

POLITECNICO DI TORINO  
DIMEAS - Dipartimento di Ingegneria Meccanica e Aerospaziale

Thesis submitted for the  
Master of Science in Aerospace Engineering

---

# Continuous Fiber Path Planning Algorithm for 3D Printed Optimal Mechanical Properties

---



Supervisors:

**Frédéric LACHAUD**  
**Joseph MORLIER**  
ISAE Supaero, Toulouse

**Marco GHERLONE**  
Politecnico di Torino

Candidate:

**Enrico STRAGIOTTI**

Academic Year 2019/2020



# Declaration of Original and Sole Authorship

I, Enrico STRAGIOTTI, declare that this thesis entitled: *Continuous Fiber Path Planning Algorithm for 3D Printed Optimal Mechanical Properties* and the data presented in it are original and my own work.

I confirm that:

- No part of this work has previously been submitted for a degree at this or any other university.
- References to the work of others have been clearly acknowledged. Quotations from the work of others have been clearly indicated, and attributed to them.
- In cases where others have contributed to part of this work, such contribution has been clearly acknowledged and distinguished from my own work.

Date: April, 2020

Signature:



# Abstract

## Continuous Fiber Path Planning Algorithm for 3D Printed Optimal Mechanical Properties

Mechanical properties of parts produced with continuous filament fabrication are influenced by the print direction, especially if the material deposited is carbon fiber or a fiber-reinforced polymer. This opens the development of new design strategies to fully exploit the unidirectionality of composites, modeling the fibers' trajectories according to external loads and the part topology. This MSc thesis presents a complete framework that calculates optimal fibers' paths that achieve minimal compliance for two-dimensional components. The framework is divided into two different sections. In the first one, an algorithm solving a Continuous Fibre Angle Optimization (CFAO) problem is described and solved using a gradient-based optimizer. In the second section, the continuous trajectories are calculated as iso-lines of a level-set surface obtained from the results of the first optimization and are then exported as G-Code. Special attention is devoted to implementing manufacturing constraints, such as fiber continuity and parallelism between the fibers. Various two-dimensional numerical examples are provided to verify the improved mechanical proprieties of the part. The trajectories are then printed using the G-code on a Fused Filament Fabrication (FFF) polymer desktop printer to validate the framework from a manufacturing point of view.

**Keywords:** material optimization; additive manufacturing; composite 3D printing; fused filament fabrication; continuous filament fabrication



# Acknowledgment

First of all, I want to thank Prof. Frédéric Lachaud and Prof. Joseph Morlier, teachers, and researchers at ISAE-SUPAERO. They were the supervisors of my thesis, and I would thank them for their availability, guidance, and encouragement. They have been able to stimulate my interest in this work and have supported me throughout all the internship. For all the time dedicated to me in the laboratory, I thank Caterina, Eric, Emilien, and Maxime, who have worked on a side project and provided me with all the materials data. A heartfelt thanks also to David, Matteo, and Kubra, colleagues, and friends, who have always been able to help me smiling and have patiently answered all my endless questions. I want to thank my academic tutor, Prof. Marco Gherlone, associate professor at Politecnico di Torino, that helped me proofing my manuscript.

Getting to the writing of this thesis would not have been easy without Alberto, Federica, Federico, Filippo, and Francesco, companions of many difficulties that I would never have been able to overcome alone. I finally thank Milena and Roberto, parents, coping with me during this time and always ready to support my choices, my brother Andrea, a sincere friend and always my biggest supporter and Andrea, without whom all this would not be possible.

Finally, I would like to thank the Politecnico di Torino that helped me financially, permitting the execution of the research.

Thank you all very much.



*To Andrea, who always pushed me  
to be my best self.*



# Contents

|   |            |
|---|------------|
| <b>Declaration of Original and Sole Authorship</b>          | <b>iii</b> |
| <b>Abstract</b>   | <b>v</b>   |
| <b>Acknowledgment</b>                                       | <b>vii</b> |
| <b>Contents</b>   | <b>xi</b>  |
| <b>List of Figures</b>                                      | <b>xv</b>  |
| <b>List of Tables</b>                                       | <b>xix</b> |
| <br>  |            |
| <b>Introduction</b>   | <b>1</b>   |
| <br>  |            |
| <b>1 The printing process</b>                               | <b>3</b>   |
| 1.1 Additive Manufacturing . . . . .                        | 3          |
| 1.1.1 Different Types of AM Processes . . . . .             | 4          |
| 1.2 Composite Materials Manufacturing . . . . .             | 9          |
| 1.2.1 Filament Winding (FW) . . . . .                       | 10         |
| 1.2.2 Automated Tape Laying (ATL) . . . . .                 | 10         |
| 1.2.3 Automated Fiber Placement (AFP) . . . . .             | 11         |
| 1.2.4 Manufacturing Constraints and Defects . . . . .       | 12         |
| 1.3 Additive Manufacturing of Composite Materials . . . . . | 13         |
| <br>  |            |
| <b>2 Optimization Algorithms</b>                            | <b>17</b>  |
| 2.1 Numerical Optimization . . . . .                        | 17         |
| 2.1.1 Terminology and Problem Statement . . . . .           | 18         |
| 2.2 Gradient-based Optimization . . . . .                   | 19         |
| 2.2.1 Gradients and Hessian . . . . .                       | 21         |
| 2.2.2 Real algorithms . . . . .                             | 22         |
| 2.2.3 Sensitivity Analysis . . . . .                        | 24         |
| 2.3 Topology and Fiber angle Optimization . . . . .         | 24         |
| 2.4 Precedent Works . . . . .                               | 26         |

|          |  |           |
|----------|--|-----------|
| <b>3</b> | <b>The FEM Simulation</b>  | <b>31</b> |
| 3.1      | FEM Linear Structural Analysis . . . . .   | 31        |
| 3.1.1    | Lamination parameters . . . . .  | 32        |
| 3.1.2    | The Stiffness Matrix . . . . .   | 33        |
| 3.1.3    | Elastic Energy . . . . .   | 34        |
| 3.1.4    | On the convexity of the lamination parameter in the optimization problem . . . . . | 34        |
| 3.2      | ABAQUS Comparison and Validation . . . . .   | 35        |
| 3.2.1    | ABAQUS . . . . .   | 35        |
| 3.2.2    | The choice of the type of element . . . . .  | 36        |
| <b>4</b> | <b>Filtered Continuous Fiber Angle Optimization</b>                                | <b>39</b> |
| 4.1      | Problem definition . . . . .   | 39        |
| 4.2      | Gradient Descent Optimization . . . . .  | 40        |
| 4.2.1    | Problem Definition . . . . .   | 40        |
| 4.2.2    | Optimization procedure . . . . .   | 40        |
| 4.2.3    | Gradient Calculation . . . . .   | 41        |
| 4.2.4    | Gaussian smoothing . . . . .   | 42        |
| 4.2.5    | Filtered Gradient Calculation . . . . .  | 44        |
| 4.2.6    | Multistart . . . . .   | 44        |
| 4.3      | Level-Set Surface Calculation . . . . .  | 45        |
| 4.3.1    | Problem description . . . . .  | 45        |
| 4.3.2    | Poisson Equation . . . . .   | 45        |
| 4.3.3    | Direct integration approach . . . . .  | 46        |
| 4.3.4    | Calculate the conservative field of the gradient . . . . .                         | 47        |
| 4.3.5    | Optimization routine . . . . .   | 47        |
| 4.4      | Fiber Propagation . . . . .  | 48        |
| 4.4.1    | The Heat Method . . . . .  | 51        |
| 4.4.2    | FEM analysis . . . . .   | 54        |
| 4.4.3    | G-code . . . . .   | 54        |
| <b>5</b> | <b>Test Case and Results</b>   | <b>57</b> |
| 5.1      | Load Cases and Materials . . . . .   | 57        |
| 5.1.1    | Load Cases . . . . .   | 57        |
| 5.1.2    | Materials . . . . .  | 60        |
| 5.2      | FEM comparison with ABAQUS . . . . .   | 61        |
| 5.2.1    | Creation of the model . . . . .  | 61        |
| 5.2.2    | Results . . . . .  | 62        |
| 5.3      | Filtered CFAO . . . . .  | 64        |
| 5.3.1    | Overview of the algorithm . . . . .  | 64        |
| 5.3.2    | Setting of the optimizer and of the filter . . . . .                               | 71        |
| 5.3.3    | Results . . . . .  | 72        |
| 5.3.4    | Comparison with the State of the art . . . . .                                     | 79        |
| 5.4      | Unfiltered CFAO . . . . .  | 80        |
| 5.4.1    | Settings of the Optimizer . . . . .  | 81        |
| 5.4.2    | Results . . . . .  | 81        |

|  |           |
|--|-----------|
| <b>Conclusion</b>  | <b>83</b> |
| <b>A Full convergence history for the FCFAO algorithm.</b> | <b>85</b> |
| <b>B Matlab Code</b>                                       | <b>89</b> |
| <b>List of Abbreviations</b>                               | <b>91</b> |
| <b>Bibliography</b>  | <b>93</b> |



# List of Figures

|      |  |    |
|------|--|----|
| 1.1  | Schematic drawing of the Selective Laser Sintering (SLS) process. The material powder is being sintered by the high power laser to form complex 3D objects. Courtesy of CustomPart Net . . . . .                                 | 5  |
| 1.2  | Different size of powder ready for Electron Beam Melting (EBM) (left) and Selective Laser Melting (SLM) (right). . . . .   | 5  |
| 1.3  | Graphical drawing of the Laminated Object Manufacturing (LOM) process. The material is being added to the 3D component sheet by sheet and is cutted by the high power laser. Courtesy of CustomPart Net . .                      | 6  |
| 1.4  | Graphical drawing of the Stereolithography (SLA) process. A UV-laser is used to trace the cross-section of each layer into a liquid bath of photosensitive polymer to fabricate a 3D object. Courtesy of CustomPart Net. . . . . | 7  |
| 1.5  | Graphical drawing of the Three Dimensional Printing (3DP) process. An inkjet print head deposits a liquid binder onto each layer of powder individually to build a 3D part. Courtesy of CustomPart Net. . . . .                  | 8  |
| 1.6  | Graphical representation of the nozzle of a Laser Engineered Net Shaping (LENS) machine for additive manufacturing (Gibou et al., 2018). . . .   | 8  |
| 1.7  | Graphical drawing of the Fused Deposition Modelling (FDM) process. The material is forced to melt and to pass trough the nozzle to be deposited on the 3D piece. Courtesy of CustomPart Net. . . . .                             | 9  |
| 1.8  | Difference between convectional and tow-steered ply of composite (Brooks and Martins, 2018). . . . .   | 10 |
| 1.9  | The head of a Automated Tape Laying (ATL) machine (Lukaszewicz et al., 2012). . . . .  | 11 |
| 1.10 | The head of a Automated Fiber Placement (AFP) machine (Lukaszewicz et al., 2012). The main difference with the ATL one is that it accepts multiple smaller tows. . . . .   | 12 |
| 1.11 | The three main defects that can occur when a tow is steered (Lukaszewicz et al., 2012). . . . .  | 13 |
| 1.12 | Example of defects of an AFP piece due to excessive curvature of the tow steer or to an incorrect planning of the trajectories: overlap (left) and puking (right). . . . .   | 14 |
| 1.13 | Physical representation of the minimum length constrain for an ATL or an AFP machine (Brooks and Martins, 2018). . . . .   | 15 |

|      |  |    |
|------|--|----|
| 1.14 | The carbon fiber is combined with a thermoset matrix to produce the prepreg then cured by the nozzle just before the deposition. Credits to Anisioprint. . . . .   | 15 |
| 1.15 | Graphical drawing of the printhead of the modified continuous fiber FDM machine. Credits to Anisioprint. . . . .   | 16 |
| 1.16 | A continuous fiber composite printed in 3D. Credits to Anisioprint. . . . .  | 16 |
| 2.1  | Graphical representation of the path followed by the person (current iteration) on the top of a hill (function to minimize) that simulate the gradient descent algorithm (Amini et al., 2018). . . . .   | 20 |
| 2.2  | Graphical representation of the topology optimization of a MBB symmetric beam using Solid Isotropic Material with Penalization method (SIMP) (Labanda, 2014). . . . .  | 25 |
| 2.3  | Result of a fiber angle optimization using Continuous Fibre Angle Optimization (CFAO) (Kiyono et al., 2017). . . . .   | 26 |
| 2.4  | Results of the topology optimization linked with the fiber angle optimization by Jiang et al. (2019). . . . .  | 26 |
| 2.5  | Printing process of the optimized topology by Jiang et al. (2019). . . . .   | 26 |
| 2.6  | Graphical representation of the spacial filter used by Kiyono et al. (2017). . . . .   | 27 |
| 2.7  | Graphical representation of a 2D mesh of 36 rectangular elements divided into 4 patches (Sørensen and Kann, 2011). . . . .   | 28 |
| 2.8  | Level-set representation of a free boundary (blue solid line) in two spatial dimensions, moving in its normal direction, and subsequent changes in topology that are handled automatically. The level-set function is depicted in red (Gibou et al., 2018). . . . .            | 29 |
| 3.1  | Components of stress in two dimensions. Credit engAPPLETS by VirginiaTech . . . . .  | 31 |
| 3.2  | Resume of the steps performed by ABAQUS to pre-process, simulate, and post-process a model. . . . .  | 36 |
| 3.3  | Graphical representation of a 2D reduced integration element. Image credits: ABAQUS Documentation. . . . .   | 37 |
| 3.4  | Graphical representation of the hourglass phenomenon over a mesh of 2D reduced integration elements. Image credits: LS-DYNA Documentation. . . . .   | 37 |
| 3.5  | Comparison between elements S4 and S4R for a simple 3 points bending problem. Even if the element S4R have the Hourglass control, it still suffer from massive deformation of the mesh. The S4 instead, using 4 different nodes for the analysis shows better results. . . . . | 38 |
| 4.1  | Graphical representation of the Gaussian filter used to achieve fiber continuity in CFAO algorithm. In this image, the dimension of the filter are 11x11 elements and $\sigma = \text{dim}/7$ . . . . .  | 43 |
| 4.2  | Graphical representation of the Convolution operation over the height and width of a matrix. Image credits: RiverTrail Github tutorial. . . . .  | 43 |
| 4.3  | Graphical representation of the Symmetrical Padding operation. Image credits: Matlab Documentation . . . . .   | 44 |
| 4.4  | Exemples of fiber propagation (see Sethian, 1999). . . . .   | 49 |

---

|      |  |    |
|------|--|----|
| 4.5  | Comparison of the different outputs that the Heat Method produces when a different number of master fibers is chosen. It is clear that having too many fibers tends to produce noisy results in the boundary and to have only one could lead to lose the optimality. . . . . | 50 |
| 5.1  | In this image is presented the first steps of the ABAQUS analysis: the geometry and the material set-up. . . . .   | 62 |
| 5.2  | In this image is presented the last steps of the ABAQUS analysis: the definition of the mesh and the set-up of the load case. . . . .  | 62 |
| 5.3  | The percentage differences is higher in models presenting high distortion of the mesh due to concentrated Boundary Conditions (BCs) or loads. .  | 63 |
| 5.4  | Graphic representation of the Messerschmitt-Bölkow-Blohm (MBB) load case. . . . .  | 64 |
| 5.5  | Starting point and optimized fiber angle directions calculated by the Filtered Continuous Fibre Angle Optimization (FCFAO) algorithm. . .  | 65 |
| 5.6  | The picture give intermediate information about the optimization using 3 different graphs: the convergence history, the optimization variables and the design variables. The results are at the 150th iteration. . . . .   | 66 |
| 5.7  | The discrete field $\theta(x, y)$ is linearly interpolated and then the contour plot is obtained. The angles are all in radians. . . . .   | 66 |
| 5.8  | The gradient $\vec{\nabla}\phi(x, y)$ vector field is obtained and plotted. . . . .  | 67 |
| 5.9  | The new gradient $\vec{\nabla}\phi(x, y)$ vector field. Thanks to its curl being almost zero everywhere, the field is from now on considered conservative. . . .   | 67 |
| 5.10 | The level set surface $\phi(x, y)$ with its iso-lines. . . . .   | 67 |
| 5.11 | The X component of the gradient of $\phi(x, y)$ . . . . .  | 68 |
| 5.12 | The Y component of the gradient of $\phi(x, y)$ . . . . .  | 68 |
| 5.13 | The iso-level lines of the surface $\phi(x, y)$ . . . . .  | 68 |
| 5.14 | The same trajectories found in img. 5.13 are plotted against the plot of the gradient. The iso-lines of the surface $\phi(x, y)$ are always perpendicular to the gradient. . . . .   | 69 |
| 5.15 | The optimized paths ready to be exported in G-code and printed. . . .  | 69 |
| 5.16 | The direction of the fibers of the optimum manufacturable case. The elastic energy is calculated performing a Finite Element Method (FEM) analysis over this mesh. Dimension and number of elements are the very same used for the FCFAO optimization. . . . .               | 70 |
| 5.17 | The trajectories found are validated trough the manufacture of the beam using a desktop 3D printer. Here the print has been interrupted to reveal the inner optimized layers. . . . .  | 70 |
| 5.18 | An example of the circular Gaussian filter used to achieve fiber continuity in the FCFAO analysis. . . . .   | 72 |
| 5.19 | Starting point and optimized fiber angle directions calculated by the FCFAO algorithm. . . . .   | 72 |
| 5.20 | Optimized paths ready to be printed for the "Cantilever beam - Distributed Load - Top" case. . . . .   | 73 |
| 5.21 | Starting point and optimized fiber angle directions calculated by the FCFAO algorithm. . . . .   | 74 |

|      |   |    |
|------|---|----|
| 5.22 | Optimized paths ready to be printed for the "Cantilever beam - Concentrated Load - Bottom" case. . . . .  | 74 |
| 5.24 | Optimized paths ready to be printed for the "Cantilever beam - Concentrated Load - Half" case. . . . .  | 75 |
| 5.25 | Starting point and optimized fiber angle directions calculated by the FCFAO algorithm. . . . .  | 76 |
| 5.26 | Optimized paths ready to be printed for the "Cantilever beam - Distributed Load - Right" case. . . . .  | 77 |
| 5.27 | Starting point and optimized fiber angle directions calculated by the FCFAO algorithm. . . . .  | 78 |
| 5.28 | Optimized paths ready to be printed for the "Messerschmitt-Bölkow-Blohm Beam - Symmetric" case. . . . .   | 78 |
| 5.29 | Graphical representation of the topological optimization of a MBB symmetric beam using SIMP (Labanda, 2014). . . . .  | 79 |
| 5.30 | FCFAO result . . . . .  | 79 |
| 5.31 | The result of the optimization on the cantilever beam by Silva et al. (2020). . . . .   | 80 |
| 5.32 | The result of the optimization on the cantilever beam by Demir et al. (2019). . . . .   | 80 |
| 5.33 | The result of the optimization on the cantilever beam by Kiyono et al. (2017). This is the first article where a spatial filter is used to achieve fibers continuity. . . . . | 80 |
| 5.34 | A comparison between the unfiltered and the filtered analysis for the "DistrTOP" load case. . . . .   | 81 |
| A.1  | Iteration n. 1 of the FCFAO algorithm used on a 60x10 mesh and MBB load case. . . . .   | 85 |
| A.2  | Iteration n. 150 of the FCFAO algorithm used on a 60x10 mesh and MBB load case. . . . .   | 86 |
| A.3  | Iteration n. 550 of the FCFAO algorithm used on a 60x10 mesh and MBB load case. . . . .   | 86 |
| A.4  | Iteration n. 924 of the FCFAO algorithm used on a 60x10 mesh and MBB load case. . . . .   | 87 |

# List of Tables

|      |   |    |
|------|---|----|
| 5.1  | Mechanical properties evaluated in the ISAE-SUPAERO testing facilities and given by Markforged (2019). . . . .  | 61 |
| 5.2  | The results of the FEM analysis performed using the proprietary MATLAB code. The values in the tables refer to the sum of the elastic energy stored in all the element of the beam. . . . . | 62 |
| 5.3  | The results of the FEM analysis performed with the commercial code ABAQUS. The values in the tables refer to the sum of the elastic energy stored in all the element of the beam. . . . .   | 63 |
| 5.4  | Percentage differences of the absolute value of the elastic energy between the proprietary code and the commercial code ABAQUS. . . . .   | 63 |
| 5.5  | Results of FCFAO algorithm for the MBB load case. . . . .   | 71 |
| 5.6  | Results of FCFAO algorithm for the Distributed Top load case and comparison with the unfiltered algorithm CFAO. . . . .   | 73 |
| 5.7  | Results of FCFAO algorithm for the Concentrated Bottom load case and comparison with the unfiltered algorithm CFAO. . . . .   | 75 |
| 5.8  | Results of FCFAO algorithm for the Concentrated Half load case and comparison with the unfiltered algorithm CFAO. . . . .   | 76 |
| 5.9  | Results of FCFAO algorithm for the Distributed Right load case and comparison with the unfiltered algorithm CFAO. . . . .   | 77 |
| 5.10 | Results of FCFAO algorithm for the MBB symmetric load case and comparison with the unfiltered algorithm CFAO. . . . .   | 79 |
| 5.11 | Results of the Unfiltered CFAO algorithm for all the different load cases. The mesh used is 60 x 20 elements. . . . .   | 81 |



# Introduction

Commercial aeronautic is a growing market. The number of air passengers is increasing worldwide by a steady 5% a year, and global demand is doubling every 15 years. However, in recent years a value different from profit and sales emerged in the aeronautic industry. The agenda of principal market players is set on environment goals and care for the planet climate. The design of efficient engines and the reduction of weight are the major fields in which R&D is focusing. Composite materials play a crucial role in both the domains, allowing the creation of lighter structures with better mechanical proprieties. Having high strength-to-weight and stiffness-to-weight ratios, they are the natural candidate to revolutionize the traditional aircraft design.

The main idea of composite structures is to utilize the material in the best possible way by tailoring it to the application. Being highly anisotropic, this class of material allows one to exploit all the characteristics of a given topology. It is known from the study of mechanics of materials that the maximum stress always occurs in a specific direction. Therefore, having isotropic mechanical proprieties in all directions leads to a natural over-sizing of the material in the non-maximum directions. This over-sizing could be strongly reduced using composite structures as the material is designed to have directional proprieties where needed. Besides stiffness and strength, a wide number of properties are improved using composite materials as well (Jones, 1999):

- Fatigue life
- Temperature-dependent behavior
- Thermal insulation and conductivity
- Wear resistance
- Corrosion resistance
- Noise insulation

Composite materials are not exactly innovative, as first uses date back to 1500 B.C. when Egyptians and Mesopotamian pioneers used a mixture of mud and straw to create stiff and durable buildings (Johnson, 2020). Neither Carbon Fiber Reinforced Polymers (CFRPs) could be considered a new topic because carbon fiber was developed around 1970, and they are recognized as a natural material by the aerospace industry by now. On new planes as the Airbus A350 and the Boeing 787, more than 50% of the structure mass is made of CFRPs. Even if the different components did not change in the last years, the field is now revamped by the use of new thermoplastic matrices

and especially the advent of new manufacturing techniques. Automated Tape Laying (ATL) and Automated Fiber Placement (AFP) are the two leading technologies that are employed today to manufacture advanced composite laminates from unidirectional prepregs (Lukaszewicz et al., 2012). In general, they place composites prepreg heated and compacted before on typically complex tooling mandrels. The material comes in the form of what is referred to as "tows" and is essentially a plain unidirectional prepreg that comes in different width. These two techniques already allow the designer to achieve better directional performances by steering the tow to meet the maximum stress directions. However, they cannot still manufacture small components and the maneuverability of the fiber deposition on a micro-scale, which is something that even a simple 3D consumer desktop printer could do. Those manufacturing constraints led in recent years to the development of a modified 3D printer capable of extruding and depositing single fiber prepreg. Companies such as Anisioprint in Russia, MOI composites in Italy, 9T labs in Switzerland, and Markforged and Continuous Composites in the USA, have all specialized in printing single CFRP filaments. They all use a similar process where a tiny bundle of fibers wrapped in thermoplastic is heated and extruded through a print-head. These new companies are competing for their share of the market and are focused on developing the fabrication technology rather than concentrating on the deposition path planning, which could further unlock the process's potential.

Allowing composite laminates to be manufactured with continuous spatial variation of fiber angle generates considerably larger design freedom as compared to conventional constant stiffness laminates. In the design of composite structures, optimization methods are often employed so that these advantages of composite materials and novel manufacturing technologies can be fully exploited.

It is evident the need for new design rules to follow and new algorithms capable of dealing with these new manufacturing techniques, to reduce the structure weight of a component further and reduce fuel consumption.

Although the manufacturing technology based on AFP and 3D printing is still undergoing developments, methods such as Filament Winding (FW) and selective ply deposition are well established. The algorithms developed in this thesis would not only be useful for AFP and 3D printing but also for these better-established methods.

# Chapter 1

## The printing process

This Chapter gives a brief description of the different techniques used to produce additive manufacturing parts. Section 1.1 focuses on additive manufacturing processes used to print components efficiently. The classic materials used for this technology are essentially polymer, metal, and ceramic. Composite materials are not yet discussed. In Section 1.2, the novel techniques used to produce composite material parts are presented. Manufacturing constraints and defects are discussed, as well. In the last section, the two topics are merged, and the new 3D printers for continuous fiber composite material are presented.

### 1.1 Additive Manufacturing

Additive Manufacturing (AM) has been described by the ASTM as: "the process of joining materials to make objects from 3D model data, usually layer upon layer". It is also known as rapid manufacturing or rapid prototyping (Huang et al., 2013), and, unlike conventional manufacturing processes that produce pieces by machining material from a larger stock or sheet, it creates the final shape by adding new material. The main advantage is that AM makes efficient use of raw materials and produce limited wastes while reaching satisfactory shape accuracy.

The development of AM technology started in the 1980s, and significant improvements have been made since then. There is the expectation that this technology can lead to a revolution of the manufacturing industry and provide various benefit to the community (Huang et al., 2013), as:

- Reduced energy consumption due to better raw material usage, which is a key contribution to environmental sustainability.
- On-demand manufacturing, which can reconfigure the manufacturing supply chain and bring cheaper products to consumers while utilizing fewer resources.
- Healthcare products customized to the needs of individuals, as prothesis or valves, which is expected to improve the population's wellbeing.

### 1.1.1 Different Types of AM Processes

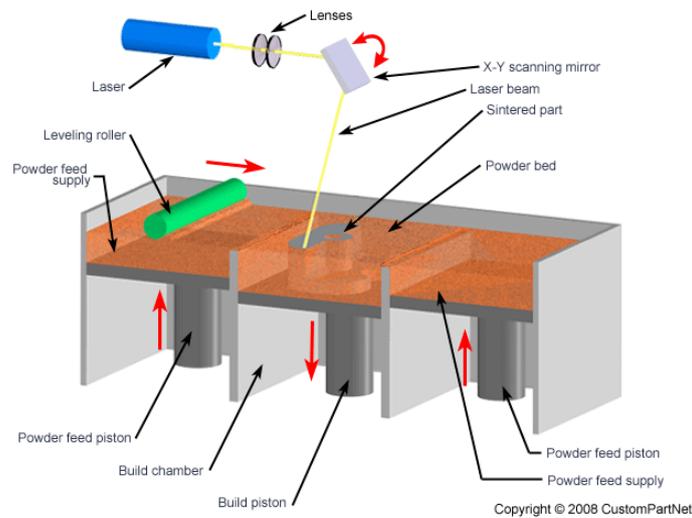
Many different types of AM processes exist, but they all operate according to the same principles. A digital drawing is first created using a Computer-Aided Design (CAD) software. One of the main advantages of additive manufacturing is the possibility to create a difficult design that cannot be produced using traditional machining. Because no tooling is needed, complex surfaces and internal features can be produced directly when making the part. Also, the complexity of a component has little effect on build times, as opposed to other manufacturing processes. It is also beneficial at this stage the use of a topology optimizer. The CAD part is then exported and imported by a slicer software. The aim of this program is the creation of the different layers according to the topology of the part and the characteristics of the printer. Each form of CAD software saves the geometric data describing the 3-D model in various ways. However, the .stl format (initially developed for stereolithography) has become the standard file format for additive methods. The .stl format renders the surfaces of the 3-D model as a set of triangles, collecting the coordinates for the vertices and normal directions for each triangle. The file that usually has a .stl extension is then exported and uploaded on the controller of the machine. Nowadays, the .stl file extension is the standard format for any AM machine.

The most popular AM processes available are Selective Laser Sintering (SLS), Laminated Object Manufacturing (LOM), Stereolithography (SLA), Three Dimensional Printing (3DP), Laser Engineered Net Shaping (LENS), and Fused Deposition Modelling (FDM). All of these different techniques are briefly described in the next section.

#### Selective Laser Sintering (SLS)

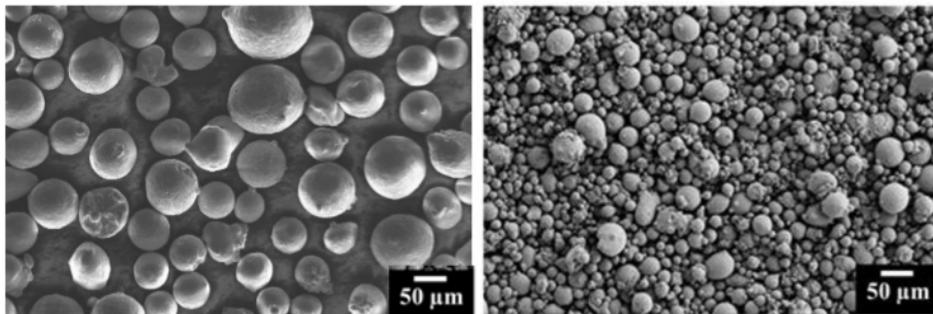
SLS uses a high power laser to fuse small particles of the build material in an inert gas environment. Usually, the powder bed is heated to minimize the thermal distortion and maximize the adhesion of the layers. The laser then supplies the final amount of heat to sinter the powder to form a solid part. The powder is not melted by the laser, since sintering is the process of forming a solid mass of material by means of heat, without liquefying the material first. With sintering, the porosity of the material can be controlled. Selective Laser Melting (SLM), on the other hand, can do the same as sintering and go further by using the laser to achieve a full melt. In this different process, the powder is not simply fused together but is melted into a regular part. That makes melting the way to go for a mono-material, as there's just one melting point, not the variety you'd find in an alloy (Simchi and Pohl, 2004). In any case, special support structures are not needed because the excess powder in each layer acts as a support to the part being built. One of the drawbacks of SLS or SLM processes is that the parts created have limited mechanical properties and poor fatigue life due to the irregular surface produced by the layer-per-layer approach.

