of the derivation. Then it can be generalized. When we do the differentiation, we get

$$\frac{dg}{dx} = \frac{\partial g}{\partial x} + \Sigma_{i=2}^{N} \frac{\partial g}{\partial u_i} \frac{du_i}{dx} \tag{3.28}$$

where N is the number of degrees of freedom. This equation can be written using vector notation as

$$\frac{dg}{dx} = \frac{\partial g}{\partial x} + Z^T \frac{du}{dx} \tag{3.29}$$

where $z \in R^n$ and its ith component is $Z_i = \frac{dg}{du_i}$. The explicit part $\frac{\partial g}{\partial x}$ is usually easy to compute or equals zero. $z^T \frac{du}{dx}$ can be computed by differentiating $Ku = f$

$$K\frac{du}{dx} = \frac{df}{dx} - \frac{dK}{dx}u \tag{3.30}$$

Multiplying both sides by$z^T K^{-1}$ from the left, we get

$$Z^T \frac{du}{dx} = Z^T K^{-1}(\frac{df}{dx} - \frac{dK}{dx}u) \tag{3.31}$$

Solving this equation is the difference between the the direct and the adjoint methods.

Direct analytical sensitivity method

Equation 3.30 is first solved for $\frac{du}{dx}$, then dotted with z. TO problems are famous for having a large number of design variables. For example, if the problem had n variables, equation 3.30 will be solved n times, which is not the most efficient way to solve this problem.

Adjoint method

The following characteristic equation is solved for the λ values:

$$K^T \lambda = z \tag{3.32}$$

$$\lambda = K^{-T} z \tag{3.33}$$

Thus, if it is dotted with $(\frac{df}{dx} - \frac{dK}{dx}u)$,(3.31) will be solved. Considering the large number of design variables, this method is much more efficient. Instead of solving (3.30) n times, (3.33) will be solved only once. The right-hand side of z is the partial derivative of the constraints. So, if the problem has m constraints, then (3.33) will be solved m times.

3.3 Optimization

As discussed in the literature, TO problems can be solved using SLP, SQP, optimality criteria, and MMA. However, MMA was chosen for this work because of its proven high performance when dealing with problems that have a large number of design variables [15]. MMA involves converting non-convex problems, as in (3.30), to small convex subproblems $\widetilde{f}_i(\mathbf{x})$ that can be easily optimized. The subproblem is calculated based on the values of the lower and upper asymptotes, L_i and U_i, respectively, and the gradient information.

Minimum compliance problem optimization using MMA can be written as follows [15]:

$$\begin{aligned} \text{Minimize}: \ & -\sum_{i=1}^{n}\left[\frac{\left(x_i^{(k)}-L_i^{(k)}\right)^2}{x_i-L_i^{(k)}}\frac{\partial c}{\partial x_i}\left(\tilde{\mathbf{x}}^{(k)}\right)\right] \\ \text{Subject to}: \ & \begin{cases} \tilde{\mathbf{x}}^{\mathrm{T}}\mathbf{v}-\bar{v}\le 0 \\ \mathbf{x}\in\mathcal{X}^{(k)}, \end{cases} \\ \text{Where}: \ & \begin{cases} \mathcal{X}^{(k)}=\Big\{\mathbf{x}\in\mathcal{X}\mid 0.9L_i^{(k)}+0.1x_i^{(k)}\le x_i\le 0.9U_i^{(k)} \\ \qquad +0.1x_i^{(k)}, i=1,\ldots,n\Big\}. \end{cases} \end{aligned} \tag{3.34}$$

where x^k is the current design, and the lower and upper asymptotes $L_i^{(k)}$ and $U_i^{(k)}$ are

updated as follows for $k = 1$ and $k = 2$:

$$\begin{aligned} U_i^{(k)} + L_i^{(k)} &= 2x_i^{(k)} \\ U_i^{(k)} - L_i^{(k)} &= 1 \end{aligned} \tag{3.35}$$

For $k \leq 3$,

$$\begin{aligned} U_i^{(k)} + L_i^{(k)} &= 2x_i^{(k)} \\ U_i^{(k)} - L_i^{(k)} &= \gamma_i^{(k)} \end{aligned} \tag{3.36}$$

where

$$\gamma_i^{(k)} = \begin{cases} 0.7 \left(x_i^{(\mathrm{k})} - x_i^{(\mathrm{k}-1)}\right)\left(x_i^{(\mathrm{k}-1)} - x_i^{(\mathrm{k}-2)}\right) < 0 \\ 1.2 \left(x_i^{(\mathrm{k})} - x_i^{(\mathrm{k}-1)}\right)\left(x_i^{(\mathrm{k}-1)} - x_i^{(\mathrm{k}-2)}\right) > 0 \\ 1 \quad \left(x_i^{(\mathrm{k})} - x_i^{(\mathrm{k}-1)}\right)\left(x_i^{(\mathrm{k}-1)} - x_i^{(\mathrm{k}-2)}\right) = 0 \end{cases} \tag{3.37}$$

This means that if the signs of three consecutive iterations are different, x_i is oscillating, and thus the two asymptotes move toward x_i^k. Also, if the signs are the same, then the asymptotes move away from x_i^k for faster convergence. The algorithm can be summarized in the following pseudo-code:

Algorithm 1: MMA algorithm.

Choose an initial feasible design x(0); set k $\leftarrow$ 0;

while *convergence criteria are not met* **do**

  if *k = 1 or k = 2* **then**

    Update L_i^k, and U_i^k using (3.35) ;

  else

    Update L_i^k, and U_i^k using (3.36) and (3.37);

  end

  Calculate sensitivity;

  Solve the MMA subproblem (3.34) to obtain $\widetilde{x}^{(k+1)}$;

  Set $x^{(k-2)} \leftarrow x^{(k-1)}$, $x^{(k-1)} \leftarrow x^k$, $x^k \leftarrow x^{(k+1)}$;

  Set k $\leftarrow$ k+1;

end

3.4 Filtering

Design filtration is an indispensable step during the TO process [14]. It helps to ensure mesh independence and removes the checker-boarding effect. Mesh independence means that results do not depend on the mesh element size. The checker-boarding effect is a phenomenon that occurs in most TO solutions, where the final design has alternating zero and one elements in certain regions that are not manufacturable. The checker-boarding effect can be seen in 3.3; it occurs mainly owing to numerical instabilities. Filters are applied by modifying the design sensitivities or design variable of a single element (density) based on a weighted average of the sensitivities of the elements in a fixed neighbourhood. Such filters are solely heuristic, but filtering yields results that are very close to those produced by a local gradient constraint, takes little extra CPU power, and is very easy to apply since it does not add to the complexity of the optimization problem.

In the literature, two types of filters are used: a density filter and a sensitivity filter. A density filter is used in **Chapter 4**, and a sensitivity filter is used in **Chapter 5** with the stress-constrained problems. Both types of filters work well; however, the sensitivity filter is used with the stress problem, as it simplifies the equations. The density filter [66] modifies the density of each element according to the following equation:

$$\tilde{x}_e = \frac{1}{\sum_{i \in N_e} H_{ei}} \sum_{i \in N_e} H_{ei} x_i \tag{3.38}$$

$$H_{ei} = \max\left(0, r_{\min} - \Delta(e, i)\right) \tag{3.39}$$

where $\tilde{x}_e$ is the filtered density, and N_e is the set of elements i that exist within the filter radius $r_{\min}$. Sensitivities are updated according to

$$\frac{\partial \psi}{\partial x_j} = \sum_{e \in N_j} \frac{\partial \psi}{\partial \tilde{x}_e} \frac{\partial \tilde{x}_e}{\partial x_j} = \sum_{e \in N_j} \frac{1}{\sum_{i \in N_e} H_{ei}} H_{je} \frac{\partial \psi}{\partial \tilde{x}_e} \tag{3.40}$$

where ψ represents the objective function or the constraint function. To apply the sensitivity filter [66], sensitivities are modified using the following equation:

$$\frac{\widehat{\partial c}}{\partial x_e} = \frac{1}{\max\left(\gamma, x_e\right) \sum_{i \in N_e} H_{ei}} \sum_{i \in N_e} H_{ei} x_i \frac{\partial c}{\partial x_i} \tag{3.41}$$

where $\gamma = 10^{-3}$ is a small positive number that is added to avoid division by zero, and H_{ei} is a weight factor defined as

$$H_{ei} = \max\left(0, r_{\min} - \Delta(e, i)\right) \tag{3.42}$$

(a) No filter (b) Filtered

Figure 3.3: Effect of filtering results of *L*-bracket structure using a sensitivity filter.

3.5 Ansys verification model

The fact that not much work is done to optimize both the orientation and the topology of the structure makes it hard to compare the present work to previous work. Moreover, commercial software still did not develop a module that performs topology and raster angle optimization. However, the Ansys TO module can be used to run different studies. Each study has its own raster angle defined for the whole mesh, such that the orientation that results in the minimum objective function will be the best orientation coupled with the resulting topology. Also, the Matlab code developed in this thesis can be adjusted to add only one angle for the whole mesh.

A 2D anisotropic material is defined in the Ansys TO module with a different raster angle in each study using 10° increments. Geometry, mesh, objective function, constraints boundary, and loading conditions are defined depending on the problem. Problem details and results are shown in **Chapter 4**.

3.6 Conclusions

This chapter discussed the main building blocks of the code that is used in the following chapters. When dealing with structures that must be placed in a specific orientation, transformation matrices are extremely useful. They allow the finite element to represent any object that can rotate, be it a fiber, truss, or beam element. The SIMP method is a TO method that is accurate as well as easy to implement and modify. Hence, it was used as the material model. The MMA method is a famous method in the field of structural optimization, as it can be easily handled and modified. Thus, it is the method on which all results will be built. Density filtering will be used in the problems of **Chapter 4**, as it is the most recently developed method, and sensitivity filtering will be used in **Chapter 5** to simplify the sensitivity analysis process.

Chapter 4

COMPLIANCE AND MASS MINIMIZATION

Summary: This chapter presents compliance and mass minimization problems subjected to global constraints. It is divided into three main sections: problem formulation, sensitivity analysis, and numerical examples.

4.1 Problem formulation

The basic compliance minimization problem presented here uses a linear volume fraction constraint and bound constraints on the density variables and the angles, which are the

main design variables. The problem statement is as follows:

$$\begin{aligned} \min_{\boldsymbol{\rho},\,\boldsymbol{\theta}} \quad & C(\boldsymbol{\rho},\boldsymbol{\theta}) = \mathbf{U}^T\mathbf{K}(\boldsymbol{\rho},\boldsymbol{\theta})\mathbf{U} \\ \text{s. t.} \quad & \sum_{i=1}^{N} V_i/V_0 \le V_f, \\ & \mathbf{K}(\boldsymbol{\rho},\boldsymbol{\theta})\mathbf{U} = \mathbf{F}, \\ & \mathbf{0} \le \boldsymbol{\rho} \le \mathbf{1}, \\ & -\pi \le \boldsymbol{\theta} \le \pi \end{aligned} \tag{4.1}$$

while the mass minimization problem is stated as follows:

$$\begin{aligned} \min_{\boldsymbol{\rho},\,\boldsymbol{\theta}} \quad & M(\boldsymbol{\rho},\boldsymbol{\theta}) = \sum_{e=1}^{N} \rho_e \\ \text{s. t.} \quad & C(\boldsymbol{\rho},\boldsymbol{\theta}) \le C_{max}, \\ & \mathbf{K}(\boldsymbol{\rho},\boldsymbol{\theta})\mathbf{U} = \mathbf{F}, \\ & \mathbf{0} \le \boldsymbol{\rho} \le \mathbf{1}, \\ & -\pi \le \boldsymbol{\theta} \le \pi \end{aligned} \tag{4.2}$$

where $\boldsymbol{\rho}$ is a vector representing the density of each finite element, $\boldsymbol{\theta}$ is a vector representing the angle of the print direction in each finite element, C is the compliance of the structure, $\mathbf{U}$ is the global displacement vector, $\mathbf{K}$ is the global stiffness matrix, N is the number of finite elements, V_i is the volume of an element i, V_0 is the volume of the original design domain, V_f is the prescribed volume fraction, $\mathbf{F}$ is the global force vector, M is the mass of the structure, ρ_e is the density of element e, and C_{max} is an upper bound on the compliance obtained from the compliance minimization problem.

4.2 Gradient calculation

Gradients are the core of the solution. They will be calculated using the adjoint variable method mentioned in section 3.2.2 [9]. First, the adjoint compliance objective function can be defined in terms of the main compliance function augmented with the global state equation multiplied by the Lagrange multiplier as

$$\bar{C}(\mathbf{U}(\rho,\theta),\rho,\theta,\lambda) = C(\rho,\theta) - \lambda * (\mathbf{K}(\rho,\theta)\mathbf{U}(\rho,\theta) - \mathbf{F}) \tag{4.3}$$

Taking the derivative of the augmented function:

$$\frac{\mathrm{D}\overline{\mathrm{C}}}{\mathrm{D}\rho_{\mathrm{e}}} = \frac{\partial \mathrm{C}}{\partial \rho_{\mathrm{e}}} + \frac{\partial \mathrm{C}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial \rho_{\mathrm{e}}} - \frac{\mathrm{D}\lambda}{\mathrm{D}\rho_{\mathrm{e}}} * (\mathbf{K}\mathbf{U} - \mathbf{F}) - \lambda * \left(\frac{\mathrm{DK}}{\mathrm{D}\rho_{\mathrm{e}}}\mathbf{U} + \mathrm{K}\frac{\mathrm{D}\mathbf{U}}{\mathrm{D}\rho_{\mathrm{e}}} - \frac{\mathrm{D}\mathbf{F}}{\mathrm{D}\rho_{\mathrm{e}}}\right) \tag{4.4}$$

Since equilibrium is satisfied and $\mathbf{KU} = \mathbf{F}$, the derivative equation can be reduced to

$$\frac{\mathrm{D}\overline{\mathrm{C}}}{\mathrm{D}\rho_{\mathrm{e}}} = \frac{\partial \mathrm{C}}{\partial \rho_{\mathrm{e}}} + \frac{\partial \mathrm{U}}{\partial \rho_{\mathrm{e}}}\left(\frac{\partial \mathrm{C}}{\partial \mathrm{U}} - \mathrm{K}\lambda\right) - \lambda * \left(\frac{\mathrm{DK}}{\mathrm{D}\rho_{\mathrm{e}}}\mathrm{U} - \frac{\mathrm{D}\mathbf{F}}{\mathrm{D}\rho_{\mathrm{e}}}\right) \tag{4.5}$$

Since λ can have any arbitrary value, it will be set to a value that makes

$$\frac{\partial C}{\partial \mathbf{U}} - \mathbf{K}\lambda = 0 \tag{4.6}$$

Since $C = \mathbf{U}^{\mathrm{T}}\mathbf{F}$, equation (4.6) can be written as

$$\frac{\partial \mathbf{U}\mathbf{F}}{\partial \mathbf{U}} - \mathbf{K}\lambda = 0 \tag{4.7}$$

$$\mathbf{F} = \mathbf{K}\lambda \tag{4.8}$$

Hence, λ can be set to $\mathbf{U}$. Moreover, as $\left(\frac{\partial C}{\partial \rho_{\mathrm{e}}} = 0\right)$ and $\left(\frac{\mathrm{D}\boldsymbol{F}}{\mathrm{D}\rho_{\mathrm{e}}} = 0\right)$, the expression can be reduced to

$$\frac{\mathrm{D}\overline{\mathrm{C}}}{\mathrm{D}\rho_{\mathrm{e}}} = -\mathrm{U}^{\mathrm{T}}\frac{\mathrm{DK}}{\mathrm{D}\rho_{\mathrm{e}}}\mathrm{U} \tag{4.9}$$

By applying the previous formulation to the finite element mesh, I determined that the derivative of compliance of each element with respect to densities of other elements is zero except for its own density. Hence, by using equation 4.9, the derivative of c w.r.t. ρ_e equals

$$\frac{\partial \mathrm{C}}{\partial \rho_{\mathrm{e}}} = -\mathrm{p}\,(\rho_{\mathrm{e}})^{\mathrm{p}-1}\,\mathbf{u}_{\mathrm{e}}^{\mathrm{T}}\mathbf{k}_{\mathrm{e}}(\theta)\mathbf{u}_{\mathrm{e}} \tag{4.10}$$

Similarly, by using equation (3.11), the derivative w.r.t. θ_e can be defined as

$$\frac{\partial C}{\partial \theta_e} = u_e^T\left\{\iint_A B^T(\frac{\partial T_1(\theta_e)^{-1}}{\partial \theta_e}D(\theta_e)T_2(\theta_e) + T_1(\theta_e)^{-1}D(\theta_e)\frac{\partial T_2(\theta_e)}{\partial \theta_e})Bdxdy\right\}u_e \tag{4.11}$$

Since the mass function was defined as $M(\boldsymbol{\rho}, \boldsymbol{\theta}) = \sum_{e=1}^{N}\rho_e$, it depends only on element density and not on element orientation. Its derivative w.r.t. ρ_e and $theta_e$ is

$$\frac{\partial M}{\partial \rho_e} = 1 \tag{4.12}$$

$$\frac{\partial M}{\partial \theta_e} = 0 \tag{4.13}$$

The sensitivities of the volume constraint can be defined as

$$\frac{\partial v}{\partial \rho} = 1 \tag{4.14}$$

$$\frac{\partial v}{\partial \theta} = 0 \tag{4.15}$$

4.3 Numerical examples

In this section, I apply the theories mentioned in the earlier sections regarding real-life applications and analyze the results. Two materials will be used in the examples. The first material is acrylonitrile butadiene styrene (ABS). It has a thickness of 0.317 mm and is a widely used 3D printing material used in FFF. Its mechanical characterization was done by [22]; the extracted properties are shown in Table 4.1. The second material is epoxy glass [17], which I chose because of its high anisotropy, as shown in Table 4.1. In the final topologies, ABS is coloured in cyan, and epoxy glass is coloured in magenta.