A similar technique is Electron Beam Melting (EBM), where parts are produced by melting metal powder, layer by layer, with an electron beam in a high vacuum, permitting higher recyclability of powders, no contamination, and a more stable process. It is characterized by high power (3 kW) beam that achieves less stress, less distortion, a finer microstructure, and better material properties when compared to SLS and SLM.



**Figure 1.1:** Schematic drawing of the SLS process. The material powder is being sintered by the high power laser to form complex 3D objects. Courtesy of CustomPart Net

Additionally, EBM generally has a higher build rate because of its higher scanning speed and energy density, which directly influence the layer thickness. For this motivation, the medium size of the grain of the powder for EBM is grosser (see fig. 1.2). EBM works only on conductive materials. Currently commercial materials for EBM include pure Titanium, Ti-6Al-4V, and Inconel.

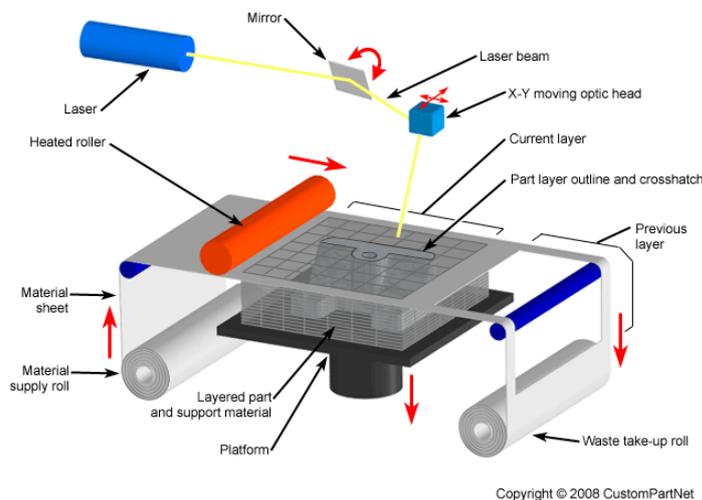


**Figure 1.2:** Different size of powder ready for EBM (left) and SLM (right).

### Laminated Object Manufacturing (LOM)

The first commercial LOM system was invented in 1991. The main elements of the system are a feed device that drives a sheet over a build stand, a heated roller to apply pressure to bond the sheet to the layer below, and a laser to cut the contour of the part in each sheet layer. When manufacturing a metal component, the sheet is rolled by passing a heated roller over the flat surface. The metal foil is coated with a heat-sensitive adhesive, which is activated by the thermal energy released by the roller during rolling and provides the appropriate adhesion with the next foil during subsequent deposition. After rolling, a laser-cut is made to produce the section, following the model

of the supplied CAD file. The power of the laser beam must be adjusted so as to obtain a cutting depth limited to the thickness of the plane so that the cut does not also affect the sections previously made. The laser removes from the tape a region of square shape containing the laser, is then executed a second cut following the perimeter of the section so as to make the desired shape. The excess material outside the perimeter is cut out by the laser into squares (cubing) to facilitate its subsequent removal. Such waste material acts as a support for the next plans during the manufacture of the object. As an alternative to adhesive, traditional welding or laser techniques or mechanical fastening using screws can be used to ensure a good bond strength between the surfaces. After all the layers have been placed, the piece is removed from the material platform in excess is removed (decubing). Only the decubing phase and the various finishing processes require the intervention of an operator; otherwise, the process is fully automated.



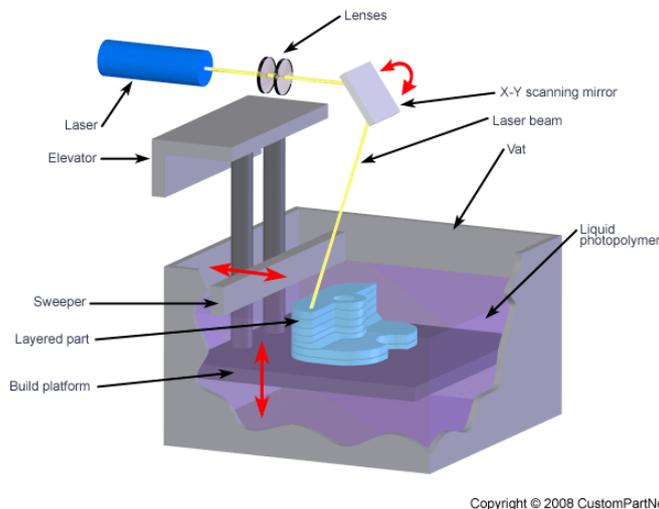
**Figure 1.3:** Graphical drawing of the LOM process. The material is being added to the 3D component sheet by sheet and is cut by the high power laser. Courtesy of CustomPart Net

Build speed is fast, but the surface finish is rough.

### Stereolithography (SLA)

Stereolithography was introduced in the mid-1980s and later established, within the following decade, as one of the basic techniques of additive manufacturing. Since then, SLA's ability to produce complex prototypes quickly and accurately has contributed to a radical change in the design industry. SLS uses a UV laser beam to allow the polymerization of thousands of individual layers until the complete piece is created. As with other additive manufacturing processes, such as SLS and SLM, SLA is largely dependent on the use of laser systems for parts manufacturing. The pieces are made through the polymerization of very thin layers of thermosetting liquid resin with the aid of an ultraviolet laser beam (UV), projected on the surface of the resin to allow the passage to the solid-state. The process is repeated for each layer of unpolymerized resin

until the complete realization of the piece. The parts manufactured by stereolithography have to undergo post-treatment, which includes a UV curing cycle to allow the complete solidification of the external surface and to meet any additional finishing requirements. Despite differences in material properties, stereolithography clearly wins

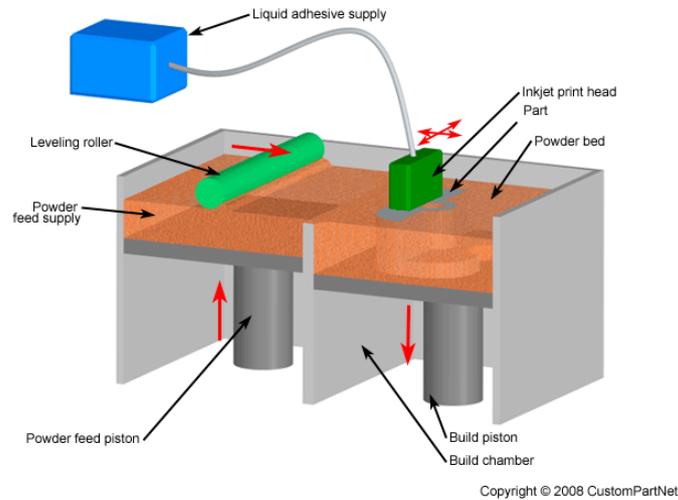


**Figure 1.4:** Graphical drawing of the SLA process. A UV-laser is used to trace the cross-section of each layer into a liquid bath of photosensitive polymer to fabricate a 3D object. Courtesy of CustomPart Net.

over SLS on the front of precision and surface finish. Usually, the production of parts with standard or high resolution, with layers of thickness between 0.1 and 0.025 mm, and details of dimensions up to 0.05 mm. This means that stereolithography makes possible the reproduction of extremely complex details and surfaces able to cope with various aesthetic demands, with finishes almost completely devoid of the "scaling" typical of processes such as casting modeling.

### Three Dimensional Printing (3DP)

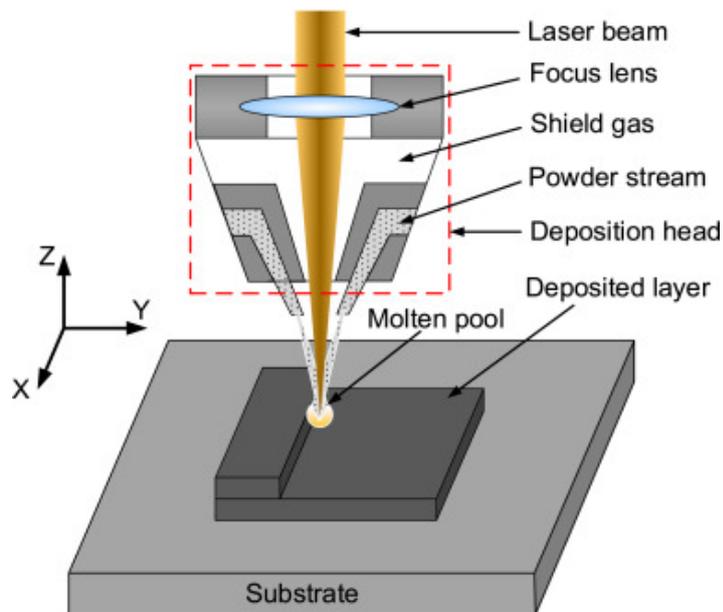
3DP functions placing powdered material on a substrate that is selectively joined using a binder diffused through a nozzle. The material is first stabilized by misting with water droplets to avoid unnecessary disturbance when it is hit by the binder. Following the sequential application of layers, the unbound powder is removed. The part may be further processed by subjecting it to firing at high temperatures to strengthen the bonding further. This process may be applied to the production of metal, ceramic, and metal/ceramic composite parts. 3DP offers the advantage of rapid fabrication, and low materials cost (Huang et al., 2013). In fact, it is probably the fastest of all AM processes. However, there are some limitations, such as rough surface finish, size limitation, and high cost.



**Figure 1.5:** Graphical drawing of the 3DP process. An inkjet print head deposits a liquid binder onto each layer of powder individually to build a 3D part. Courtesy of CustomPart Net.

### Laser Engineered Net Shaping (LENS)

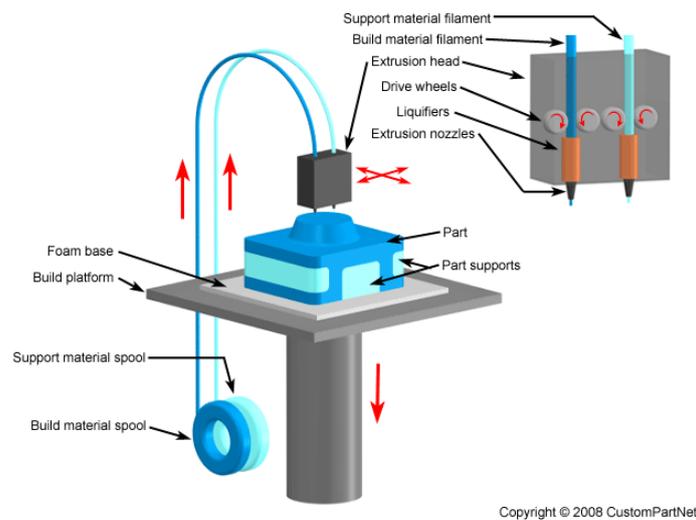
The LENS process is an injected powder process that allows the fully automated manufacture of geometrically complex metal components from the CAD model. The dust jet is usually coaxial with the laser beam and perpendicular to the plane of the substrate towards which it is sprayed, however, the directions of the laser and the dust jet may also have a certain inclination, usually between  $0$  and  $45^\circ$ . The powders are sprayed onto a substrate by one or more nozzles and fused by a high-power laser beam.



**Figure 1.6:** Graphical representation of the nozzle of a LENS machine for additive manufacturing (Gibou et al., 2018).

### Fused Deposition Modelling (FDM)

FDM is a manufacturing process used for prototyping and small-scale production. FDM uses an additive function, depositing the material in layers, to form the part. A plastic or metal filament that is initially stored in rolls is inserted into a mouthpiece. The nozzle is above the melting temperature of the material and can be moved on three electronically controlled axes. The nozzle is usually moved by motors in steps or servomotors. The piece is constructed with fine material threads that solidify immediately after leaving the mouthpiece.



**Figure 1.7:** Graphical drawing of the FDM process. The material is forced to melt and to pass through the nozzle to be deposited on the 3D piece. Courtesy of CustomPart Net.

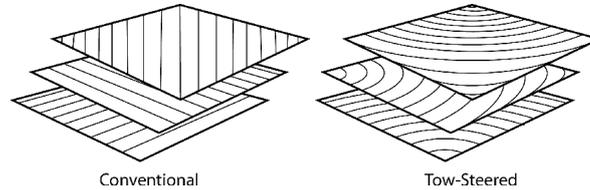
The FDM process begins with a software process, which starts from a stereolithographic file (.stl). The file is oriented so that it can be printed, divided into layers, and the trajectories that the nozzle must follow to deposit the material, layer by layer, to form the piece are calculated. If necessary, support structures can be generated and, after the part is created, they are being removed. These support structures can even be made with other materials if the printer allows, for example, the change of print-head. The thermoplastic is melted and deposited by the nozzle in layers of the required thickness (finer layers mean better quality in the final piece but longer printing time) one by one. The layers are deposited from the bottom up.

Although FDM is a very flexible technology, and is capable of making very diverse parts, there are some restrictions on the characteristics of what can be made with this technique, especially as regards that you can not start printing in the air, without supports below, and the maximum slope of the overhanging or hollow parts.

## 1.2 Composite Materials Manufacturing

In this Section, the most important technique to tape continuous composite materials are presented. Only the manufacturing processes that are suitable to create large pieces

for the aeronautical world are described, and therefore techniques such as vacuum forming, pultrusion, bladder-assisted molding are not discussed.



**Figure 1.8:** Difference between conventional and tow-steered ply of composite (Brooks and Martins, 2018).

### 1.2.1 Filament Winding (FW)

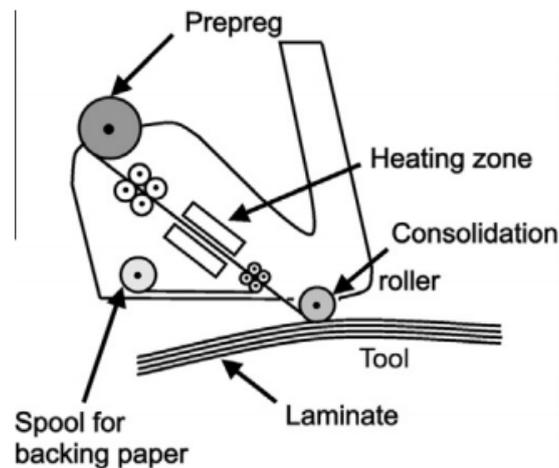
Filament Winding (FW) is a manufacturing technique for the production of composite materials, usually in the form of cylindrical structures.

The process consists of winding filaments subjected to various stresses on a male mold or mandrel. The mandrel rotates while a trolley moves horizontally, depositing the fibers according to the desired pattern. The most common filaments are made of carbon or glass and are embedded in synthetic resin while being deposited and wound. Once the mandrel is completely covered with the desired thickness, the mandrel is placed in an autoclave to solidify (cure) the resin. Once the resin has been cured, the mandrel is removed, leaving the final product with its inner hole.

The filament winding process is very suitable for automating the filament winding process, as this makes it possible to control the tension of the filaments precisely. If the filaments are placed with high tension, then a final product with higher rigidity and strength is achieved, while if the tension applied to the filaments is low, then more flexible products are obtained. It is possible to control the orientation of the filaments in such a way that successive layers are with different orientations from the previous. The angle at which the fiber is placed determines the mechanical properties of the finished product. Generally, high angles provide greater resistance against crushing, while a low-angle pattern (so-called closed or helical) provides higher tensile strength.

### 1.2.2 Automated Tape Laying (ATL)

Automated Tape Laying (ATL) has been developed since the 1970s as an automated version of manual tape laying and offers high productivity and reliability for simple or low complexity components. It is, in particular, highly productive for large simple flat components and able to handle high areal weight materials with few modifications. ATL can be interpreted as a form of additive manufacturing or inverse machining since the part is built up by adding material, as opposed to material removal during machining. The ATL head handles the prepreg tape, which is typically 75, 150, or 300 mm wide and supplied on a cardboard core similar to the prepreg used for the manual layup (Lukaszewicz et al., 2012). However, the prepreg can be modified for automated layup by changing the backing paper or degree of impregnation. Several commercially available systems store the prepreg material directly in the layup head. A picture of

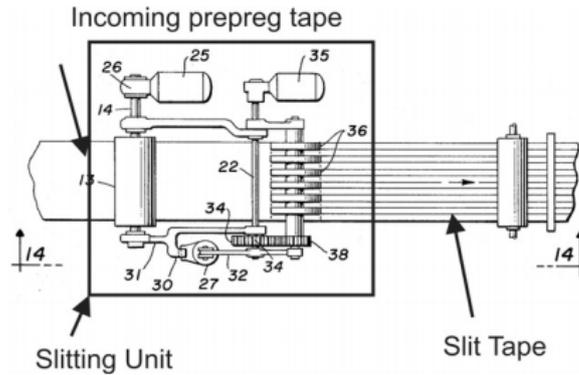


**Figure 1.9:** The head of a ATL machine (Lukaszewicz et al., 2012).

ATL head is shown in fig. 1.9. Due to the mass of both the head and material, as well as the size of the parts typically manufactured, ATL machines are mounted on straight gantries or a vertical column system. For most aerospace structures, courses consist of ramps and gaps as well as ply terminations, resulting in complex surface topologies. ATL's are Computer Numerical Control (CNC) systems that follow predefined paths accurately and reproducibly, allowing the elimination of layup errors that could occur in the manual layup. Material tolerances are commonly sufficiently small to reduce the impact of gaps on mechanical performance. At the end of the ply path, the print head decelerates just before finishing and cuts the tape, using rotating or pinching blades. The distance between the blade position and the roller contact point is called the "minimal course length", and it is used as a lower bound on the part sizes that can be manufactured. After cutting the tape, the remaining or minimal course length is delivered to finish the ply route. This process is replicated course by course until the ply is completed, the system is stopped by the program, user intervention, or if an automated fault detection system has recognized a layup error.

### 1.2.3 Automated Fiber Placement (AFP)

Automated Fiber Placement (AFP) systems were commercially introduced towards the end of the 1980s and were described as a logical combination of ATL and FW. AFP systems differ from ATL in the width of the material that is laid down with typical material widths of 3.2 mm, 6.4 mm, and 12.7 mm. However, AFP generally delivers several tows in a single sequence, termed tows. A band then forms a course, while a sequence of courses is called ply. Presently, AFP can deliver up to 32 tows in parallel at linear speeds of up to 1 ms<sup>-1</sup> (Lukaszewicz et al., 2012). The systems also tend to have higher acceleration in the linear axes with typical values around 2 ms<sup>-2</sup>. Rotational speeds and accelerations are more varied by the company and, therefore, not quoted. However, it is important to note that rotational speed and acceleration can have a great impact on layup productivity for complicated parts, and are therefore more relevant to AFP than ATL. The width and number of tows delivered depend strongly on the complexity



**Figure 1.10:** The head of a AFP machine (Lukaszewicz et al., 2012). The main difference with the ATL one is that it accepts multiple smaller tows.

and local geometry of the part that the course is to be laid over. Thus material width and tow count affect productivity. Each tow is normally driven individually and can be clamped, cut, and restarted during manufacture, enabling the production of complex parts. This makes it possible to deliver each tow at distinct speeds, enabling layup over complex geometries and tow steering, and is beneficial, for example, in structures such as fuselage sections with window cut-outs, or wing skins with numerous pad-ups and valleys (Lukaszewicz et al., 2012). While steering was initially conceived to improve layup over surfaces with double-curvature, the individual tow payout may improve productivity and reduced materials wastage rates, and mechanical properties, too. An important consideration is the amount of gap between the tows, which is much larger than for ATL and typically scales with the amount of steering. This may affect mechanical performance detrimentally and is often countered by transversely offsetting subsequent plies by half a tow-widths.

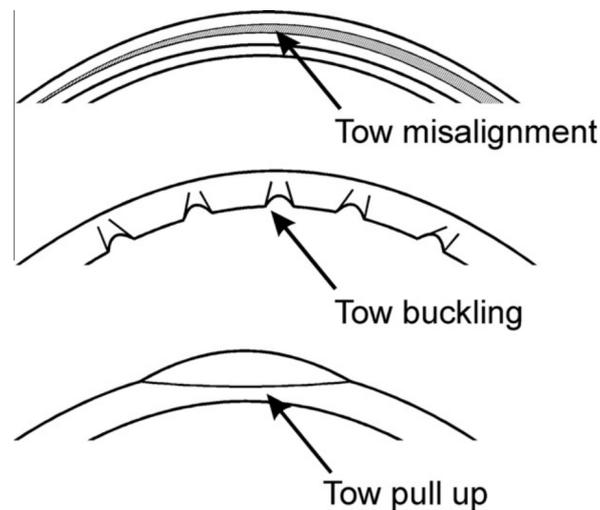
#### 1.2.4 Manufacturing Constraints and Defects

The available degree of steering in AFP and ATL layup is often reported to be the smallest possible radius fibers can be laid into without significant defect development, such as detachment from the tool and ply wrinkling. There are, in principle, three main tow steering defects: tow buckling, tow pull-up, and tow misalignment (see fig. 1.11) (Lukaszewicz et al., 2012). Twisted tows can also occur but are less common. Tow buckling occurs on the inner radius of a tow if pressing forces are too high. Similarly, tow pull-up happens on the outside of a tow due to excessive tensile forces. Ultimately, tow misalignment is the effect of variability in the layup system, layup control, or prepreg material.

Hale et al. (2002) reported the successful layup of AFP tows into radii as small as 50.8 cm compared to 610 cm for ATL layup of 150 mm wide tape.

Typical manufacturing constraints of AFP components are:

- A maximum curvature to respect in order to mitigate tow buckling and tow pull up.
- A minimum length of a tow must be deposited (fig 1.13).

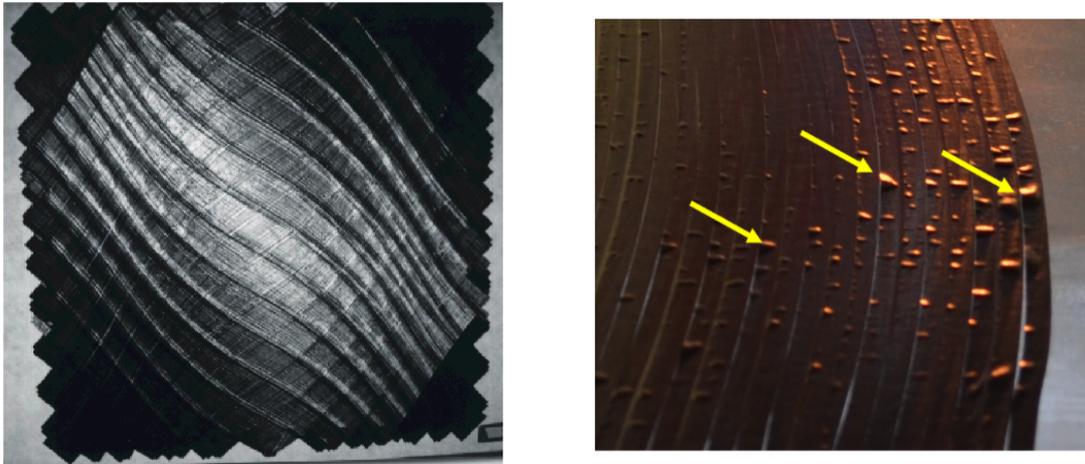


**Figure 1.11:** The three main defects that can occur when a tow is steered (Lukaszewicz et al., 2012).

- The towpaths must be as parallel as possible to avoid excessive misalignment and gaps.

### 1.3 Additive Manufacturing of Composite Materials

Continuous fiber-reinforced thermoplastic composites are becoming more significant in industrial applications because of their inherent advantages, such as excellent mechanical and chemical performance, recycling, and potential light-weight structures. For years, several fabrication methods for Carbon Fiber Reinforced Polymer (CFRP) have been developed, such as vacuum forming, filament winding, pultrusion, bladder-assisted molding, ATL, and AFP. Nevertheless, the limitations of these processes lie in the high cost of molds, the manufacturing boundedness of the parts with complex constructions, and the inability of particular fiber alignment, leading to the bottlenecks for the full applications of CFRP in industrial production and people's daily life. FDM, as one of the most commonly used low-cost 3D printing technologies, has been a significant method to realize the transformation from conceptualization to the products. The parts can be manufactured rapidly and directly from CAD model without geometry limitation and specific tooling and with high material utilization. However, the mechanical properties of products fabricated by conventional FDM 3D printing are intrinsically weak because of the thermoplastic polymers used, although the optimization of processing parameters, such as the printing direction and laminate thickness, has been investigated to improve the mechanical properties of thermoplastic resin parts (Matsuzaki et al., 2016). 3D printing has primarily been used for trial products or toys, without application to the manufacture of structural components for aerospace or automotive products. Expanding the applicability of 3D printing to obtain mechanically robust components for aerospace and automotive structures is an essential goal of industrial fabrication. In the FDM process, the polymeric fiber is continuously fed into the outlet and heated to a semiliquid state, and then the thermoplastic material

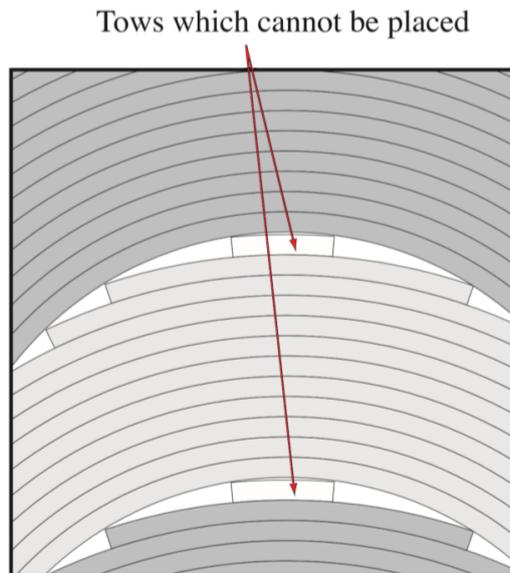


(a) Thickness build up due to overlaps (Brooks and Martins, 2018). (b) Tow puckering for curved fiber path (Tatting and Gurdal, 2002).

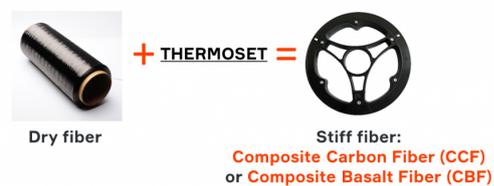
**Figure 1.12:** Example of defects of an AFP piece due to excessive curvature of the tow steer or to an incorrect planning of the trajectories: overlap (left) and puking (right).

is extruded onto the previous layer along the cross-section contour and the filling trajectory. At the same time, the extruded material rapidly solidifies and adheres with the surrounding material to accumulate the plastic parts. Hence, in comparison with the conventional fabrication process of composites characterized by impregnation and crystallization of the matrix, FDM provides a possibility for manufacturing complex functional and structural parts with CFRP.

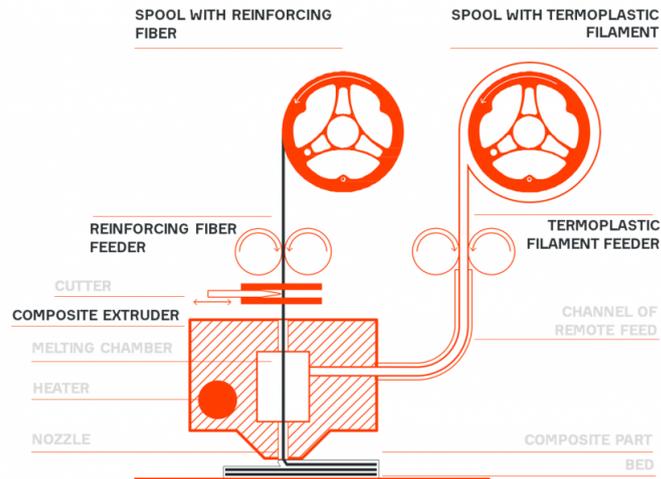
Hybrid approaches between AFP and FDM are described by Rakhshbahar and Sinapius (2018). The research is focused on the mitigation of gaps when using AFP. The gaps are especially visible on complex and double-curved surfaces and reduce the mechanical properties of the composite part. To cover the unavoidable weak area of this effect, a plurality of fiber composite layers is laid on top of one another to improve the mechanical properties of the component. This makes the elements heavier and more expensive to manufacture. In this new method, the holes are detected by the profile sensor after placement of the tape on the mold. The gaps are filled with the aid of a 3D printer with carbon continuous-fiber reinforced plastics. By combining the 3D printing and AFP technology, composite elements can be manufactured more homogeneously. Subsequently, the components are produced faster, cheaper, and even lighter because of the avoidance of the additional layers.



**Figure 1.13:** Physical representation of the minimum length constrain for an ATL or an AFP machine (Brooks and Martins, 2018).



**Figure 1.14:** The carbon fiber is combined with a thermoset matrix to produce the prepreg then cured by the nozzle just before the deposition. Credits to Anisioprint.



**Figure 1.15:** Graphical drawing of the printhead of the modified continuous fiber FDM machine. Credits to Anisioprint.



**Figure 1.16:** A continuous fiber composite printed in 3D. Credits to Anisioprint.

# Chapter 2

## Optimization Algorithms

The scope of this Chapter is to present the problem of fiber trajectories optimization from a mathematical point of view. In the first two sections (2.1 and 2.2), the general optimization algorithms, and gradient-based optimization techniques are presented, explaining why they are chosen among other types of optimizers. In section 2.3, the concept of topology and fiber angle optimization is discussed, and the differences and points in common are listed. In the last section (2.4), a brief state-of-the-art analysis is carried out, presenting the most important discoveries in the field of optimization of fiber trajectories and focusing on why there is the need to develop this topic further.

### 2.1 Numerical Optimization

Numerical optimization consists in the use of algorithms to minimize or maximize a given function by varying several variables. The problem might or not be subject to constraints.