Material	E_1 (Pa.)	E_2 (pa.)	G_{12} (pa.)	v_{12}	v_{21}
ABS	2.6e6	7.8e6	1.3e6	0.25	0.75
EG	38.6e9	8.27e9	4.14e9	0.27	0.0578

Table 4.1: Mechanical properties of ABS and EG

4.3.1 Verification of results

Given that not much work was required to optimize both the orientation and the topology of the structure, it is difficult to verify the new methods. The code was divided into three versions to be able to determine whether the results are valid and whether the optimization is working properly. The validation process was performed on a problem that is defined as follows: A cantilever beam fixed from the left side with a length of 64 mm and a height of 40 mm is subjected to a point load of 80 N at its lower-right corner (Fig. 4.1). The problem is minimization compliance of the structure. The constraint function is the volume fraction, which is set to 25% of the original design space. The mesh is composed of 2400 elements, with 60 elements along the x axis and 40 elements

along the y axis. This problem is then solved as a plane stress problem, and the material was defined as ABS.

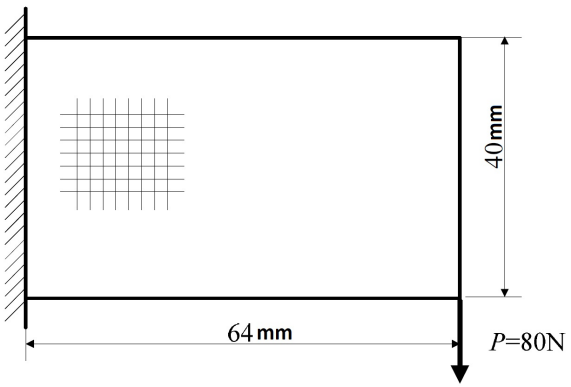


Figure 4.1: Cantilever beam fully fixed from the left and showing the design space dimensions and load.

The results were validated in the following sequence, with different TO simulations being run on the Ansys TO module for different orientations of fibers. These data were used to determine the optimal fiber angle, if the whole geometry is oriented in one direction. The code was run to optimize topology only. This topology was then extracted, and only one angle optimization was done for the extracted geometry. This resulted in an optimum angle equal to the optimum angle of the commercial software. After that, the code of coupling topology and orientation optimization was run to solve the same problem, and a similar topology and optimal angle were found.

After that, a couple of examples were run for optimization of both topology and the angles of the whole mesh. But this time, each element had its own angle as a unique design variable. The results were intuitively right, as a lower compliance was achieved than for the one-angle problem, and compression- and tension-only members were detected in the fields that must have tension or compression members only. The topology was the same as the fixed-angle problem; however, the stiffness increased because

this problem was optimizing compliance, only adding the volume fraction as a constraint.

Ansys was used to solve this problem for fibers having different orientations, with the angle varying from 0°–360° with 10° increments. The mesh independence study was completed first. It was found that 60 x 40 elements is sufficient to obtain stable results.

Each analysis had a predetermined raster angle, which is defined in the software material library. The SIMP solver was used, and the minimum compliance results are shown in Fig. 4.2. The minimum was obtained when the fibers are oriented at 90° or 270°. This was expected because that is the direction that has a higher Young's modulus. The minimum compliance value was 0.114 Nm. The final topology is shown in Fig. 4.2.

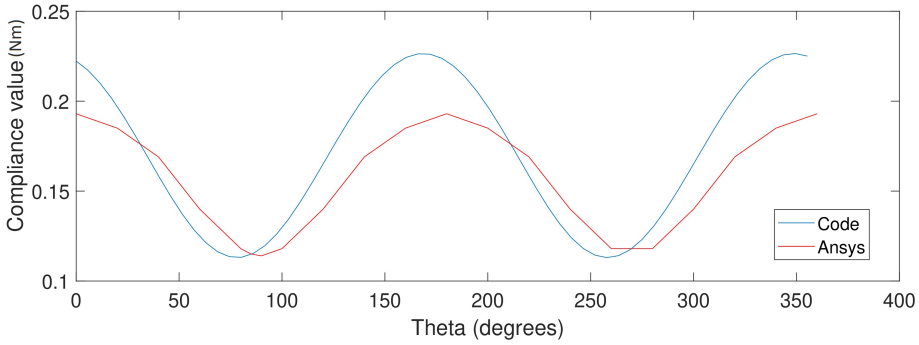
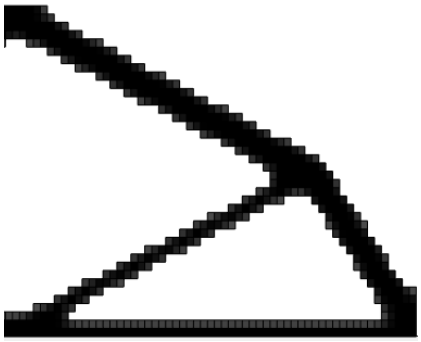


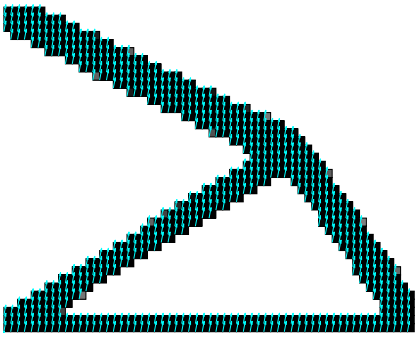
Figure 4.2: Compliance comparison from a study done on Ansys with 20 degrees increment and on our algorithm

The topology-only optimization problem was run on the same problem with the fibers oriented along the y axis ($\theta = 90°$). A similar topology was achieved as shown in 4.3a with a compliance of 0.1153 Nm. The error between the Ansys results and the TO-only code was found to be 1.13%. The topology matrix was imported from the topology-only code to orientation-only optimization code: the optimized angle was found to be $\theta = 84°$) with a compliance result of 0.113 Nm. The orientation-only optimization result is shown in 4.3b with a compliance value of 0.113 Nm. Finally, a similar study to the one

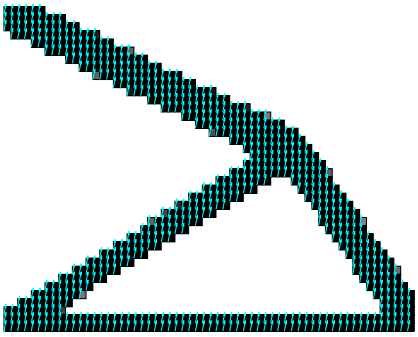
made in Ansys was done in the code by varying the angles with a smaller increment of 0.1 rad for a fixed topology obtained from the topology-only code. Because of the high computational efficiency of the code, a small-increment study could be completed in 45 seconds. The results are shown in 4.2, which is the same as the curve obtained from the Ansys study. Finally, the code was run in the case of coupling both topology and orientation in one angle optimization problem: a compliance of 0.1135 Nm was achieved at an angle of 80°. The results of the topology and orientation angle problem are shown in 4.3c. From these separate small studies, it can be concluded that the compliance calculation, angle calculation, and optimization parts of the code are working well to produce the right answer. The next part of the results section uses the code to solve the same problem but defines a separate angle for each element in the mesh and new materials as the code is validated to be working properly.



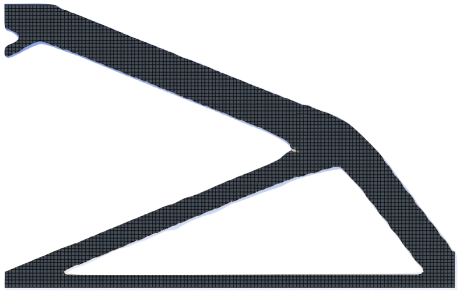
(a) Topology-only result when oriented at 90°.



(b) Orientation-only optimization for the extracted topology.



(c) Coupling topology and one angle result.



(d) Final topology of the structure after optimization at the optimal angle on Ansys

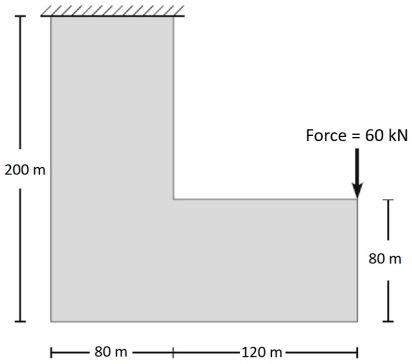
Figure 4.3: Structures generated from our code and from Ansys at the optimal angle

4.3.2 Case studies

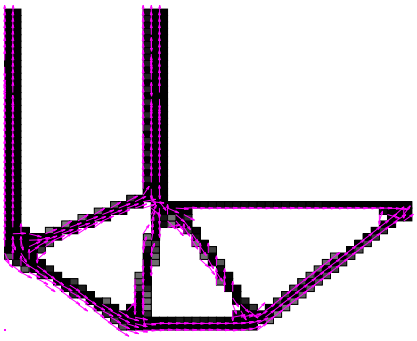
L-bracket compliance minimization

The first example is a compliance minimization or stiffness maximization problem, as presented in equation (4.1), with the maximum volume of the final structure set to be 0.25% of the original structure. The design domain is the L-bracket (Fig. 4.4a) made of epoxy glass, fixed from the upper-left edge with a force of 60 kN applied to the node the right corner. This L-bracket is used in the medical, food, and beverage industries as well as in specialty equipment [67]. The initial density was set to be 0.25, and the initial angle was -0.1 rad. Compliance decreased from 234.068 Nm until it reached 0.652 Nm on the optimal design. Compliance decreased significantly as each element was

oriented in the direction that leads to minimum compliance. A decrease of 358.8 % was achieved after 89 iterations, as shown in Fig. 4.5c. The minimum compliance value was reached after 30 iterations; however, minimal decreases kept occurring and were adjusted for 70 more iterations until the maximum changes in density and angle were minimal (Figs. 4.5a, 4.5b). The final topology with the angle orientation is shown in Fig. 4.4b. Note that near the support and the force application node, the density increased. In addition, the fibers are usually oriented along the element because all elements carry either tension or compression.

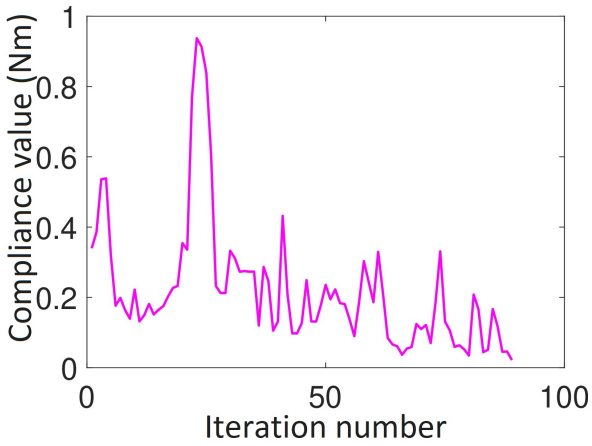


(a) L-bracket structure subjected to 60 kN loading and fixed from the upper-left corner.

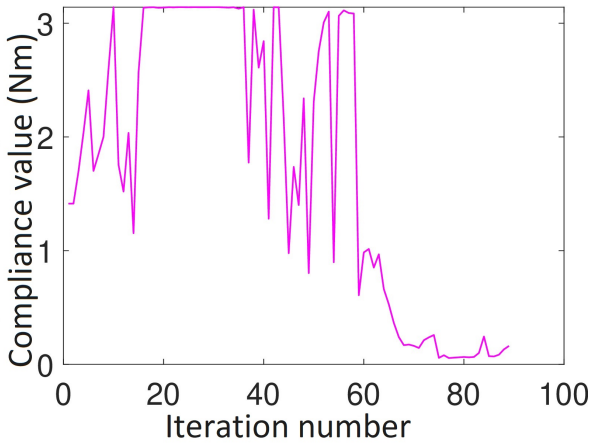


(b) Final topology and fiber orientation after solving the compliance minimization problem.

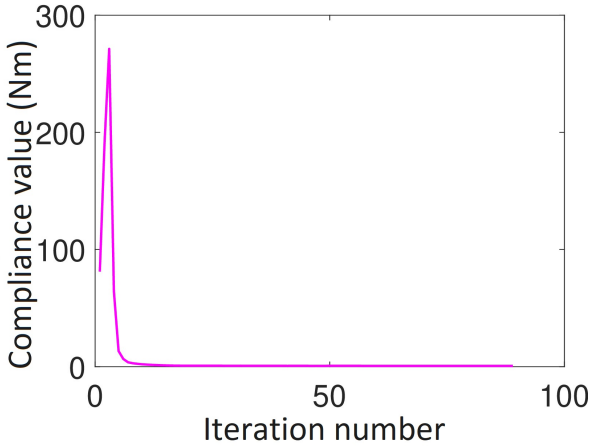
Figure 4.4: Case study 1, compliance minimization of L-bracket structure with volume fraction = 0.25



(a) Maximum change in ρ for the compliance minimization problem.



(b) Maximum change in θ for the compliance minimization problem.

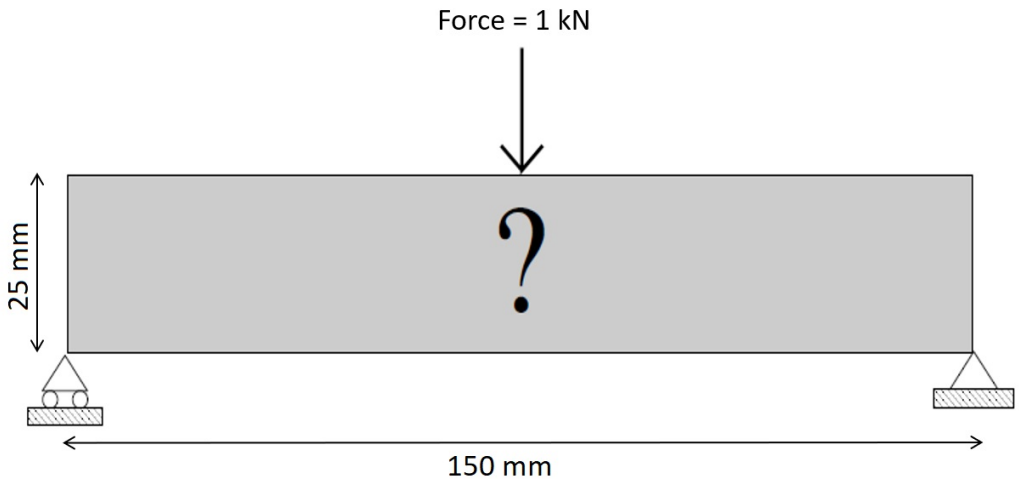


(c) Convergence curve for the compliance minimization problem.

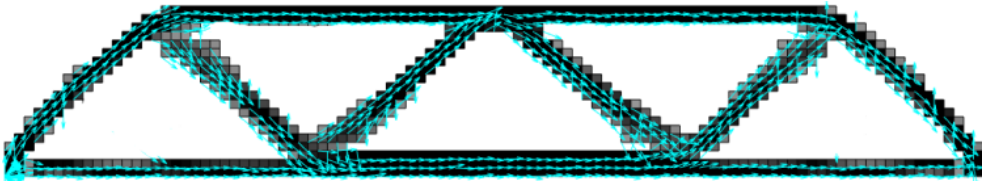
Figure 4.5: Graphs of case study 1

Messerschmitt–Bölkow–Blohm mass minimization

The second example is a mass minimization problem as presented in equation (4.2), with the maximum compliance limit for the final structure set to be 0.1082 Nm of the original structure. The design domain is the famous Messerschmitt–Bölkow–Blohm (MBB) problem shown in Fig. (4.6a). The structure is made of ABS material and is fixed from the lower-right corner node, with a roller on the lower-left corner node. A force of 1 kN is applied to the middle of the structure. This MBB problem is a famous problem in the field of TO that produces a truss-like structure [68]. The initial density was set to be 0.25, with an initial angle of 0.4 rad. The mass decreased from 3.75 kg until it reached 1.37 kg in the optimal design. The mass decreased significantly, as each element was oriented in a direction that leads to minimum mass. The mass savings were calculated to be 63.8 % after 1200 iterations, as shown in Fig. 4.7c. Mass minimization problems usually take significantly more iterations than mass minimization. That is not specific to the MBB problem; however, it comes from the mass minimization using the MMA method. This will be clarified in the next study, as it will be a comparison between the compliance and mass minimization problems for the same design domain. By observing the convergence graphs shown in Fig. 4.7, the mass value kept decreasing for 1200 iterations, until the density changes reached their minimum. Note that changes in ρ and θ continued fluctuating between high and low values. Hence, the convergence criteria set to be a change in ρ should be less than 0.003. Eventually, the fibers are usually oriented along the element because all elements carry either tension or compression, as was the case in the previous example.

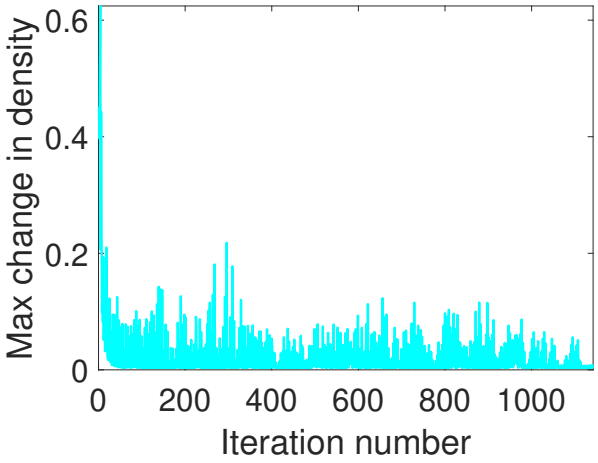


(a) MBB beam design space with boundary and loading conditions

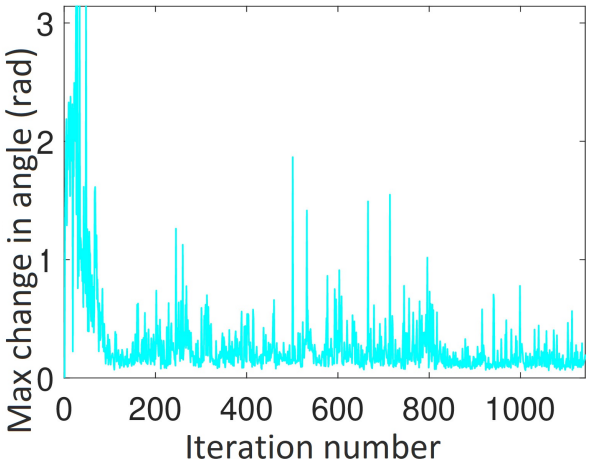


(b) Optimized topology and fibers orientation of MBB beam

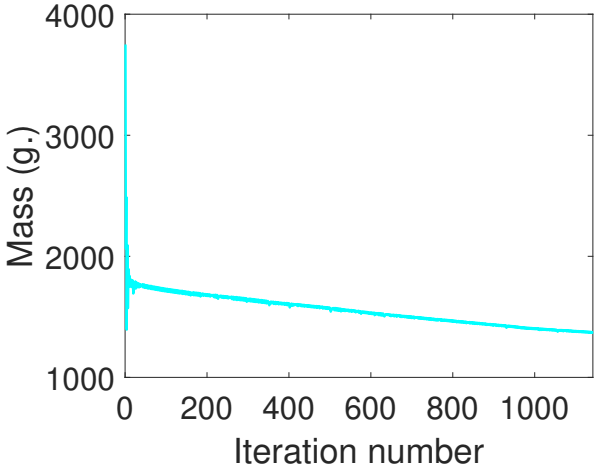
Figure 4.6: Case study 2. Mass minimization of MBB beam with compliance limit = 0.1082 Nm



(a) Maximum change in ρ for the mass minimization problem.