Since many engineering design problems seek to minimize or maximize an objective function linked to some measure of performance, the application of these algorithms became extremely useful. For example, a typical structural design optimization problem is to minimize the thickness of some components subject to maximum stress constraints achieving more lightweight parts. Multidisciplinary Design Optimization (MDO) emerged following the success of the application of numerical optimization techniques to structural design. Early aircraft design was one of the first applications of MDO because there is much to be gained by the simultaneous consideration of the various disciplines involved (structures, aerodynamics, propulsion, stability and controls, aeroelasticity, etc.), which are tightly coupled. Supposing that the algorithm saves a unit of weight in the structure because of the coupled nature of all the aircraft weight dependencies and the reduction in induced drag, the total reduction in the aircraft's gross weight is several times the structural weight reduction (about 5 for a typical airliner) (Martins, 2012).

### 2.1.1 Terminology and Problem Statement

The formulation of an optimization problem is an important step that permits to avoid common conceptual errors, as confusing constraints with objective functions. If the problem formulation is wrong, the solution could fail or simply return a mathematical optimum that is not engineering-wise feasible.

#### Objective Function

In the convention of numerical optimization, the objective function  $f$  is the scalar that we want to minimize. If one would instead maximize a function, it is simply inverted or multiplied by  $-1$ . This new function measures the "badness" that one would minimize. Common objective functions in the structural design are the weight or the structural stiffness of a structure. The objective function can be modeled as an explicit function or even be the result of a highly complex computational procedure. The choice of the objective function is usually trivial, and an error could lead to a minimum that is non-optimal from the engineering point of view, no matter how efficient or precise is the optimization scheme used.

Optimization problems are classified in the literature with respect to the objective function by determining how the function is linked to design variables (linear, quadratic, or generally non-linear).

If one would optimize a part using multiple objective functions, it is possible, but this results in a family of optimum designs with different emphasis on the various objectives. Usually, it is far better to convert these different objectives into constraints. In the end, it is possible to make only one thing best at a time.

#### Design Variables

The design variables  $x$  are the parameters that the optimizer algorithm can change to minimize the objective function. Design variables could be continuous or discrete if only some distinct values are allowed (for example, only a certain size for a hole in a structural analysis). The optimization problem formulation allows for the lower and upper boundary for each design variable.

#### Constraints

The constraints are function used to restrict in some way the design variables. They are used to prevent the algorithm from finding a numerical minimum that it is not feasible because of physical and engineering constraints. As in the case of the objective function, the constraint functions can be linear, quadratic, or generally non-linear and different algorithms must be used to take into account the differences.

The constraint functions can be further classified: if the design variables are restricted to be equal to a fixed quantity, the function is called *equality constraint*. When instead the design variables are enforced to be greater or equal to a certain quantity the function is called *inequality constraint*.

### Optimization Problem Statement

Once that the major components of the problem are defined, it is now possible the formulation of the optimization problem.

$$\begin{aligned} & \text{minimize : } f(x) \\ & \text{by varying : } x \in [l^-, l^+] \\ & \text{subject to : } c_e(x) = 0 \\ & \quad c_i(x) > 0 \end{aligned}$$

where

- $f(x)$  is the objective function to minimize
- $x$  is the vector of design variables bounded between  $l^-$  and  $l^+$
- $c_e$  are the equality constraints and  $c_i$  the inequality constraints.

## 2.2 Gradient-based Optimization

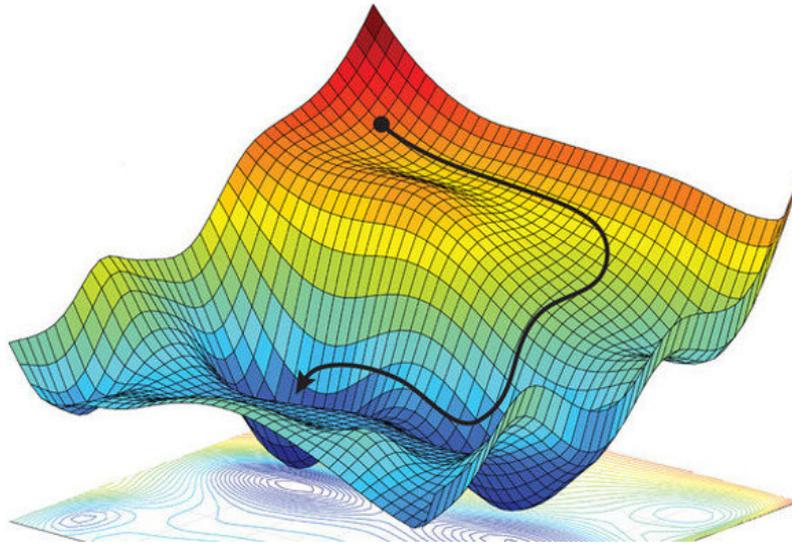
Gradient-based optimization or Gradient descent is a first-order iterative optimization algorithm for finding the local minimum of a differentiable function. A step proportional to the negative of the gradient (or approximate gradient) of the function at the current point is taken to find a local minimum of a function using gradient descent.

A simple allegory can illustrate the basic intuition behind gradient descent. A person is at the top of a hill and is trying to get down and trying to find the global minimum. The visibility is extremely low, and therefore the path down the mountain is not visible, so he must use local information to find the minimum. He can use the method of gradient descent, which involves looking at the steepness of the hill at his current position, then proceeding in the direction with the steepest descent. Using this algorithm, he would find his way down the mountain or possibly get stuck in some hole (local minima). Let us assume that the steepness of the hill is not natural to calculate, but instead, it requires an instrument to measure that demands much time to use. Thus, the person should minimize the use of the instrument if he wanted to get down the mountain quickly (see img. 2.1).

The difficulty then is choosing the frequency at which he should measure the gradient to not go off track. In this analogy, the person represents the algorithm, and the path represents the sequence of steps that the algorithm will suggest. The steepness of the hill represents the slope of the surface at that point. The instrument used to measure steepness is differentiation using, for example, finite differences.

For large numbers of variables  $n$ , gradient-based methods are usually the most efficient algorithms (Martins, 2012; Sigmund, 2011).

For the sake of simplicity, only the optimization of smooth unconstrained problems is described here. The vast majority of gradient-based algorithm uses two different major steps. The first one uses the gradient of the objective function to determine the most promising direction along which the minimum has to be searched. The second



**Figure 2.1:** Graphical representation of the path followed by the person (current iteration) on the top of a hill (function to minimize) that simulate the gradient descent algorithm (Amini et al., 2018).

one uses a line search algorithm to find the best point in this direction in  $n$ -dimensional space. The design variables are updated at each iteration  $k$  using

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

where  $p_k$  is the search direction given by the gradient, and  $\alpha_k$  is the step length given by the line search algorithm. Line search accounts for additional iterations that are usually not taken into account in the major iteration count. By the way, this is an important distinction that needs to be considered when comparing different approaches. Even if line search algorithms greatly affect the quality of gradient-based schemes, in the literature they are classified based on the method that is used for computing the search direction.

---

**Algorithm 1:** General algorithm for smooth functions.

---

**input** : Vector of starting points  $x_0$  and tolerances  
**output:** Vector of optimized design variables  $x^*$   
 $k \rightarrow 0$ ;  
 evaluate  $f(x_0)$ ;  
**while** *tolerances not met* **do**  
     compute search direction  $p_k$  ;  
     find step length  $\alpha_k$  such that  $f(x_k + \alpha_k p_k) < f(x_k)$ ;  
     update the vector of design variables  $x_{k+1} \rightarrow x_k + \alpha_k p_k$ ;  
      $k \rightarrow k + 1$ ;  
**end**

---

### 2.2.1 Gradients and Hessian

In vector calculus, the gradient of a scalar-valued differentiable function  $f(x)$  of several variables  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is the vector field, or more simply a vector-valued function  $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , whose value at a point  $p$  is the vector whose components are the partial derivatives of  $f$  at  $p$ .

$$\nabla f(p) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(p) \\ \vdots \\ \frac{\partial f}{\partial x_n}(p) \end{bmatrix}.$$

The gradient is closely related to the derivative, but it is not itself a derivative: the value of the gradient at a point is a tangent vector: a vector at each point; while the value of the derivative at a point is a cotangent vector: a function of vectors at each point. They are related because the dot product of the gradient of  $f$  at a point  $p$  with another tangent vector  $v$  equals the directional derivative of  $f$  at  $p$  of the function along  $v$ . The gradient can be interpreted as the "direction and rate of fastest increase". If at a point  $p$ , the gradient of a function of several variables is not the zero vector, the direction of the gradient is the direction of the fastest increase of the function at  $p$ , and its magnitude is the rate of increase in that direction. In the multivariate case, the gradient vector is always perpendicular to the hyperplane tangent to the contour surfaces of constant  $f$ .

Higher derivative of multi-variable function functions are needed by some particular algorithm and can be defined as in the single-variable case. It is however important to note that the size of the matrix increase by an order of  $n$ . Suppose  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a function taking as input a vector  $x \in \mathbb{R}^n$  and outputting a scalar  $f(x) \in \mathbb{R}$ . If all second partial derivatives of  $f$  exist and are continuous over the domain of the function, then the Hessian matrix  $\mathbf{H}$  of  $f$  is a square  $n \times n$  matrix, usually defined and arranged as follows:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

or, by stating an equation for the coefficients using indices  $i$  and  $j$ :

$$\mathbf{H}_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}.$$

### 2.2.2 Real algorithms

The classification of a method is based on how the best direction  $\alpha_k$  is evaluated. The most common gradient-based algorithms are

#### Steepest Descent Method

: This technique uses the gradient vector at  $x_k$  as the search direction for every major iteration  $k$ . The gradient vector is always orthogonal to the plane tangent to the isosurfaces of the function. Once being normalized, the direction  $p_k$  is used to complete the line search to achieve the step length  $\alpha_k$  finally. This rate of change is given by the norm of the gradient  $\|g(x_k)\|$ . The steepest descent direction at each iteration is orthogonal to the previous. Because of this characteristic of the search directions, the steepest descent method "zigzags" in the design space and is ineffective. Although a significant decrease may be observed in the first few iterations, the method is usually very slow to meet at a local optimum. The rate of convergence is linear. To stop the optimizer, three different convergence tolerances are used  $\epsilon_a, \epsilon_r$  and  $\epsilon_g$  (respectively absolute, relative and gradient tolerance).

$$|f(x_{k+1}) - f(x_k)| < \epsilon_a + \epsilon_r |f(x_k)|$$

$$\|g(x_{k-1})\| < \epsilon_g$$

Here is the summary of the algorithm:

---

#### Algorithm 2: Steepest Descent algorithm

---

**input** : Vector of starting points  $x_0$  and tolerances  $\epsilon_a, \epsilon_r$  and  $\epsilon_g$   
**output**: Vector of optimized design variables  $x^*$   
 $k \rightarrow 0$ ;  
 evaluate  $f(x_0)$ ;  
**while** *tolerances not met* **do**  
     compute the normalized search direction  $p_k \rightarrow -g(x_k)/\|g(x_k)\|$ ;  
     find step length  $\alpha_k$  performing line search;  
     verify that  $f(x_k + \alpha_k p_k) < f(x_k)$ ;  
     update the current point  $x_{k+1} \rightarrow x_k + \alpha_k p_k$ ;  
      $k \rightarrow k + 1$ ;  
**end**

---

#### Conjugate Gradient Method

The Conjugate Gradient method slightly modify the steepest descent taking into account the history of the gradient to move faster to the minimum. Usually, a restart is performed every  $n$  iterations for computational stability, using a steepest descent direction. The conjugate gradient method does not produce well-scaled search directions, so it is possible to use same strategy to choose the initial step size as for steepest descent.

---

**Algorithm 3:** Nonlinear conjugate gradient method

---

**input** : Vector of starting points  $x_0$  and tolerances  $\epsilon_a, \epsilon_r$  and  $\epsilon_g$   
**output**: Vector of optimized design variables  $x^*$   
 $k \rightarrow 0$ ;  
evaluate  $f(x_0)$ ;  
**while** *tolerances not met* **do**  
    **if**  $k=0$  **then**  
        Compute the normalized steepest descent direction  
         $p_k \rightarrow -g(x_k)/\|g(x_k)\|$   
    **else**  
        compute  $\beta_k \rightarrow \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}$ ;  
        compute the conjugate gradient direction  $p_k = -g(x_k)/\|g(x_k)\| + \beta_k p_{k-1}$ ;  
    **end**  
    find step length  $\alpha_k$  performing line search;  
    verify that  $f(x_k + \alpha_k p_k) < f(x_k)$ ;  
    update the current point  $x_{k+1} \rightarrow x_k + \alpha_k p_k$ ;  
     $k \rightarrow k + 1$ ;  
**end**

---

**Newton's Method**

The steepest descent and conjugate gradient methods only use the first derivative term in the Taylor series to obtain a local model of the function. Newton methods instead use a second-order Taylor series expansion of the function about the current design point.

---

**Algorithm 4:** Newton's method

---

**input** : Vector of starting points  $x_0$  and tolerances  $\epsilon_a, \epsilon_r$  and  $\epsilon_g$   
**output**: Vector of optimized design variables  $x^*$   
 $k \rightarrow 0$ ;  
evaluate  $f(x_0)$ ;  
**while** *tolerances not met* **do**  
    compute the gradient of the objective function  $g(x_k)$ ;  
    compute the Hessian of the objective function  $H(x_k)$ ;  
    compute the search direction  $p_k = -H^{-1}g_k$  find step length  $\alpha_k$  performing  
    line search;  
    verify that  $f(x_k + \alpha_k p_k) < f(x_k)$ ;  
    update the current point  $x_{k+1} \rightarrow x_k + \alpha_k p_k$ ;  
     $k \rightarrow k + 1$ ;  
**end**

---

As in the singular variable case, problems and even failure may occur when the quadratic model is a poor approximation of the function near the current point. Another disadvantage of Newton's method is the need to compute not only the gradient

but also the Hessian, which contains  $n(n + 1)/2$  second order derivatives.

### Quasi-Newton Method

Quasi-Newton methods create second-order information using only first-order information. An approximate Hessian based on the sequence of function values and gradients of previous iterations is calculated and used to perform a Newton step. The difference between the methods is in how they update the approximate Hessian.

### Trust Region Method

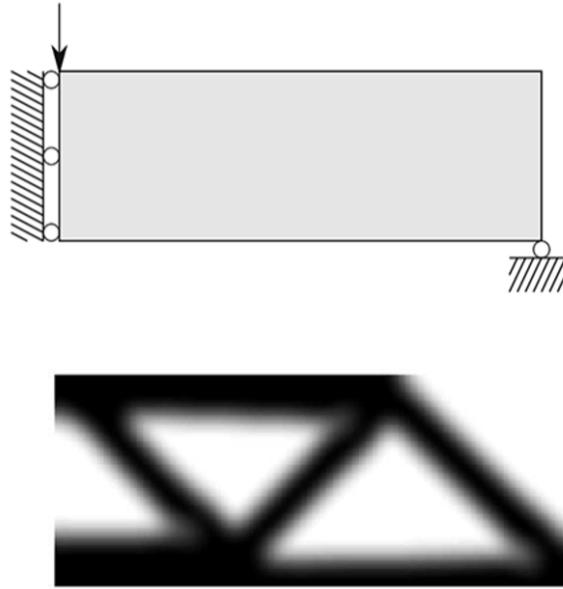
Trust region methods are a different approach to resolving the weaknesses of Newton's method, arising from a Hessian that is not positive definite or a highly non-linear function. These problems arise when the quadratic approximation is no more reasonable. Thus, a quadratic function within a region around  $x_k$  is minimized instead. A ratio  $r_k$  will after evaluating the goodness of the actual reduction to the predicted one.

### 2.2.3 Sensitivity Analysis

Sensitivity analysis is the optimization branch that studies how the inputs of the model control modifications in the outputs; it plays a key role in gradient-based optimization. Usually, it is possible to classify the sensitivity analysis between local and global analysis. Local sensitivity analysis aims to evaluate the response for a fixed set of inputs or everywhere where the uncertainties are pretty low. General sensitivity analysis quantifies the response for inputs that tend to span over a wide range of values, and it is better suited for models with high uncertainties, as non-physical models. Knowing the exact analytical sensitivity permit to save a lot of processing time, avoiding the use of finite differences in the gradient-based algorithm. The accuracy of the derivative computation affects the convergence behavior of the solver used in the algorithm. In the case of gradient-based optimization, accurate derivatives are important to ensure robust and efficient convergence, especially for problems with large numbers of constraints. The precision of the gradients limits that of the optimum solution, and inaccurate gradients can cause the optimizer to stop prematurely or to take a less direct route to the optimum.

## 2.3 Topology and Fiber angle Optimization

According to Bendsøe (1989): "Shape optimization in its most general setting should consist of a determination for every point in space whether there is material in that point or not.". Topology optimization is a mathematical method that optimizes the material layout within a well-defined design space for a set of Boundary Conditions (BCs), loads, and additional constraints given by the user (Sigmund and Maute, 2013). The goal is set defining an objective function that usually translates in better mechanical properties (see fig. 2.2). Usually, a Finite Element Method (FEM) analysis is carried out to evaluate the evolution of mechanical performance during the optimization. Topology optimization permits to achieve any shape within the design space

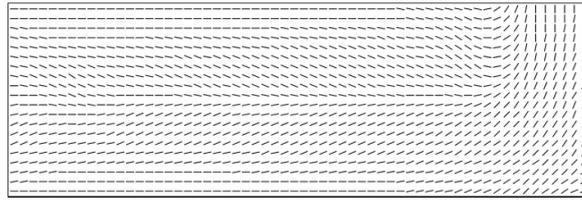


**Figure 2.2:** Graphical representation of the topology optimization of a MBB symmetric beam using Solid Isotropic Material with Penalization method (SIMP) (Labanda, 2014).

instead of dealing with predefined configuration, and this is a clear advantage over the more classical shape optimization and sizing optimization. The shape is optimized using either gradient-based algorithms such as the ones described in Section 2.2, the method of moving asymptotes (Svanberg, 1987) or non-gradient-based algorithms such as genetic algorithms, even if each one has its drawback. The most common algorithm used to perform topology optimization is SIMP.

From the studies of mechanics of materials, it is well known that the maximum stresses occur in a particular specific direction. Thus, having uniform stiffness and strength in all directions of a component leads to an over-sizing of the material in the other directions. Topology optimization focuses on how to use the material to achieve better performances macroscopically. A completely different approach is carried out by Fiber Angle optimization that instead focuses on microscopic properties of the material. The primary purpose of the algorithm is to exploit the intrinsic anisotropy of composites materials to achieve better mechanical properties without changing the macroscopic topology (see img. 2.3). This new approach opens new possibilities of how to optimize different components. For example, there are plenty of pieces as the skin of the fuselage of a plane that simply cannot be optimized using topology optimization because their shape is highly linked to the function they need to perform. For such pieces, 3D printing linked to Fiber Angle optimization could potentially revolutionize the way they are conceived and manufactured.

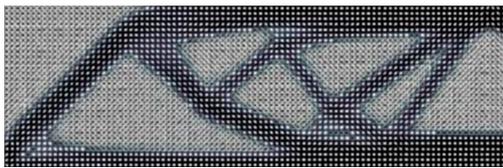
However, one must not consider these two topics as totally unrelated. The two algorithms are indeed linked when a composite piece that does not have substantial shape constraints is conceived to be produced using AM. On these components, the two algorithms work in synergy and produce pieces that are optimal from a micro and a macro point of view. To sum up, Topology optimization works best when there are



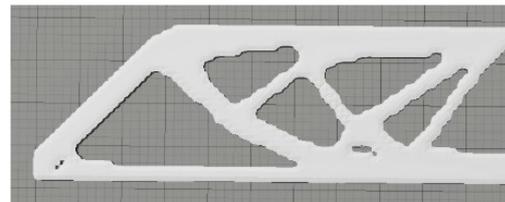
**Figure 2.3:** Result of a fiber angle optimization using Continuous Fibre Angle Optimization (CFAO) (Kiyono et al., 2017).

not substantial shape constraints, and the material is isotropic. When instead the material used is highly anisotropic, and the topology can't be changed, Fiber Angle optimization is the tool to use. Finally, the two algorithms can be used together if the material used is anisotropic, and the topology could be optimized.

Example of the two algorithms used together can be found in Luo et al. (2020); Jiang et al. (2019); Chandrasekhar et al. (2020) or in figs. 2.4 and 2.5.

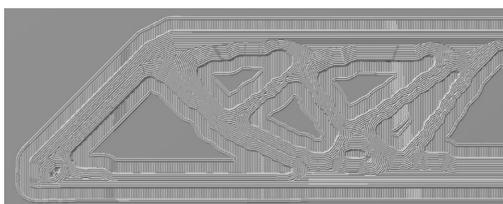


(a) Optimized topology with fiber direction angle for every element.



(b) Representation of the topology in the program used to slice the 3D object.

**Figure 2.4:** Results of the topology optimization linked with the fiber angle optimization by Jiang et al. (2019).



(a) Fiber path of the proposed optimization.



(b) Printed part ready to mechanical testing.

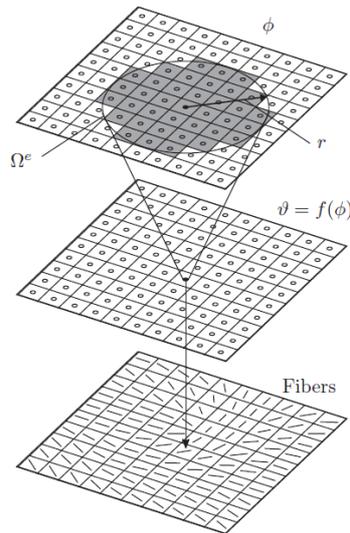
**Figure 2.5:** Printing process of the optimized topology by Jiang et al. (2019).

## 2.4 Precedent Works

The increasing availability of new materials and the accessibility to novel manufacturing processes have increased the interest in the application of composite structural parts in various engineering fields. AFP permits to print layer-by-layer composites components laying down the fiber creating custom shapes. AFP is an automated composites

manufacturing process of heating and compacting resin pre-impregnated non-metallic fibers on typically complex tooling mandrels. The fiber usually comes in the form of tow, a bundle of carbon fibers impregnated with epoxy resin. This process gets rid of the typical manufacturing constraint of straight parallel fiber. The concept of optimization applied to composite materials allows finding the optimum use of material (topology optimization) or/and the optimum fiber path for a given load. AFP permits to lay down single fiber on a custom path, and it opens the possibility to manufacture optimized composite parts.

In the literature, there is no unanimity on the best method to optimize the fiber path. There are different approaches possible: optimization of angle and position in the stack of different ply of composite material (Liu et al., 2004; Karakaya and Soykasap, 2009; Bohrer et al., 2015) or optimization of the fiber path on a single ply. In the first category falls the work of (Liu et al., 2004) which discuss how unstiffened composite panels are optimized by using flexural lamination parameters (Hahn and Tsai, 1980) as continuous design variables for the case in which the amounts of  $0^\circ$ ,  $45^\circ$ , and  $90^\circ$  plies are given. Later works use genetic algorithms and generalized pattern search algorithm for optimal stacking sequence of a composite panel (Karakaya and Soykasap, 2009). The problem has several global optimum configurations in the vicinity of local optima since it was decided to use fiber orientation angles as design variables instead of lamination parameters. The linearized buckling problem is convex in the lamination parameters space (Bohrer et al., 2015). Although techniques based on lamination parameters can be successfully used in different applications, its use makes it difficult to impose manufacturing constraints on the fiber angles, since they do not generate a sequence of the stacking plies (Kiyono et al., 2017).



**Figure 2.6:** Graphical representation of the spatial filter used by Kiyono et al. (2017).

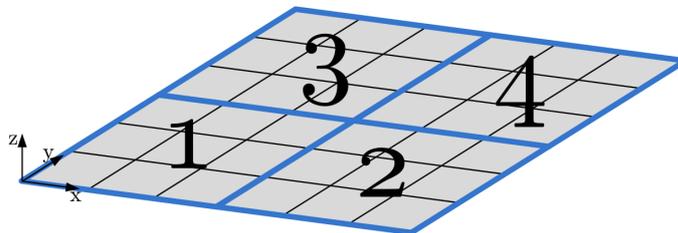
The allowable fiber angles of all of these studies have been restricted due to manufacturing to a finite set of values. Thus, given the discrete nature of the problem, the intensive use of the heuristic algorithm is found in the literature. However, Sigmund (2011) incites the scientific community to solve the optimization problem using

only gradient-based algorithms. He demonstrates how genetic algorithms are inefficient when numerous variables are involved. Gradient-based algorithms have been applied to optimize the fiber angle of elements in a continuum domain on a single ply. Luo and Gea (1998) introduced an energy-based method to determine the optimal orientation of orthotropic materials under static loading. An implicit method was used to estimate the effect of changing the fiber angle within an element on the stress and strain fields. However, as fiber angle optimization is a non-convex problem, solutions are influenced by the initial solutions.

Additionally, the solution tends to be hardly manufacturable due to an abrupt change of angle between adjacent elements or non-parallel fiber direction. To achieve fiber continuity and ensuring manufacturability, a spatial filter based on the projection technique could be used (Kiyono et al., 2017). Besides the optimization variables, another set of variables called design variables is defined for each element. The value of the orientation variables is then calculated as the mean of the design variables around the element (Fig. 2.6). Such a filter is advisable also to avoid stress concentration at discontinuous paths, which makes it challenging to converge in stress minimization problems.

Another approach found in the literature to solve the manufacturability problem is Xia and Shi (2017). The essential idea is to construct a continuous global function interpolating scattered design points to represent the fiber angle plan throughout the design domain. The Shepard interpolation method is applied to construct the interpolation function. The angles at discrete design points are taken as the design variables to be optimized. The Shepard interpolation essentially ensures spatial continuity of fiber angle.

To address the issue of high computation cost for the optimization, one could use a patch approach for the design variables. A patch is a group of elements that share the same set of design variables. This enables the engineers to use a fine mesh to obtain a good approximation of the displacement field while keeping the number of design variables low. In figure 2.7, a square plate is modeled with four patches where each patch covers nine elements. In this example, each element contains one lamina

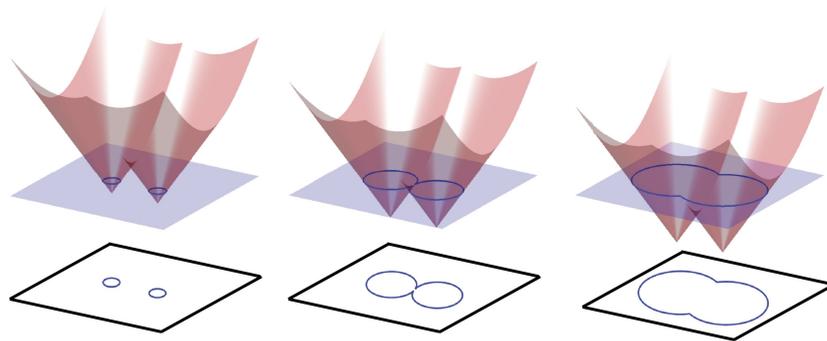


**Figure 2.7:** Graphical representation of a 2D mesh of 36 rectangular elements divided into 4 patches (Sørensen and Kann, 2011).

where the fiber orientation is applied as a design variable. The patch model thus has four design variables. If the patches were removed, the model would have 36 design variables, which would increase the amount of time needed for convergence. Another

benefit of applying a patch formulation is that the engineer may use less time on post-processing the optimized design. This is because larger areas of the structure will have the same value of the design variable, making it easier for the engineer to translate the result into a manufacturable design. The problem using a patch formulation is that each patch represents the average of the covering elements. Thus some of the elements in the patch may be assigned a value of the design variable, which may differ from what the optimum value is for that particular element. Thus the final design may not represent the same optimum design that one would obtain without the use of patches. It is thus up to the engineer to determine how small each patch has to be in order to obtain a good result, while also keeping the patch big enough so as to use less time on post-processing and waiting for the optimization to converge.

Aiming at addressing the issue of local optimal solution, Stegmann and Lund (2005) proposed the Discrete Material Optimization (DMO) method. The fiber angle at any element is a weighted sum of several discrete candidate angles. The weights are taken as the design variables. It is required in the resulting optimal laminate that only one of these weights is equal to one and the others zero, which means that only one of the candidate angles is selected. This method has gained much popularity, and several variants were proposed, for instance the Shape Functions with Penalization (SFP) (Bruyneel, 2011) and the Bi-Value Coding Parameterization (BCP) (Gao et al., 2013), which effectively reduced the number of design variables. However, the success of these methods relies on the ability of optimizer to drive the weights to their bounds: the intermediate value of weights or mixture of fiber angles has no physical meaning. Another limitation of these methods is that continuous spatial variation of fiber angle is not considered as of CFAO methods. Kiyono et al. (2017) proposes another DMO algorithm based on the optimized selection of discrete angles. The method uses the normal distribution function as the angle selection function, which requires only one variable to select the optimized angle among any number of discrete candidate angles.



**Figure 2.8:** Level-set representation of a free boundary (blue solid line) in two spatial dimensions, moving in its normal direction, and subsequent changes in topology that are handled automatically. The level-set function is depicted in red (Gibou et al., 2018).

Another promising technique that achieve fiber continuity is level-set method (Fig. 2.8), initially proposed by Osher and Sethian (1988). It has become popular in many disciplines, such as image processing, computer graphics, computational geometry, optimization, computational fluid dynamics, and computational biophysics. The advan-

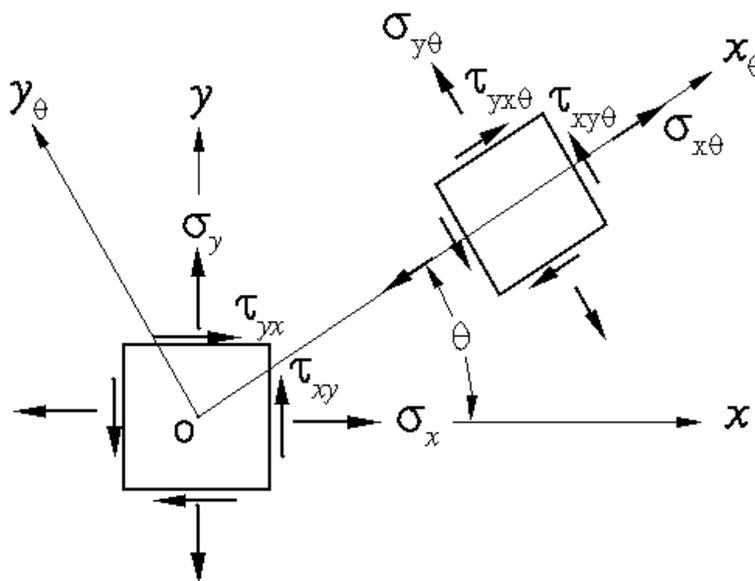
tage of the level-set method is that one can perform numerical computations involving curves and surfaces on a fixed Cartesian grid without having to parameterize these objects (Sethian, 1999). Recently the level-set method was successfully applied to address the fiber continuity (Brampton et al., 2015); however, the algorithm seems to be extremely sensitive to the initial angle configuration. Brampton et al. (2015) proposes to solve this problem by using the level set topology optimization solution for isotropic material as the initial fiber configuration, and thus, additional work needs to be done to achieve good fiber orientation solutions with the level set method. Liu and Yu (2017) developed another level set algorithm that also performs topology optimization, simplifying the sensitivity result by eliminating the domain integration term. The numerical results have proved this simplification to be reasonable because structural topology optimization produces thin rib-based design, and the majority of the level set contours follow the longitudinal direction, which well reinforces the ribs of the structure. Similar works can be found on Ranaivomiarana et al. (2019), Jantos et al. (2020), and Silva et al. (2020).

# Chapter 3

## The FEM Simulation

This chapter gives a description of the script used by the optimizer to evaluate displacements, strains, and the total elastic energy stored in the analyzed beam. In the first section, a brief recap for orthotropic materials is carried out using classic and new theories. Section 3.2 is instead dedicated to the numerical correlation between the FEM code in MATLAB and the commercial code ABAQUS, to verify the effective concordance of results. The numerical results are discussed in a separate chapter, in Section 5.2.

### 3.1 FEM Linear Structural Analysis



**Figure 3.1:** Components of stress in two dimensions. Credit engAPPLETS by VirginiaTech

The optimal orientation problem is solved for a two-dimensional linear elastic struc-

ture. The mesh is made up by bi-linear rectangular elements. In a two-dimensional orthotropic elastic material, according to image 3.1, it is possible to write

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix} \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{pmatrix} = \bar{Q} \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{pmatrix}$$

where  $\sigma$  and  $\epsilon$  represent respectively stress and strain components and  $\bar{Q}$  the rotated orthotropic stiffness matrix of the element. The relationship between the rotated matrix  $\bar{Q}$  and non-rotated matrix  $Q$  can be written using the standard rotational matrix  $T(\theta)$

$$\bar{Q} = T^T(\theta) Q T(\theta),$$

where

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} = \begin{bmatrix} 1/E_l & -\nu_{lt}/E_l & 0 \\ -\nu_{lt}/E_l & 1/E_t & 0 \\ 0 & 0 & 1/G_{lt} \end{bmatrix}^{-1}$$

$$T(\theta) = \begin{bmatrix} \cos^2 \theta & \sin^2 \theta & \cos \theta \sin \theta \\ \sin^2 \theta & \cos^2 \theta & -\cos \theta \sin \theta \\ -2 \cos \theta \sin \theta & 2 \cos \theta \sin \theta & \cos^2 \theta - \sin^2 \theta \end{bmatrix}$$

where  $E_l$  represent Young's modulus for the main axis and  $E_t$  for the perpendicular one.  $\nu_{lt}$  it is the Poisson's ratio. This approach was discarded because the aim of the algorithm is to be later extended to multi-ply composites and the lamination parameters better describe this class of materials.

### 3.1.1 Lamination parameters

According to Hahn and Tsai (1980) it is possible to define the **Directional Parameters**

$$U_1 = (3Q_{11} + 3Q_{22} + 2Q_{12} + 4Q_{66})/8$$

$$U_2 = (Q_{11} - Q_{22})/2$$

$$U_3 = (Q_{11} + Q_{22} - 2Q_{12} - 4Q_{66})/8$$

$$U_4 = (Q_{11} + Q_{22} + 6Q_{12} - 4Q_{66})/8$$

$$U_5 = (Q_{11} + Q_{22} - 2Q_{12} + 4Q_{66})/8$$

and the **Material Invariants** as well

$$T_0 = \begin{bmatrix} U_1 & U_4 & 0 \\ U_4 & U_1 & 0 \\ 0 & 0 & U_5 \end{bmatrix}, T_1 = \begin{bmatrix} U_2 & 0 & 0 \\ 0 & -U_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 0 & 0 & U_2/2 \\ 0 & 0 & U_2/2 \\ U_2/2 & U_2/2 & 0 \end{bmatrix}, T_3 = \begin{bmatrix} U_3 & -U_3 & 0 \\ -U_3 & U_3 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$T_4 = \begin{bmatrix} 0 & 0 & U_3 \\ 0 & 0 & -U_3 \\ U_3 & -U_3 & 0 \end{bmatrix}$$

The main characteristic of these parameters is that they are not dependent on the angle of the plies. They only rely on the engineering moduli ( $E, G$  and  $\nu$ ), and they are then a perfect candidate for the optimization problem. The matrix  $\bar{Q}$  could then be written as

$$\bar{Q} = T_0 + T_1 \cos 2\theta + T_2 \sin 2\theta + T_3 \cos 4\theta + T_4 \sin 4\theta \quad (3.1)$$

Usually, to define the constitutive matrix, lamination parameters are used instead of the rotation matrix  $T(\theta)$  because they are scalable to multilayered problems and provide a convex formulation of the optimization problem, as discussed in section 3.1.4. However, the use of lamination parameters makes it difficult to impose manufacturing constraints on the design variables, since they do not directly generate a direct description of the plies or its order. It is, in fact, necessary to add a further step where a Genetic Algorithm (GA) is used to generate the stack.

In case that all plies have the same material and thickness, for a membrane problem they are defined as follow:

$$\xi_{[1,2,3,4]}^A = \frac{1}{h} \int_{-h/2}^{h/2} \cos(2\theta(z)), \sin(2\theta(z)), \cos(4\theta(z)), \sin(4\theta(z)) dz$$

### 3.1.2 The Stiffness Matrix

In this section, the global stiffness matrix is calculated using a hybrid formulation. In fact, either the classic theory that uses the matrix 3.1 and lamination parameters have its pros and cons. Since the actual goal of the algorithm is to optimize a single ply, it was concluded that the use of the lamination parameters would have rendered the framework too complex. Even if the optimization algorithm with lamination parameters would have benefited from the convexity of the objective function (see subsection 3.1.4), the computational cost added by the use of a GA to generate the stacking ply is not justified. Therefore, a hybrid formulation using the aforementioned Material invariants and theta as a design variable is used. Even if it does not provide a substantial improvement in performance over the classic theory, it permits to easily upgrade the code to accept lamination parameters as design variables to optimize multiple plies later.

The global stiffness matrix  $K$  depends on the material orientation because it is an assembly of the local stiffness matrices of each element. The local stiffness matrix for

small displacement problems is commonly defined in linear elasticity as follows:

$$K^e = \int_{\Omega_e} B^T \bar{Q} B d\Omega_e \quad (3.2)$$

where  $B$  is the shape function derivative matrix, and  $\bar{Q}$  is the local material constitutive matrix.

Here, the matrices are obtained by analytical integration to reduce the computational cost caused by exact Gauss integration, typically used in conventional commercial FEM codes.

$$K^e = \begin{bmatrix} A\bar{Q}_{11} + B\bar{Q}_{33} + C\bar{Q}_{13} + C^T\bar{Q}_{31} & A\bar{Q}_{13} + B\bar{Q}_{32} + C\bar{Q}_{12} + C^T\bar{Q}_{33} \\ A\bar{Q}_{31} + B\bar{Q}_{23} + C\bar{Q}_{33} + C^T\bar{Q}_{21} & A\bar{Q}_{33} + B\bar{Q}_{22} + C\bar{Q}_{32} + C^T\bar{Q}_{23} \end{bmatrix}$$

where  $A$ ,  $B$  and  $C$  are 4x4 matrices defined as follows:

$$A = \frac{1}{12} \frac{\Delta y}{\Delta x} \begin{bmatrix} 4 & -4 & -2 & 2 \\ -4 & 4 & 2 & -2 \\ -2 & 2 & 4 & -4 \\ 2 & -2 & -4 & 4 \end{bmatrix}, \quad B = \frac{1}{12} \frac{\Delta x}{\Delta y} \begin{bmatrix} 4 & 2 & -2 & 4 \\ 2 & 4 & -4 & -2 \\ -2 & -4 & 4 & 2 \\ -4 & -2 & 2 & 4 \end{bmatrix}$$

$$C = \frac{1}{12} \begin{bmatrix} 3 & 3 & -3 & -3 \\ -3 & -3 & 3 & 3 \\ -3 & -3 & 3 & 3 \\ -3 & -3 & 3 & 3 \end{bmatrix}$$

The global stiffness matrix  $K$  is obtained by constructing the local stiffness matrices using the assembly operator (Akin, 1995)

$$K = \sum_{e=1}^{n^e} K_e.$$

### 3.1.3 Elastic Energy

The objective function of the optimizer to minimize is expressed as a function of the discrete displacements and stiffness matrix as presented in equation

$$J = \frac{1}{2} q^T K(\theta) q \quad (3.3)$$

, where  $q$ , represents the displacement field given solving the classic FEM equation

$$K(\theta)q = f.$$

### 3.1.4 On the convexity of the lamination parameter in the optimization problem

Once that the elastic energy is defined in eq. 3.3, is now possible to better understand why the choice of lamination parameters makes the optimization convex. As

seen in section 3.1.2, the stiffness matrix can be expressed in terms of the lamination parameters, and it is linear with respect to them.

$$J = \frac{1}{2}q(\xi^n)^T K(\xi)q(\xi^n) \quad (3.4)$$

where

$X(\xi)$  indicates that  $X$  depends on  $\xi$  in first order.

$X(\xi^n)$  indicates that  $X$  depends on  $\xi$  in some order  $n$ .

The elastic energy stored in the structure depends on lamination parameters in an order of  $2n + 1$ . However, Svanberg (1994) showed that stiffness optimization problems in a finite element context are convex if the stiffness matrix is linear with respect to the design variables.

## 3.2 ABAQUS Comparison and Validation

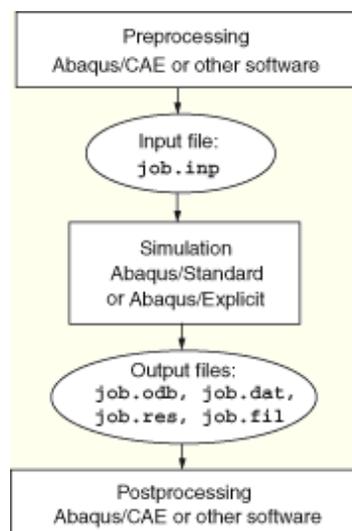
In this section, a comparison between the Matlab code and a commercial FEM code (Dassault Systemes ABAQUS) is carried out to validate the analysis.

### 3.2.1 ABAQUS

Abaqus/CAE® is a powerful software tool for simulating various engineering problems involving dynamic forces, vibrations, thermal effects, and impact loading. As other FEM software, the program requires a few steps to be completed to simulate a model successfully.

1. **Draw or import the CAD file.** The user is free to choose to create the parts in ABAQUS itself or to import the CAD models from another 3D CAD software package. If one would draw the component inside ABAQUS, he works in the *Part* module of the software.
2. **Create the material data.** The material data could consist of elastic, plastic, thermal, and failure data. For the stiffness optimization of a part, only the linear elastic component is used. The module to use is *Property*.
3. **Create an assembly.** Even when the model only consists of one part, an assembly of parts has to be created. In the *assembly* module, all the individual parts of the model are combined. If the model consists of multiple parts, interactions have to be implemented between these parts. These interactions are, e.g., contact forces, welds, and fluid contacts.
4. **Define the steps of the simulation.** Besides the initial step, in which the model initialization takes place, at least one additional step has to be created to be able to perform a simulation. This can be achieved using the *Step* module of the software.

5. **Define loads and boundary condition.** The boundary conditions can apply to the initial step; however, loads require to start at one of the additionally defined steps. Additional multi-body constraints can be applied, as of friction and magnetic interaction. The modules to use are *Load* and *Interaction*
6. **Select a type of element and mesh the assembly.** The defining of the mesh is a crucial step since the mesh type, and its size determines the outcome and the time to complete a simulation significantly. This is achieved using the *Mesh* module of ABAQUS/CAE.
7. **Create the job file.** The creation of the job is straightforward, creating a file for the ABAQUS environment to use to write results in the *Job* module. Which results are written to the output file has to be defined manually. Possible outputs are von Mises stresses, linear elastic strains, and energy stored in the element.
8. **Run the analysis.** The analysis can be run directly in Abaqus/CAE® or using the .inp file.
9. **Post-processing.** The visualization of the results and the creation of charts, tables, and graphs can be achieved using the *Visualization* module.

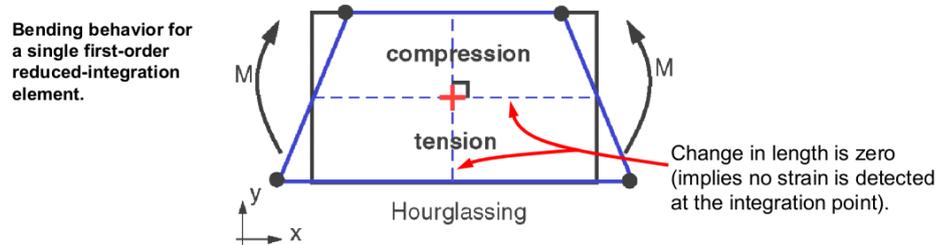


**Figure 3.2:** Resume of the steps performed by ABAQUS to pre-process, simulate, and post-process a model.

### 3.2.2 The choice of the type of element

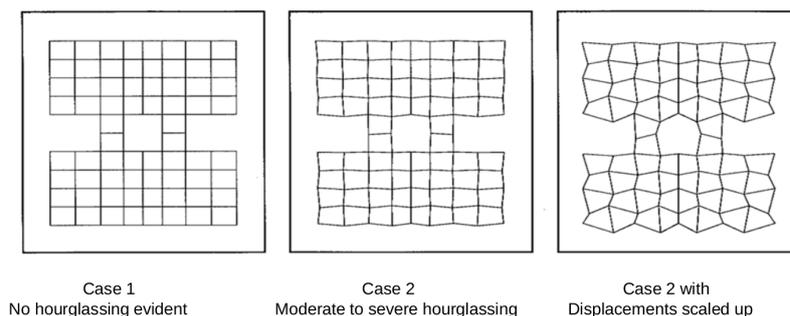
In Abaqus/CAE® each part can be meshed using a different element type, and each type has its code. The most used element types are S4R, C3D8R, B31, M3D4R, and C3D4. These codes stand for the family of the part, the number of nodes, and the type of integration. For example, S4R is a Shell 4-nodes Reduced integration, and B31 stands for Beam 3-D 1st order interpolation. When creating a part, the user initially

chooses between 2D and 3D elements, after which the choice is made between shell, solid, wire, or point shapes.



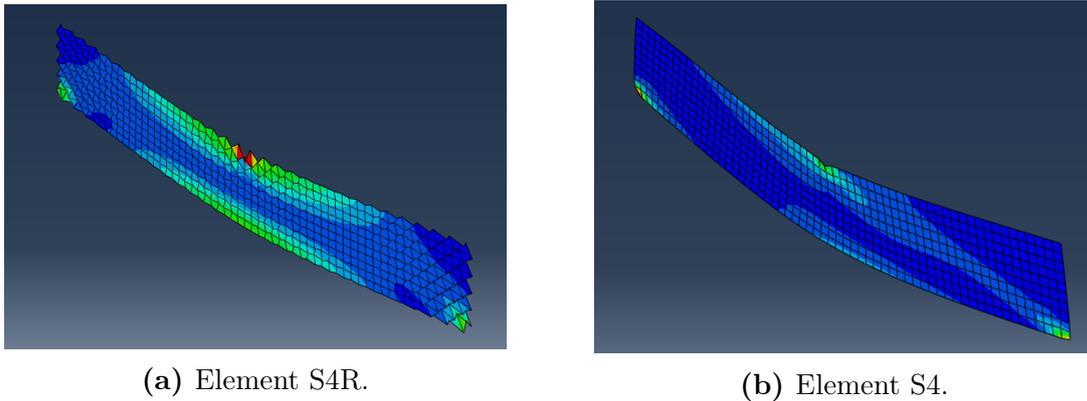
**Figure 3.3:** Graphical representation of a 2D reduced integration element. Image credits: ABAQUS Documentation.

Let us consider a 4 node quadrilateral shell element. In ABAQUS, this is called an S4 element. It may have either 4 or one number of integration points (depending on if full or reduced integration is chosen). Stresses and strains are captured by integrating the values that the element has at these integration points. One must remember that values are more accurate at the exact location of the integration points and not so accurate as we move away from them. For the sake of lesser computational time, it is possible to use the S4R element, in which R means Reduced Integration and where only one integration point is used, and it is ideally at the center of the element. In the case of these Reduced Integration elements, while post-processing results, it is possible to observe a false or fake deformation mode (zig-zag lines). Such a thing is the Hourglass phenomenon. The image 3.1 is an example of a single integration point shell element. The values captured at the single integration point remain the same as the dotted lines have not changed in magnitude or angle. However, the element as a whole has shown a change in configuration. This is because elements with minor integration points are less stiff as there is no stiffness in this deformation mode.



**Figure 3.4:** Graphical representation of the hourglass phenomenon over a mesh of 2D reduced integration elements. Image credits: LS-DYNA Documentation.

To mimic precisely the behavior of the code presented in Section 3.1, the shell S4 is chosen, a 4-node doubly curved thin or thick shell finite membrane strains. The S4 was chosen instead of the more common S4R because, in such a simple problem, there is not the need to reduce the computation time by introducing fictitious external forces that could introduce some error in the energy computation. There is no



**Figure 3.5:** Comparison between elements S4 and S4R for a simple 3 points bending problem. Even if the element S4R have the Hourglass control, it still suffer from massive deformation of the mesh. The S4 instead, using 4 different nodes for the analysis shows better results.

need to use an element that achieves reduced integration and hourglass control. Even the ABAQUS documentation suggests that in elastic bending problems with a coarse mesh, the hourglass energy may be higher than the recommended limit. This does not necessarily mean the results are adversely affected; however, one must use engineering judgment to assess the validity of the results.

Since the MATLAB FEM simulation doesn't allow the nodes to move out of the plane, the same logic is implemented in ABAQUS changing the "Section Integration" option of the 2D beam to "Before Analysis" and choosing "Membrane Only" to account only for the membrane behavior of the model.

# Chapter 4

## Filtered Continuous Fiber Angle Optimization

In this Chapter, the optimizer algorithm is presented and described. The first Section 4.1 describes the problem and the algorithm used to solve it. Section 4.2 focuses on the gradient-based optimization and the interpretation of its results. A spatial filter is used to enforce the continuity between near elements. The goal of Section 4.3 is to present how to generate a Level-Set surface for the problem. Its iso-lines are the fiber trajectories that approximate the angles found by the elemental optimization. The main challenge is to generate trajectories as smooth as possible without moving far away from the optimum. In the last section (4.4), manufacturing constraints - for instance, the need for equispaced fibers - are implemented using an algorithm coming from computer graphics called the Heat Method.

### 4.1 Problem definition

The objective of the optimization is to maximize the stiffness of a 2D plate printed with a fiber that has a diameter  $h$ . This problem is equivalent to minimize the total elastic energy resulting from the applied loads. Let  $\mathbf{G}$  be a rectangular domain  $[0, a] \times [0, b]$  with equally spaced grid made by  $N \times M$  elements. A discrete field  $\theta(x, y)$  that represents the direction of the fiber is defined for every element of the mesh.

The algorithm can be divided into three different independent sub-routine

- **Gradient descent optimization:** Optimization routine that uses a gradient descent method to minimize the total elastic energy of a 2D plate. This is achieved by modifying the angle of the fiber in every element of the mesh. A Gaussian filter is used to promote fiber continuity.
- **Creation of a level-set surface:** Taking the discrete optimized angles as the input, the algorithm generates a 3D surface that, once sliced with horizontal planes, gives an approximation of fibers trajectories.
- **Fiber propagation:** Algorithm that takes the continuous trajectories as an input, it propagates them and creates the points that the machine must follow

to print the layer. Use of the computer graphics algorithm named Heat Method: robust, linear, and easy to parallelize.

The following algorithm describes the step taken to optimize and generate the code used to 3D print the component.

---

**Algorithm 5:** Fiber Continuous Fiber Angle Optimization

---

**Result:** Find the optimized fibers' trajectories in a 2D plate under a specified load.

initialization of the load case and the mesh;

initialization of the continuity filter;

optimization of the angle  $\theta(x, y)$  for every element of the mesh;

creation of continuous trajectories over the entire domain;

select a number  $n$  of generating fiber;

**for**  $n = 1 : N_{n,k}$  **do**

    propagate the fibers in every direction using the fiber diameter  $h$ ;

    perform FEM analysis;

    compare elastic energy;

**end**

choose best  $i$  cases and write G-Code;

---

## 4.2 Gradient Descent Optimization

The algorithm aims at finding the optimal trajectories of the fibers of a composite in-plane loaded plate. A gaussian spatial filter is used to achieve fiber continuity, and a parallel *multistart* approach is implemented to reduce the influence of initial conditions.

### 4.2.1 Problem Definition

In fiber path optimization literature, two main methods are found. The former is known as DMO and accepts only a discrete set of angles for every element of the structure. The latter is CFAO, and it permits to describe the angle  $\theta(x, y)$  of the fiber in the elements continuously. On the other hand, increased sensibility to the initial conditions and more local minima are to be taken into account. Both use FEM analysis applied to structured grids with bi-linear shape functions. Static linear analysis is carried out to find the discrete displacement field  $q$ . The problem to solve is

$$K(\theta)q = f$$

where  $K$  is the global stiffness matrix, and  $f$  is the discrete applied force. The CFAO approach is chosen because it can adequately describe the uninterrupted and unconstrained paths that a FDM machine can perform. Each element has a material direction, and a gradient descent method is used to reduce the total elastic energy iteratively.

### 4.2.2 Optimization procedure

Let us define the optimization procedure as follow

$$\begin{aligned} & \text{minimize : } J = \frac{1}{2}q^T K(\theta(x, y))q \\ & \text{subject to : } \begin{cases} f = K(\theta(x, y))q \\ -\pi/2 \leq \theta(x, y) \leq \pi/2 \end{cases} \end{aligned}$$

The scalar field  $\theta(x, y)$  is the design variable. A gradient descent method is used to have the angles evolving towards minimal compliance iteratively. This optimization procedure is implemented in MATLAB through the *fmincon* function in the optimization toolbox. The algorithm used is *interior point*.

### 4.2.3 Gradient Calculation

The gradient of the optimization is calculated analytically, evaluating the derivative of the objective function ( $J$ ) with respect to the design variable ( $\theta(x, y)$ ).

$$\frac{\partial J}{\partial \theta} = \frac{\partial}{\partial \theta} \left( \frac{1}{2}q^t K q \right) = \frac{1}{2} \frac{\partial q^t}{\partial \theta} K q + \frac{1}{2} q^t \frac{\partial K}{\partial \theta} q + \frac{1}{2} q K \frac{\partial q}{\partial \theta} \quad (4.1)$$

where, remembering that  $q = K^{-1}f$  and  $q = f^t K^{-1}$

$$\begin{aligned} \frac{\partial q}{\partial \theta} &= \frac{\partial K^{-1}f}{\partial \theta} = \frac{\partial K^{-1}}{\partial \theta} f \\ \frac{\partial q^t}{\partial \theta} &= \frac{\partial f^t K^{-1}}{\partial \theta} = f^t \frac{\partial K^{-1}}{\partial \theta}. \end{aligned}$$

Knowing that

$$\frac{\partial A^{-1}(t)}{\partial t} = A^{-1}(t) \frac{\partial A(t)}{\partial t} A^{-1}(t)$$

it is possible to write

$$\begin{aligned} \frac{\partial q}{\partial \theta} &= -K^{-1} \frac{\partial K}{\partial \theta} K^{-1} f = -K^{-1} \frac{\partial K}{\partial \theta} q \\ \frac{\partial q^t}{\partial \theta} &= f^t K^{-1} \frac{\partial K}{\partial \theta} K^{-1} = q^t \frac{\partial K}{\partial \theta} K^{-1}. \end{aligned}$$

This leads to rewrite eq. 4.1 as follow

$$\begin{aligned} \frac{\partial J}{\partial \theta} &= -\frac{1}{2} K^{-1} \frac{\partial K}{\partial \theta} q K q + \frac{1}{2} q^t \frac{\partial K}{\partial \theta} q - \frac{1}{2} q K K^{-1} \frac{\partial K}{\partial \theta} q \\ & \frac{\partial J}{\partial \theta} = -\frac{1}{2} q^t \frac{\partial K}{\partial \theta} q \end{aligned} \quad (4.2)$$

Recalling the assembly operator and observing that the only local stiffness matrix depending on  $\theta(e)$  is  $K_e$  belonging to the same element  $e$ , the derivative is limited to differentiating eq. 3.2 with respect to  $e$ .

$$\frac{\partial K}{\partial \theta} = \frac{\partial K_e}{\partial \theta_e}$$

Differentiating eq. 3.2 with respect to  $\theta_e$  gives

$$\frac{\partial K_e}{\partial \theta_e} = \int_{\Omega_e} B^T \frac{\partial \bar{Q}}{\partial \theta_e} B d\Omega_e \quad (4.3)$$

where, recalling eq. 3.1 it is possible to write

$$\frac{\partial \bar{Q}}{\partial \theta_e} = -2 \sin(2\theta_e) T_1 + 2 \cos(2\theta_e) T_2 - 4 \sin(4\theta_e) T_3 + 4 \cos(4\theta_e) T_4. \quad (4.4)$$

The integral can be evaluated using the same method used to calculate  $K_e$  in eq. 3.2.

The gradient is therefore calculated assembling eqs. 4.2, 4.3 and 4.4

$$\frac{\partial J}{\partial \theta} = -\frac{1}{2} q^t \left( \int_{\Omega_e} B^T (-2 \sin(2\theta_e) T_1 + 2 \cos(2\theta_e) T_2 - 4 \sin(4\theta_e) T_3 + 4 \cos(4\theta_e) T_4) B d\Omega_e \right) q. \quad (4.5)$$

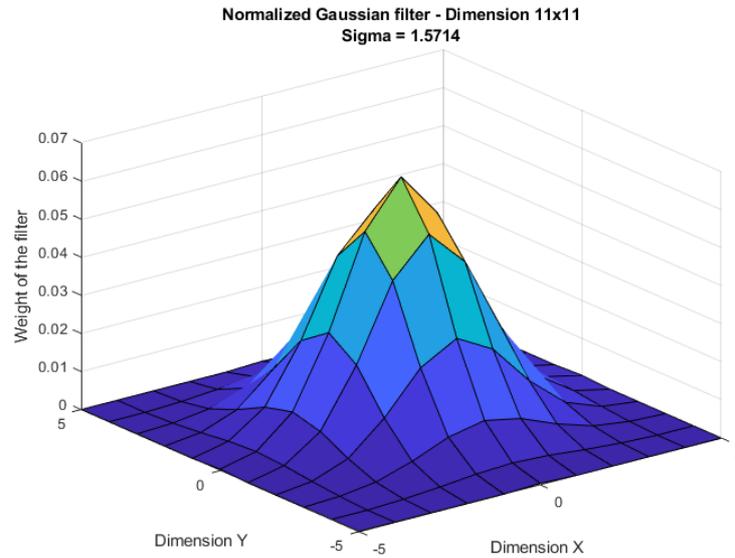
#### 4.2.4 Gaussian smoothing

Usually found in image processing, the Gaussian smoothing is the result of blurring a scalar field using a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce details. The effect of this blurring technique is a smooth blur that helps to achieve fiber continuity at the cost of a slightly increased computational cost and elastic energy. The filter is implemented in the code by building a square matrix of a given odd dimension called  $\chi$ . The Gaussian function for a 2D domain is

$$f(x, y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{(x-\mu)+(y-\mu)}{\sigma}\right)^2}$$

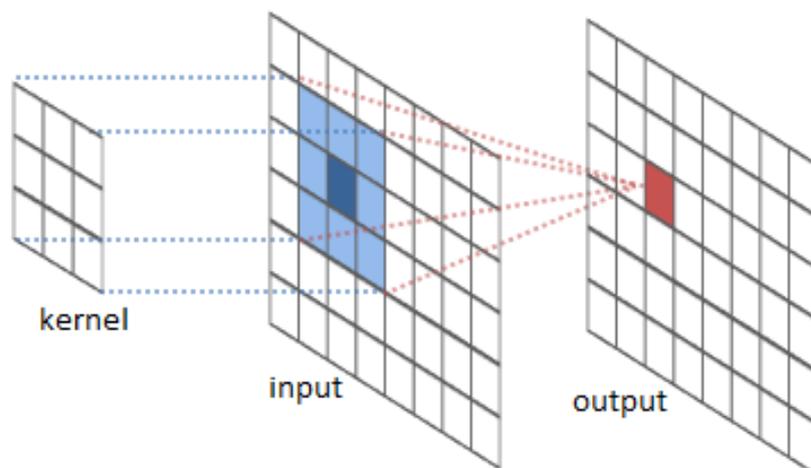
where the parameter  $\mu$  represent the mean or expectation of the distribution (and also its median) and  $\sigma$  is its standard deviation.

The Gaussian function is then normalized to avoid an unnecessary scaling of the design variables. The peak of the function is found in the center of the kernel  $\chi$ . Such a filter is advisable also to avoid stress concentration at discontinuous paths, which makes it difficult to converge in stress minimization problems. Here is one example of a 11x11 kernel with  $\sigma = \text{dim}/7$ .