(b) Maximum change in θ for the mass minimization problem.



(c) Convergence curve for the mass minimization problem.

Figure 4.7: Graphs of case study 2

Comparison between mass and compliance minimization for a rectangular domain

This section will show the differences between solving a problem from the compliance minimization or mass minimization perspective. The application design space is shown in Fig. 4.8. A force of 1000 N was applied in both the mass and compliance minimization problems. A compliance minimization problem with a volume fraction of 0.3 is solved first; the minimum compliance calculated was used as the compliance threshold of the subsequent mass minimization problem.

For the compliance problem, because of the low volume fraction, the structural members are considered to be carrying only tensile or compressive loads (i.e., a truss-like structure). As a result, all the fibers are oriented in a direction aligned with the load carried by each element. The compliance of the optimized structure is 0.1082 Nm.

Figure 4.9b shows that the compliance nearly reached the minimum after 50 iterations; however, the code had to continue in the optimization process because there were fine changes occurring in the densities and angles of each element; these changes can further enhance the compliance value. The changes continued decreasing until a predefined tolerance was reached. The changes in density and angle can be seen in Figs. 4.9c and 4.9d.

As for the mass minimization problem, the compliance limit was set to 0.1082, which is the minimum reached in the first problem. At the end of the optimization process, the mass reached was 31% of the original mass. The final topology and raster angle are shown in Fig. 4.10a. As shown in Fig. 4.10b, the objective function zigzags and reaches the optimum after close to 1200 iterations. This convergence scheme is similar to the steepest descent method convergence, which is linear and incredibly time consuming.

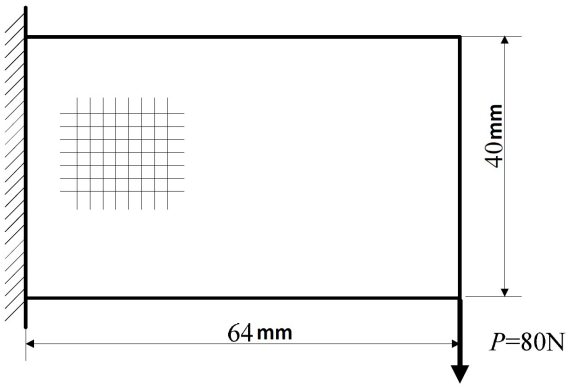
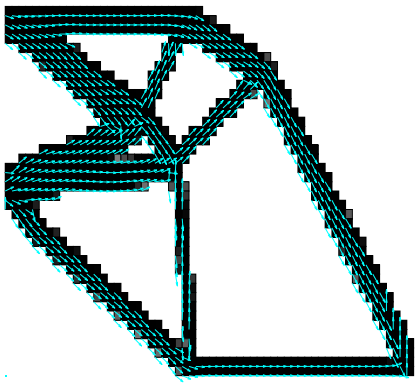


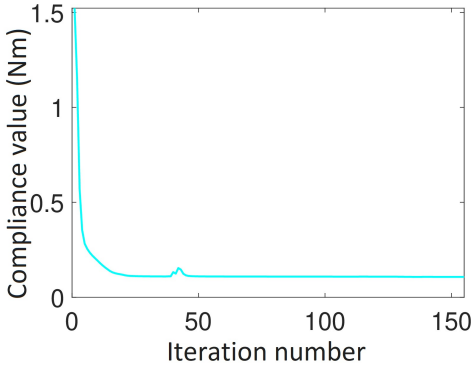
Figure 4.8: Cantilever beam half fixed from the left and showing the design space dimensions and load.

Unlike with compliance minimization, changes in angles and densities reached low values, as shown in Figs. 4.10c and 4.10d, while the mass continued decreasing until it reached 31 % at convergence.

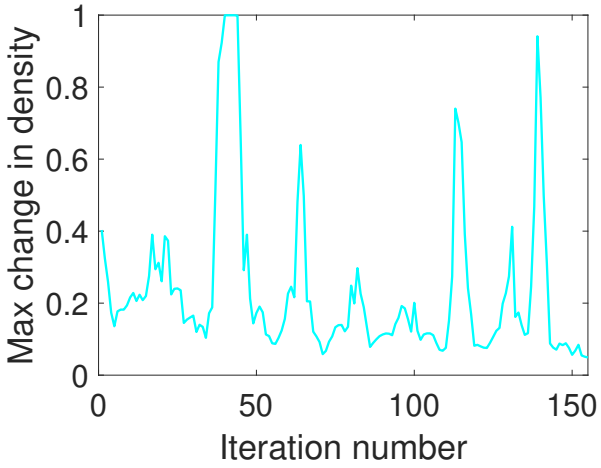
As shown, both problems reached nearly the same final topology with only minor differences. The advantage of the compliance minimization problem was that the minimum was reached much faster than it was in the mass minimization problem, with no intermediate elements. However, looking at the final topology of both methods, the elements in the compliance minimization problem resulted in a structure with sharper corners, meaning that the corners were not as smooth as those produced in the mass minimization problem. This result favours the mass minimization method over the compliance minimization method, as sharp corners lead to the initiation of stress concentrations in the structure.



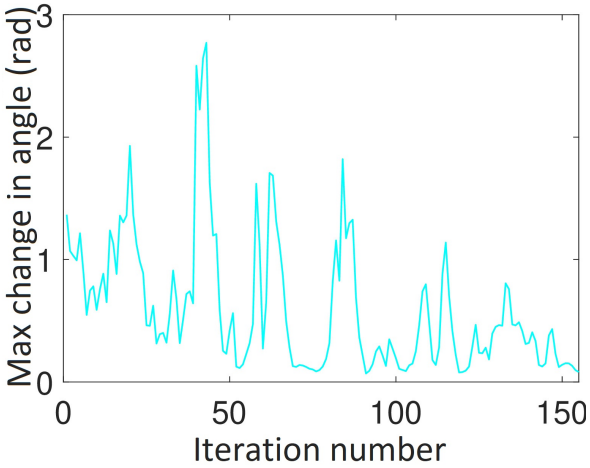
(a) Final topology and raster angle after solving the compliance minimization problem.



(b) Convergence curve for the compliance minimization problem.

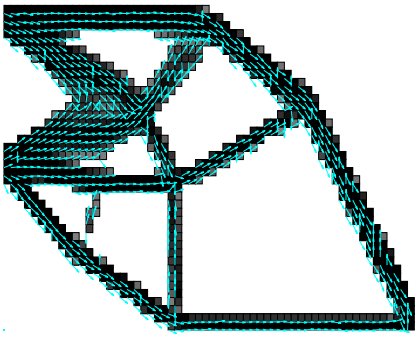


(c) Change in ρ for the compliance minimization problem.

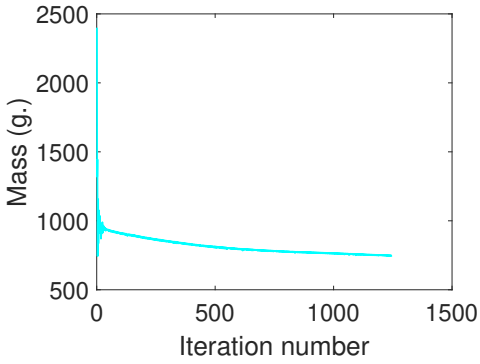


(d) Change in θ for the compliance minimization problem.

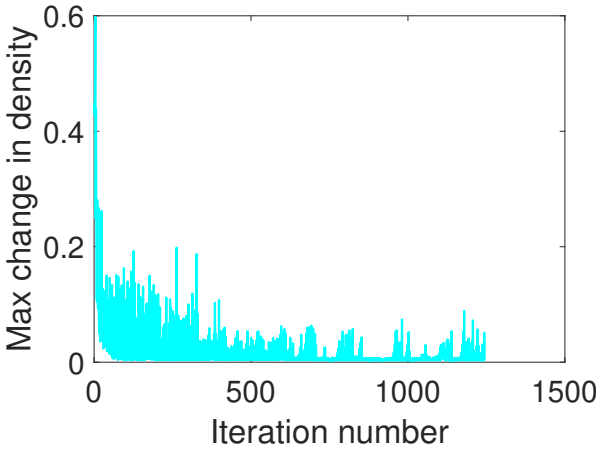
Figure 4.9: Case study 3. Compliance minimization of half fixed rectangular structure with volume fraction = 0.3.



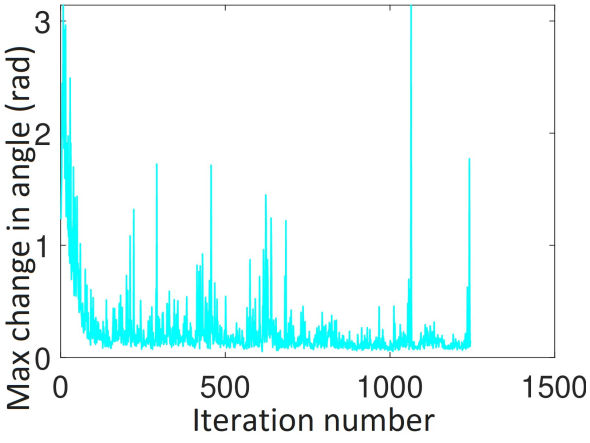
(a) Final topology and raster angle after solving the mass minimization problem



(b) Convergence curve for the mass minimization problem.



(c) Maximum change in ρ for the mass minimization problem.



(d) Maximum change in θ for the mass minimization problem.

Figure 4.10: Case study 3. Mass minimization of half fixed rectangular structure with maximum compliance = 0.1082.

4.4 Conclusions

After optimization, the finite elements align themselves along the contours of the structural element, especially if the element is carrying pure tension or pure compression. This demonstrates how effective it is to use element transformation angles to represent physical fiber angles. If the compliance minimization problem is solved, the mass minimization problem of polymers is usually straightforward to deduce. Adding the fiber angles to the optimization problem results in a greater decrease in mass by orienting the fibers in the correct way while constraining compliance in order not to increase above a certain threshold. This method has great potential for future use in the AM industry.

Chapter 5

DIRECTION-DEPENDANT STRESS CONSTRAINTS

Summary: This chapter discusses compliance minimization problems subjected to local constraints. It is divided into five main sections: problem formulation, stress measure, objective function and sensitivity analysis, modelling limitations and numerical examples.

5.1 Problem formulation

The advanced form of TO problems involve local constraints as the stress constraints. In this chapter, a compliance minimization problem subjected to stress and volume constraints will be solved, taking into account that each element has density ρ_e and angle θ_e as the main design variables. Stress constraints are applied along the fibers "σ_1" and perpendicular to the fibers "σ_2". This can be defined mathematically as follows:

$$\text{Minimize } C(\boldsymbol{\rho},\boldsymbol{\theta}) = \mathbf{U}(\boldsymbol{\rho}\boldsymbol{\theta})^{\mathrm{T}}\mathbf{F}$$

$$\text{Subject to } \begin{cases} \sum_{i=1}^{N} V_i/V_0 \leq V_f \\ \mathbf{K}(\boldsymbol{\rho},\boldsymbol{\theta})\mathbf{U}(\boldsymbol{\rho},\boldsymbol{\theta}) = \mathbf{F} \\ -(\sigma_1^C)_{ult} \leq \sigma_1^{PN}(x) \leq (\sigma_1^T)_{ult} \\ -(\sigma_2^C)_{ult} \leq \sigma_2^{PN}(x) \leq (\sigma_2^T)_{ult} \\ \rho_{\min} \leq \rho_{\mathrm{e}} \leq 1 \\ -\pi \leq \theta_{\mathrm{e}} \leq \pi \end{cases} \tag{5.1}$$

where design variables of the whole domain are collected in a vector $\boldsymbol{x}$, C is the compliance of the structure, $\mathbf{U}$ is the global displacement vector, written here as a function of the density and material angle design variable vectors ρ and θ, respectively. $\mathbf{K}$ is the global stiffness matrix, N is the number of finite elements, V_i is the volume of an element i, V_0 is the volume of the original design domain, V_f is the prescribed volume fraction, $\mathbf{F}$ is the global force vector, and ρ_e is the density of element e. The vectors $\rho = [\rho_1, \rho_2, .., \rho_N]$ and $\theta = [\theta_1, \theta_2, .., \theta_N]$ are defined, respectively, by element density design variables $\rho_e, e = 1, ..., N$ and element material angle design variables $\theta_e, e = 1, ..., N$, where N is the number of elements in the finite element model. σ_1, and σ_2 are the principal stresses along the print fibers and perpendicular to the fibers, respectively. $\sigma_1^{PN}(x)$ is the modified P-norm stress measure for the cluster of stresses along the fibers, and $\sigma_2^{PN}(x)$ is the modified P-norm stress measure for the cluster of stresses perpendicular to the fibers, which will be discussed in detail in subsection 5.2.2; σ^C and σ^T are the ultimate compressive and tension stresses, respectively.

5.2 Stress measure

5.2.1 Failure criteria

Failure expectancy can be calculated using many theories, including maximum stress, maximum pressure, Tsai–Hill, and the tensor polynomial failure criterion [17]. In this work, the focus is on the maximum stress failure criterion owing to its effectiveness in predicting failure and its ease of implementation. The maximum stress failure theory states that failure can be determined from stress values in the X and Y directions directly without any further calculations. Shear values are ignored in this study.

According to maximum stress loss theory, all individual stress components must be less than the ultimate stress value in the assumed direction. As a result, the failure criterion may be written as follows:

$$\begin{aligned} X_C < \sigma_1 < X_T \\ Y_C < \sigma_2 < Y_T \\ |\tau_{23}| < Q \end{aligned} \tag{5.2}$$

where X_T and X_C are the maximum allowed tension and compression stresses in the first principal direction, respectively. Similarly, Y_T, and Y_C are the maximum allowed tension and compression stresses in the second principal direction, respectively. σ_1 is the stress along the fibers, and σ_2 is the stress perpendicular to the fibers. τ_{23} and Q are the shear stress and the maximum allowed shear, respectively, but these constraints are out of scope for this study.

To reduce the number of stress limits, the absolute values of the stresses will be used, with only one ultimate stress assumed for tension and compression. As a result, the stress constraints for the following analysis would be:

$$|\sigma_1| < X$$
$$|\sigma_2| < Y \tag{5.3}$$

Shear stresses are not taken into account in this study, but they can easily be added using the same technique of X and Y stresses.

5.2.2 Stress measure

Stress penalization

As mentioned in the literature review, singularity problems arise because of strain generation in elements with low density. Such elements can still experience strain, resulting in a high stress value in this area when it should be zero. This can be solved by stress penalization.

A stress tensor of a 2D finite element is defined as follows [65]:

$$\hat{\sigma}_a(x) = (\hat{\sigma}_{ax}, \hat{\sigma}_{ay}, \hat{\tau}_{axy})^T = DB_a u(x) \tag{5.4}$$

Stresses are penalized as well to obtain intermediate stress values depending on the fictitious density of each element. Stress penalization functions in the literature are diverse; the method chosen here is the one suggested by Le et al. [18], as it proved its efficiency when coupled with the P-norm stress measure in [57]. Penalization is done as follows:

$$\sigma_a(x) = \eta_S\left(\rho_e(x)\right)\hat{\sigma}_a(x) \tag{5.5}$$

where $\eta_S\left(\rho_e(\boldsymbol{x})\right) = \left(\rho_e(\boldsymbol{x})\right)^{\frac{1}{2}}$. This method will help to prevent high stress values in

areas that do not have material [69], since:

$$\lim_{\rho_e \to 0} \sigma_a(x) = \mathbf{0} \tag{5.6}$$

Maximum stress cluster technique

As mentioned in the literature review, because of the local nature of stresses, a global measure stress technique is used here in conjunction with stress agglomeration techniques [57] to constrain stresses with a low number of constraints while still obtaining accurate results. Stress constraints can be applied using one of three techniques: local, global, or clusters. Local stress measures imply that each element is accompanied by two stress restrictions, which is inefficient in terms of computation. The term "global constraining" refers to the addition of only one constraint to the entire mesh using a global measure such as P-norm [55]. The P-norm measure is expressed as follows:

$$\sigma^{PN}(x) = \left(\frac{1}{N} \sum_{a=1}^{N} (\sigma_1^a(x))^P \right)^{\frac{1}{P}} \tag{5.7}$$

where σ^{PN} is the global stress measure, N is the number of elements, P is the P-norm coefficient, and σ_1^a is the stress value in the first principal direction of the stress evaluation point a. The global stress measure forces the σ^{PN} to approach the maximum stress value in the entire mesh, but it only does so when $P = \infty$ [55]:

$$\lim_{p \to \infty} \left(\frac{1}{N} \sum_{a=1}^{N} (\sigma_1^a(x))^p \right)^{\frac{1}{p}} = \max \sigma_1^a(x) \tag{5.8}$$

As a result, we can conclude that the global stress measure will be effective when used with a small number of stress evaluation points, but will be inaccurate when used with the entire mesh. This means that the global stress technique works well when the stresses are divided into groups known as clusters. Each cluster will have its own P-norm

measure, which provides a better approximation of the highest stress in the cluster. However, it would still underestimate the stress values in the cluster depending on the amount of stress evaluation point values that dominate the cluster. The first method for arranging stress evaluation points in clusters is to use the stress level technique. This technique involves arranging stresses in descending order and then adding them to clusters. The number of stress evaluation points in each cluster will be equal to the number of elements n_e divided by the number of constraints n_c. It can be expressed mathematically as follows [57]:

$$\underbrace{\sigma_1^1 \geq \sigma_1^2 \geq \sigma_1^3 \geq \ldots.. \geq \sigma_1^{\frac{n_e}{n_c}}}_{Cluster1} \geq \underbrace{\cdots\cdots \geq \sigma_1^{\frac{2*n_e}{n_c}}}_{Cluster2} \geq \underbrace{\cdots\cdots \geq \sigma_1^{\frac{(n_c-1)n_e}{n_c}}}_{Cluster(n_c-1)} \geq \underbrace{\cdots \geq \sigma_1^{n_e}}_{n_c} \quad (5.9)$$

Cluster 1	Cluster 2	...	Cluster n_c
σ_1^1	$\sigma_1^{(\frac{n_e}{n_c}+1)}$	.	$\sigma_1^{(n_e-(\frac{n_e}{n_c}))}$
σ_1^2	$\sigma_1^{(\frac{n_e}{n_c}+2)}$	.	.
σ_1^3	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
$\sigma_1^{(\frac{n_e}{n_c})}$	$\sigma_1^{(\frac{2*n_e}{n_c})}$	.	$\sigma_1^{n_e}$

Table 5.1: Stress clustering using stress level technique.

The second technique is the distributed stress method [18], where elements are arranged in descending order as in (5.9) and then each cluster receives an element. That is, the first element is inserted into the first cluster, the second element is inserted into the second, and so on until each cluster has one stress evaluation point, at which point the loop begins again until all of the evaluation points are placed in their corresponding clusters. This method aids in the averaging of the global stress measure in each cluster.

Cluster 1	Cluster 2	Cluster 3	...	Cluster n_c
σ_1^1	σ_1^2	σ_1^3	...	$\sigma_1^{(\frac{n_e}{n_c})}$
$\sigma_1^{(\frac{n_e}{n_c}+1)}$	$\sigma_1^{(\frac{n_e}{n_c}+2)}$	...	...	$\sigma_1^{(\frac{2*n_e}{n_c})}$
...	...	...	...	...
...	...	...	...	...
$\sigma_1^{(n_e-(\frac{n_e}{n_c}))}$	...	...	...	$\sigma_1^{n_e}$

Table 5.2: Stress clustering using stress averaging technique.

In this work, the stress leveling technique is used together with the minmax method. Only the first cluster will be used as a stress constraint in each of the two principal directions. This method is more efficient because it decreases the number of constraints while keeping the stresses within the constraint as each iteration's stresses are rearranged and the maximum stresses go in the first cluster. It is hereinafter referred to as the "maximum stress level technique."

5.3 Objective function and gradients calculation

Since MMA is a gradient-based method, gradients of the objective function and constraints must be calculated before proceeding with the optimization process. Many of the gradients calculated in this section are for the stresses along the fiber direction. For the perpendicular direction, stress constraints and its derivatives can be calculated using the same formula used for the first principal direction. Compliance can be defined as follows:

$$c(\boldsymbol{\rho}, \theta) = \sum_{\mathrm{e}=1}^{\mathrm{N}} (\rho_{\mathrm{e}})^{\mathrm{p}} \, \mathbf{u}_{\mathrm{e}}^{\mathrm{T}} \mathbf{k}_{\mathrm{e}} \left(\theta_{\mathrm{e}}\right) \mathbf{u}_{\mathrm{e}} \tag{5.10}$$

Its derivative with respect to density and theta is:

$$\frac{\partial c}{\partial \rho_e} = -p\left(\rho_e\right)^{p-1}\mathbf{u}_{\mathrm{e}}^{\mathrm{T}}\mathbf{k}_{\mathrm{e}}\mathbf{u}_{\mathrm{e}} \tag{5.11}$$

$$\frac{\partial c}{\partial \theta_e} = \rho_e^p \mathbf{u}_{\mathrm{e}}^{\mathrm{T}}\left\{\iiint_{\Omega_{\mathrm{e}}}\mathbf{B}^{\mathrm{T}}\left(\frac{\partial \mathrm{T}\left(\theta_e\right)^{-1}}{\partial \theta_e}\mathrm{D}\,\mathrm{T}\left(\theta_e\right)^{-\mathrm{T}} + \mathrm{T}\left(\theta_e\right)^{-1}\mathrm{D}\frac{\partial \mathrm{T}\left(\theta_e\right)^{-\mathrm{T}}}{\partial \theta_e}\right)\mathrm{B}\mathrm{d}\Omega\right\}\mathbf{u}_{\mathrm{e}} \tag{5.12}$$

Using the max stress level technique mentioned in section 5.2.2, the maximum stress cluster can be defined as:

$$\sigma_{max}^{PN}(x) = \left(\frac{1}{N}\sum_{a\in\Omega_{max}}\left(\sigma_1^a(x)\right)^p\right)^{\frac{1}{p}} \tag{5.13}$$

where $\sigma_1^a(x)$ is the stress value along the fibers, Ω_{max} is the cluster domain that holds the maximum stress values, and $\sigma_{max}^{PN}(x)$ is the global stress measure of maximum stresses. To decrease the number of constraints, only one stress limit will be added for stresses in both compression and tension. Hence, the absolute value of the stress value in each stress evaluation point will be calculated. To simplify the derivatives, the square root of the stress squared in each direction will be taken instead of the absolute value.

$$\sigma_1^a(x) = \sqrt{\sigma_{1a}^2} \tag{5.14}$$

The chain rule shall be used to measure the derivative of the clustered P-norm stresses described in equation (5.13) w.r.t. the design variables vector $\boldsymbol{x}$:

$$\begin{aligned}\frac{\partial \sigma_{max}^{PN}(\boldsymbol{x})}{\partial \mathbf{x}} &= \sum_{a\in\Omega_{max}}\frac{\partial \sigma_{max}^{PN}(\boldsymbol{x})}{\partial \sigma_1^a}\frac{\partial \sigma_1^a(\boldsymbol{x})}{\partial \mathbf{x}} \\ &= \sum_{a\in\Omega_{max}}\frac{\partial \sigma_{max}^{PN}(\boldsymbol{x})}{\partial \sigma_1^a}\left(\frac{\partial \sigma_1^a(\boldsymbol{x})}{\partial \sigma_a}\right)^T\frac{\partial \sigma_a(\boldsymbol{x})}{\partial \mathbf{x}}\end{aligned} \tag{5.15}$$

The equation (5.15) is made up of three terms, the first of which is the derivative of the

equation (5.13) w.r.t. the stress evaluation point:

$$\begin{aligned}\frac{\partial \sigma_{max}^{PN}(\boldsymbol{x})}{\partial \sigma_1^a} =& \frac{1}{p}\left(\frac{1}{N}\sum_{a\in\Omega_{max}}(\sigma_1^a(\boldsymbol{x}))^p\right)^{\left(\frac{1}{p}-1\right)} \\ & \times \frac{1}{N}p\left(\sigma_1^a(\boldsymbol{x})\right)^{p-1} \\ =& \left(\frac{1}{N}\sum_{a\in\Omega_{max}}(\sigma_1^a(\boldsymbol{x}))^p\right)^{\left(\frac{1}{p}-1\right)} \\ & \times \frac{1}{N}\left(\sigma_1^a(\boldsymbol{x})\right)^{p-1}\end{aligned} \tag{5.16}$$

The second term is the derivative of the stress vector at the measurement point w.r.t. the stress components:

$$\frac{\partial \sigma_1^a(x)}{\partial \sigma_a} = (\sigma_{1a}/\sqrt{\sigma_{1a}^2}, 0, 0) \tag{5.17}$$

$$\frac{\partial \sigma_2^a(x)}{\partial \sigma_a} = (0, \sigma_{2a}/\sqrt{\sigma_{2a}^2}, 0) \tag{5.18}$$

The third term is the derivative of the penalized stress vector (5.5) w.r.t. the design variables. To measure it, the design variables vector $\boldsymbol{x}$ will be divided into two vectors, with one consisting of the element densities $\boldsymbol{\rho}$ and the other consisting of the element angles $\boldsymbol{\theta}$.

$$\frac{\partial \sigma_a(\boldsymbol{x})}{\partial \rho_e} = \frac{\partial \eta_S\left(\rho_e(\boldsymbol{x})\right)}{\partial \rho_e}\boldsymbol{D}\boldsymbol{B}_a\boldsymbol{u}(\boldsymbol{x}) + \eta_S\left(\rho_e(\boldsymbol{x})\right)\boldsymbol{D}\boldsymbol{B}_a\frac{\partial \boldsymbol{u}(\boldsymbol{x})}{\partial \rho_e} \tag{5.19}$$

$$\frac{\partial \sigma_a(\boldsymbol{x})}{\partial \theta_e} = \eta_S\left(\rho_e(\boldsymbol{x})\right)\frac{\partial D}{\partial \theta_e}\boldsymbol{B}_a\boldsymbol{u}(\boldsymbol{x}) + \eta_S\left(\rho_e(\boldsymbol{x})\right)\boldsymbol{D}\boldsymbol{B}_a\frac{\partial \boldsymbol{u}(\boldsymbol{x})}{\partial \theta_e} \tag{5.20}$$

The derivative of the stress vector exists only at the stress evaluation point taking the derivative w.r.t its design variables, i.e., $\partial \eta_S\left(\rho_e(\boldsymbol{x})\right)/\partial \rho_r \neq 0$ and $\partial E/\partial \theta r \neq 0$ only for

$r = e$. The global state equation is used to determine the derivative of displacements:

$$\sum_{r=1}^{n_e} \frac{\partial \boldsymbol{K}(\rho,\theta)}{\partial \rho_r}\boldsymbol{u}(\boldsymbol{x}) + \boldsymbol{K}(\rho,\theta)\frac{\partial \boldsymbol{u}(\boldsymbol{x})}{\partial \rho_e} = 0 \tag{5.21}$$

$$\frac{\partial \boldsymbol{u}(\boldsymbol{x})}{\partial \rho_e} = -\boldsymbol{K}^{-1}(\rho,\theta)\left[\sum_{r=1}^{n_e} \frac{\partial \boldsymbol{K}(\rho,\theta)}{\partial \rho_r}\boldsymbol{u}(\boldsymbol{x})\right] \tag{5.22}$$

Similarly,

$$\sum_{r=1}^{n_e} \frac{\partial \boldsymbol{K}(\rho,\theta)}{\partial \theta_r}\boldsymbol{u}(\boldsymbol{x}) + \boldsymbol{K}(\rho,\theta)\frac{\partial \boldsymbol{u}(\boldsymbol{x})}{\partial \theta_e} = 0 \tag{5.23}$$

$$\frac{\partial \boldsymbol{u}(\boldsymbol{x})}{\partial \theta_e} = -\boldsymbol{K}^{-1}(\rho,\theta)\left[\sum_{r=1}^{n_e} \frac{\partial \boldsymbol{K}(\rho,\theta)}{\partial \theta_r}\boldsymbol{u}(\boldsymbol{x})\right] \tag{5.24}$$

The derivative in equation (5.15) w.r.t. ρ and θ can now be written by back substitution as:

$$\begin{aligned}\frac{\partial \sigma_{max}^{PN}(x)}{\partial \rho_e} = \sum_{a\in\Omega_{max}} \frac{\partial \sigma_{max}^{PN}(\boldsymbol{x})}{\partial \sigma_1^a}\left(\sigma_{1a}/\sqrt{\sigma_{1a}^2}, 0, 0\right)^T \times \Bigg(\frac{\partial \eta_S\left(\rho_e(\boldsymbol{x})\right)}{\partial \rho_e}\boldsymbol{D}\boldsymbol{B}_a\boldsymbol{u}(\boldsymbol{x}) \\ - \eta_S\left(\rho_e(\boldsymbol{x})\right)\boldsymbol{D}\boldsymbol{B}_a\boldsymbol{K}^{-1}(\rho,\theta) \times \left[\sum_{r=1}^{n_e} \frac{\partial \boldsymbol{K}(\rho,\theta)}{\partial \rho_r}\boldsymbol{u}(\boldsymbol{x})\right]\Bigg)\end{aligned} \tag{5.25}$$

$$\begin{aligned}\frac{\partial \sigma_{max}^{PN}(x)}{\partial \theta_e} = \sum_{a\in\Omega_{max}} \frac{\partial \sigma_{max}^{PN}(\boldsymbol{x})}{\partial \sigma_1^a}\left(\sigma_{1a}/\sqrt{\sigma_{1a}^2}, 0, 0\right)^T \times \Bigg(\eta_S\left(\rho_e(\boldsymbol{x})\right)\frac{\partial \boldsymbol{D}}{\partial \theta_e}\boldsymbol{B}_a\boldsymbol{u}(\boldsymbol{x}) \\ - \eta_S\left(\rho_e(\boldsymbol{x})\right)\boldsymbol{D}\boldsymbol{B}_a\boldsymbol{K}^{-1}(\rho,\theta) \times \left[\sum_{r=1}^{n_e} \frac{\partial \boldsymbol{K}(\rho,\theta)}{\partial \theta_r}\boldsymbol{u}(\boldsymbol{x})\right]\Bigg)\end{aligned} \tag{5.26}$$

The adjoint method is used in this step in order to avoid calculating the inverse of the stiffness matrix n times. Instead, it will be calculated only once, as mentioned in section 3.2.2. The adjoint function is mathematically defined as follows:

$$\boldsymbol{K}(\rho,\theta)\boldsymbol{\lambda} = \sum_{a\in\Omega_{max}} \frac{\partial\sigma_{max}^{PN}(\boldsymbol{x})}{\partial\sigma_1^a}\boldsymbol{B}_a^T\boldsymbol{D}^T\left(\sigma_{1a}/\sqrt{\sigma_{1a}^2},0,0\right) \tag{5.27}$$

The adjoint variable is the same whether the derivative is taken w.r.t. ρ or θ and can be computed as:

$$\lambda^T = \sum_{a\in\Omega_{max}} \frac{\partial\sigma_{max}^{PN}(\boldsymbol{x})}{\partial\sigma_1^a}\left(\sigma_{1a}/\sqrt{\sigma_{1a}^2},0,0\right)^T\boldsymbol{D}\boldsymbol{B}_a\boldsymbol{K}^{-1}(\rho,\theta) \tag{5.28}$$

Eventually, equation (5.15) will be:

$$\begin{aligned}\frac{\partial\sigma_{max}^{PN}(\boldsymbol{x})}{\partial\rho_e} =& \sum_{a\in\Omega_i} \frac{\partial\sigma_{max}^{PN}(\boldsymbol{x})}{\partial\sigma_1^a}\left(\sigma_{1a}/\sqrt{\sigma_{1a}^2},0,0\right)^T \times \frac{\partial\eta_S\left(\rho_e(\boldsymbol{x})\right)}{\partial\rho_e}\boldsymbol{D}\boldsymbol{B}_a\boldsymbol{u}(\boldsymbol{x}) \\ &- \eta_S\left(\rho_e(\boldsymbol{x})\right)\boldsymbol{\lambda}^T \times \left[\sum_{r=1}^{n_e}\frac{\partial\boldsymbol{K}(\rho,\theta)}{\partial\rho_r}\boldsymbol{u}(\boldsymbol{x})\right]\end{aligned} \tag{5.29}$$

$$\begin{aligned}\frac{\partial\sigma_{max}^{PN}(\boldsymbol{x})}{\partial\theta_e} =& \sum_{a\in\Omega_{max}} \frac{\partial\sigma_{max}^{PN}(\boldsymbol{x})}{\partial\sigma_1^a}\left(\sigma_{1a}/\sqrt{\sigma_{1a}^2},0,0\right)^T \times \eta_S\left(\rho_e(\boldsymbol{x})\right)\frac{\partial\boldsymbol{D}}{\partial\theta_e}\boldsymbol{B}_a\boldsymbol{u}(\boldsymbol{x}) \\ &- \eta_S\left(\rho_e(\boldsymbol{x})\right)\boldsymbol{\lambda}^T \times \left[\sum_{r=1}^{n_e}\frac{\partial\boldsymbol{K}(\rho,\theta)}{\partial\theta_r}\boldsymbol{u}(\boldsymbol{x})\right]\end{aligned} \tag{5.30}$$

5.4 Modelling limitations

This section covers two algorithm limitations. The first is how initial discretization affects the results, and the second is how to deal with stress sparks in certain elements. Because of the locality of stresses, a mesh dependency test is required before the results can be relied upon. Mesh dependency testing entails decreasing the element size until the results are no longer dependent on the mesh size. If the elements are large, a significant number of intermediate elements will result with stress values less than the stress limit.

Using small elements, on the other hand, will result in solid structure elements.

Another issue rises when modelling these types of problems is that some elements in the vicinity of load application can have a stress spark, which is not always the case in real applications because load is not always concentrated at a single point. This can be resolved by distributing the load across multiple nodes. Another recommendation is to avoid adding elements close to the load application point to the clusters when calculating the P-norm value. This has no effect on the outcome because the elements in the loading region represent a very small proportion of the total mesh elements as shown in Figure 5.1.

Finally, it is usually preferable to normalize the stresses before assigning them to the MMA to avoid numerical errors during the optimization process, which also occurs when assigning all the P-norm values as constraints to the MMA.

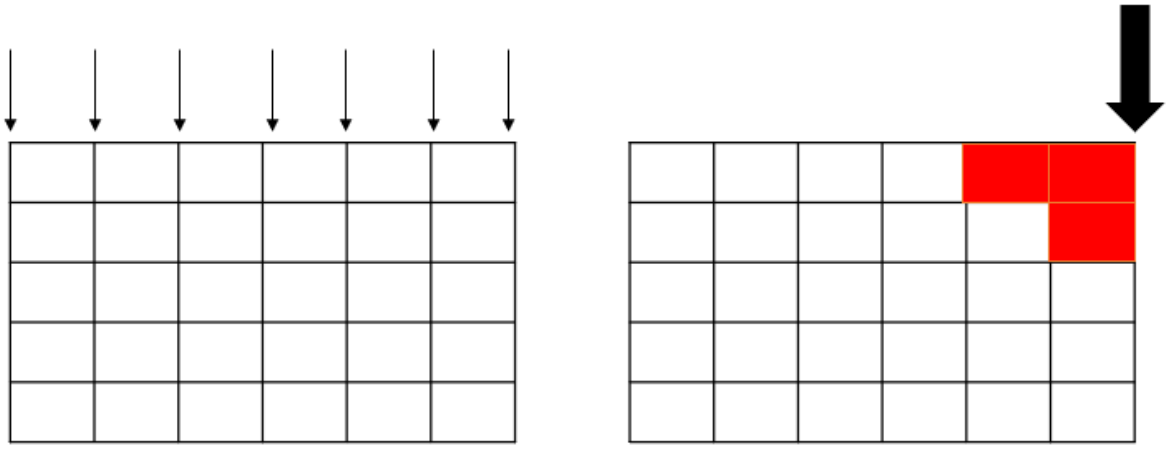


Figure 5.1: Difference between adding a load at one point and distributing the load.