**Figure 4.1:** Graphical representation of the Gaussian filter used to achieve fiber continuity in CFAO algorithm. In this image, the dimension of the filter are 11x11 elements and  $\sigma = \text{dim}/7$

Besides the design variables  $\theta(x, y)$ , another set of variables called optimization variables ( $\alpha(x, y)$ ) is defined for each element. The value of the orientation variables is then calculated convoluting the newly created scalar field  $\alpha(x, y)$  with the kernel  $\chi$ . A 2D convolution is an operation concerning a kernel, which is simply a small matrix of weights and an input matrix ( $\chi$  and  $\alpha(x, y)$ ). This kernel "slides" over the 2D input data, performing an element-wise multiplication with the part of the input it is currently on, and then summing up the results into a single output element.



**Figure 4.2:** Graphical representation of the Convolution operation over the height and width of a matrix. Image credits: RiverTrail Github tutorial.

The kernel repeats this process for every location it slides over, converting a 2D matrix of features into yet another 2D matrix of features (see img. 4.2). The output

features are essentially the weighted sums (with the weights being the values of the kernel itself) of the input features located roughly in the same location of the output element on the input layer. Whether or not an input feature falls within this "roughly same location", gets determined directly by whether it's in the area of the kernel that produced the output or not. It means that the size of the kernel directly determines how many input features are employed.

The matrix being convoluted is padded symmetrically before the 2D convolution to take into account the shrinking of the resulting matrix (see img. 4.3),

$$\begin{array}{c}
 \left[ \begin{array}{cccc|cccc|cccc}
 a_{11} & a_{12} & a_{12} & a_{11} & a_{10} & a_{10} & a_{11} & a_{12} & a_{12} & a_{11} & a_{10} & a_{10} & a_{11} & a_{12} \\
 a_{01} & a_{02} & a_{02} & a_{01} & a_{00} & a_{00} & a_{01} & a_{02} & a_{02} & a_{01} & a_{00} & a_{00} & a_{01} & a_{02} \\
 a_{01} & a_{02} & a_{02} & a_{01} & a_{00} & a_{00} & a_{01} & a_{02} & a_{02} & a_{01} & a_{00} & a_{00} & a_{01} & a_{02} \\
 a_{11} & a_{12} & a_{12} & a_{11} & a_{01} & a_{10} & a_{11} & a_{12} & a_{12} & a_{11} & a_{10} & a_{10} & a_{11} & a_{12} \\
 a_{21} & a_{22} & a_{22} & a_{21} & a_{20} & a_{20} & a_{21} & a_{22} & a_{22} & a_{21} & a_{20} & a_{20} & a_{21} & a_{22} \\
 a_{21} & a_{22} & a_{22} & a_{21} & a_{20} & a_{20} & a_{21} & a_{22} & a_{22} & a_{21} & a_{20} & a_{20} & a_{21} & a_{22} \\
 a_{11} & a_{12} & a_{12} & a_{11} & a_{01} & a_{01} & a_{11} & a_{12} & a_{12} & a_{11} & a_{10} & a_{10} & a_{11} & a_{12}
 \end{array} \right] \text{Input matrix}
 \end{array}$$

**Figure 4.3:** Graphical representation of the Symmetrical Padding operation. Image credits: Matlab Documentation

The algorithm that is based on CFAO but uses this convolution step is called Filtered Continuous Fibre Angle Optimization (FCFAO).

Since the angle variable has  $2\pi$  period, the main concern for the whole algorithm is that the problem cannot be convex. The non-convex complexity can potentially increase the solving time significantly, and one should be careful to avoid over-constraining the design space.

### 4.2.5 Filtered Gradient Calculation

The filtered gradient is analytically calculated using the convolution operator, noticing that every previous equation that depends on  $\theta(x, y)$ , now depends on  $\alpha(x, y)$ . Using the chain rule and recalling eq. 4.5 it is possible to write

$$\frac{\partial J}{\partial \alpha} = \frac{\partial J}{\partial \theta_e} \frac{\partial \theta_e}{\partial \alpha}$$

and it can be evaluated convoluting with the very same kernel  $\chi$  the matrix  $\partial J / \partial \theta_e$ . Once again the matrices are padded symmetrically to keep equal dimensions.

### 4.2.6 Multistart

The code is provided with the "multi-start" option to help to find global minimum. It starts a local solver (such as *fmincon*) from multiple start points. The algorithm uses multiple start points to sample multiple basins of attraction. This is done to

demonstrate the robustness of the algorithm and to reduce the influence of the starting point.

### 4.3 Level-Set Surface Calculation

The goal of this section is to generate a Level-Set surface for our problem. Its iso-lines are the fiber trajectories that approximate the angles found by FCFAO. The main challenge is to generate trajectories as smooth as possible without moving far away from the optimum found.

#### 4.3.1 Problem description

Let  $\mathbf{G}$  be a rectangular domain  $[0, a] \times [0, b]$  with equally spaced grid made by  $N \times M$  elements ( $(N + 1) \times (M + 1)$  nodes). Looking at the gradient is possible to say that the gradient vector is always orthogonal to the plane tangent to the iso-lines of a function. The equation that links the angle of the fiber  $\theta(x, y)$  in an element with the corresponding Level-set surface  $\phi$ , tho, is:

$$\theta(x, y) = \frac{\pi}{2} + \arctan \left( \frac{\partial\phi(x, y)/\partial y}{\partial\phi(x, y)/\partial x} \right) \quad (4.6)$$

The equation admits infinite solutions since there is just one relation for the two unknowns  $\partial\phi/\partial x$  and  $\partial\phi/\partial y$ . Finally, let us define

$$f = \begin{pmatrix} f_x \\ f_y \end{pmatrix} = \begin{pmatrix} \partial\phi/\partial x \\ \partial\phi/\partial y \end{pmatrix} = \vec{\nabla}\phi$$

#### 4.3.2 Poisson Equation

To find  $\phi$ , one would like to solve the Partial Differential Equation (PDE)

$$\vec{\nabla}\phi = f$$

using MATLAB PDE internal solver *pdepe*. To further simplify the problem, the divergence of

$$\vec{\nabla}\phi = f$$

is taken and the Poisson's equation is obtained, since  $\vec{\nabla} \cdot (\vec{\nabla}\phi) = \vec{\nabla}^2\phi = \Delta\phi$ . Poisson equation is a quite well-covered topic in any numerical methods book and extensively studied.

$$\Delta\phi = \vec{\nabla} \cdot f \text{ in } \mathbf{G}.$$

To make a well-posed problem for Poisson's equation, one needs to impose some boundary conditions on  $\partial \mathbf{G}$ . The natural conditions for that would be Neumann type conditions

$$\frac{\partial \phi}{\partial n} = \vec{n} \vec{f} \text{ on } \partial \mathbf{G}.$$

This approach works well for the problems in which the explicit equation for the gradient  $\vec{f}$  is known. It is challenging to find suitable BCs because eq. 4.6 alone doesn't allow to know the two components of  $\vec{f}$  at the same time.

### 4.3.3 Direct integration approach

Another approach to solve the equation 4.6 would be the approximation of the potential field  $\phi$  by direct gradient approximation.

For every point  $(x_i, y_j)$  let us write

$$\phi_{i,j} = \left( \phi_{i-1,j} + \int_{x_{i-1}}^{x_i} f_x dx \right) |x_i - x_{i-1}|, \quad i = 1, \dots, M, j = 0, \dots, N$$

$$\phi_{i,j} = \left( \phi_{i,j-1} + \int_{y_{j-1}}^{y_j} f_y dy \right) |y_j - y_{j-1}|, \quad i = 0, \dots, M, j = 1, \dots, N$$

where  $f_x$  and  $f_y$  must be at least continuous function.

That is direct gradient approximation applied to

$$\frac{\partial \phi}{\partial x} = f_x \tag{4.7}$$

$$\frac{\partial \phi}{\partial y} = f_y \tag{4.8}$$

on every segment of the mesh. There are too many equations ( $2NM + N + M$ ) for  $(N+1)(M+1)$  unknowns. That is because they are not independent and are consistent only if some condition imposed on  $(f_x, f_y)$  is true. That condition is conservative imposed on  $f$ , that in a simply connected domain like  $\mathbf{G}$  became just  $\vec{\nabla} \times f = 0$ . Numerical integration over any path consisting of grid segments should now have the same value. Extra equations could now be dropped and compute  $\phi_{i,j}$  in following steps:

Since  $\phi$  is defined up to an additive constant, it needs to be fixed at some point, for example

$$\phi_{0,0} = 0.$$

$\phi_{0,j}$  are computed for  $j = 1, \dots, N$  using the first difference equation. For each  $j$ ,  $\phi_{i,j}$  is now computed with  $i = 0, \dots, M, j = 1, \dots, N$  using the second difference equation. All the integrations are performed using trapezoidal rule that it is exact for linear functions.

The only drawback of this method is that different values of  $\phi_{i,j}$  would be obtained if a different order of evaluation is chosen, and  $f$  isn't conservative. But the calculation time is indeed much faster than solving the PDE if the conservative field  $\vec{f}$  is found.

### 4.3.4 Calculate the conservative field of the gradient

A conservative vector field (also called a path-independent vector field) is a vector field  $\vec{F}$  whose line integral  $\int_C \vec{F} d\vec{s}$  over any curve  $C$  depends only on the endpoints of  $C$ . The integral is independent of the path that  $C$  takes going from its starting point to its ending point. Directly checking to see if a line integral doesn't depend on the path is nearly impossible, as an infinite number of paths between any pair of points to check is needed. If  $\vec{F}$  is a two-dimensional vector field,  $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , it is conservative if and only if  $F = \vec{\nabla}f$  for some potential function. The curl of a gradient is zero,  $\vec{\nabla} \times (\vec{\nabla}f) = 0$ , for any twice continuously differentiable  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ . Therefore, if  $\vec{F}$  is conservative, then its curl must be zero, as  $\vec{\nabla} \cdot F = \vec{\nabla} \times (\vec{\nabla}f) = 0$ .

$$\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} = \frac{\partial f^2}{\partial x \partial y} - \frac{\partial f^2}{\partial y \partial x} = 0.$$

It is correct to state that if  $\vec{F}$  is conservative, then its curl must be zero. Without additional conditions on the vector field, the opposite may not be exact, so it's not possible to conclude that  $\vec{F}$  is conservative just from its curl being zero. The inverse is valid only if the domain it's simply connected.

To find the conservative field  $\vec{f}$  this system of PDEs is then written using equations 4.6, 4.7, and 4.8.

$$\begin{cases} \theta(x, y) = \frac{\pi}{2} + \arctan\left(\frac{f_y(x, y)}{f_x(x, y)}\right) \\ \frac{\partial f_y(x, y)}{\partial x} - \frac{\partial f_x(x, y)}{\partial y} = 0 \end{cases}$$

that translates into a first-order PDE. Unfortunately, once again the system is impossible to solve because of BCs. Hence a numerical approach is used.

### 4.3.5 Optimization routine

The conservative field  $\vec{f}$  is found using an optimization routine. Its calculation permits to apply later either the PDE or the direct gradient approach.

The problem is now reformulated. The equation 4.6 is equivalent to

$$\frac{\partial \phi}{\partial x} \cot(\theta(x, y)) = \frac{\partial \phi}{\partial y} \quad (4.9)$$

To take into account the change of the domain of the equation, the scalar field  $\theta(x, y)$  that was bound between  $[-\frac{\pi}{2}; \frac{\pi}{2}]$  it is now converted into  $[0; \pi]$ . This permits to have a simply connected field as it is the case for  $f(\theta(x, y))$ .

Since there is an infinite number of vector fields  $\vec{f}$  that verify equation 4.9, one must somehow constrain the field. Concerning the Level-set surface, the choice made is:

- The surface must be monotonic descending respect to  $X \rightarrow f_x \leq 0$
- The sign of  $f_y$  changes according to  $\begin{cases} f_y(x, y) > 0 & \text{if } \theta(x, y) \in [0, \pi/2] \\ f_y(x, y) \leq 0 & \text{if } \theta(x, y) \in [\pi/2, \pi] \end{cases}$

Once these conditions are fixed, it is possible to calculate the angle of the gradient written in polar coordinates.

$$\begin{cases} f_x(x, y) = \rho(x, y) \cos(\Gamma(x, y)) \\ f_y(x, y) = \rho(x, y) \sin(\Gamma(x, y)) \end{cases}$$

where  $\Gamma(x, y) = \theta(x, y) + \pi/2$ , condition that respects the precedent conditions.

At this stage of the analysis the  $\Gamma$  found is the only important component: since in equation 4.9  $f_x, f_y$  appear only as a fraction, the  $\rho$  simplifies. But changing  $\rho$  influences if the curl of the vector field is or not zero. To summarize, it is possible to fix the  $\Gamma$  found using a constrain 4.9 that doesn't set to zero everywhere the curl of the vector field. Later, a minimization problem to find the  $\rho$  that set the curl to zero everywhere is solved. Doing so, it is finally possible to find a vector field ( $\vec{f}$ ) that is conservative, that verify equation 4.9, and that could be integrated to find its potential ( $\phi$ ).

Here is the mathematical formulation of the minimization problem:

$$\text{minimize : } \sum_{j=0}^N \sum_{i=0}^M (\vec{\nabla} \times f_{i,j}(\rho(i, j)))^2$$

$$\text{by varying : } \rho(i, j) \in [0, +\infty]$$

$$\text{subject to : } \Gamma(x, y) = \text{const.}$$

The conservative vector field is later integrated using the direct gradient approach as it is considerably faster than the PDE approach.

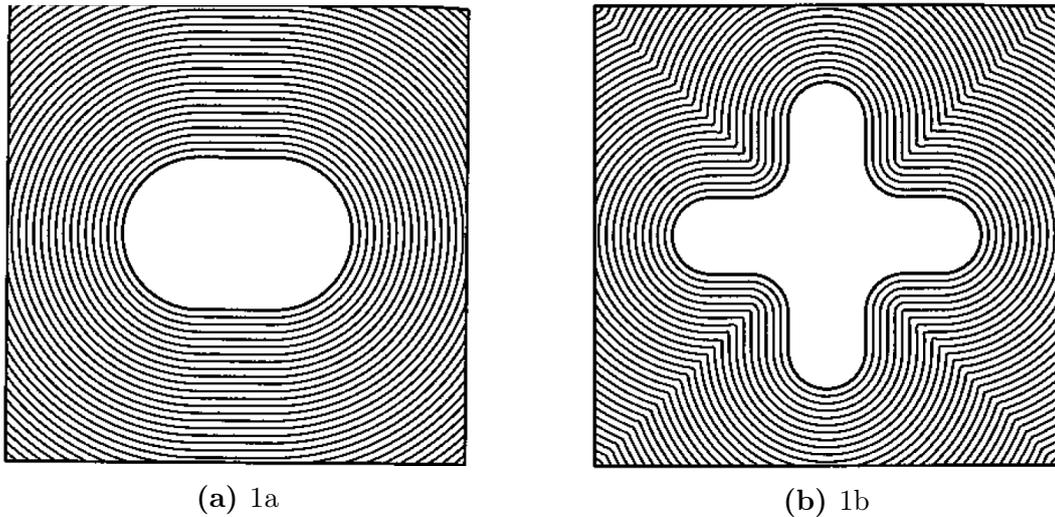
Once the Level-set surface is generated, its contour is plotted and compared with the original trajectory and the gradient  $f$ . The trajectories achieved give an adequate estimation of the directions, but they are not yet ready to be converted into G-Code.

## 4.4 Fiber Propagation

The propagation of the fibers is an essential step in ensuring the manufacturability of the ply. Having equispaced trajectories is needed to create an even distribution of the fiber on the layer. This problem is known in the scientific community as the problem of propagating surfaces and is extensively studied by the American professor James Sethian (1999). In 1995 he created the numerical model called Fast Marching Method (Sethian, 1995), an algorithm capable of dealing with boundary value problems of the Eikonal equation:

$$|\nabla u(x)| = \frac{1}{f(x)}, \quad x \in \Omega \quad (4.10)$$

where  $\Omega$  is an open set in  $\mathbb{R}^n$  with well-behaved boundary and  $f(x)$  is a function with positive values, and it is an indication of the speed of the moving boundary. The right-hand side  $f(x)$  is typically supplied as known input. Physically, the solution  $u(x)$  is the shortest time needed to travel from the boundary  $\partial\Omega$  to  $x$  inside  $\Omega$ , with  $f(x)$  being the speed at  $x$ . In the special case of  $f = 1$  in all  $\Omega$ , the equation tells us that the distance  $u(x)$  must change at one unity per unity of length. In practice, the solution



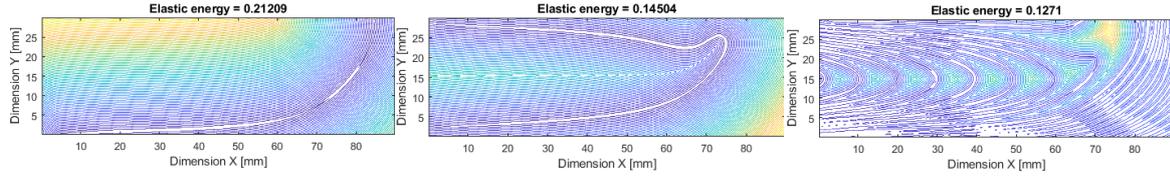
**Figure 4.4:** Exemples of fiber propagation (see Sethian, 1999).

gives the signed distance from the boundary of the set. The signed distance plays a key role in the fiber propagation problem because plotting the isoline at a specific value allows to determine the fibers parallel to a starting one (see img. 4.4). The values chosen range between 0 and the maximum distance with a step equal to the diameter of the fiber  $h$ .

- If a great quantity of starting fibers are chosen and propagated, one obtains a solution that perfectly corresponds to the optimized case. However, it would have a lot of bad-behaved boundaries where the propagated fibers meet.
  
- If a single fiber is chosen among all the others, an optimal mathematical solution is obtained, but it could be different from the optimized case.
  
- If two to three starting fibers are chosen among the bundle of the optimized fibers, then the mathematical solution doesn't misbehave. The optimality is conserved if a criterion to choose among the generating fibers is fixed. The minimum elastic energy is chosen.

The comparison between the different implementations can be observed in fig.4.5.

Here is presented the algorithm chosen to find the optimal solution of the propa-



(a) Fiber propagation output when a single fiber is used. (b) Fiber propagation output when a couple of fiber are used. (c) Fiber propagation output when a bunch of fiber is used.

**Figure 4.5:** Comparison of the different outputs that the Heat Method produces when a different number of master fibers is chosen. It is clear that having too many fibers tends to produce noisy results in the boundary and to have only one could lead to lose the optimality.

gating fiber problem.

---

#### Algorithm 6: Fiber Propagation algorithm

---

**Result:** Find the best  $i$  solution based on FEM analysis

initialization;

set the fiber diameter  $h$ ;

calculation of all the possible combination  $N_{n,k}$  of fibers;

**for**  $n = 1 : N_{n,k}$  **do**

    propagate the fibers in every direction using the step  $h$ ;

    retrieve the scalar field  $\theta(x, y)$  from the new propagated fibers;

    perform FEM analysis;

    compare elastic energy;

**end**

choose best  $i$  cases;

write G-Code for the selected  $i$  cases;

---

Since the order is not important, the calculation of the candidate fibers among the initial trajectories is done using the combinatorial combination without repetition. This can be achieved knowing that

$$N_{n,k} = \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

In the literature there are at least three different algorithm that could calculate the signed distance for our problem:

- **Dijkstra's algorithm:** invented by the Dutch computer scientist Edsger Dijkstra in 1956 and published in 1959, it is a graph-based searching algorithm that solves the single-source shortest path problem. This algorithm is often used in routing. Dijkstra's algorithm is used for finding the shortest path with minimum cost. However, this algorithm only evaluates the graph distance that overestimates the straight-line Euclidean distance, no matter how fine the grid becomes (see Dijkstra, 1959).
- **Exact schemes:** the formulation of "exact" algorithms represents an important development, which leads to the evaluation of the distance on a continuous do-

main, where paths can cut through the faces of a triangulation (see Chen and Han, 1990). However these schemes are often  $\mathcal{O}(n^2)$ .

- **Fast Marching Method:** The algorithm is somehow similar to Dijkstra’s algorithm and uses the fact that the information only flows outward from the seeding area. What makes it more interesting is that the value given to every node is calculated differently, giving it a far better approximation of the true Geodesic distance compared to Dijkstra’s algorithm. It is really fast and well studied (see Sethian, 1995).
- **The Heat Method:** New algorithm for solving the single or multiple-source shortest path problem. The distance computation - a non-linear and hyperbolic problem - is solved by splitting it into two linear stages. The heat method is robust, efficient, and simple to implement since it is based on solving two sparse linear systems. Moreover, these systems can be factored once and then solved for a set of different starting points. The Heat method is inspired by Varadhan’s known result (Varadhan, 1967) in differential geometry, which relates heat diffusion to geodesic distance (see Crane et al., 2017).

#### 4.4.1 The Heat Method

The algorithm is based upon the idea that the geodesic distance can be calculated by analyzing how the heat diffuses in a short period of time over a fixed domain  $\Omega$ . The first to bring forth such a hypothesis was the Indian American mathematician S.R.S Varadhan who wrote the relation in a closed-form (Varadhan, 1967). Over time heat spreads out over the domain and can be described by the function  $k_t(x, y)$  called *heat kernel*, which measures the heat transferred from a source to a destination in the domain after time  $t$ . The distance  $\phi(x, y)$  can be recovered via a simple point-wise transformation of the heat kernel:

$$\phi(x, y) = \lim_{t \rightarrow 0} \sqrt{-4t \log k_t(x, y)} \quad (4.11)$$

This relation, unfortunately, requires a precise numerical calculation of the heat kernel, which is trivial to obtain even for a 2D domain as our plate. The Heat method addresses this issue using the direction of the gradient of the scalar field that is parallel to the exact geodesic distance one.

Relative to existing algorithms, the heat method offers two major advantages:

1. It can be applied to virtually any type of geometric discretization where the laplacian ( $\Delta$ ), the divergence ( $\vec{\nabla} \cdot$ ) and the gradient ( $\vec{\nabla} f$ ) operators are well defined.
2. It involves only the resolution of sparse linear systems that can be prefactorized and solved for different starting points. The standard linear PDEs solved here are widespread in scientific computing and can be easily parallelized to reduce CPU’s time even more.

Since our problem is intrinsically parallel, the heat method is chosen over the others.

### The Heat Equation

The first step to implement the heat method and solve the fiber propagation problem is to calculate the heat kernel  $k_t$  over the domain  $\Omega$  that in the presented research is a 2D plate. The heat equation is

$$\frac{\partial u_t}{\partial t} = \alpha \Delta u_t \quad (4.12)$$

and is discretized using a single backwards Euler step for some fixed time  $t$ . The equation 4.11 can be rewritten as the linear equation

$$(\text{id} - t\Delta)u_t = u_0 \quad (4.13)$$

where  $\text{id}$  is the identity operator,  $t$  is the chosen time step and  $u_t$  is the approximation of the heat kernel. The vector of constant value is defined

$$u_0 = \delta_{ij} \text{ on } \gamma \quad (4.14)$$

where  $\gamma$  represent the starting fibers and  $\delta_{ij}$  is the Kronecker delta.

Equation 4.13 can be solved remembering that  $\Delta$  represent the matrix of second order central differencing. For the 1D case it is possible to write:

$$\Delta_{1D,d} = \begin{pmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & b_2 & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ & & & b_{n-1} & a_n \end{pmatrix} \quad (4.15)$$

with  $a = 2$  and  $b = -1$ . In a general situation the multidimensional discrete Laplacian is the Kronecker sum of 1D discrete Laplacians. The 2D discrete Laplacian on a regular grid with the homogeneous Dirichlet boundary condition is:

$$\Delta_{2D,d} = \Delta_{xx} \oplus \Delta_{yy} = \Delta_{xx}I + I\Delta_{yy}$$

where  $\Delta_{xx}$  and  $\Delta_{yy}$  are the 1D discrete Laplacians in the X and Y directions, correspondingly, and  $I$  is the identities matrix. Both  $\Delta_{xx}$  and  $\Delta_{yy}$  correspond to the case of the homogeneous Dirichlet boundary condition at end points of the x- and y-intervals, in order to generate the 2D discrete Laplacian  $\Delta_{2D}$  corresponding to the homogeneous Dirichlet boundary condition everywhere on the boundary of the rectangular domain.

The choice of the time step is made taking into consideration the sensibility analysis made by Crane et al. (2017) and is fixed at

$$t = \frac{Dx + Dy}{2}.$$

Once the heat kernel is calculated, it is possible to evaluate the geodesic distance approximation  $\phi(x, y)$  made by eq. 4.11.

It's essential to note that solving eq. 4.13 is easy to parallelize when multiple starting points are given: one can prefactor using the Cholesky decomposition the

matrix  $(\text{id} - t\Delta)$  and solve it multiple times using backward substitution. Let us take a linear system

$$Ax = b$$

where  $A$  is symmetric and positive definite. The Cholesky factorization of the matrix  $A$  is:

$$A = LL^t.$$

where  $L$  is a lower triangular matrix with real and positive diagonal entries, and  $L^t$  denotes the conjugate transpose of  $L$ . Calling  $y = L^t x$ , the system  $Ax = b$  can be solved first evaluating  $Ly = b$  for  $y$  by a forward substitution, and finally solving  $L^t x = y$  for  $x$  by a backward substitution.

### Normalizing the Gradient

Once that the scalar field  $\phi(x, y)$  is evaluated, it is possible to calculate the gradient vector field  $\vec{\nabla}\phi$  parallel to the one of the true geodesic distance. Eq. 4.10 tells that the gradient of the geodesic distance function must be equal to one everywhere. The gradient of  $\phi(x, y)$  is then normalized and changed in sign in order to better reflect the true distance behaviour.

$$\vec{f} = -\frac{\vec{\nabla}\phi}{|\vec{\nabla}\phi|}$$

### Poisson Equation

One possible choice to integrate the vector field  $\vec{f}(x, y)$  is to conduce it to the form of Poisson equation and to solve it with Neumann BCs. Applying the divergence vector operator to

$$\vec{\nabla}\phi = \vec{f} \text{ in } \Omega$$

one would obtain

$$\Delta\phi = \vec{\nabla} \cdot \vec{f} \text{ in } \Omega. \quad (4.16)$$

To make a well-posed problem for Poisson's equation, one needs to impose some boundary conditions on  $\partial\Omega$ . The natural conditions for that would be Neumann type BCs

$$\frac{\partial\phi}{\partial n} = \vec{n} \cdot \vec{f} \text{ on } \partial\Omega.$$

Once again, the matrix used to solve the Poisson equation is calculated as the Kronecker sum of the 1D Laplacian with Neumann BCs.

$$\Delta_{1D,n} = \begin{pmatrix} 2 & -2 & & & \\ -1 & 2 & -1 & & \\ & -1 & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -2 & 2 \end{pmatrix} \quad (4.17)$$

A PDE with only Neumann BCs will always depend on an additive constant and

so the method of Lagrange multipliers is used, enforcing that eq. 4.16 respects

$$\sum \phi(x, y) = 0 \text{ in } \Omega.$$

The trajectories of the fibers are finally found tracing the iso-lines of the scalar field  $\phi(x, y)$  with a step  $h$  given by the diameter of the selected fiber.

#### 4.4.2 FEM analysis

A standard linear FEM analysis is carried out for every combination of fibers. The mesh used to perform the analysis is the very same used in the first optimization (CFAO), and the angle  $\theta(x, y)$  for every element of the mesh is found using the eq. 4.6. This equation links the gradient of a surface with its iso-lines. The outlines of the FEM algorithm are presented in Chapter 3.

#### 4.4.3 G-code

G-code is the name for the most widely used Numerical Controlling (NC) programming language. It is used mainly in Computer-Aided Manufacturing (CAM) environments to control cutting and non-cutting CNC. G-code is a language used to command a machine sequentially, telling it how and when making something. Code instructions are provided to a machine controller that instructs motors where and how fast to move, and what path to follow. For conventional machines, a cutting tool is controlled according to the received instructions through a path, cutting away material from the workpiece. Alternatively, the workpiece can be precisely positioned in any of the axes that the machine has and, either or both can move relative to each other, creating more difficult paths. The very same concept also extends to non-cutting tools such as forming or burnishing tools, additive manufacturing methods such as FDM, and measuring devices. While the G-code is standard in theory, many different manufacturers and different milling machines adopt different nuances of the language to serve their desired functionality.

The G-code used to print the plastic specimens was generated by a MATLAB code developed by a team of students of ISAE-SUPAERO. The work of my algorithm was to give them the coordinates to follow to print every optimal trajectory. Here is an example of the output generated after the propagation of the fibers:

```
0.975, 1.4088, 1
0.825, 1.4124, 1
0.675, 1.4162, 1
0.525, 1.4183, 1
0.375, 1.4106, 1
0.225, 1.3747, 1
0.075, 1.3258, 0
0.075, 28.444, 1
0.225, 28.436, 1
0.375, 28.432, 1
```

The first column represents the X-axis, the second column the Y-axis, and the third column store a parameter used to tell at the machine if the next movement is an extrusion of the fiber (Value 1) or just a movement of the print-head (Value 0).



# Chapter 5

## Test Case and Results

In this Chapter, the optimization framework presented in Chapter 4 is applied to a 2D beam having various loading cases and different dimensions. In Section 5.1, the different load cases and the materials used for the optimization are presented. In the second section (5.2), the results of the FEM comparison between MATLAB and ABAQUS are presented to verify the accordance of the algorithm to commercial software. In Section 5.4, the first results from the unfiltered algorithm are analyzed and discussed. The unsatisfying high percentage of noncontinuous fibers found during the optimization leads to Section 5.3, where the very same beams are optimized using a spatial filter to promote the continuity of the trajectories.

### 5.1 Load Cases and Materials

Here all the load cases used to validate the algorithm are presented. Where not specified, concentrated forces are equal to 10 N, and Distributed forces are chosen to verify that  $\sum_{f \in \Omega} f_i = 10 \text{ N}$ . The choice was made to find a compromise between the speed of convergence of the optimization and the avoidance of non-linear behavior.