5.5 Numerical examples

5.5.1 Case study 1: Verification of the method

The first case study demonstrates the effectiveness of the stress constraints and how they can affect the final topology. A compliance minimization problem subjected to volume constraint is solved twice, once without adding the stress constraints and then with adding the stress constraints. The second time, the stress constraints are set to a value lower than the maximum stresses calculated in the first case. Mathematically, the first case can be written as equation (4.1) while setting $V_f = 0.25$. The second case can be represented as equation (5.1) while setting $V_f = 0.25$, $(\sigma_1^T)_{ult} = (\sigma_1^C)_{ult} = 60$ kPa and $(\sigma_2^T)_{ult} = (\sigma_2^C)_{ult} = 25$ kPa. Both problems start from the same initial conditions $\rho = 1$ and $\theta = 1$. The material is set to epoxy glass, and the geometry is the same as the one shown in Figure 4.1; however, dimensions are in (m.) and the load value is changed to 60 kN distributed along six consecutive nodes, with 10 kN per node, as shown in Figure 5.2.

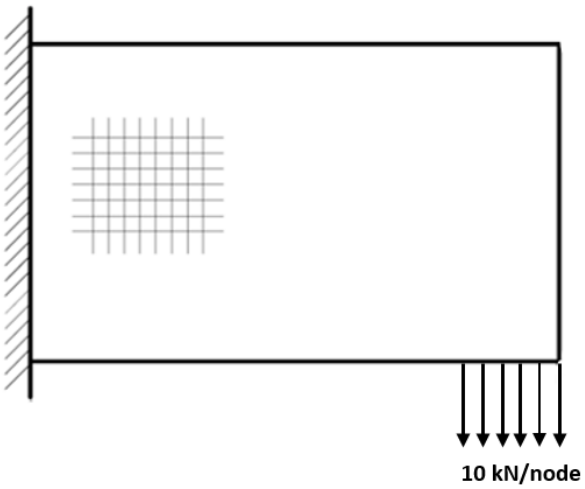
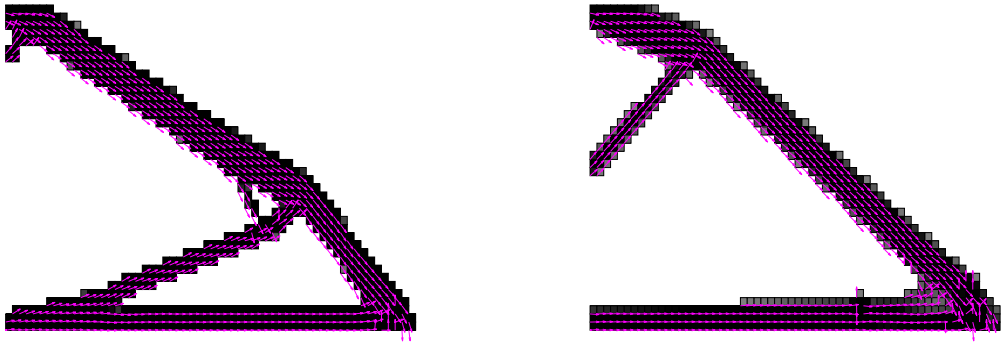


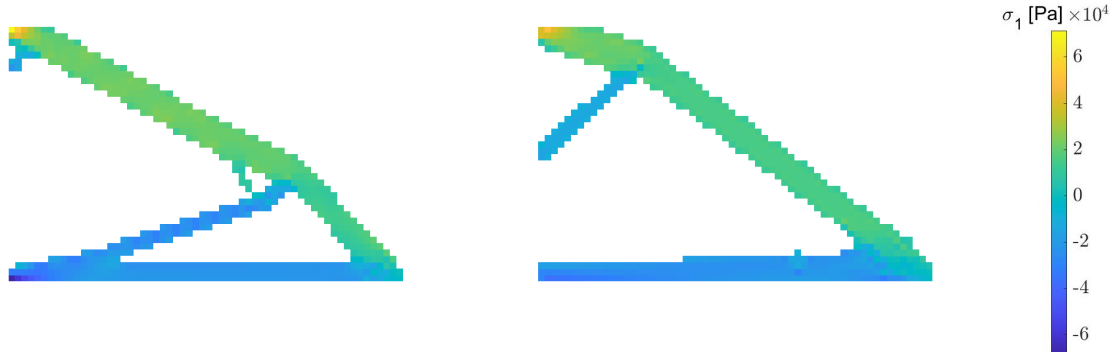
Figure 5.2: Cantilever beam fully fixed from the left and a distributed load on six consecutive nodes in the bottom-right corner.

Both problems showed that fibers should be aligned with the structure contours to obtain minimum compliance, as shown in Figure 5.3. The results show a compliance of 8.59 Nm for the volume only constrained problem, with a maximum stress value along the fibers of 71 kPa in tension and 68 kPa in compression. For stresses perpendicular to the fibers (bonding strength), the stress values reached 28 kPa in tension and 11.7 kPa in compression. However, after adding the stress constraint, a new topology was designed by the software, which yielded a compliance value of 8.226 Nm and stresses that fell within the constraints. In reviewing the new topology, it is clear that the position of the supporting member in the middle of the domain changed from the bottom right to the top-left corner to support the members that had the highest stress values on the left. Hence, the stresses were lowered to 50.5 and 37 kPa in tension and compression, respectively, along the fibers. For the stress values perpendicular to the fibers, they were reduced to 28 and 13 kPa in tension and compression, respectively. Figures 5.4 and 5.5 illustrate the stress distribution of both topologies. For this problem, the design domain was discretized to 60 elements in x and 40 elements in y, a total of 2400 elements. The stress cluster used had 240 stress evaluation points. One cluster was used for σ_1^{PN}, and one was used for σ_2^{PN}. It is noted that stresses perpendicular to fibers did not fall within the constrain limit for only one element and this errors result from using large number of stress evaluation points within the cluster. A study on the number of stress evaluation points and number of clusters is discussed in the subsequent case studies.



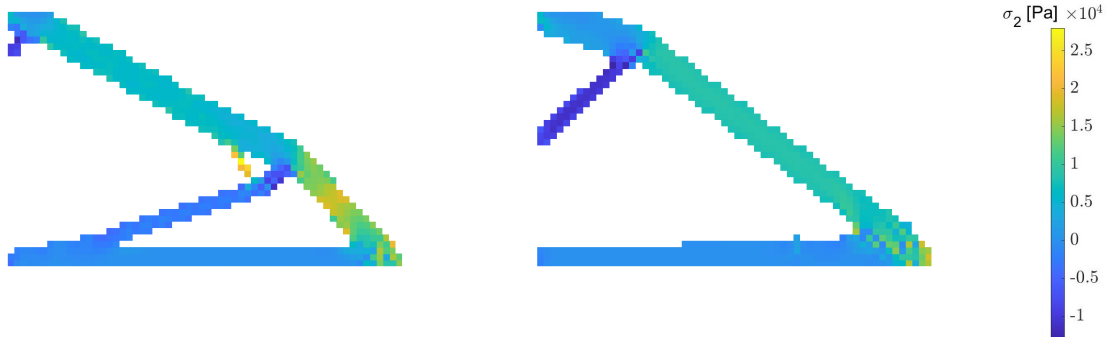
(a) Without stress constraints. (b) With stress constraints.

Figure 5.3: Optimized topology and raster angle.



(a) Without stress constraints (b) With stress constraints

Figure 5.4: Stresses along the fibers



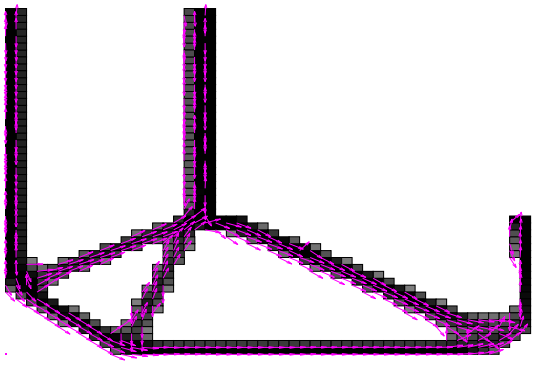
(a) Without stress constraints (b) With stress constraints

Figure 5.5: Stresses perpendicular to the fibers

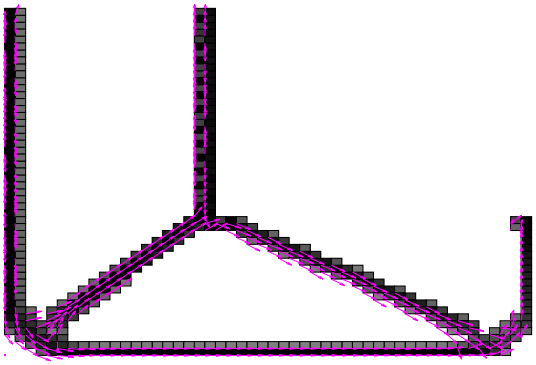
5.5.2 Case study 2: Effect of number of stress evaluation points on final topology

This Case study evaluates the effect of the number of stress evaluation points included in the cluster used as a constraint for each of the material local axes on the accuracy of the result and convergence scheme. Two studies are run: one consists of adding 80 stress evaluation points in each cluster, and the other one involves adding only 40 stress evaluation points in each cluster. The L-bracket design domain shown in Figure 4.4a is used with the same boundary and loading conditions, and the material is epoxy glass. Both problems can be represented as equation (5.1) while setting $V_f = 0.25$, $(\sigma_1^T)_{ult} = (\sigma_1^C)_{ult} = 5$ kPa and $(\sigma_2^T)_{ult} = (\sigma_2^C)_{ult} = 4$kPa. They start from the same initial conditions, where $\rho = 1$ and $\theta = -0.1$.

The results show that adding too many stress evaluation points to the cluster may decrease the efficiency of the code, as the P-norm value — and hence the constraint value — will not be a good indicator of the stress values throughout the domain, as indicated in Table 5.3. This scenario was clear in the case with 80 points, as stresses did not fall within the constraint range and some outliers existed. On the other side, when using 40 evaluation points, all the stresses fell within the constraint range as shown in Figures 5.7 and 5.8. Moreover, compliance values were 1.0583 and 1.5923 Nm for the 40 evaluation points and the 80 evaluation points, respectively. This finding confirms that the lower number of evaluation points is more efficient. The case with a higher number of evaluation points was better in one respect, which is the convergence speed, as it converged in only 3905 iterations, whereas in the former case, 6176 iterations were required, as shown in Fig. 5.9. However, accurate results, not faster convergence, is the main aim of this study.

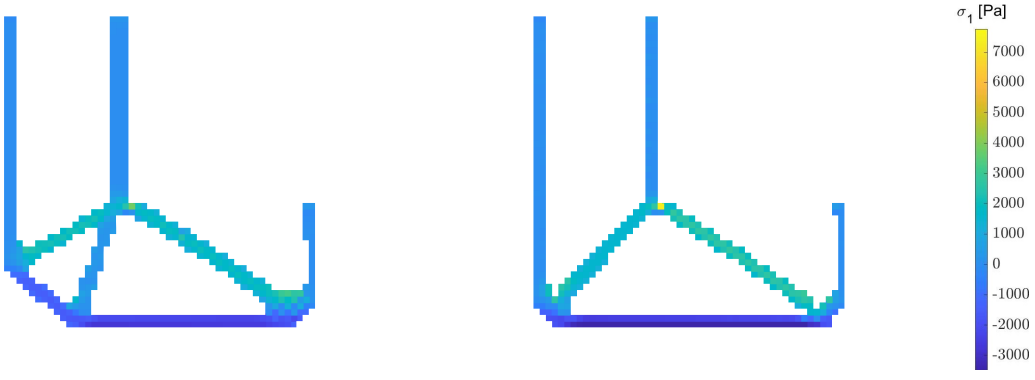


(a) 40 stress evaluation points in the stress cluster



(b) 80 stress evaluation points in the stress cluster

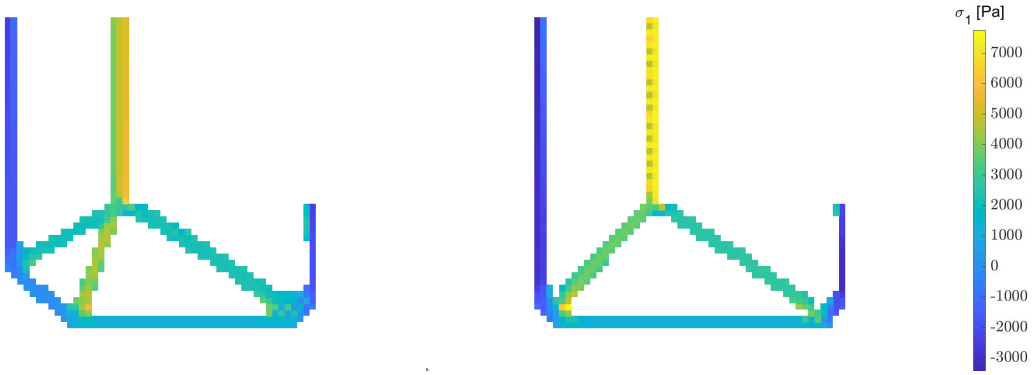
Figure 5.6: Final topology and raster angle.



(a) 40 stress evaluation points in the stress cluster

(b) 80 stress evaluation points in the stress cluster

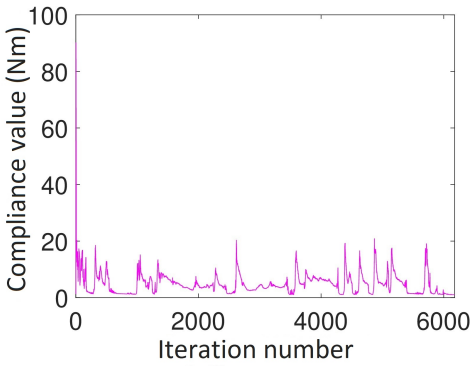
Figure 5.7: Stresses along the fibers



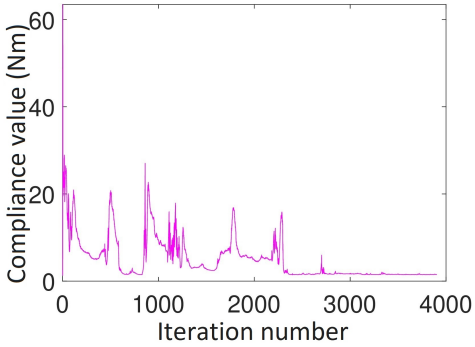
(a) 40 stress evaluation points in the stress cluster

(b) 80 stress evaluation points in the stress cluster

Figure 5.8: Stresses perpendicular to the fibers



(a) 40 stress evaluation points in the stress cluster



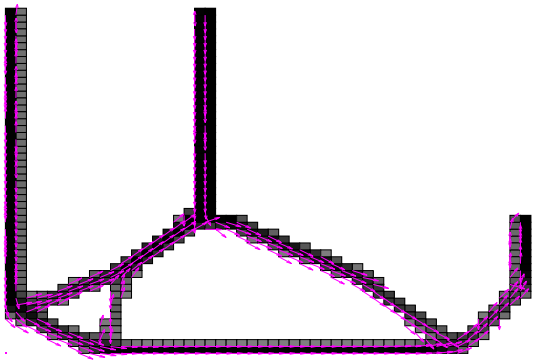
(b) 80 stress evaluation points in the stress cluster

Figure 5.9: Convergence scheme of objective function

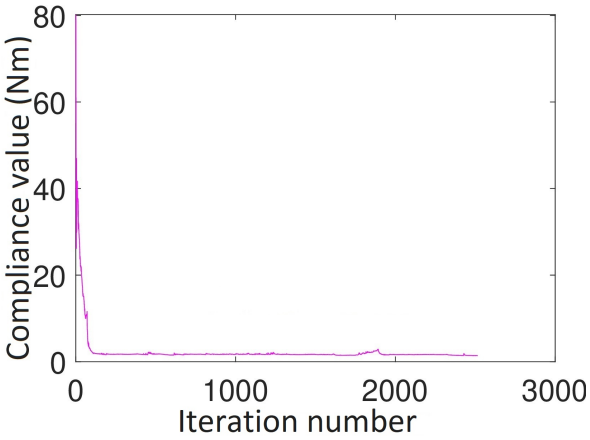
5.5.3 Case study 3: Adding one more stress cluster in each direction

The previous case study demonstrated that decreasing the number of stress evaluation points in the maximum stress cluster leads to more accurate results but also a longer convergence time, on the order of 6000 iterations. One solution to this problem is to constrain the first two clusters in each evaluation direction, instead of taking only one cluster. By doing so, the solver is constraining 80 stress evaluation points at the same time but dividing them into two clusters. That proved to be the most efficient way of dealing with a problem like the L-bracket, which is famous for producing different topologies depending on the constraints. In this case, we are going to resolve the L-bracket example from case study 2, while constraining two clusters in each direction instead of one.

The results in figure 5.10a show that this method is the most efficient, as it produced a topology with more solid elements under the loading point. Moreover, all stress values fell within the constraint limit and converged in 2515 iterations, which is a faster convergence compared with the two cases shown in case study 2. The compliance value was 1.49 Nm, and the constraints were satisfied as shown by the P-norm values in Table 5.3.



(a) Topology and raster angle.

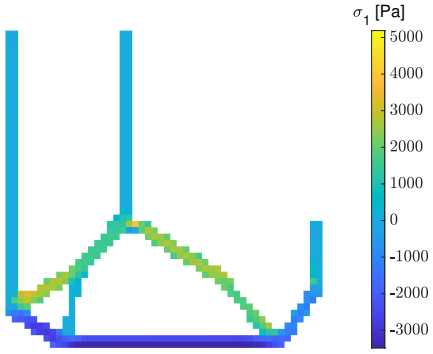


(b) Convergence scheme.