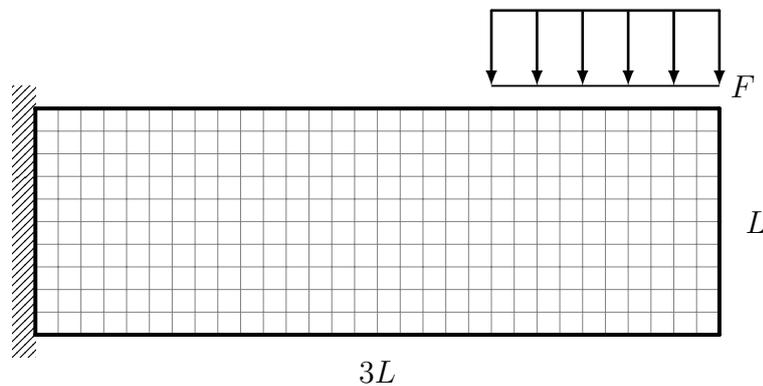
#### 5.1.1 Load Cases

The load cases used to verify the optimization algorithm are presented here. Since the amount of time available to develop the whole algorithm was limited and since the difficulty to extend the capability of the FEM routine, the cases implemented are all 2D rectangular beam. Nevertheless, it is important to note that 3D printing usually works slicing the component on many horizontal planes and that, thus, makes the simplification acceptable. The choice of the load cases was made to simulate as different conditions as possible: concentrated and distributed loads and BCs, horizontal and vertical symmetry, and cases studied in the literature are all implemented to verify the algorithm.

#### **Cantilever beam - Distributed Load - Top**

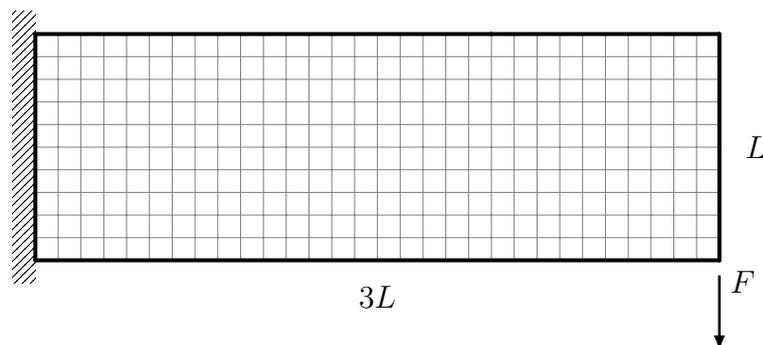
The cantilever beam with the distributed load on the upper edge represents one of the most studied cases in the state of the art of fiber angle optimization. In section 5.3.4,

the results of the optimization algorithm are compared with the results found by the literature, and its similarities are discussed. The beam is anchored at one end to a support from which it protrudes. The concentrated load is smeared on the upper edge, starting from the end for a length equal to  $L/2$ . The size of the beam is 90 mm x 30 mm, and the mesh is always made by square elements.



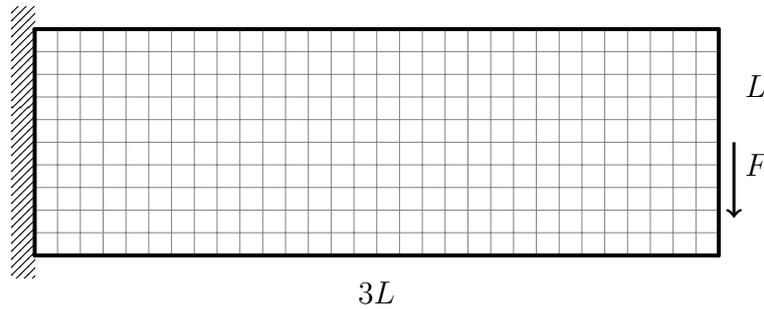
### Cantilever beam - Concentrated Load - Bottom

The cantilever beam with the concentrated load on the lower corner is a case used to check the behavior of the optimizing algorithm with loads that are concentrated on a single integration point of the mesh. The beam is once again anchored at one end to a support from which it protrudes. The concentrated load is applied on the very last point of the corner. The dimensions of the beam are 90 mm x 30 mm, and square elements always make the mesh.



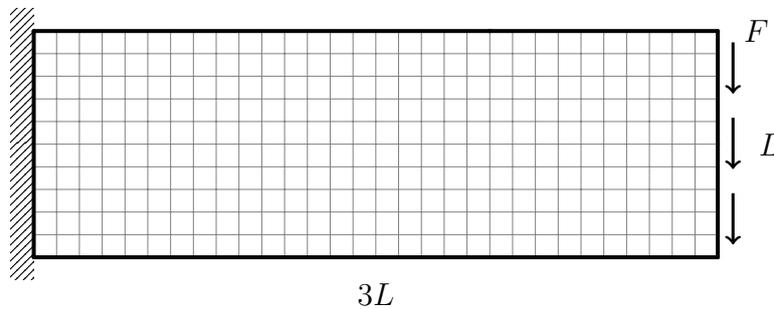
### Cantilever beam - Concentrated Load - Half

The cantilever beam with the concentrated load on the lower corner is once again one of the most studied cases in the state of the art of fiber angle optimization. It is used to check the behavior of the optimizing algorithm with concentrated loads and is used as well to verify the symmetries of the result. The beam is anchored at one end to a support from which it protrudes. The concentrated load is applied to the half-length of the right edge. The size of the beam is once again 90 mm x 30 mm, and the mesh is always made by square elements.



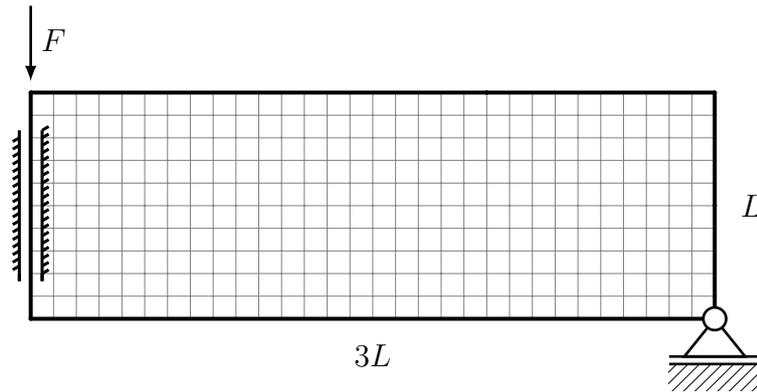
### Cantilever beam - Distributed Load - Right

The cantilever beam with the distributed load on the right edge is used to test the differences with the precedent case. In general, it has permitted to verify the behavior of the symmetries of the result. The beam is anchored at one end to a support from which it protrudes, and the distributed load is applied on the right edge. The size of the beam is 90 mm x 30 mm, and the mesh is always made by square elements.



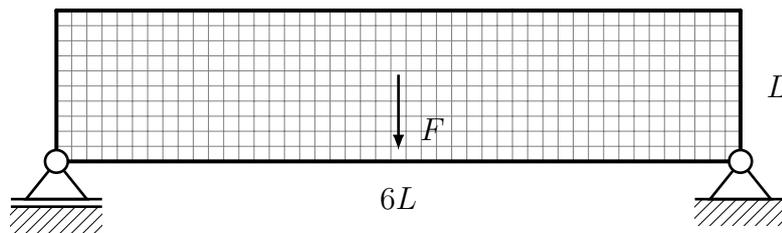
### Messerschmitt-Bölkow-Blohm Beam - Symmetric

The Messerschmitt-Bölkow-Blohm (MBB) beam case is implemented because it is one of the most popular load cases used to validate topology optimization algorithms. Since one of the objectives of the research was also the comparison between the two approaches and since it is straightforward to find reliable results of the topology optimization in the literature, the MBB beam is one of the most studied cases in the research (see Section 5.3). The beam is constrained on the right corner and on the left edge to move only in the  $y$ -axis to mimic the vertical symmetry. The rotation as well is locked down. The size of the beam is 90 mm x 30 mm, and the mesh is always made by square elements.



### Messerschmitt-Bölkow-Blohm Beam - Full

The beam is constrained on the left and the right corner. It mimics the 3-points bending test case. The size of the beam is 120 mm x 20 mm, and the mesh is always made by square elements.



### 5.1.2 Materials

The materials modeled for the optimization process are highly anisotropic composite materials, like the carbon-fiber-reinforced Onyx of Markforged (Markforged, 2019) or ABS reinforced with carbon chopped fiber (Cantrell et al., 2016). The framework doesn't limit only to these materials and works for every anisotropic composite, regardless of the fiber, the matrix, or the deposition method used.

The characterization of the continuous fiber material was already carried out when I arrived at the research laboratory of ISAE-SUPAERO. The continuous carbon fiber composite was tested on specimens printed using the semi-professional 3D printer Mark-Forged Mark II. The mechanical properties were determined through micrographies and traction tests. The printer is capable of depositing a variety of materials proposed by the company itself. A combination of Nylon with shredded carbon fiber (Onyx) and continuous carbon fiber pre-impregnated with thermoplastic was tested for the research purpose. This revealed the fiber fraction and porosity of the composite, as well as the diameter and stiffness of the fibers. All specimens were printed using the proprietary *eiger.io* Markforged software.

The tests concluded that the fiber volume fraction is slightly above 50% in regions filled with carbon fiber reinforced composite filament. There is a small increase in porosity, depending on the deposition direction. A 50% fiber fraction is quite acceptable, given that the process does not undergo high-pressure processing and does not require additional absorbing material to manufacture. On the other hand, the porosity

is high, which reduces the mechanical properties and creates additional locations prone to crack initiation. In addition, the material quality might depend on the deposition angle, but more data is required to validate this statement. The longitudinal tensile modulus found through traction testing is lower than announced by Markforged in their material data-sheet (Markforged, 2019). On the other hand, other papers have also indicated lower results. Dickson et al. (2017) found a tensile modulus of 30 GPa.

Here are the values used to create all the results presented in this chapter:

### Traction Test Results

| Property                              | Measured | MarkForged |
|---------------------------------------|----------|------------|
| Longitudinal Modulus [GPa]            | 44.8     | 60         |
| Transverse Modulus [GPa]              | 4.2      | -          |
| Shear Modulus [GPa]                   | 1.9      | -          |
| Longitudinal-Transverse Poisson Ratio | 0.49     | -          |
| Transverse-Longitudinal Poisson Ratio | 0.09     | -          |

**Table 5.1:** Mechanical properties evaluated in the ISAE-SUPAERO testing facilities and given by Markforged (2019).

## 5.2 FEM comparison with ABAQUS

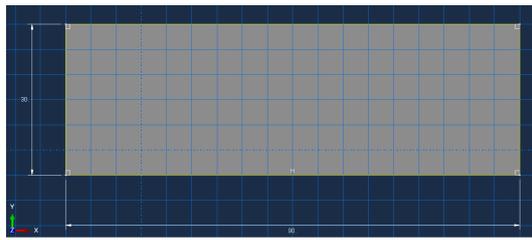
In this Section the steps followed to compare the MATLAB and the ABAQUS FEM analysis are presented.

### 5.2.1 Creation of the model

The FEM model is created and analyzed using the ABAQUS/CAE software. Following the procedure described in Section 3.2, the component is first drawn in the *Part* module of the software. Its dimension varies according to the load case, and it is fixed to 120x20 mm for the full MBB beam and 90x30 mm for all the other cases (see img. (a) of fig. 5.1). The mechanical proprieties used are those of the carbon fiber described in Section 5.1.2. They are modeled in the FEM program defining assigning to the part an orthotropic material and by giving it an orientation angle (see img. (b) of fig. 5.1). This is done to simulate all the different orientations used by the *multistart* approach of the FCFAO optimization.

The next step is the set-up of the mesh: the element chosen is the S4, a bi-linear first order quadrilateral element with 4 points of integration. The choice is discussed in section 3.2.2. The number of elements is fixed to 120x20 for the MBB beam and 60x20 for all the other (see img. (a) of fig. 5.2). The last module of the program to use is *Load* and is used to add the BCs that are implemented according to the load cases mentioned above.

The test is conducted over all the load cases presented in Section 5.1.

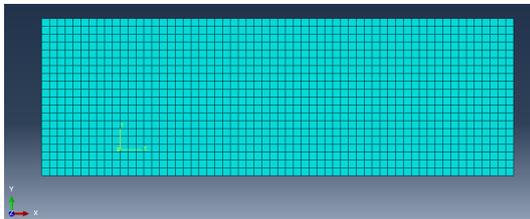


(a) Geometry modelling.



(b) Rotation of the reference system of the material to simulate the different rotation angles of the fiber.

**Figure 5.1:** In this image is presented the first steps of the ABAQUS analysis: the geometry and the material set-up.



(a) Mesh of the part using S4 elements.



(b) BCs and loads implementation.

**Figure 5.2:** In this image is presented the last steps of the ABAQUS analysis: the definition of the mesh and the set-up of the load case.

## 5.2.2 Results

The ABAQUS analysis is carried out for all the load cases and 5 different orientations of the fiber relative to the horizontal. The angle spans between  $+\pi/2$  and  $-\pi/2$ .

Here the results are presented. The numerical values are all the evaluation of the total elastic energy stored in the structure.

FCFAO- Elastic Energy evaluation

| Angle [Deg] | DistrTOP | ConcHALF | ConcLOW. | DistrRIGHT | SymmMBB | ComplMBB |
|-------------|----------|----------|----------|------------|---------|----------|
| -45         | 0,775    | 1,005    | 1,026    | 0,995      | 0,409   | 0,268    |
| -22,5       | 0,352    | 0,457    | 0,487    | 0,442      | 0,152   | 0,154    |
| 0           | 0,182    | 0,229    | 0,273    | 0,214      | 0,240   | 0,103    |
| 22,5        | 0,348    | 0,457    | 0,510    | 0,442      | 0,461   | 0,153    |
| 45          | 0,769    | 1,005    | 1,056    | 0,995      | 0,729   | 0,268    |

**Table 5.2:** The results of the FEM analysis performed using the proprietary MATLAB code. The values in the tables refer to the sum of the elastic energy stored in all the element of the beam.

## ABAQUS - Elastic Energy evaluation

| Angle [Deg] | DistrTOP | ConcHALF | ConcLOW. | DistrRIGHT | SymmMBB | ComplMBB |
|-------------|----------|----------|----------|------------|---------|----------|
| -45         | 0,780    | 1,011    | 1,032    | 1,001      | 0,418   | 0,280    |
| -22,5       | 0,351    | 0,456    | 0,486    | 0,441      | 0,159   | 0,160    |
| 0           | 0,180    | 0,225    | 0,268    | 0,211      | 0,237   | 0,110    |
| 22,5        | 0,347    | 0,456    | 0,515    | 0,441      | 0,453   | 0,160    |
| 45          | 0,774    | 1,011    | 1,072    | 1,001      | 0,720   | 0,280    |

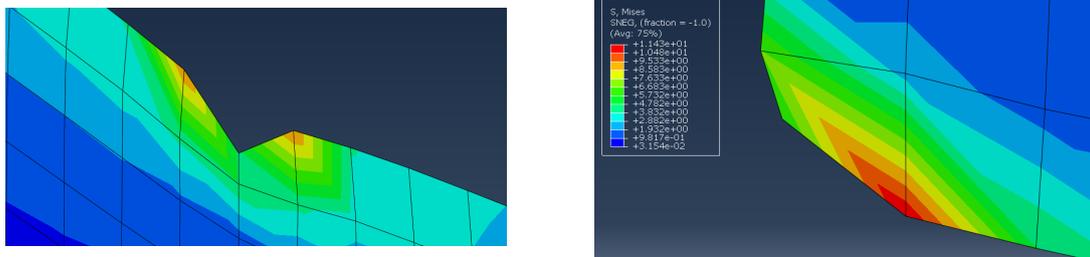
**Table 5.3:** The results of the FEM analysis performed with the commercial code ABAQUS. The values in the tables refer to the sum of the elastic energy stored in all the element of the beam.

## Percentage Differences Elastic Energy evaluation

| Angle [Deg] | DistrTOP | ConcHALF | ConcLOW. | DistrRIGHT | SymmMBB | ComplMBB |
|-------------|----------|----------|----------|------------|---------|----------|
| -45         | 0,61%    | 0,57%    | 0,53%    | 0,57%      | 2,17%   | 4,21%    |
| -22,5       | 0,27%    | 0,15%    | 0,20%    | 0,29%      | 4,65%   | 4,41%    |
| 0           | 1,15%    | 1,45%    | 1,66%    | 1,40%      | 1,51%   | 6,59%    |
| 22,5        | 0,18%    | 0,15%    | 1,08%    | 0,29%      | 1,73%   | 4,52%    |
| 45          | 0,64%    | 0,57%    | 1,54%    | 0,57%      | 1,18%   | 4,47%    |

**Table 5.4:** Percentage differences of the absolute value of the elastic energy between the proprietary code and the commercial code ABAQUS.

As is possible to notice from the Table 5.4, the correlation between the MATLAB model and the commercial software is good and pretty consistent, regardless of the fiber angle, topology, and load case. The percentage difference is never higher than 10% and hardly over 2%. The only outliers between the tested cases are the symmetric and the complete MBB. Here the Matlab model consistently underestimates the energy, and this is probably due to the excessive distortion of the mesh near the BCs (see img. 5.3). It is important to notice that the two are the only ones that have concentrated BCs.



(a) Detail of the concentrated load application point. (b) Detail of the point where the structure is constrained.

**Figure 5.3:** The percentage differences is higher in models presenting high distortion of the mesh due to concentrated BCs or loads.

## 5.3 Filtered CFAO

In this section, the results of the whole algorithm described in Chapter 4 are presented. The full MBB beam test case is used to go into all the details and steps necessary to achieve the optimized state. For all the other load cases, only the starting point and the final optimized design are reported, together with a resuming table of numerical results.

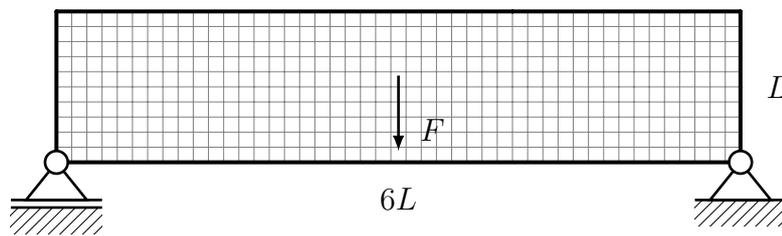
### 5.3.1 Overview of the algorithm

#### Messerschmitt-Bölkow-Blohm Beam - Full

As presented in Section 4.1, the optimizing algorithm is split onto 3 separate sub-problems:

- **Gradient descent optimization:** Optimization routine that uses a gradient descent method to minimize the total elastic energy of a 2D plate.
- **Creation of a level-set surface:** Taking the discrete optimized angles as the input, it generates a 3D surface that, once sliced with horizontal planes, gives an approximation of fibers trajectories.
- **Fiber propagation:** Algorithm that takes the continuous trajectories as an input, it propagates them and creates the points that the machine must follow to print the layer.

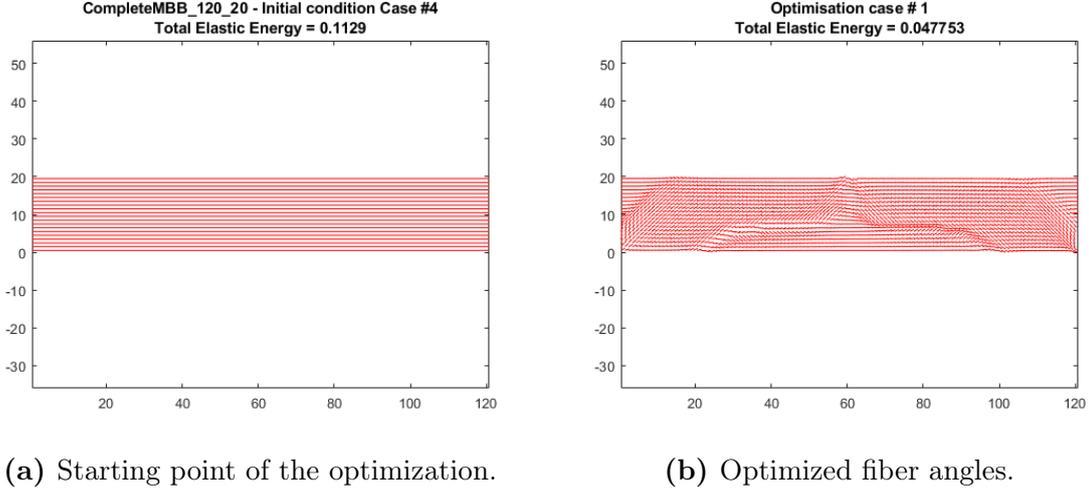
The load case used to present the algorithm is the complete MBB beam to simulate a generic 3-points bending test. The beam is simply supported on the two lower corners, and a concentrated force is applied on the half point of their lower edge (see img. 5.4). The mesh is 120x20 elements, and the dimensions of the beam are 120x20 mm. The thickness is set to 1 mm.



**Figure 5.4:** Graphic representation of the MBB load case.

Once that the load case is defined, it is possible to set up the FCFAO algorithm. The numerical parameters used for the *fmincon* optimizer and for the spatial filter are discussed in Section 5.3.2. On the left side of fig. 5.5, one can observe one of the starting points of the optimization with the evaluation of the objective function. The *multistart* approach simulates multiple starting points, but here, for the sake of brevity, only the one with the minimum compliance is shown. The fibers are all horizontally orientated. On the right end side of the image, the results of the optimizing algorithm

are presented. It is drawn a small trait that indicates the optimal direction for every element or the analyzed mesh. The solution is entirely symmetric, even if no additional constraints are imposed to force that.

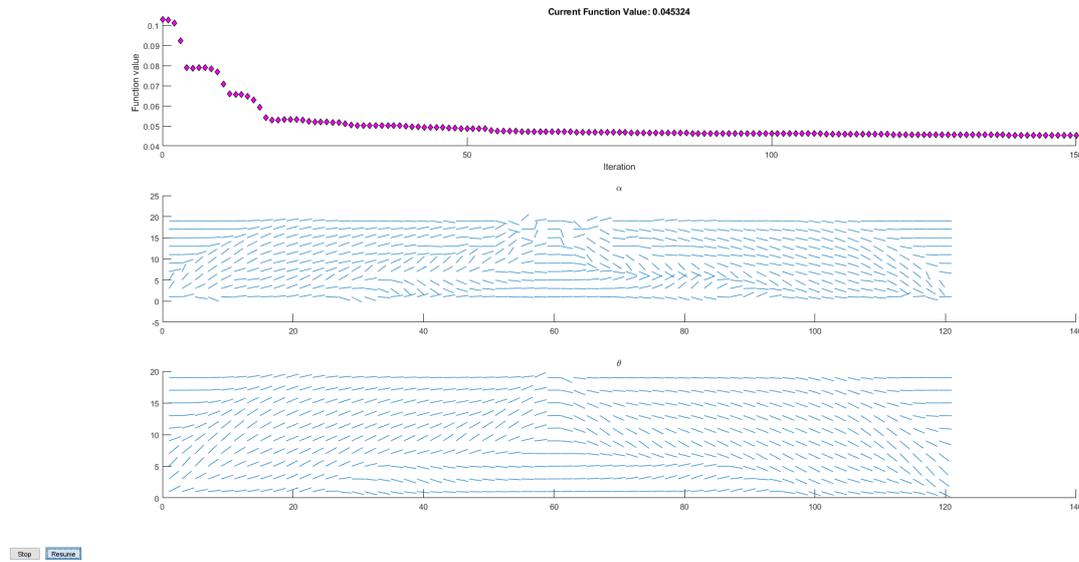


**Figure 5.5:** Starting point and optimized fiber angle directions calculated by the FCFAO algorithm.

The optimization took 924 iterations and 4725 seconds (CPU time) using a Workstation equipped with an Intel® Core™ i7-8700K and 32 Gb of RAM and Matlab distribution r2018a. Intermediate results can be observed in img. 5.6 where three sub-figures are presented:

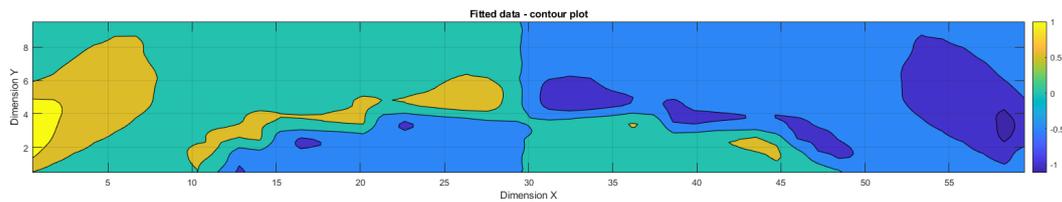
- The convergence history of the objective function of the gradient-based optimizer. It is a graphical representation of the evolution of the optimization as a function of the number of individuals evaluated. The convergence history helps the designer to know how the optimization problem is converging to the optimal solution. It can also be used if the problem has converged quickly, and it is not optimizing anymore.
- The optimization variables  $\alpha(x, y)$  actually being processed by *fmincon*. These are the variables directly modified by the optimizing algorithm, and they can be checked to understand better how the optimization is going or why it is not converging. Severe discontinuities are allowed at this stage. They are allowed to sweep between an upper and a lower bound defined by the designer (usually called  $u_b$  and  $l_b$ ). It is possible to obtain faster and more reliable solutions by explicitly including these bounds in the problem formulation.
- The design variables  $\theta(x, y)$  obtained convoluting the optimization variables with the Gaussian filter and used to evaluate the objective function. They are the actual output of the optimizing algorithm. Due to the spatial Gaussian filter, severe discontinuities are not accepted anymore.

The whole convergence history is presented in Appendix A for iteration number 1, 150, 550, and 924.



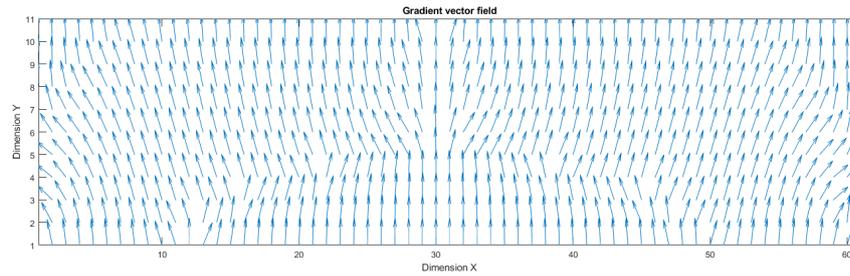
**Figure 5.6:** The picture give intermediate information about the optimization using 3 different graphs: the convergence history, the optimization variables and the design variables. The results are at the 150th iteration.

Once that the CFAO algorithm finished, it is possible to start post-processing the results to make them as easy as possible to manufacture. The two following algorithms all work with nodes and not with the center of the elements. The optimized discrete field  $\theta(x, y)$  is, therefore, linearly interpolated to calculate the value in the nodes of the elements. The new continuous field is plotted in img. 5.7.



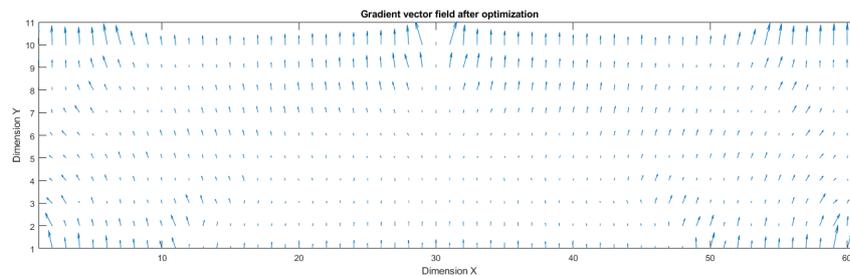
**Figure 5.7:** The discrete field  $\theta(x, y)$  is linearly interpolated and then the contour plot is obtained. The angles are all in radians.

The next step is now the evaluation of the gradient of the level-set surface. It is possible to say that the gradient vector is always orthogonal to the plane tangent to the iso-lines of a function. The equation that links the angle of the fiber  $\theta(x, y)$  to the corresponding level-set surface  $\phi$  is eq. 4.6. The gradient  $\vec{\nabla}\phi(x, y)$  vector field is obtained using the  $\theta(x, y)$  scalar field and eq. 4.6.



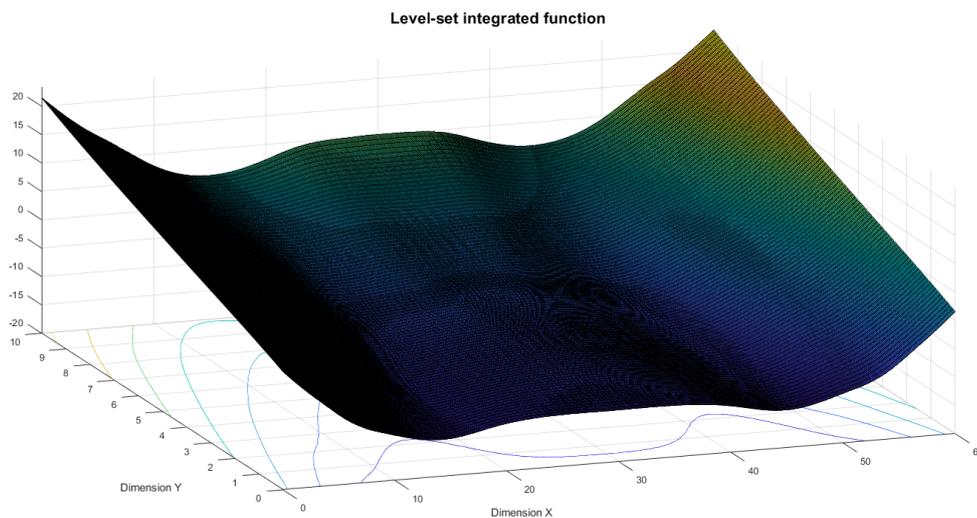
**Figure 5.8:** The gradient  $\vec{\nabla}\phi(x, y)$  vector field is obtained and plotted.

Unfortunately, as extensively explained in Section 4.3.4, the vector field obtained is not yet integrable (it is non-conservative) thanks to its curl not being zero everywhere. But, since only the direction of the gradient is used by the eq. 4.6 to link to the angles  $\theta(x, y)$ , one can freely modify the value of the moduli of the gradient to minimize the curl. A minimization procedure is then set up, and the results are plotted in fig. 5.9.

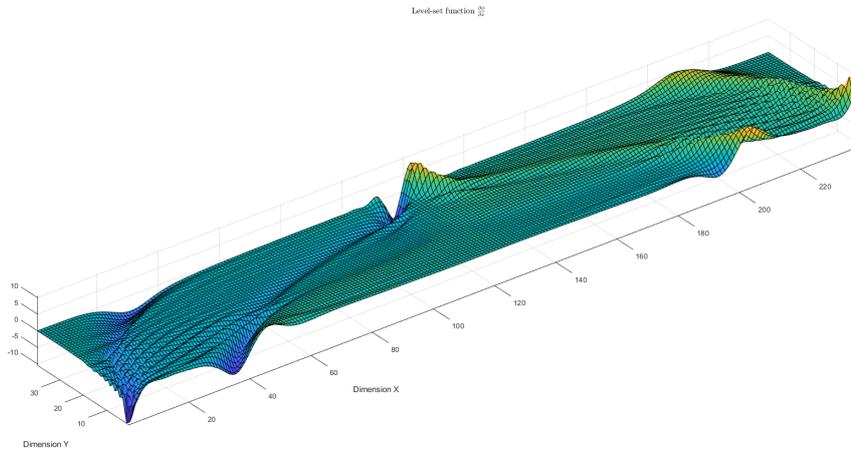


**Figure 5.9:** The new gradient  $\vec{\nabla}\phi(x, y)$  vector field. Thanks to its curl being almost zero everywhere, the field is from now on considered conservative.

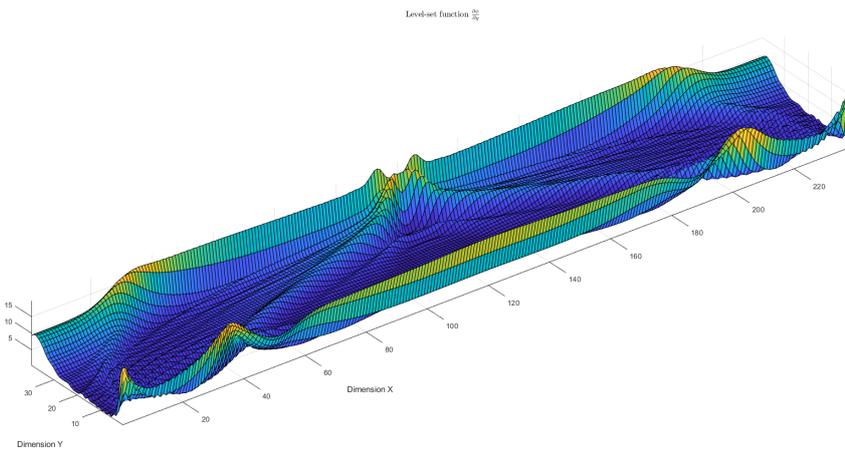
Thanks to its curl being almost zero everywhere, the field is from now on considered conservative, and it can be integrated to finally obtain the surface  $\phi(x, y)$  (fig. 5.10).



**Figure 5.10:** The level set surface  $\phi(x, y)$  with its iso-lines.

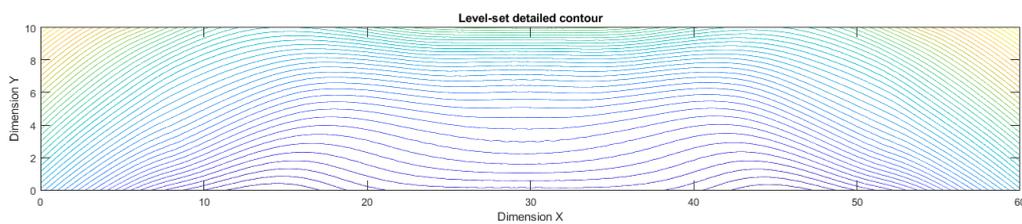


**Figure 5.11:** The X component of the gradient of  $\phi(x, y)$ .



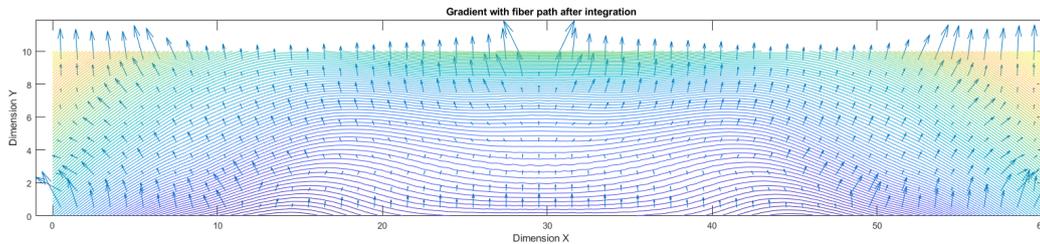
**Figure 5.12:** The Y component of the gradient of  $\phi(x, y)$ .

Once the surface  $\phi(x, y)$  is generated, the approximation of the trajectories are calculated cutting with many equispaced horizontal planes the surface itself (see img. 5.13). The result is perfectly symmetric.



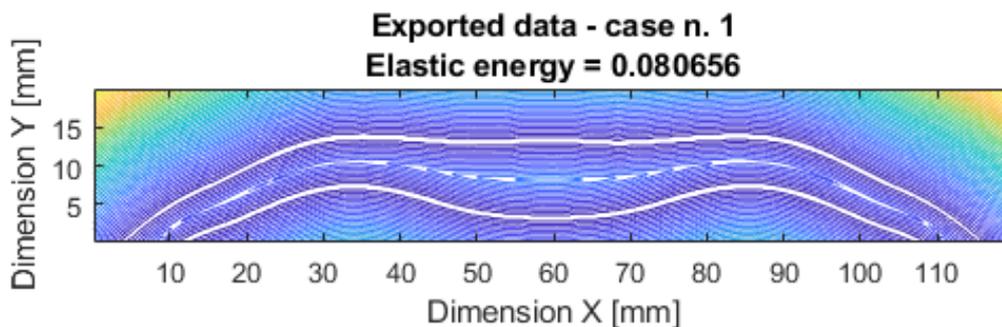
**Figure 5.13:** The iso-level lines of the surface  $\phi(x, y)$ .

To further check the results obtained by the level-set integration, the trajectories are plotted against a vector plot of the gradient obtained from the optimization. The vector field must be perpendicular to the iso-lines in every integration point.



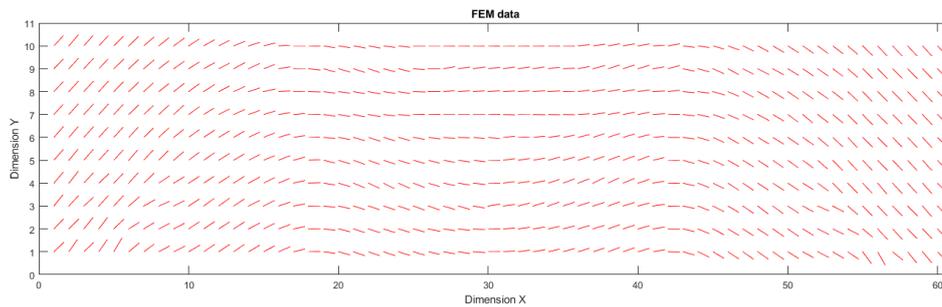
**Figure 5.14:** The same trajectories found in img. 5.13 are plotted against the plot of the gradient. The iso-lines of the surface  $\phi(x, y)$  are always perpendicular to the gradient.

Once that the approximation of the trajectories are obtained, it is now possible to calculate the paths that the printer needs to follow to produce the part. The main parameters to give to the algorithm are the diameter of the fiber  $h$ , the number of total fibers to choose from  $k$  and the number of fibers to choose  $n$  (see Section 4.4). The algorithm then tests all the possible combinations for the parameters given, and the best candidate is chosen according to the minimum value of elastic energy. It is now possible to plot the manufacturable trajectories, obtained with a fiber step  $h$  (fig. 5.15).



**Figure 5.15:** The optimized paths ready to be exported in G-code and printed.

In fig. 5.16 the mesh of the optimum case with the direction of the fiber for all the elements is plotted, ready to perform the FEM analysis. The mesh is obtained using once again eq. 4.6.



**Figure 5.16:** The direction of the fibers of the optimum manufacturable case. The elastic energy is calculated performing a FEM analysis over this mesh. Dimension and number of elements are the very same used for the FCFAO optimization.

Once that the optimum case is chosen, it is possible to proceed to the exportation of the coordinates as described in Section 4.4.3. The piece is then printed on a desktop 3D printer. The voids present represent a defect that can be easily solved, filling them with the thermoplastic matrix used by the continuous fiber printer.



**Figure 5.17:** The trajectories found are validated through the manufacture of the beam using a desktop 3D printer. Here the print has been interrupted to reveal the inner optimized layers.

The complete results of the MBB case are presented in table 5.5. Three different meshes are used to be sure that consistent data is found. In the first column, the elastic energy is evaluated for the starting points where all the fibers are horizontally placed. In the second column, the written value is the objective function of the FCFAO optimized component. For the MBB case, there is nearly a 60% improvement that effectively confirms that the fiber angle optimization is a perfect tool to use when the topology of the component is fixed. However, one must not forget that this is only a theoretical value that can't be reached using the actual manufacturing technique. Once the manufacturing constraints are added, the elastic energy stored jumps to around 30% less than the non-optimized case, asserting the usefulness of the algorithm anyways.

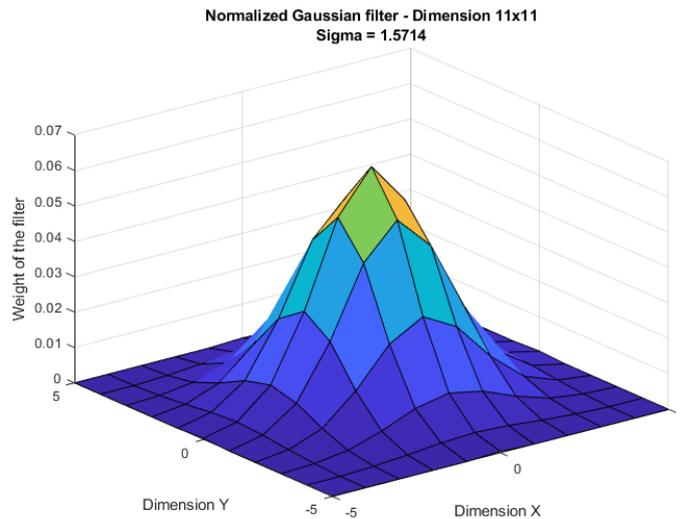
## Messerschmitt-Bölkow-Blohm Beam - Full

| Mesh size | X0    | Optimized | Optimized with constr. |
|-----------|-------|-----------|------------------------|
| 36x6      | 0,095 | 0,041     | 0,071                  |
| 60x10     | 0,103 | 0,044     | 0,073                  |
| 120x20    | 0,110 | 0,045     | 0,080                  |

**Table 5.5:** Results of FCFAO algorithm for the MBB load case.**5.3.2 Setting of the optimizer and of the filter**

The settings used to obtain the results here discussed are:

- The mesh is always made by square elements of different sizes. The loads are chose to verify  $\sum_{f \in \Omega} f_i = 10 \text{ N}$ .
- The algorithm used for the *fmincon* algorithm is *interior-point*. It is an algorithm used for solving large nonlinear programming problems. It follows a barrier approach that employs Sequential Quadratic Programming (SQP) and trust regions to solve the sub-problems occurring in the iteration.
- **StepTolerance** is set to 1e-8. **StepTolerance** is a lower bound on the size of a step, meaning the norm of  $(x_i - x_{i+1})$ . If the solver attempts to take a step that is smaller than **StepTolerance**, the iteration ends. **StepTolerance** is used as a relative bound, meaning iterations end when  $|(x_i - x_{i+1})| < \text{StepTolerance} \times (1 + |x_i|)$ .
- **MaxFunEvals** and **MaxIter** are set to **inf**. They represent the maximum number of iterations (the big loop) of the optimizer and the maximum number of the objective function evaluations, respectively.
- The chosen upper and lower bound for the design variables are  $[-\pi/2; \pi/2]$ .
- There is not the need to use non-linear constraints that tend to slow down the convergence rate of the optimizer.
- The filter dimension is always set to half of the shorter side of the component. The size is measured as the number of elements of the square that perfectly enclose the filter circle. The  $\sigma$  used is set to the number of elements of the filter divided by a factor 7.  $\sigma = \text{dim}/7$ .



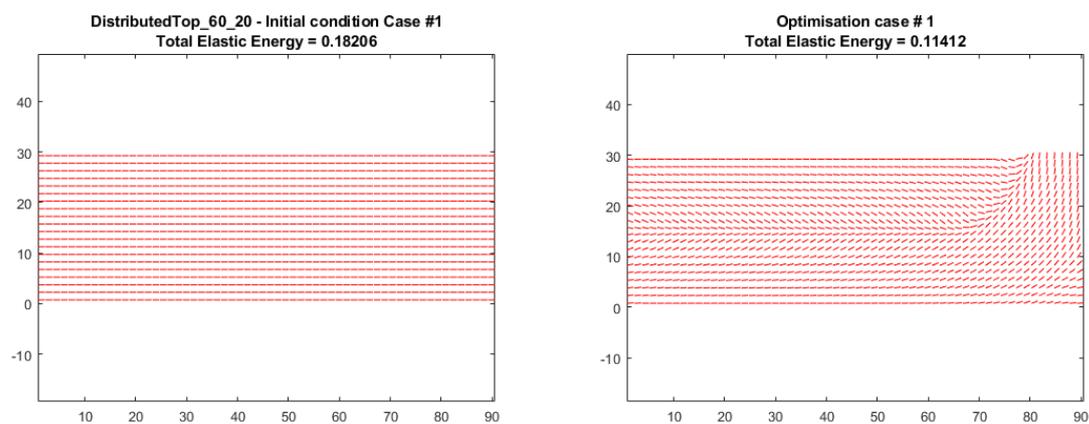
**Figure 5.18:** An example of the circular Gaussian filter used to achieve fiber continuity in the FCFAO analysis.

### 5.3.3 Results

This Section is entirely dedicated to the presentation of the results obtained by the optimizing algorithm against the load cases presented in Section 5.1.1.

A set of images representing the starting point, the discrete field  $\theta(x, y)$  obtained after the FCFAO optimization and the printing paths are presented for every load case. To complete them, a table resuming the numerical results for different mesh dimensions is added. The numerical values are compared with the unfiltered algorithm to complete the results.

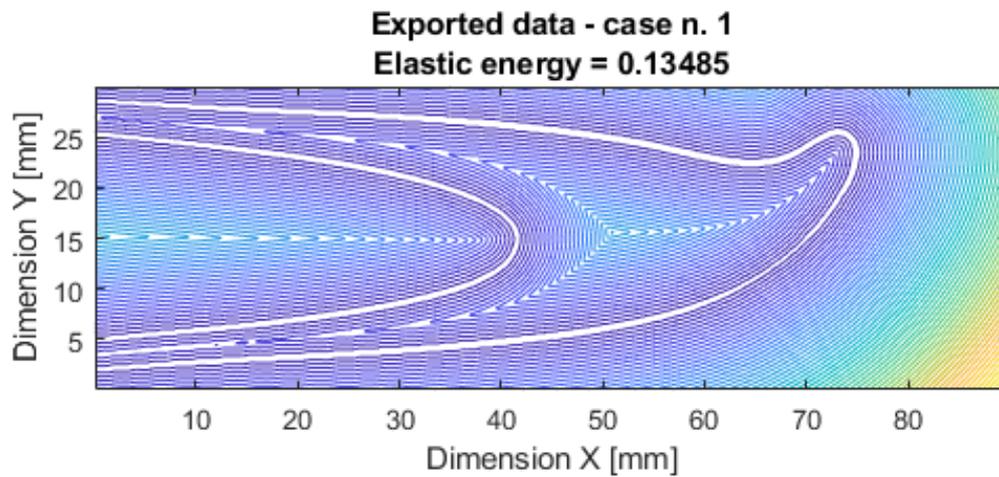
#### Cantilever beam - Distributed Load - Top



(a) Starting point of the optimization.

(b) Optimized fiber angles.

**Figure 5.19:** Starting point and optimized fiber angle directions calculated by the FCFAO algorithm.



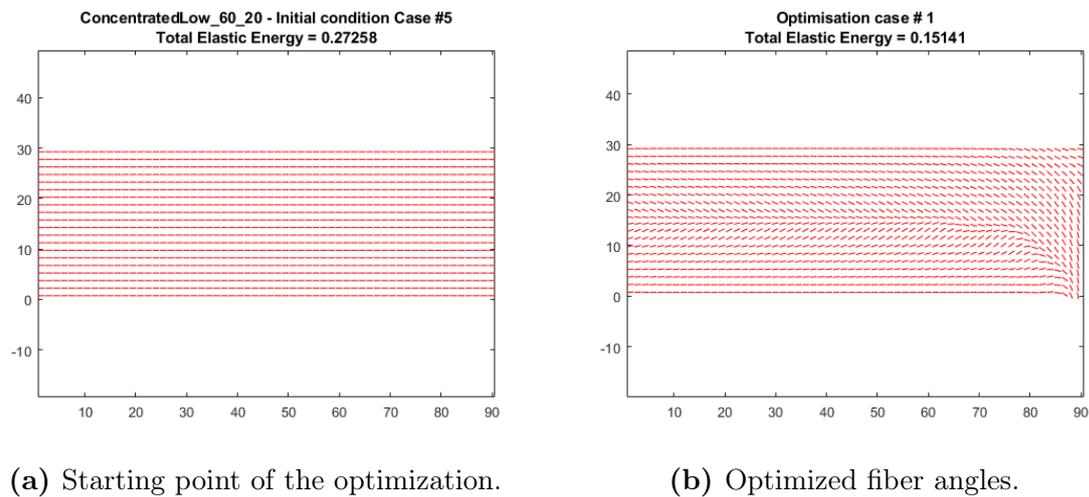
**Figure 5.20:** Optimized paths ready to be printed for the "Cantilever beam - Distributed Load - Top" case.

Cantilever beam - Distributed Load - Top

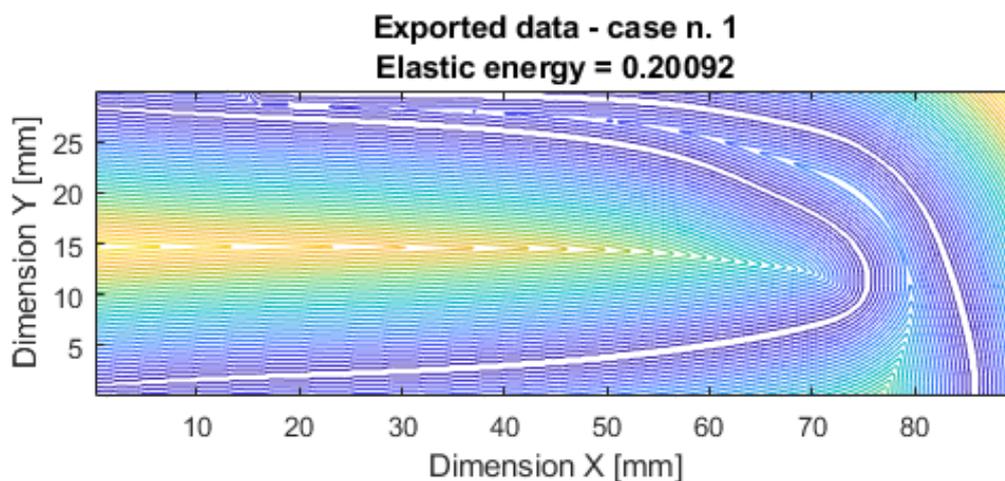
| Mesh size  | X0    | Optimized | Optimized with constr. |
|------------|-------|-----------|------------------------|
| 15x5       | 0,180 | 0,111     | 0,131                  |
| 30x10      | 0,181 | 0,112     | 0,136                  |
| 60x20      | 0,182 | 0,114     | 0,135                  |
| CFAO 60x20 | 0,182 | 0,112     | -                      |

**Table 5.6:** Results of FCFAO algorithm for the Distributed Top load case and comparison with the unfiltered algorithm CFAO.

## Cantilever beam - Concentrated Load - Bottom



**Figure 5.21:** Starting point and optimized fiber angle directions calculated by the FCFAO algorithm.



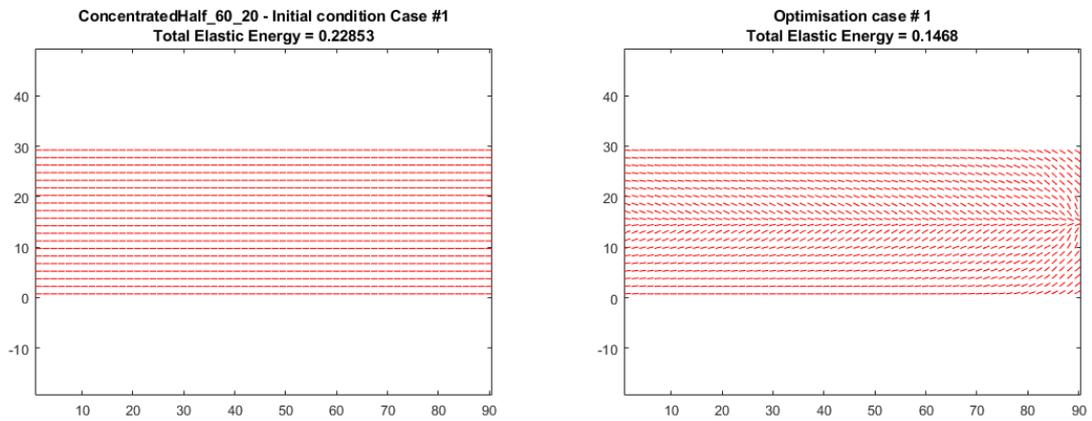
**Figure 5.22:** Optimized paths ready to be printed for the "Cantilever beam - Concentrated Load - Bottom" case.

Cantilever beam - Concentrated Load - Bottom

| Mesh size  | X0    | Optimized | Opt. with constr. |
|------------|-------|-----------|-------------------|
| 15x5       | 0,244 | 0,148     | 0,175             |
| 30x10      | 0,258 | 0,149     | 0,181             |
| 60x20      | 0,273 | 0,151     | 0,200             |
| CFAO 60x20 | 0,273 | 0,151     | -                 |

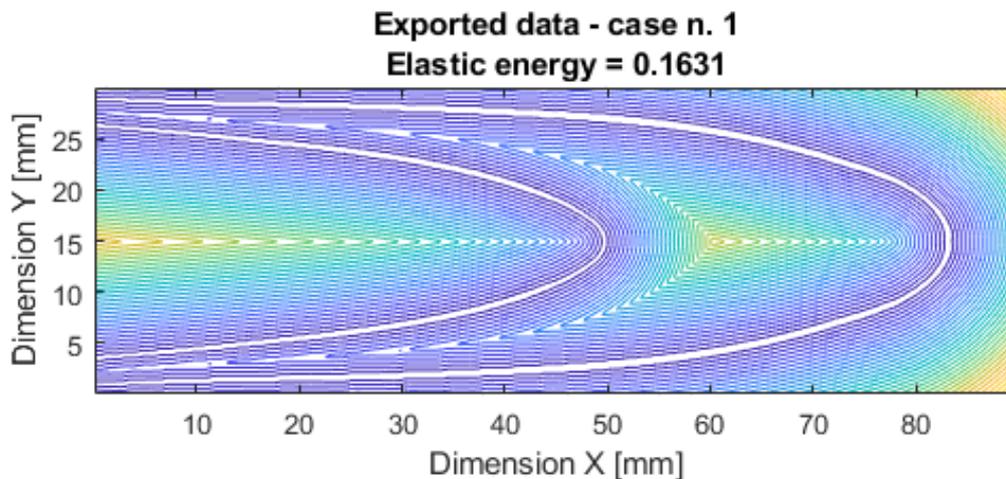
**Table 5.7:** Results of FCFAO algorithm for the Concentrated Bottom load case and comparison with the unfiltered algorithm CFAO.

Cantilever beam - Concentrated Load - Half



(a) Starting point of the optimization.

(b) Optimized fiber angles.



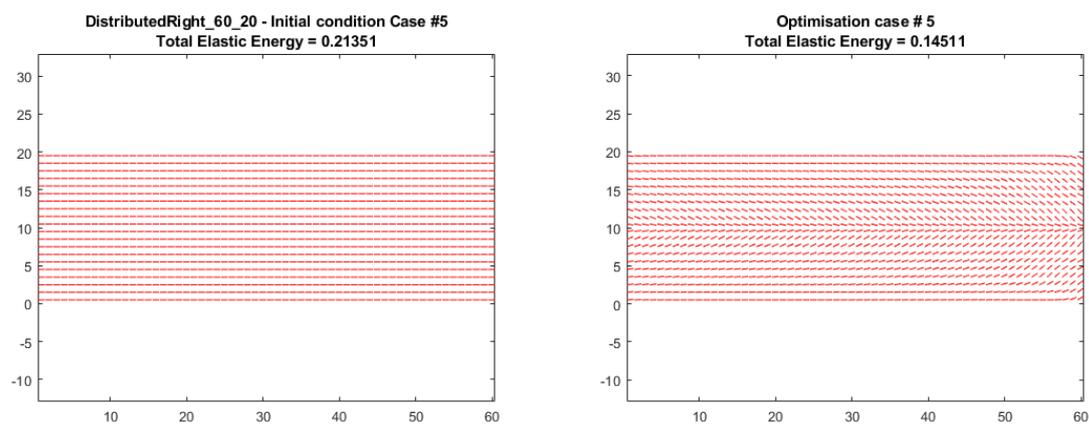
**Figure 5.24:** Optimized paths ready to be printed for the "Cantilever beam - Concentrated Load - Half" case.

## Cantilever beam - Concentrated Load - Half

| Mesh size  | X0    | Optimized | Optimized with constr. |
|------------|-------|-----------|------------------------|
| 15x5       | 0,218 | 0,140     | 0,153                  |
| 30x10      | 0,224 | 0,139     | 0,157                  |
| 60x20      | 0,229 | 0,141     | 0,163                  |
| CFAO 60x20 | 0,229 | 0,140     | -                      |

**Table 5.8:** Results of FCFAO algorithm for the Concentrated Half load case and comparison with the unfiltered algorithm CFAO.

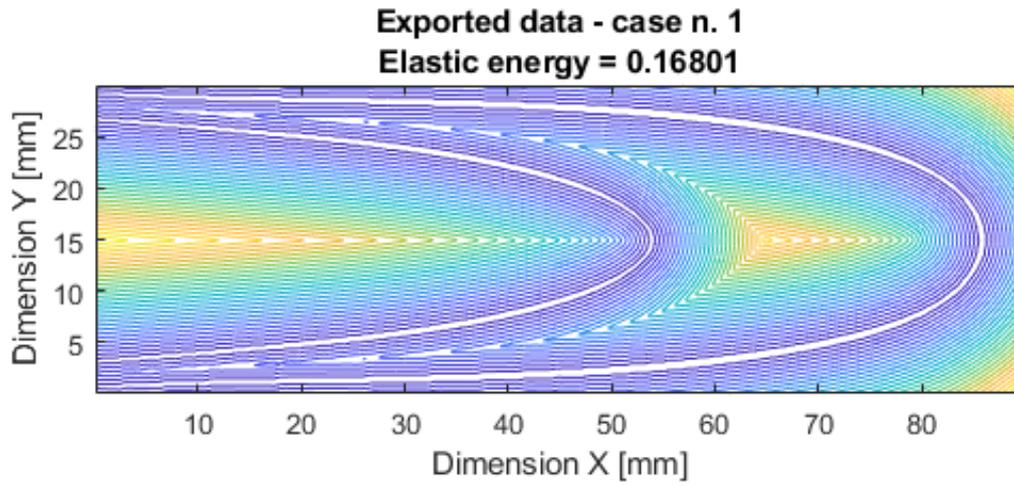
## Cantilever beam - Distributed Load - Right



(a) Starting point of the optimization.

(b) Optimized fiber angles.

**Figure 5.25:** Starting point and optimized fiber angle directions calculated by the FCFAO algorithm.



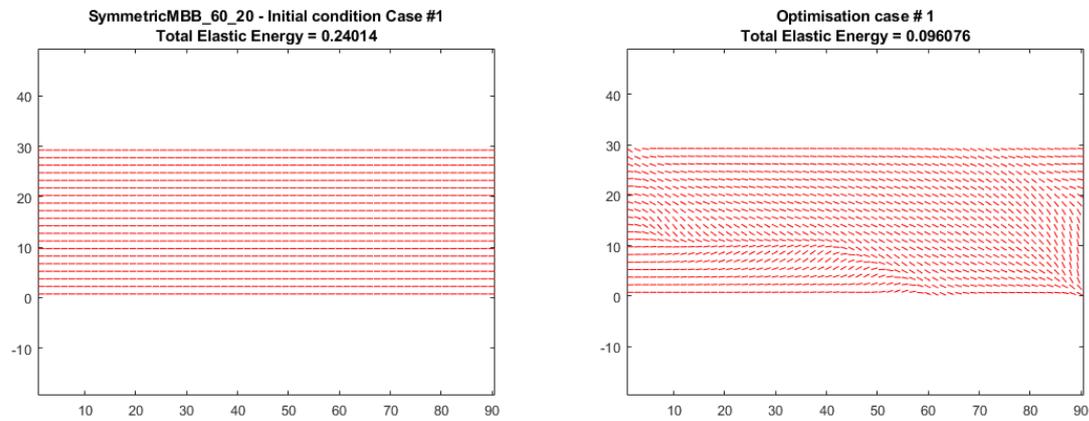
**Figure 5.26:** Optimized paths ready to be printed for the "Cantilever beam - Distributed Load - Right" case.

Cantilever beam - Distributed Load - Right

| Mesh size  | X0    | Optimized | Optimized with constr. |
|------------|-------|-----------|------------------------|
| 15x5       | 0,210 | 0,140     | 0,162                  |
| 30x10      | 0,213 | 0,142     | 0,162                  |
| 60x20      | 0,214 | 0,145     | 0,168                  |
| CFAO 60x20 | 0,214 | 0,140     | -                      |

**Table 5.9:** Results of FCFAO algorithm for the Distributed Right load case and comparison with the unfiltered algorithm CFAO.