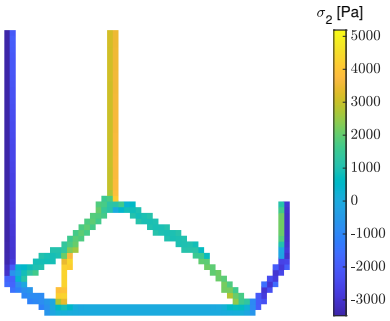
Figure 5.10: Results of L-bracket stress-constrained topology optimization when using two stress clusters in each of the material coordinate directions (final topology, fibers orientation, and convergence scheme)

No. of clusters in each direction	1	1	2
No. of stress evaluation points per cluster	40	80	40
Max. stress along the fibers (kPa)	4.5	7.4	5.2
Max. stress perpendicular to the fibers (kPa)	4.01	5.1	4.2
Compliance (Nm)	1.05	1.592	1.49
σ_1^{PN1} (kPa)	5.04	6.23	4.62
σ_2^{PN1} (kPa)	4.2	4.5	3.84
σ_1^{PN2} (kPa)	N/A	N/A	3.86
σ_2^{PN2} (kPa)	N/A	N/A	3.58
No. of iterations	6176	3905	2515

Table 5.3: Comparing different stress clustering techniques.



(a) Stresses along the fibers.



(b) Stresses perpendicular to the fibers.

Figure 5.11: Results of L-bracket stress-constrained topology optimization when using two stress clusters in each of the material coordinate directions (stress values)

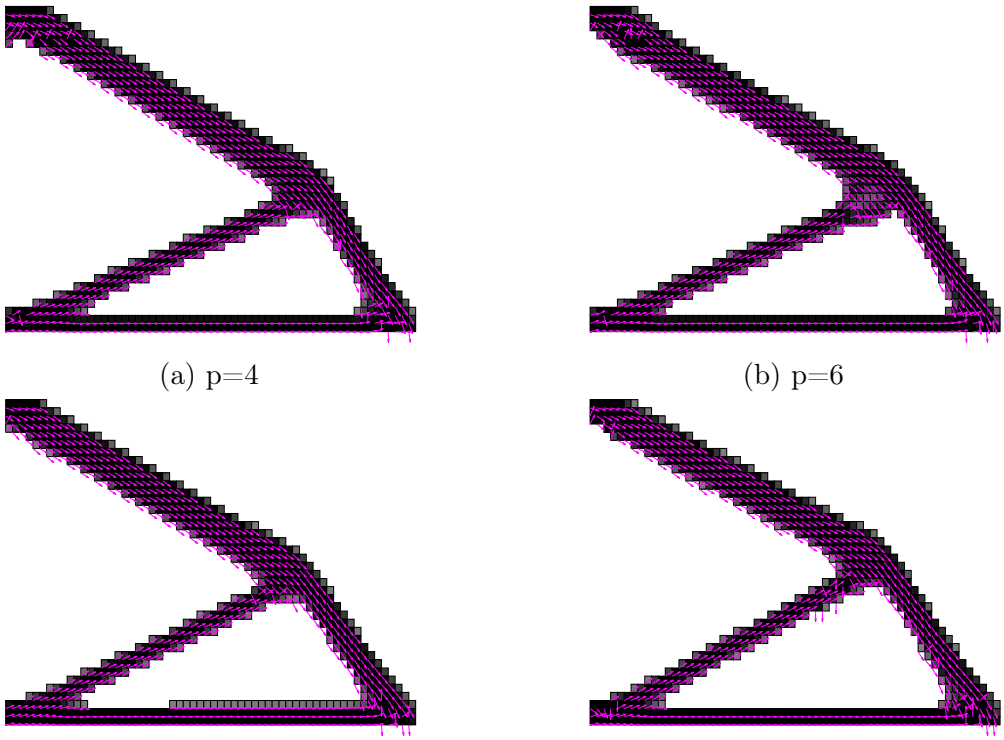
5.5.4 Case study 4: Effect of P-values on the stress results

The P-norm method is so called because the arithmetic series of equation (5.13) is raised to the power of p [57]. This number is chosen based on trial and error. From the literature, it is usually set to $p = 8$ [57]. However, that value is for an isotropic material and constrains all the stress clusters. Moreover, none of the previous studies showed the basis of choosing $p = 8$. Case study 4 tests the effect of the P-value on stresses generated in the principal material coordinates of an anisotropic material, namely epoxy glass. This study was performed on the same geometry, constraints, boundary, and loading conditions as those used in case study 1.

P-value	4	6	8	10
Max. stress along the fibers (kPa)	73	61.8	59.6	80.2
Max. stress perpendicular to the fibers (kPa)	34.3	27.3	23	22.2
Compliance (Nm)	7.93	7.823	7.62	7.749
No. of iterations	156	103	109	143
No. of stress evaluation points per cluster	120	120	120	120

Table 5.4: Effect of P-value on the efficiency of the results of stress-constrained problem with $X = 60$ kPa and $Y = 25$ kPa.

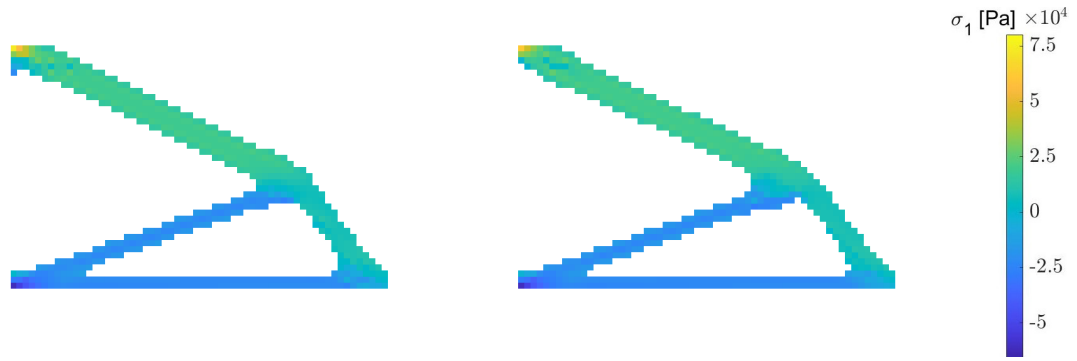
In observing the results shown in Table 5.4, it is clear that $p = 8$ is also the best value for the method used in this study when applied on an anisotropic material. This method resulted in the lowest compliance and stress values that fell within the stress constraints with no outliers. The same topology and raster angle were formed in all cases; however, the intermediate elements are not the same. In observing stresses in both directions, for P-values other than 8, one can find that stress concentrations exist in the generated topologies. However, when there are elements that have stress concentrations with values more than the stress constraint, the number of elements does not exceed 10 in each case.



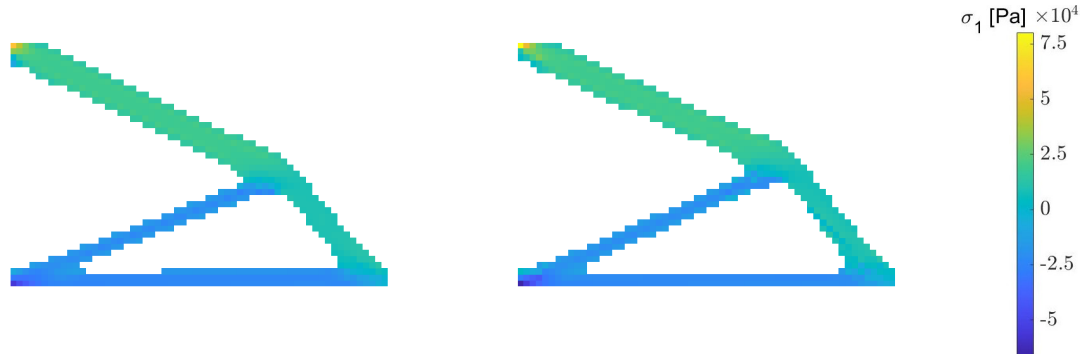
(a) p=4 (b) p=6

(c) p=8 (d) p=10

Figure 5.12: Effect of P-value on topology and raster angle.

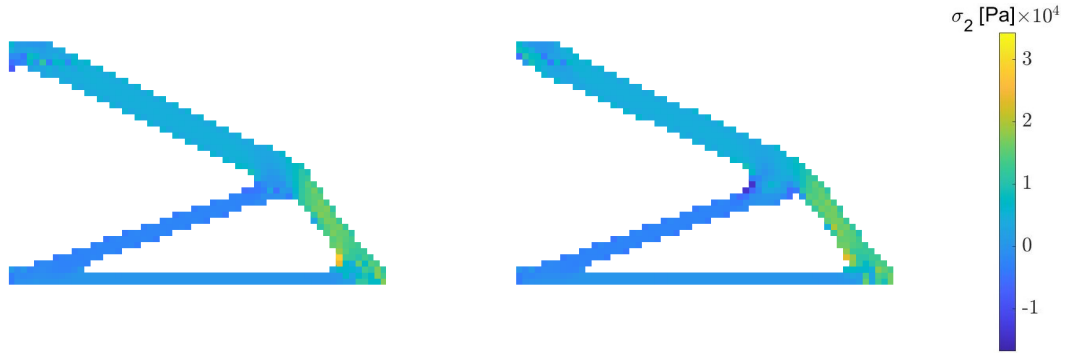


(a) P = 4 (b) P = 6

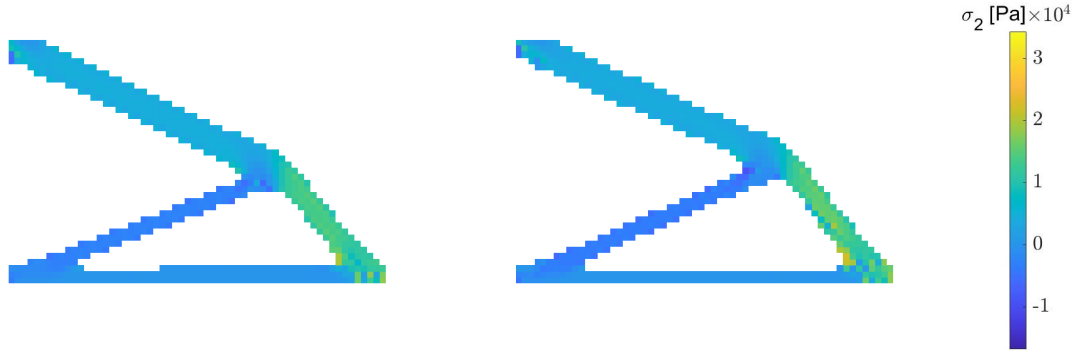


(d) P = 8 (e) P = 10

Figure 5.13: Effect of P-value on stresses along the fibers



(a) P = 4 (b) P = 6



(d) P = 8 (e) P = 10

Figure 5.14: Effect of P-value on stresses perpendicular to the fibers

5.6 Conclusions

This chapter investigated the method of adding stress constraints to the topology and raster angle optimization problem. Constraining all the stress clusters is no longer necessary, since it causes numerical errors, inflates the compliance values. Constraining only one or two clusters of stresses in each of the principal material coordinates is sufficient to keep the stresses within the stress limit when using the stress level technique. This method resulted in structures that were not seen in the field of TO, since this is the first time that stress limitations were applied to a composite material. Using two clusters in each direction with a low number of stress evaluation points is better than using one cluster with a high number of stress evaluation points. The P-norm exponent value of 8 was the optimal value to use for this type of problem.

Chapter 6

CONCLUSIONS AND FUTURE WORK

This chapter summarizes the problem, objectives, contributions of this work to the field, and potential future work.

6.1 Statement of the problem

Despite the numerous advantages of AM, anisotropy in the material characteristics of 3D printed components remains a major concern among manufacturers, as this property controls the mechanical properties of the final part. The main processing parameters that affect the mechanical properties of 3D printed parts are layer thickness, filament width and orientation, and air gap. In addition to anistropy, manufacturers are also greatly concerned with raster angle while printing, as it affects the stiffness of the final product. Furthermore, TO is an important step that must be completed before 3D printing the parts in order to maximize the material savings associated with mass manufacturing. However, each fibre orientation configuration will have a distinctive topology result from

TO, and each TO design will yield a unique fibre orientation. There is no commercial tools that are capable of addressing these challenges. Therefore, it is important to develop a tool that would develop the printing strategies. The research question that motivated this work is therefore as follows: How can the structural topology and raster orientation optimization problems for AM process be coupled into a single problem that is subject to local and global constraints?

6.2 Objectives

The objective of this work was to improve the design process of 3D printed polymers by combining the topology and raster orientation optimization process in one step. Special consideration was given to the following:

1. Derive the isoparametric Q4 element constitutive matrix, with the fiber angle **theta** defined as a design variable.

2. Develop a robust and generic in-house code that solves the compliance minimization problem, subject to volume constraints, where the produced topologies are a good start in the conceptual design process for AM.

3. Solve mass minimization problems, subject to compliance constraints.

4. Derive of stress constraints for AM polymers, both along the fibers and perpendicular to the fibers.

5. Solve compliance minimization problems, subject to stress constraints, to reduce the formation of stress concentration regions in the final topology.

6.3 General conclusions

6.3.1 Adding the fiber angle as a design variable to the TO problem

Transformation matrices are very useful when dealing with structures that need to be placed in a specific orientation. They give the finite element the freedom to mimic any object that can rotate, be it a fiber, truss, or beam element. After optimization, the finite elements align themselves along the contours of the structural element, especially if the element is carrying pure tension or pure compression. This shows how effective this procedure is to use the element transformation angles in representing physical fiber angles.

6.3.2 Mass savings and stiffness increase improvements

Derivation of the mass minimization problem of polymers is usually straightforward if the compliance minimization problem is solved. By incorporating fiber angles into the optimization problem, more mass is reduced by correctly orienting the fibers, while compliance is limited to a certain value. This demonstrates the great future potential of using this method in the AM industry.

6.3.3 Direction-dependant stress constraints

Adding stress constraints to TO problems is challenging because of the local nature of the stresses. In addition, fiber stresses should be controlled in two directions, not just one. Stress aggregation approach was adopted here. However, instead of aggregating stresses or "clustering stresses" in two directions and using all the clusters as constraints, the minmax approach was used, meaning that only the clusters that had the highest

stress values in both directions were used. This method resulted in novel structures in the field of TO, as this is the first time that stress constraints have been added to a composite material. It is preferable to use two clusters in each direction with a modest number of stress evaluation points rather than one cluster with a high number of stress evaluation points. A P-norm exponent value of 8 is best to use in this scenario because it produced the best outcomes.

6.3.4 General comments on code performance

Algorithm convergence for non-stress-constrained problems was stable, with most of the compliance minimization problems converging in 100–250 iterations, and mass minimization problems converging in 1000–1500 iterations. However, for the stress-constrained problems, convergence was highly dependant on the stress constraint value. If the constraint is a value that is not easily achieved at the defined loading condition, the code requires more iterations to converge, that may reach 6000 iterations if only one cluster with a high number of stress evaluation points was used. If stress constraints are easily achieved, then the code typically converges in 100–300 iterations depending on the problem geometry.

6.4 Thesis main contributions

The main research contributions of this work are:

1. Developed mass minimization algorithms for AM orthotropic structures.

2. Derived and implemented directed stress constraints approach which aids in limiting maximum stress values to prevent composite failure during the optimization process.

3. Employed maximum clustering technique in principal material directions, to improve convergence.

4. Applied those developed algorithms to several numerical examples.

6.5 Future work

The following areas are worthy of further research:

1. Extend the code to incorporate other constraints for example shear stress.

2. Explore the stress calculations at the nodes and perform sensitivity study.

3. Develop layer independent algorithm for multilayered structures.

4. Incorporate other failure criteria for example Tsai–Hill and tensor polynomial failure criteria.

Bibliography

[1] Lucien A Schmit. "Structural design by systematic synthesis". In: *Proceedings of the Second National Conference on Electronic Computation, ASCE, Sept., 1960*. 1960.

[2] Martin Philip Bendsøe and Noboru Kikuchi. "Generating optimal topologies in structural design using a homogenization method". In: *Computer methods in applied mechanics and engineering* 71.2 (1988), pp. 197–224.

[3] Tuan D Ngo et al. "Additive manufacturing (3D printing): A review of materials, methods, applications and challenges". In: *Composites Part B: Engineering* 143 (2018), pp. 172–196.

[4] Peng Wu, Jun Wang, and Xiangyu Wang. "A critical review of the use of 3-D printing in the construction industry". In: *Automation in Construction* 68 (2016), pp. 21–31.

[5] Jeffrey W Stansbury and Mike J Idacavage. "3D printing with polymers: Challenges among expanding options and opportunities". In: *Dental materials* 32.1 (2016), pp. 54–64.

[6] Pedram Parandoush and Dong Lin. "A review on additive manufacturing of polymer-fiber composites". In: *Composite Structures* 182 (2017), pp. 36–53.

[7] Xin Wang et al. “3D printing of polymer matrix composites: A review and prospective”. In: *Composites Part B: Engineering* 110 (2017), pp. 442–458.

[8] Jaroslaw Kotlinski. “Mechanical properties of commercial rapid prototyping materials”. In: *Rapid Prototyping Journal* (2014).

[9] Delin Jiang. “Three dimensional topology optimization with orthotropic material orientation design for additive manufacturing structures.” PhD thesis. 2017.

[10] Jie Gao et al. “Concurrent topology optimization of multiscale composite structures in Matlab”. In: *Structural and Multidisciplinary Optimization* 60.6 (2019), pp. 2621–2651.

[11] Hai Peng Jia et al. “Topology Optimization of Orthotropic Material Structure”. In: *Materials Science Forum*. Vol. 575. Trans Tech Publ. 2008, pp. 978–989.

[12] Martin-Pierre Schmidt et al. “Structural topology optimization with smoothly varying fiber orientations”. In: *Structural and Multidisciplinary Optimization* 62.6 (2020), pp. 3105–3126.

[13] Daryl L Logan. *A first course in the finite element method*. Nelson Education, 2017.

[14] Martin Philip Bendsoe and Ole Sigmund. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2013.

[15] Krister Svanberg. “MMA and GCMMA, versions September 2007”. In: *Optimization and Systems Theory* 104 (2007).

[16] Josephine V Carstensen. “Topology optimization with nozzle size restrictions for material extrusion-type additive manufacturing”. In: ().

[17] Carl T Herakovich. “Mechanics of fibrous composites”. In: (1998).

[18] Chau Le et al. “Stress-based topology optimization for continua”. In: *Structural and Multidisciplinary Optimization* 41.4 (2010), pp. 605–620.

[19] Terry Wohlers and Tim Gornet. “History of additive manufacturing”. In: *Wohlers report* 24.2014 (2014), p. 118.

[20] Tim Caffrey, Terry Wohlers, and Ian Campbell. *Executive summary of the Wohlers Report 2016*. 2016.

[21] Ferdinand P Beer et al. “additive manufacturing market forecast to reach $51 billion by 2030_2021”. In: *Metal Additive Manufacturing* (2021).

[22] Madhukar Somireddy. “Multiscale Material Modeling of Additively Manufactured Composite Laminates”. In: (2019).

[23] Peter W Christensen and Anders Klarbring. *An introduction to structural optimization*. Vol. 153. Springer Science & Business Media, 2008.

[24] Robin Larsson. “Methodology for topology and shape optimization: Application to a rear lower control arm”. MA thesis. 2016.

[25] K Maute and Gregory W Reich. “Integrated multidisciplinary topology optimization approach to adaptive wing design”. In: *Journal of Aircraft* 43.1 (2006), pp. 253–263.

[26] Xinhao Zhao et al. “Finite element analysis and topology optimization of a 12000KN fine blanking press frame”. In: *Structural and multidisciplinary optimization* 54.2 (2016), pp. 375–389.

[27] Lars Krog et al. “Topology optimisation of aircraft wing box ribs”. In: *10th AIAA/ISSMO multidisciplinary analysis and optimization conference*. 2004, p. 4481.

[28] Ole Sigmund and Kurt Maute. “Topology optimization approaches”. In: *Structural and Multidisciplinary Optimization* 48.6 (2013), pp. 1031–1055.

[29] Martin P Bendsøe. "Optimal shape design as a material distribution problem". In: *Structural optimization* 1.4 (1989), pp. 193–202.

[30] M Zhou and GIN Rozvany. "The COC algorithm, Part II: Topological, geometrical and generalized shape optimization". In: *Computer methods in applied mechanics and engineering* 89.1-3 (1991), pp. 309–336.

[31] HP Mlejnek. "Some aspects of the genesis of structures". In: *Structural optimization* 5.1-2 (1992), pp. 64–69.

[32] Martin P Bendsøe and Ole Sigmund. "Material interpolation schemes in topology optimization". In: *Archive of applied mechanics* 69.9 (1999), pp. 635–654.

[33] Mathias Stolpe and Krister Svanberg. "An alternative interpolation scheme for minimum compliance topology optimization". In: *Structural and Multidisciplinary Optimization* 22.2 (2001), pp. 116–124.

[34] Niels L Pedersen. "Maximization of eigenvalues using topology optimization". In: *Structural and multidisciplinary optimization* 20.1 (2000), pp. 2–11.

[35] G Allaire and GA Francfort. "A numerical algorithm for topology and shape optimization". In: *Topology design of structures*. Springer, 1993, pp. 239–248.

[36] Yi M Xie and Grant P Steven. "A simple evolutionary procedure for structural optimization". In: *Computers & structures* 49.5 (1993), pp. 885–896.

[37] Osvaldo M Querin, Grant P Steven, and Yi Min Xie. "Evolutionary structural optimisation (ESO) using a bidirectional algorithm". In: *Engineering computations* (1998).

[38] Stanley Osher and James A Sethian. "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations". In: *Journal of computational physics* 79.1 (1988), pp. 12–49.

[39] Daniël Peeters, Daniel van Baalen, and Mostafa Abdallah. "Combining topology and lamination parameter optimisation". In: *Structural and Multidisciplinary Optimization* 52.1 (2015), pp. 105–120.

[40] Liang Xia and Piotr Breitkopf. "Design of materials using topology optimization and energy-based homogenization approach in Matlab". In: *Structural and multidisciplinary optimization* 52.6 (2015), pp. 1229–1241.

[41] Jackson Sawyer Ramsey. "Topology optimization of weakly coupled thermomechanical systems for additive manufacturing." PhD thesis. 2019.

[42] K Svanberg. "The method of moving asymptotes (MMA) with some extensions". In: *Optimization of large structural systems*. Springer, 1993, pp. 555–566.

[43] Ole Sigmund. "On the usefulness of non-gradient approaches in topology optimization". In: *Structural and Multidisciplinary Optimization* 43.5 (2011), pp. 589–596.

[44] G Allaire and RV Kohn. "Topology optimization and optimal shape design using homogenization". In: *Topology design of structures*. Springer, 1993, pp. 207–218.

[45] Raphael T Haftka, Zafer Gürdal, and Manohar P Kamat. "Linear Mathematical Programming". In: *Elements of Structural Optimization*. Springer, 1990, pp. 61–98.

[46] R Timothy Marler and Jasbir S Arora. "Survey of multi-objective optimization methods for engineering". In: *Structural and multidisciplinary optimization* 26.6 (2004), pp. 369–395.

[47] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[48] Vijay Krishna Yalamanchili and Ashok V Kumar. "Topology optimization of structures using a global stress measure". In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 45011. American Society of Mechanical Engineers. 2012, pp. 1321–1328.

[49] Ole Sigmund. "A 99 line topology optimization code written in Matlab". In: *Structural and multidisciplinary optimization* 21.2 (2001), pp. 120–127.

[50] Kai Liu and Andres Tovar. "An efficient 3D topology optimization code written in Matlab". In: *Structural and Multidisciplinary Optimization* 50.6 (2014), pp. 1175–1196.

[51] Aaditya Chandrasekhar, Tej Kumar, and Krishnan Suresh. "Build optimization of fiber-reinforced additively manufactured components". In: *Structural and Multidisciplinary Optimization* 61.1 (2020), pp. 77–90.

[52] Josephine V Carstensen. "Topology optimization with nozzle size restrictions for material extrusion-type additive manufacturing". In: *Structural and Multidisciplinary Optimization* 62 (2020), pp. 2481–2497.

[53] Susana Rojas-Labanda and Mathias Stolpe. "Benchmarking optimization solvers for structural topology optimization". In: *Structural and Multidisciplinary Optimization* 52.3 (2015), pp. 527–547.

[54] Alexander Verbart. "Topology optimization with stress constraints". In: (2015).

[55] Pierre Duysinx and Ole Sigmund. "New developments in handling stress constraints in optimal material distribution". In: *7th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization*. 1998, p. 4906.

[56] Josi Paris et al. "Block aggregation of stress constraints in topology optimization of structures". In: *Advances in Engineering Software* 41.3 (2010), pp. 433–441.

[57] Erik Holmberg, Bo Torstenfelt, and Anders Klarbring. "Stress constrained topology optimization". In: *Structural and Multidisciplinary Optimization* 48.1 (2013), pp. 33–47.

[58] BY Chen et al. "Modelling the tensile failure of composites with the floating node method". In: *Computer Methods in Applied Mechanics and Engineering* 308 (2016), pp. 414–442.

[59] Gengdong Cheng and Zheng Jiang. "Study on topology optimization with stress constraints". In: *Engineering Optimization* 20.2 (1992), pp. 129–148.

[60] X Guo, G Cheng, and K Yamazaki. "A new approach for the solution of singular optima in truss topology optimization with stress and local buckling constraints". In: *Structural and Multidisciplinary Optimization* 22.5 (2001), pp. 364–373.

[61] GIN Rozvany and T Birker. "On singular topologies in exact layout optimization". In: *Structural optimization* 8.4 (1994), pp. 228–235.

[62] Uri Kirsch. "On singular topologies in optimum structural design". In: *Structural optimization* 2.3 (1990), pp. 133–142.

[63] Yongsheng Han et al. "Topology optimization of material nonlinear continuum structures under stress constraints". In: *Computer Methods in Applied Mechanics and Engineering* 378 (2021), p. 113731.

[64] Christophe Geuzaine and Jean-François Remacle. "Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities". In: *International journal for numerical methods in engineering* 79.11 (2009), pp. 1309–1331.

[65] Klaus-Jurgen Bathe. *Finite element procedures*. Klaus-Jurgen Bathe, 2006.

[66] Erik Andreassen et al. "Efficient topology optimization in MATLAB using 88 lines of code". In: *Structural and Multidisciplinary Optimization* 43.1 (2011), pp. 1–16.

[67] Ferdinand P Beer et al. "Mechanics of materials, mcgraw-hill". In: *New York* (1981), pp. 150–233.

[68] Delin Jiang, Robert Hoglund, and Douglas E Smith. "Continuous fiber angle topology optimization for polymer composite deposition additive manufacturing applications". In: *Fibers* 7.2 (2019), p. 14.

[69] Michal Kočvara and Michael Stingl. "Solving stress constrained problems in topology and material optimization". In: *Structural and Multidisciplinary Optimization* 46.1 (2012), pp. 1–15.

Appendix A

MATLAB Code for Stress Constrained TO of Orthotropic Material

---

```
% That is the main file of the code, it includes all the functions used
% Except MMA method because of confidentielity
% New variables in the field
% Sigmax --> stress constraint value along the fibers
% Sigmay --> Stress constraint value perpendicular to the fibers
% 64mm_40mm --> Is the solid works file
% clts ---> Number of clusters used in the optimization method
% dkdrho ---> derivative of stiffness w.r.t density
% dkdtheta ---> derivative of stiffness w.r.t theta
% All other variables are famous literature variables
clear;clc;close all;
global penal p E1 E2 G12 v12 Force Emin clts volfrac H Hs nelx nely sigmax
    sigmay nc maxloop nodes conn nele

clts=2; sigmax = 60e3;sigmay = 25e3; nelx=60; nely=40; nc=20; maxloop=80;
```

```
E1 = 38.6e9;E2 = 8.27e9;G12 = 4.14e9; v12 = 0.37; Force = -10000;volfrac =
    0.25; Emin = 1e-9; p = 4;penal = 3;rmin = 2;

nele=nelx*nely;[connec,nodes,elements]=Unstruct('64mm_40mm',nelx+1,nely+1
    );[nodes,conn]=arrange_nodes(nodes,nelx,nely);nodes=nodes*10^3;
[H,Hs] = fil(nelx, nely, rmin);

% Initial conditions
x = repmat(0.25,nele,1);theta=repmat(-0.1,nele,1);xPhys=(x(:));

% MMA initialization
m=1+(2*clts);n=2*nele;xmin=[zeros(n/2,1);-pi/2*ones(n/2,1)]; ...
    xmax=[ones(n/2,1);pi/2*ones(n/2,1)];xold1=x(:);xold2=x(:); ...
    low=xmin;upp=xmax;a0=1;a=zeros(m,1);c_MMA=10000*ones(m,1);d=zeros(m,1);

% Loop Initialization
change=100; loop=0;
while change > 0.001
    warning('off');
    loop = loop + 1;
    [U,edofMat,KEE,Kg,dkdrho,dkdthetaa,B] = displacements(theta,x);
    [f0val,df0dx] = Obj(U,edofMat,KEE,dkdrho,dkdthetaa,x);
    [fval,dfdx,p1,p2,sigX,sigY] =
        constraints(theta,x,U,edofMat,Kg,dkdrho,dkdthetaa,B);
    [xmma, ~, ~, ~, ~, ~, ~, ~, ~, low,upp] = ...
        mmasub(m, n, loop, [xPhys;theta(:)] , xmin, xmax, xold1, xold2, ...
        f0val,df0dx,fval,dfdx,low,upp,a0,a,c_MMA,d);
    xnew     = reshape(xmma(1:nele),nely,nelx);
```

```
    thetanew    = reshape(xmma(nele+1:end),nely,nelx);
    xold2   = xold1(:);
    xold1   = [x(:);theta(:)];
    change = max(abs(xnew(:)-x(:)));
    x=xnew;
    compp(loop)=f0val;
    theta=thetanew(:);
    xPhys= (x(:));
    fs(loop,:)=fval;
    p1s(loop)=max(p1);
    p2s(loop)=max(p2);
    fprintf(' It.:%5i Obj.:%11.4f Vol.:%7.3f max_P1.:%7.3f max_P2.:%7.3f
        change.:%7.3f\n',loop,f0val, ...
        mean(x(:)), max(p1)/(10^6),max(p2)/(10^6), change);
    colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-6);
end
plotstresses(nodes(:,2:3),conn(:,2:5),sigX,20,'stresses along fibers (pa)')
plottopology(x,theta,0.5)

function [connectivity,nodess,elements] = Unstruct(filename,n1,n2 )
% This function extracts the nodes and
%connectivity matrix by using Gmsh
%%%% Inputs: 1- Filename of your geometry file
%%%%% 2- spanwise sections
%%%%% 3- lengthwise sections
geoflat( n1,n2 );
[a,b] = system(['gmsh.exe -watch pattern ', filename,'.step geometry.geo
    -2']);
```

```
file= [filename ,'.msh'];
% no of nodes is mentioned in 5th row and first column
N_n  = dlmread(file,'',[5-1 1-1 5-1 1-1]);
nodes      = dlmread(file,'',[5 1 4+N_n 3]);
N_e = dlmread(file,'',[7+N_n 0 7+N_n 0]);
node_id = dlmread(file,'',[5 0 4+N_n 0]);
elements = dlmread(file,'',[8+N_n 0 7+N_n+N_e 8]);
if min(nodes(:,1))<0
    nodes(:,1)=nodes(:,1)-min(nodes(:,1));
end
if min(nodes(:,2))<0
    nodes(:,2)=nodes(:,2)-min(nodes(:,2));
end
nodes(:,1)=nodes(:,1)/10^3;
nodes(:,2)=nodes(:,2)/10^3;
x=0;
for i=1:length(elements(:,1))
    if elements(i,2)~=3
        x=x+1;
    end
end
u=1;
count=1;
for i=1:length(elements(:,1))
    conn(u,[2:5])=elements(i,[6:9]);
    conn(u,1)=u;
    u=u+1;
    if elements(i,2)==3
```

```
        connectivity(count,1)=count;
        connectivity(count,2:5)=elements(i,[6:9]);
        count=count+1;
    end
end
for i=1:x
    Panel_coor(i,[1:3])=[nodes(conn(i,2),1), ...
        nodes(conn(i,2),2),nodes(conn(i,2),3)];
    Panel_coor(i,[4:6])=[nodes(conn(i,3),1), ...
        nodes(conn(i,3),2),nodes(conn(i,3),3)];
    Panel_coor(i,[7:9])=[0 0 0];
    Panel_coor(i,[10:12])=[0 0 0];
    if elements(i,4)==6
        Panel_coor(i,13)=3;
    elseif elements(i,4)==7
        Panel_coor(i,13)=4;
    end
end
for i=x+1:length(conn(:,1))
    Panel_coor(i,[1:3])=[nodes(conn(i,2),1), ...
        nodes(conn(i,2),2),nodes(conn(i,2),3)];
    Panel_coor(i,[4:6])=[nodes(conn(i,3),1), ...
        nodes(conn(i,3),2),nodes(conn(i,3),3)];
    Panel_coor(i,[7:9])=[nodes(conn(i,4),1), ...
        nodes(conn(i,4),2),nodes(conn(i,4),3)];
    Panel_coor(i,[10:12])=[nodes(conn(i,5),1), ...
        nodes(conn(i,5),2),nodes(conn(i,5),3)];
    if elements(i,4)==5
```

```
        Panel_coor(i,13)=6;
    elseif elements(i,4)==4
        Panel_coor(i,13)=2;
    end
end
for u=1:length(nodes)
    nodess(u,1)=u;
    nodess(u,2:3)=nodes(u,1:2);
end
end
```

```
function [] = geoflat( n1,n2 )
%%%%% Creating the geo file for 2D structures
fid=fopen('geometry.geo','wt'); %creates a text file
if fid<0 %makes sure file open succesful
    fprintf('error opening file \n'); %if not send an error message
end
fprintf(fid,'Physical Surface(5) = {1,2};\n');
fprintf(fid,'Physical Line(6) = {6,2}; \n');
fprintf(fid,'Physical Line(7) = {8,4}; \n');
fprintf(fid,'Transfinite Surface {1,2}; \n');
fprintf(fid,'Transfinite Line {5, 3, 1} = %d Using Progression 1; \n',n1);
    %%% spanwise
fprintf(fid,'Transfinite Line {8,6,4, 2} = %d Using Progression 1; \n',n2);
    %%% chordwise
fprintf(fid,'Recombine Surface {1,2};');
```

```
end

function [k, dk, dktheta]= complete_stiff(theta,x,n1,n2,n3,n4)
% This function calculates stiffness matrix and
% its derivatives w.r.t rho and theta
global penal E1 E2 G12 v12 Emin
%penalizing
E1p = (Emin+x.^penal*(E1-Emin));
E2p = (Emin+x.^penal*(E2-Emin));
G12p= (Emin+x.^penal*G12);
dE1p = (-penal*x^(penal-1))*E1;
dE2p = (-penal*x^(penal-1))*E2;
dG12p= (-penal*x^(penal-1))*G12;
[D, dD_dtheta, D_zero] = Rotation(theta, E1p, E2p, G12p, v12);
[dD, dddD_dtheta, ddD_zero] = Rotation(theta, dE1p, dE2p, dG12p, v12);
%syms theta
W=[1 1 1 1];
%thickness
h=1;
s1=-0.5773; s2=-0.5773; s3=0.5773; s4=0.5773; t1=-0.5773; t2= 0.5773;...
    t3=0.5773; t4=-0.5773;
s=[s1 s2 s3 s4];t=[t1 t2 t3 t4];
% element nodes
Xc=[n1(1);n2(1);n3(1);n4(1)];Yc=[n1(2);n2(2);n3(2);n4(2)];
%initiating stiffness matrix
k=zeros(8,8);
dk=zeros(8,8);
dktheta=zeros(8,8);
```

```
% k1=zeros(8,8);
for i=1:4
    j=i;
    %%% Jacobian
    J=(1/8)*Xc'*[0 1-t(j) t(j)-s(i) s(i)-1; t(j)-1 ...
        0 s(i)+1 -s(i)-t(j); s(i)-t(j) -s(i)-1 0 t(j)+1 ...
        ; 1-s(i) s(i)+t(j) -t(j)-1 0]*Yc;
%     % Shape functions
%     N1=(1-s(i))*(1-t(j))/4;
%     N2=(1+s(i))*(1-t(j))/4;
%     N3=(1+s(i))*(1+t(j))/4;
%     N4=(1-s(i))*(1+t(j))/4;
    % Derivatives of shape function
    N1s=(1/4)*(t(j)-1);
    N1t=(1/4)*(s(i)-1);
    N2s=(1/4)*(-t(j)+1);
    N2t=(1/4)*(-s(i)-1);
    N3s=(1/4)*(t(j)+1);
    N3t=(1/4)*(s(i)+1);
    N4s=(1/4)*(-t(j)-1);
    N4t=(1/4)*(-s(i)+1);
    Ns=[N1s N2s N3s N4s];
    Nt=[N1t N2t N3t N4t];
    % B matrix coefficients
    a= (1/4)*((Yc(1)*(s(i)-1))+(Yc(2)*(-s(i)-1)) ...
        +(Yc(3)*(s(i)+1))+(Yc(4)*(-s(i)+1)));
    b= (1/4)*((Yc(1)*(t(j)-1))+(Yc(2)*(-t(j)+1)) ...
        +(Yc(3)*(t(j)+1))+(Yc(4)*(-t(j)-1)));
```

```
    c= (1/4)*((Xc(1)*(t(j)-1))+(Xc(2)*(-t(j)+1)) ...
        +(Xc(3)*(t(j)+1))+(Xc(4)*(-t(j)-1)));
    d= (1/4)*((Xc(1)*(s(i)-1))+(Xc(2)*(-s(i)-1)) ...
        +(Xc(3)*(s(i)+1))+(Xc(4)*(-s(i)+1)));
    for z=1:4
        Bz(:,:,z)=[a*Ns(z)-b*Nt(z) 0 ...
            ;0 c*Nt(z)-d*Ns(z); c*Nt(z)-d*Ns(z) a*Ns(z)-b*Nt(z)];
    end
    B=(1/det(J))*[Bz(:,:,1) Bz(:,:,2) Bz(:,:,3) Bz(:,:,4)];
    k=k+B'*D*B*J*h*W(i)*W(i);
    dk=dk+B'*dD*B*J*h*W(i)*W(i);
    dktheta=dktheta+B'*dD_dtheta*B*J*h*W(i)*W(i);
end
end

function [f,dfdx,p1,p2,sigX,sigY] =
    constraints(theta,x,U,edofMat,Kg,dkdrho,dkdthetaa,B)
% This function calculates the constraints values and its derivatives
global nele p E1 E2 G12 v12 volfrac nc sigmax sigmay clts
Ni=2*nele/nc; %number of stress evaluation points
for i=1:nele
    [C(:,:,i) dCdth(:,:,i)] = Rotation(theta(i), E1, E2, G12, v12);
    stress(i,:) = C(:,:,i)*B*U(edofMat(i,:));
    stress_val(i,:) = x(i)^0.5*stress(i,:);
    sigX(i)=stress_val(i,1);
    sigY(i)=stress_val(i,2);
    p1(i) = sqrt(stress_val(i,1)^2);
    p2(i) = sqrt(stress_val(i,2)^2);
```

```
    von_mises(i,1) = sqrt((stress_val(i,1)^2)+...
       (stress_val(i,2)^2)+(3*stress_val(i,3)^2)...
       -stress_val(i,1)*stress_val(i,2));
end
[v1 i1]=max(p1);
p1(i1)=0;
[v2 i2]=max(p1);
p1(i2)=0;
[v3 i3]=max(p2);
p2(i3)=0;
[v4 i4]=max(p2);
p2(i4)=0;
% constraints calculation
[p1_desc,sort_index] = sort(p1,'descend');
cluster = zeros(nc/2,Ni);
cluster = reshape(p1_desc,Ni,(nc/2));
cluster_p = (cluster).^p;
cluster_sum = sum(cluster_p,1);
sigmapn = (cluster_sum./Ni).^(1/p);
[p2_desc,sort_indexx] = sort(p2,'descend');
clusterr = zeros(nc/2,Ni);
clusterr = reshape(p2_desc,Ni,(nc/2));
clusterr_p = (clusterr).^p;
clusterr_sum = sum(clusterr_p,1);
sigmapnn= (clusterr_sum./Ni).^(1/p);
f=[(sum(x(:))/(volfrac*nele))-1; ((sigmapn(1:clts)'./sigmax) ...
    -(ones(clts,1))); ((sigmapnn(1:clts)'./sigmay)-(ones(clts,1)))];
dv = ones(nele,1);
```

```
dv_dtheta= zeros(nele,1);

dv(:) = dv(:);

dpndx1=zeros(1,nele);

dpndtheta1=zeros(1,nele);

dppndx1=zeros(1,nele);

dppndtheta1=zeros(1,nele);

count=1;

Kinv = inv(Kg);

for i=1:clts

    for j=1:(2*nele/nc)

        dpndx1(i,sort_index(count))= ((cluster_sum(i)/Ni)^((1/p)-1))*(1/Ni)*

            ...

            ((p1(sort_index(count)))^(p-1))*...

            (0.5*x(sort_index(count))^-0.5)*[stress_val(sort_index(count),1)...

            /sqrt(stress_val(sort_index(count),1)^2) 0

                0]*C(:,:,sort_index(count))* ...

            B*U(edofMat(sort_index(count),:));

        dpndtheta1(i,sort_index(count))=

            ((cluster_sum(i)/Ni)^((1/p)-1))*(1/Ni) ...

            *((p1(sort_index(count)))^(p-1))*...

            (x(sort_index(count))^0.5)*[stress_val(sort_index(count),1)/ ...

            sqrt(stress_val(sort_index(count),1)^2) 0

                0]*dCdth(:,:,sort_index(count))...

            *B*U(edofMat(sort_index(count),:));

        prelambda(j,:)=((cluster_sum(i)/Ni)^((1/p)-1))*(1/Ni)* ...

            ((p1(sort_index(count)))^(p-1))*...

            [stress_val(sort_index(count),1)/ ...

            sqrt(stress_val(sort_index(count),1)^2) ...
```

```
        0 0]*C(:,:,sort_index(count))*B*...
        Kinv(edofMat(sort_index(count),:), ...
        edofMat(sort_index(count),:));
    dppndx1(i,sort_index(count))= ((clusterr_sum(i)/Ni)^((1/p)-1))*(1/Ni)*
        ...
        ((p2(sort_indexx(count)))^(p-1))*...
        (0.5*x(sort_indexx(count))^-0.5)*[0
            stress_val(sort_index(count),2)...
        /sqrt(stress_val(sort_index(count),2)^2)...
        0]*C(:,:,sort_indexx(count))*B*U(edofMat(sort_indexx(count),:));
    dppndtheta1(i,sort_indexx(count))=
        ((clusterr_sum(i)/Ni)^((1/p)-1))*(1/Ni)...
        *((p2(sort_indexx(count)))^(p-1))*...
        (x(sort_indexx(count))^0.5)*[0 stress_val(sort_index(count),2)/...
        sqrt(stress_val(sort_index(count),2)^2) ...
        0]*dCdth(:,:,sort_indexx(count))*B*...
        U(edofMat(sort_indexx(count),:));
    prelambdaa(j,:)=((clusterr_sum(i)/Ni)^((1/p)-1))*(1/Ni)*...
        ((p2(sort_indexx(count)))^(p-1))*...
        [0 stress_val(sort_index(count),2)/...
        sqrt(stress_val(sort_index(count),2)^2)...
        0]*C(:,:,sort_indexx(count))*B*Kinv...
        (edofMat(sort_indexx(count),:),...
        edofMat(sort_indexx(count),:));
    count=count+1;
end
Lambda(i,:)=sum(prelambda,1);
Lambdaa(i,:)=sum(prelambdaa,1);
```

```
end
count=1;
dpndx2=zeros(1,nele);
dpndtheta2=zeros(1,nele);
dppndx2=zeros(1,nele);
dppndtheta2=zeros(1,nele);
for  i=1:clts
    for j=1:(2*nele/nc)
        dpndx2(i,sort_index(count))=(x(sort_index(count))^0.5)*Lambda(i,:) ...
            *dkdrho(:,:,sort_index(count))*U(edofMat(sort_index(count),:));
        dpndtheta2(i,sort_index(count))=(x(sort_index(count))^0.5)*Lambda(i,:)
            ...
            *dkdthetaa(:,:,sort_index(count))*U(edofMat(sort_index(count),:));
        dppndx2(i,sort_indexx(count))=(x(sort_indexx(count))^0.5)*Lambdaa(i,:)
            ...
            *dkdrho(:,:,sort_indexx(count))*U(edofMat(sort_indexx(count),:));
        dppndtheta2(i,sort_indexx(count))=(x(sort_indexx(count))^0.5)...
            *Lambdaa(i,:)*dkdthetaa(:,:,sort_indexx(count))...
            *U(edofMat(sort_indexx(count),:));
        count=count+1;
    end
end
dpndx=dpndx1-dpndx2;
dpndth=dpndtheta1-dpndtheta2;
dppndx=dppndx1-dppndx2;
dppndth=dppndtheta1-dppndtheta2;
for i=1:clts
    dpndx(i,:)=dpndx(i,:)';
```

```
    dppndx(i,:)=dppndx(i,:)';
    dpndth(i,:)=dpndth(i,:)';
    dppndth(i,:)=dppndth(i,:)';
end
dfdx=[(dv(:)'/(volfrac*nele)) dv_dtheta(:)' ;dpndx./sigmax dpndth./sigmax ...
    ;dppndx./sigmay dppndth./sigmay];
end

function [D, dD_dtheta, D_zero] = Rotation(theta, E1, E2, G12, v12)
% Rotation function
% Takes mechanical properties gives S and D of rotation
% 2 methods can be used and both gives the same result
% Output D rotated and derivative of D rotated and D zero angle
m=cos(theta);
n=sin(theta);
% method 1
% sb11= (1/E1)*(m^4+(m^2*n^2*(-2*v12+(E1/G12)))+(n^4*E1/E2));
% sb12= (1/E1)*((n^2*m^2*(1+(E1/E2)-(E1/G12)))-((n^4+m^4)*v12));
% sb16= (n*m/E1)*((m^2*(2+2*v12-E1/G12))+(n^2*(-2*v12-(2*E1/E2)+(E1/G12))));
% sb22= (n^4 + (m^2*n^2*(-2*v12+E1/G12))+(m^4*E1/E2))*(1/E1);
% sb26= (n*m/E1)*((n^2*(2+2*v12-E1/G12))+(m^2*(-2*v12-(2*E1/E2)+(E1/G12))));
% sb66= (1/E1)*((4*n^2*m^2*(1+2*v12+E1/E2)) + ((n^2-m^2)^2*E1/G12));
% S=[sb11 sb12 sb16; sb12 sb22 sb26; sb16 sb26 sb66];
% D=inv(S);
% % method 2
v21=v12*E2/E1;
s11=1/E1;
s12=-v21/E2;
```

```
s22=1/E2;

s66=1/G12;

s16=0;

s26=0;

S=[s11 s12 s16;s12 s22 s26; s16 s26 s66];

D_zero=inv(S);

T1=[m^2 n^2 2*n*m; n^2 m^2 -2*n*m; -n*m n*m m^2-n^2];

T2= [m^2 n^2 m*n; n^2 m^2 -m*n; -2*m*n 2*m*n m^2-n^2];

D=inv(T1)*D_zero*T2;

S=inv(D);

% This will be used to determine the derivative of constitutive matrix

dT1_inv=[-(2*cos(theta)*sin(theta))/(cos(theta)^4 ...

+sin(theta)^4+2*cos(theta)^2*sin(theta)^2) ...

    (2*cos(theta)*sin(theta))/(cos(theta)^4 ...

    + sin(theta)^4 + 2*cos(theta)^2*sin(theta)^2) ...

    (2*sin(theta)^2)/(cos(theta)^4 + sin(theta)^4 ...

    + 2*cos(theta)^2*sin(theta)^2) - (2*cos(theta)^2)/(cos(theta)^4 ...

    + sin(theta)^4 + 2*cos(theta)^2*sin(theta)^2)...

    ; (2*cos(theta)*sin(theta))/(cos(theta)^4 + sin(theta)^4 ...

    + 2*cos(theta)^2*sin(theta)^2) ...

    -(2*cos(theta)*sin(theta))/(cos(theta)^4 + sin(theta)^4 ...

    + 2*cos(theta)^2*sin(theta)^2) ...

    (2*cos(theta)^2)/(cos(theta)^4 + sin(theta)^4 + ...

    2*cos(theta)^2*sin(theta)^2) - (2*sin(theta)^2)/(cos(theta)^4 ...

    + sin(theta)^4 + 2*cos(theta)^2*sin(theta)^2) ...

    ; cos(theta)^2/(cos(theta)^4 + sin(theta)^4 + ...

    2*cos(theta)^2*sin(theta)^2) - sin(theta)^2/(cos(theta)^4 ...

    + sin(theta)^4 + 2*cos(theta)^2*sin(theta)^2) ...
```

```
    sin(theta)^2/(cos(theta)^4 + sin(theta)^4 + ...
    2*cos(theta)^2*sin(theta)^2) - cos(theta)^2/(cos(theta)^4 ...
    + sin(theta)^4 + 2*cos(theta)^2*sin(theta)^2) ...
    -(4*cos(theta)*sin(theta))/(cos(theta)^4 + ...
    sin(theta)^4 + 2*cos(theta)^2*sin(theta)^2)];
dT2= [ -2*cos(theta)*sin(theta) 2*cos(theta)*sin(theta)
    cos(theta)^2-sin(theta)^2 ...
    ; 2*cos(theta)*sin(theta) -2*cos(theta)*sin(theta)
        sin(theta)^2-cos(theta)^2 ...
    ;2*sin(theta)^2-2*cos(theta)^2 2*cos(theta)^2-2*sin(theta)^2
        -4*cos(theta)*sin(theta)];
dD_dtheta= (dT1_inv*D_zero*T2) + (inv(T1)*D_zero*dT2);
end

function [U,edofMat,KEE,Kg,dkdrho,dkdthetaa,B]=displacements(theta,x)
% This function calculates the displacements of each node
global nelx nely nele Force nodes conn
col_x=x(:);
col_theta=theta(:);
nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nelx*nely,1);
edofMat = repmat(edofVec,1,8)+repmat([0 1 2*nely+[2 3 0 1] -2
    -1],nelx*nely,1);
iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);
F = sparse([2*(nelx+1)*(nely+1)...
    ,2*(nelx)*(nely+1)...
    ,2*(nelx-1)*(nely+1)...
```

```
    ,2*(nelx-2)*(nely+1)...
    ,2*(nelx-3)*(nely+1)...
    ,2*(nelx-4)*(nely+1)...
    ],[1 1 1 1 1 1],[Force Force Force Force Force Force] ...
    ,2*(nely+1)*(nelx+1),1);
U = zeros(2*(nely+1)*(nelx+1),1);
%fixeddofs = [1:nely+1];
fixeddofs  = [1:2*(nely+1)];
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs,fixeddofs);
for i=1:nele
    n4=[nodes(conn(i,2),2) nodes(conn(i,2),3)];
    n1=[nodes(conn(i,3),2) nodes(conn(i,3),3)];
    n3=[nodes(conn(i,4),2) nodes(conn(i,4),3)];
    n2=[nodes(conn(i,5),2) nodes(conn(i,5),3)];
    % Penalization is done inside the function
    [KE, dk, dktheta] = complete_stiff(col_theta(i),col_x(i),n1,n2,n3,n4);
    B = B_mat(n1,n2,n3,n4);

    KEE(:,:,i)=KE;
    dkdrho(:,:,i)=dk;
    dkdthetaa(:,:,i)=dktheta;
    sK((((i-1)*64)+1:(i)*64),1) = reshape(KE,64,1);
end
K = sparse(iK,jK,sK);
Kg = (K+K')/2;
U(freedofs) = Kg(freedofs,freedofs)\F(freedofs);
end
```

```
function [H,Hs] = fil(nelx, nely, rmin)
% PREPARE FILTER
iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for i1 = 1:nelx
  for j1 = 1:nely
    e1 = (i1-1)*nely+j1;
    for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
      for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
        e2 = (i2-1)*nely+j2;
        k = k+1;
        iH(k) = e1;
        jH(k) = e2;
        sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
      end
    end
  end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);
end


function plotstresses(n,connec,obj,n_pts,name)
% n--> nodes coordinates matrix
% conn ---> connectivity matrix
```

```
% n_pts ---> number of points in color bar
% name ---> name of the calculated value
% Initialization of the required matrices
n_nodes=length(connec(1,:));
n_elem=length(connec(:,1));
X = zeros(n_nodes,n_elem);
Y = zeros(n_nodes,n_elem);
main = zeros(n_nodes,n_elem);
conn=zeros(length(connec(:,1)),2);
conn(:,1)=connec(:,3);
conn(:,2)=connec(:,4);
conn(:,3)=connec(:,2);
conn(:,4)=connec(:,1);
% calculating x and y and corresponding objective
for i=1:n_elem
    for j=1:n_nodes
        node(j)=conn(i,j); % extract connected node of element i
        X(j,i)=n(node(j),1); % extract x value of the node
        Y(j,i)=n(node(j),2); % extract y value of the node
    end
    main(:,i) = obj(i) ; % extract component value of the node
end
figure
plot(X,Y,'r')
fill(X,Y,main)
axis off ;
c_bar=colorbar;
set(get(c_bar,'title'),'string',name);
```

```
% get the color limits
lim = caxis;
ylim(c_bar,[lim(1) lim(2)]);
ticks = linspace(lim(1),lim(2),n_pts);
set(c_bar,'YtickMode','manual','YTick',ticks); % Set the tickmode to manual
for i = 1:n_pts
    imep = num2str(ticks(i),'%+3.3E');
    values(i) = {imep} ;
end
set(c_bar,'YTickLabel',values(1:n_pts),'fontsize',9);
end

function plottopology(x,theta,threshold)
global conn nodes
obj1=1-x(:);
obj2=theta(:);
figure
grid off
uu=1;
for i=1:length(conn(:,1))
    if obj1(i)<threshold
        x0=nodes(conn(i,3),2);
        y0=nodes(conn(i,3),3);
        x1=nodes(conn(i,5),2);
        y1=nodes(conn(i,2),3);
        dx=abs(x1-x0);
        dy=abs(y1-y0);
        rectangle('Position',[x0 y0 dx dy],...
```

```
            'FaceColor',obj1(i)*[1 1 1],'EdgeColor','k');
        hold on;
        r(i)=cos(obj2(i));
        s(i)=sin(obj2(i));
        anglee(uu)= obj2(i);
        uu=uu+1;
        xvec(i)=x0;
        yvec(i)=y0;
    end
end
quiver(xvec,yvec,r,s,'m')
end
```