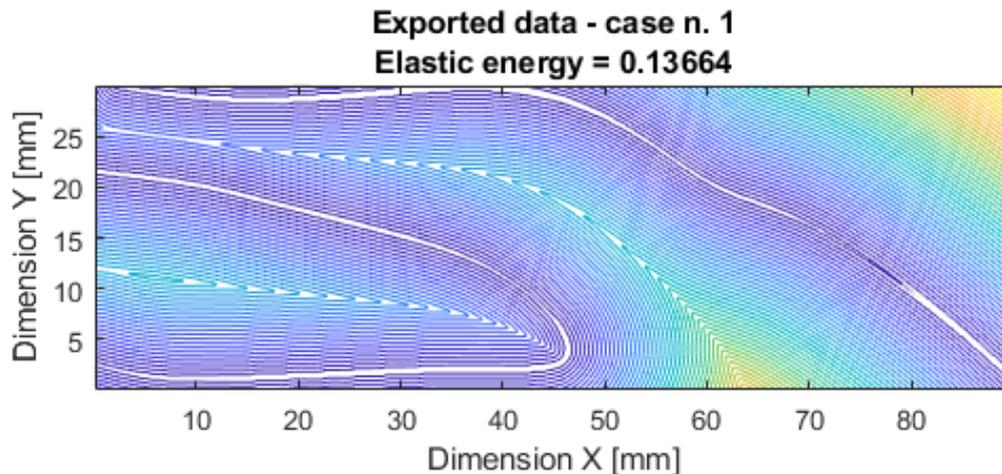
## Messerschmitt-Bölkow-Blohm Beam - Symmetric



(a) Starting point of the optimization.

(b) Optimized fiber angles.

**Figure 5.27:** Starting point and optimized fiber angle directions calculated by the FCFAO algorithm.



**Figure 5.28:** Optimized paths ready to be printed for the "Messerschmitt-Bölkow-Blohm Beam - Symmetric" case.

The MBB beam case was chosen to see how similar the fiber angle algorithm performs compared to the state of the art of topology optimization. The results in figures 5.5 and 5.29 confirm that the two algorithms are trying to reinforce the beam where the stress is maximum, and the correlation between the two is indeed perfect.

Messerschmitt-Bölkow-Blohm Beam - Symmetric

| Mesh size  | X0    | Optimized | Optimized with constr. |
|------------|-------|-----------|------------------------|
| 15x5       | 0,207 | 0,084     | 0,152                  |
| 30x10      | 0,224 | 0,089     | 0,142                  |
| 60x20      | 0,242 | 0,101     | 0,136                  |
| CFAO 60x20 | 0,242 | 0,098     | -                      |

**Table 5.10:** Results of FCFAO algorithm for the MBB symmetric load case and comparison with the unfiltered algorithm CFAO.



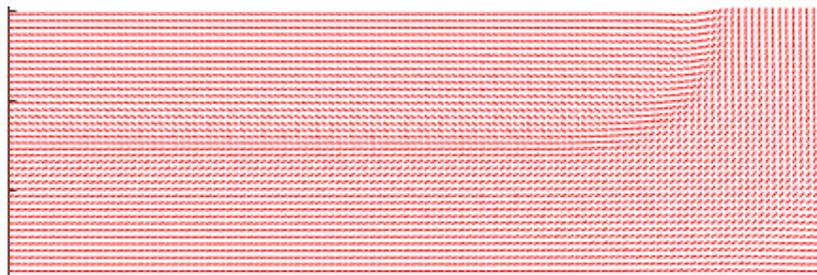
**Figure 5.29:** Graphical representation of the topological optimization of a MBB symmetric beam using SIMP (Labanda, 2014).

### 5.3.4 Comparison with the State of the art

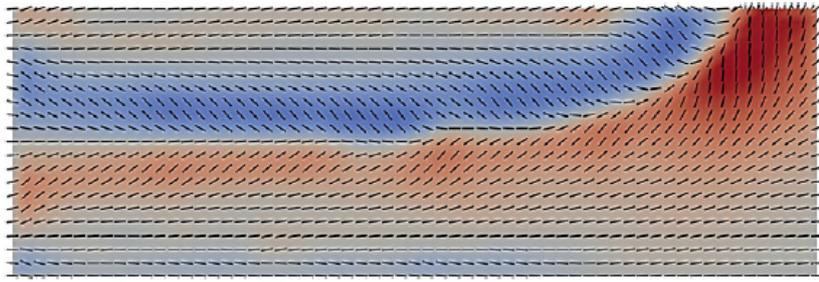
In this Section, a quick comparison with state of the art is carried out to validate the FCFAO algorithm. Since the impossibility to obtain from all the publications the exact value of the dimension of the piece, the forces applied, and the number of elements of the mesh, the comparison is only qualitative. The articles are chosen among state of the art according to two different requirements:

- They must be at maximum three years old.
- They all make use of mathematical formulations to promote continuous fibers.

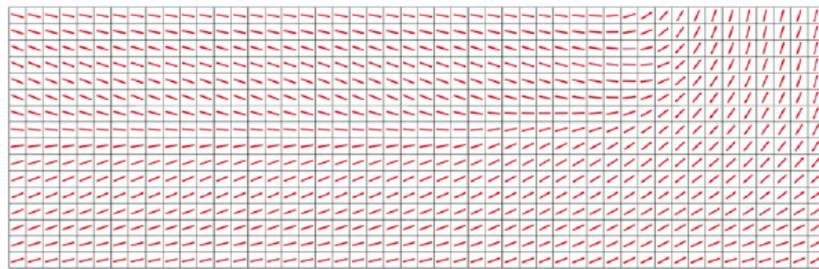
The chosen load case is what is here called "Cantilever beam - Distributed Load - Top". This case represents a perfect test for the optimizing algorithm because it produces bad results when a filter is not used.



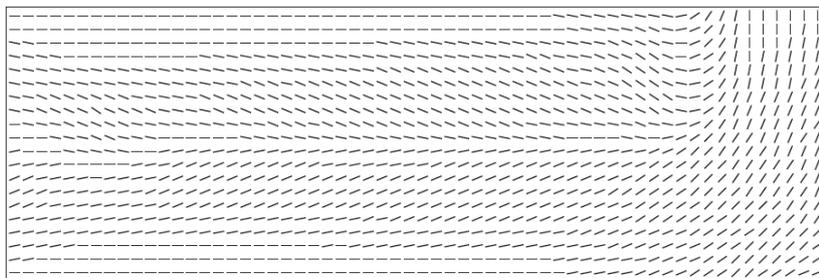
**Figure 5.30:** FCFAO result



**Figure 5.31:** The result of the optimization on the cantilever beam by Silva et al. (2020).



**Figure 5.32:** The result of the optimization on the cantilever beam by Demir et al. (2019).



**Figure 5.33:** The result of the optimization on the cantilever beam by Kiyono et al. (2017). This is the first article where a spatial filter is used to achieve fibers continuity.

Even if it is not possible to numerically compare the results, they look nearly identical. Having already tested the FEM analysis with ABAQUS, it is now possible to validate the FCFAO algorithm.

## 5.4 Unfiltered CFAO

In this section, all the cases mentioned above are tested using a different mathematical formulation. The CFAO algorithm does not utilize a filter to achieve the continuity of the fibers.

### 5.4.1 Settings of the Optimizer

The settings used to obtain the results presented in the next section are the very same in section 5.3.2. Obviously, the filter size and the  $\sigma$  parameter are not defined in this case. The size of the mesh is 60 x 20 elements.

### 5.4.2 Results

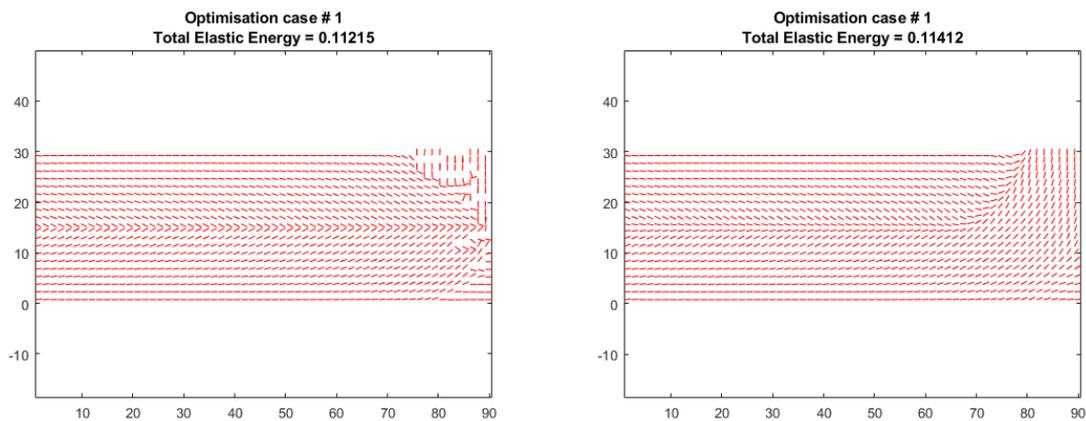
The numerical results are presented in table 5.11. As one would expect, the unfiltered algorithm produces better results from the objective function point of view. However, the stiffness gain is low and never higher than 10 %.

#### Unfiltered CFAO

|            | DistrTOP | ConcHALF | ConcRIGHT | DistrRIGHT | SymmMBB |
|------------|----------|----------|-----------|------------|---------|
| x0         | 0,182    | 0,229    | 0,273     | 0,212      | 0,240   |
| Unfiltered | 0,112    | 0,140    | 0,151     | 0,140      | 0,098   |
| Filtered   | 0,114    | 0,141    | 0,151     | 0,145      | 0,101   |

**Table 5.11:** Results of the Unfiltered CFAO algorithm for all the different load cases. The mesh used is 60 x 20 elements.

As it is possible to note on img. 5.34, the little stiffness gain of the unfiltered method is overcome by the elevate number of discontinuities produced. They tend to make the piece impossible to manufacture, even if they are post-processed with the level-set algorithm.



(a) Unfiltered Optimized fiber angles.

(b) Filtered Optimized fiber angles.

**Figure 5.34:** A comparison between the unfiltered and the filtered analysis for the "DistrTOP" load case.



# Conclusion and Future Works

In the present thesis, a complete framework to optimize and manufacture 3D printed composite parts is presented. A hybrid approach using the mathematics of lamination parameters and the classic composite theory, together with a spatial filter and a gradient-based optimizer was used to formulate the algorithm named FCFAO. The results of the algorithm are post-processed to create the G-code used to 3D print the specimens. In Chapter 4, the complete mathematical formulation is described, and it is later applied in Chapter 5. The load cases tested, together with the printed specimens, verify the full functionality of the framework. The mechanical testing of specimens is not possible because, in the laboratory of ISAE-SUPAERO, the only printer capable of dealing with CFRP is, at the date of writing of this document, a Markforged Mark Two. This machine accepts commands only via the proprietary *Eiger* software that does not permit to customize the continuous fiber paths.

The next logical steps to take to improve the optimizing algorithm are:

- Formulate the extension of the model to deal with composite materials with multiple layers and to optimize the stacking sequence using lamination parameters. However, one should take into account the added computational time due to the algorithm that converts the lamination parameters in the actual stacking sequence.
- Even if the Matlab FEM routine was tested and verified using commercial software, the use of ABAQUS to perform the FEM analysis can be beneficial. This will be especially true when a sub-routine permitting to give a different orientation for every element of the mesh will be written. Even if this approach would mean to cope with higher computational time, it will be useful for pieces with higher complexity and when different test criteria - as buckling or stress performances - are used.
- The algorithm needs to be extended with out-of-plane loads. This will permit to test and verify even further the algorithm, using multiple loads that lead to complex tension cases.
- The algorithm has only been applied to minimize the compliance of a 3D printed part. Other relevant objective functions could be implemented in future versions to extend the usefulness of the code. Other standard objective functions, such as maximizing the lowest buckling load, are just as essential when dealing with thin composite structures such as laminated plates and shells. Additionally, the

maximization of the minimum buckling load according to Svanberg (1994) also displays optimization convexity when formulated with lamination parameters as design variables, and, thus, the performance of such a formulation can be expected to be similar. For unidirectional fibrous composites, a common failure criterion is the Tsai-Wu failure criterion from Tsai and Wu (1970).

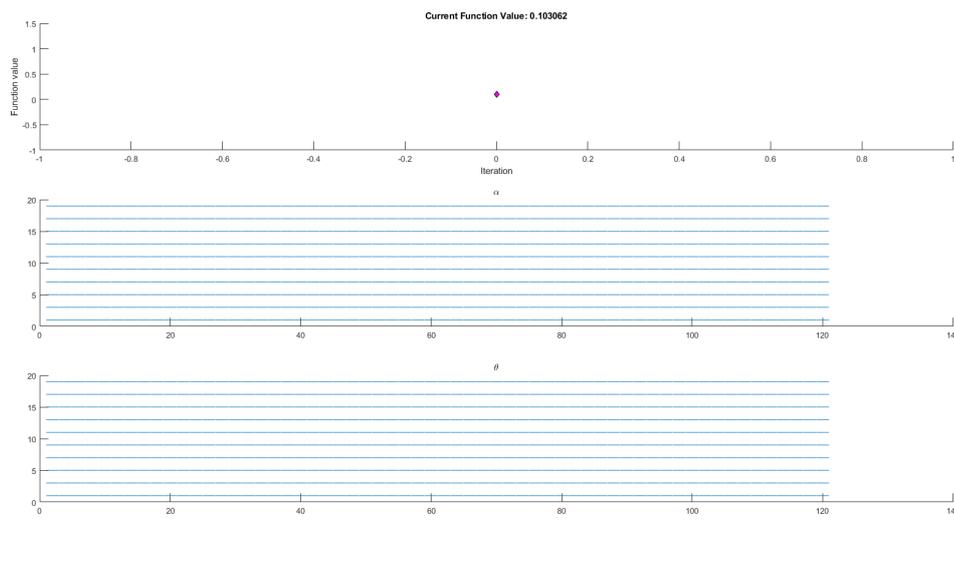
- More sophisticated solution algorithms for the optimization - as SQP or Method of Moving Asymptotes (MMA) - globalization techniques, and filtering techniques - as gradient-based regularization, density filters, and morphological-based filters - could be combined with the proposed material orientation optimization further to improve the convergence and the overall quality of results.
- Further studies should focus on enhancing the quality of the mesh. An unstructured mesh may present some advantages. In this way, it will be possible to refine the mesh in regions of interest, such as the interface or the zones where the curves show high curvature or where thin walls are used.
- Mechanical testing with continuous fiber-reinforced polymer is mandatory to validate the framework. Seen that at the moment of writing this document is difficult to 3D print custom path with continuous carbon fibers, a first test using chopped fiber-reinforced polymer can be performed.

It is the author's hope that this thesis work will help the 3D printing community and the aerospace industry better to understand the process of carbon fiber additive manufacturing. New manufacturing methods are being developed fast these days, and the formulation of the dedicated designing algorithm is the key to the spreading of these technologies. Continuous CFRP additive manufacturing is a new technique, but much work is still needed to be done before the printing process can achieve the needed quality to be aerospace qualified.

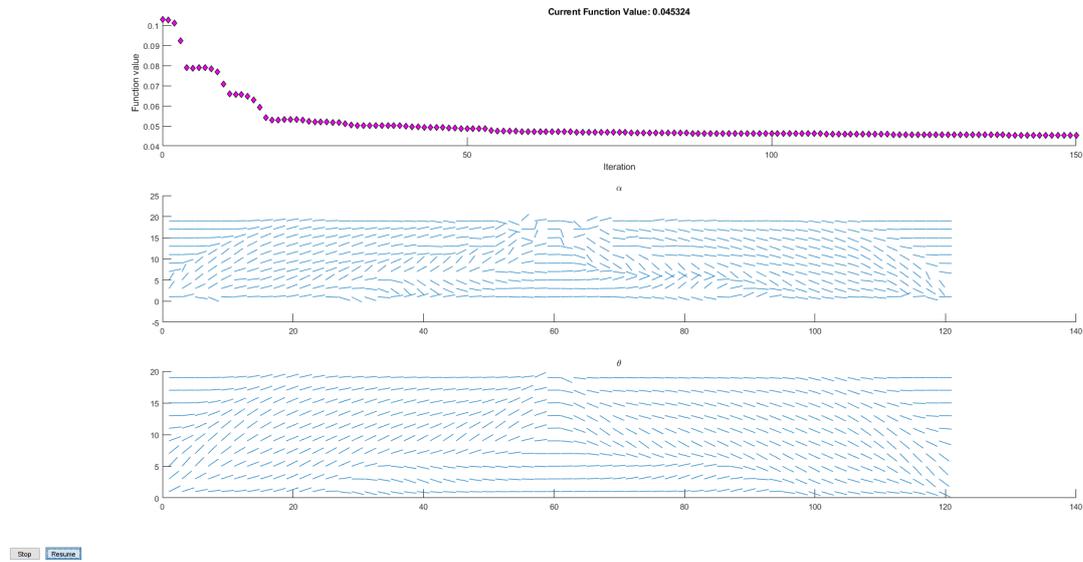
# Appendix A

## Full convergence history for the FCFAO algorithm.

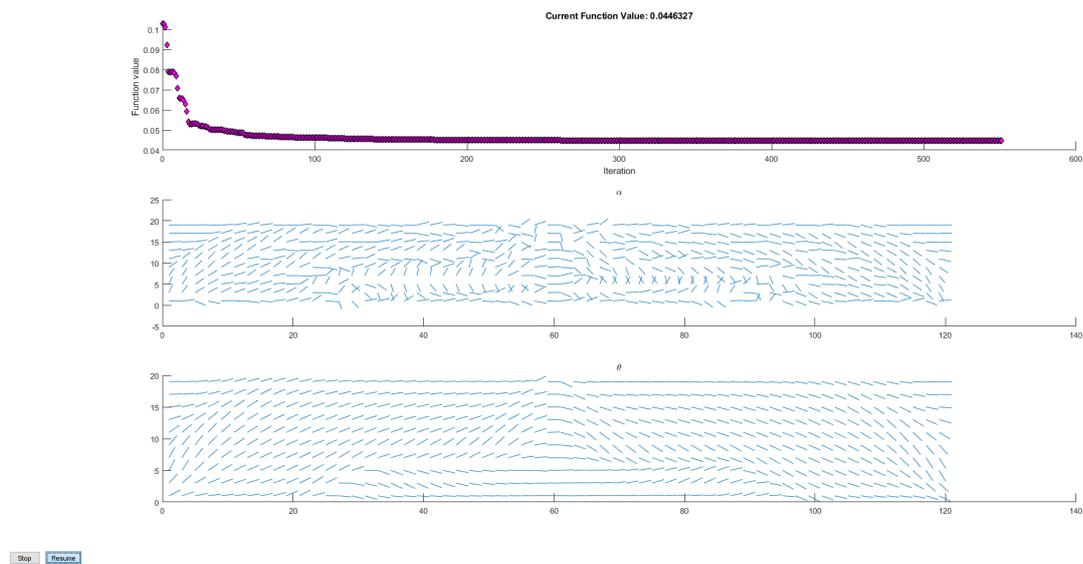
In this appendix is presented the convergence history for the FCFAO algorithm. The full analysis took 924 iterations and here the imagers present the iteration n. 1, 150, 550, and 924. Together with the history of the



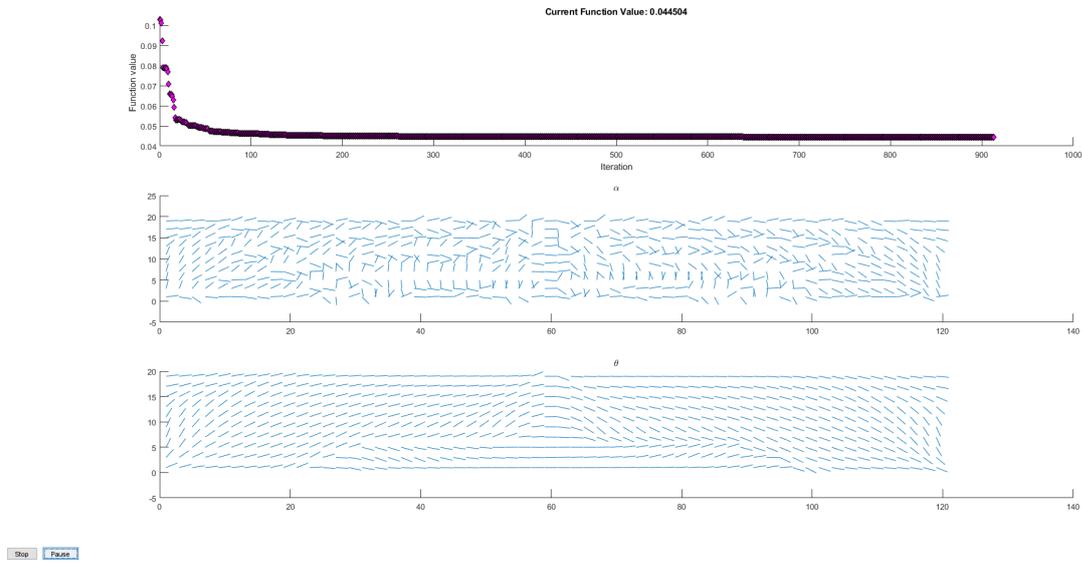
**Figure A.1:** Iteration n. 1 of the FCFAO algorithm used on a 60x10 mesh and MBB load case.



**Figure A.2:** Iteration n. 150 of the FCFAO algorithm used on a 60x10 mesh and MBB load case.



**Figure A.3:** Iteration n. 550 of the FCFAO algorithm used on a 60x10 mesh and MBB load case.



**Figure A.4:** Iteration n. 924 of the FCFAO algorithm used on a 60x10 mesh and MBB load case.



# Appendix B

## Matlab Code

All code can be found at <https://github.com/enricostragiotti/FCFAO>



# List of Abbreviations

|              |   |
|--------------|---|
| <b>3DP</b>   | Three Dimensional Printing                        |
| <b>AFP</b>   | Automated Fiber Placement                         |
| <b>AM</b>    | Additive Manufacturing                            |
| <b>ATL</b>   | Automated Tape Laying                             |
| <b>BCP</b>   | Bi-Value Coding Parameterization                  |
| <b>BCs</b>   | Boundary Conditions                               |
| <b>CAD</b>   | Computer-Aided Design                             |
| <b>CAM</b>   | Computer-Aided Manufacturing                      |
| <b>CFAO</b>  | Continuous Fibre Angle Optimization               |
| <b>CFRP</b>  | Carbon Fiber Reinforced Polymer                   |
| <b>CNC</b>   | Computer Numerical Control                        |
| <b>DMO</b>   | Discrete Material Optimization                    |
| <b>EBM</b>   | Electron Beam Melting                             |
| <b>FCFAO</b> | Filtered Continuous Fibre Angle Optimization      |
| <b>FDM</b>   | Fused Deposition Modelling                        |
| <b>FEM</b>   | Finite Element Method                             |
| <b>FFF</b>   | Fused Filament Fabrication                        |
| <b>FW</b>    | Filament Winding                                  |
| <b>GA</b>    | Genetic Algorithm                                 |
| <b>LENS</b>  | Laser Engineered Net Shaping                      |
| <b>LOM</b>   | Laminated Object Manufacturing                    |
| <b>MBB</b>   | Messerschmitt-Bölkow-Blohm                        |
| <b>MDO</b>   | Multidisciplinary Design Optimization             |
| <b>MMA</b>   | Method of Moving Asymptotes                       |
| <b>NC</b>    | Numerical Controlling                             |
| <b>PDE</b>   | Partial Differential Equation                     |
| <b>SFP</b>   | Shape Functions with Penalization                 |
| <b>SIMP</b>  | Solid Isotropic Material with Penalization method |
| <b>SLA</b>   | Stereolithography                                 |
| <b>SLM</b>   | Selective Laser Melting                           |
| <b>SLS</b>   | Selective Laser Sintering                         |
| <b>SQP</b>   | Sequential Quadratic Programming                  |



# Bibliography

- J. E. Akin. *Finite elements for analysis and design*. Computational mathematics and applications. Acad. Press, London, 2. printing edition, 1995. ISBN 978-0-12-047653-4 978-0-12-047654-1. OCLC: 258678806.
- A. Amini, A. Soleimany, S. Karaman, and D. Rus. Spatial uncertainty sampling for end-to-end control. 05 2018.
- M. Bendsøe. Optimal shape design as a material distribution problem. page 10, 1989.
- R. Z. G. Bohrer, S. F. M. de Almeida, and M. V. Donadon. Optimization of composite plates subjected to buckling and small mass impact using lamination parameters. *Composite Structures*, 120:141–152, Feb. 2015. ISSN 0263-8223. doi: 10.1016/j.compstruct.2014.09.043. URL <http://www.sciencedirect.com/science/article/pii/S0263822314004917>.
- C. J. Brampton, K. C. Wu, and H. A. Kim. New optimization method for steered fiber composites using the level set method. *Structural and Multidisciplinary Optimization*, 52(3):493–505, Sept. 2015. ISSN 1615-147X, 1615-1488. doi: 10.1007/s00158-015-1256-6. URL <http://link.springer.com/10.1007/s00158-015-1256-6>.
- T. Brooks and J. Martins. On Manufacturing Constraints for Tow-steered Composite Design Optimization. *Composite Structures*, 204, Aug. 2018. doi: 10.1016/j.compstruct.2018.07.100.
- M. Bruyneel. SFP-a new parameterization based on shape functions for optimal material selection: Application to conventional composite plies. *Structural and Multidisciplinary Optimization*, 43:17–27, Jan. 2011. doi: 10.1007/s00158-010-0548-0.
- J. Cantrell, S. Rohde, D. Damiani, R. Gurnani, L. DiSandro, J. Anton, A. Young, A. Jerez, D. Steinbach, C. Kroese, and P. Ifju. Experimental Characterization of the Mechanical Properties of 3D-Printed ABS and Polycarbonate Parts. page 19, Sept. 2016.
- A. Chandrasekhar, T. Kumar, and K. Suresh. Build optimization of fiber-reinforced additively manufactured components. *Structural and Multidisciplinary Optimization*, 61(1):77–90, Jan. 2020. ISSN 1615-1488. doi: 10.1007/s00158-019-02346-z. URL <https://doi.org/10.1007/s00158-019-02346-z>.

- J. Chen and Y. Han. Shortest paths on a polyhedron. In *Proceedings of the sixth annual symposium on Computational geometry*, SCG '90, pages 360–369, Berkley, California, USA, May 1990. Association for Computing Machinery. ISBN 978-0-89791-362-1. doi: 10.1145/98524.98601. URL <https://doi.org/10.1145/98524.98601>.
- K. Crane, C. Weischedel, and M. Wardetzky. The Heat Method for Distance Computation. *Commun. ACM*, 60(11):90–99, Oct. 2017. ISSN 0001-0782. doi: 10.1145/3131280. URL <http://doi.acm.org/10.1145/3131280>.
- E. Demir, P. Yousefi-Louyeh, and M. Yildiz. Design of variable stiffness composite structures using lamination parameters with fiber steering constraint. *Composites Part B: Engineering*, 165:733–746, May 2019. ISSN 1359-8368. doi: 10.1016/j.compositesb.2019.02.004. URL <http://www.sciencedirect.com/science/article/pii/S1359836818336631>.
- A. N. Dickson, J. N. Barry, K. A. McDonnell, and D. P. Dowling. Fabrication of continuous carbon, glass and Kevlar fibre reinforced polymer composites using additive manufacturing. *Additive Manufacturing*, 16:146–152, Aug. 2017. ISSN 22148604. doi: 10.1016/j.addma.2017.06.004. URL <https://linkinghub.elsevier.com/retrieve/pii/S2214860416301889>.
- E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, Dec. 1959. ISSN 0945-3245. doi: 10.1007/BF01386390. URL <https://doi.org/10.1007/BF01386390>.
- T. Gao, W. H. Zhang, and P. Duysinx. Simultaneous design of structural layout and discrete fiber orientation using bi-value coding parameterization and volume constraint. *Structural and Multidisciplinary Optimization*, 48(6):1075–1088, Dec. 2013. ISSN 1615-1488. doi: 10.1007/s00158-013-0948-z. URL <https://doi.org/10.1007/s00158-013-0948-z>.
- F. Gibou, R. Fedkiw, and S. Osher. A review of level-set methods and some recent applications. *Journal of Computational Physics*, 353:82–109, Jan. 2018. ISSN 0021-9991. doi: 10.1016/j.jcp.2017.10.006. URL <http://www.sciencedirect.com/science/article/pii/S0021999117307441>.
- H. T. Hahn and S. W. Tsai. *Introduction to Composite Materials*. CRC Press, Jan. 1980. ISBN 978-0-87762-288-8.
- R. Hale, R. Moon, K. Lim, K. Schueler, and M. Wiehn. Current Progress on Integrated Design and Analysis Tools for Fiber Steered Composites. Apr. 2002. doi: 10.2514/6.2002-1763. URL <http://arc.aiaa.org/doi/10.2514/6.2002-1763>.
- S. H. Huang, P. Liu, A. Mokasdar, and L. Hou. Additive manufacturing and its societal impact: a literature review. *The International Journal of Advanced Manufacturing Technology*, 67(5-8):1191–1203, July 2013. ISSN 0268-3768, 1433-3015. doi: 10.1007/s00170-012-4558-5. URL <http://link.springer.com/10.1007/s00170-012-4558-5>.

- 
- D. R. Jantos, K. Hackl, and P. Junker. Topology optimization with anisotropic materials, including a filter to smooth fiber pathways. *Structural and Multidisciplinary Optimization*, Feb. 2020. ISSN 1615-147X, 1615-1488. doi: 10.1007/s00158-019-02461-x. URL <http://link.springer.com/10.1007/s00158-019-02461-x>.
- D. Jiang, R. Hoglund, and D. E. Smith. Continuous Fiber Angle Topology Optimization for Polymer Composite Deposition Additive Manufacturing Applications. *Fibers*, 7(2):14, Feb. 2019. doi: 10.3390/fib7020014. URL <https://www.mdpi.com/2079-6439/7/2/14>.
- Johnson. History of composites, Feb. 2020. URL <https://www.thoughtco.com/history-of-composites-820404>.
- R. M. Jones. *Mechanics of composite materials*. Taylor & Francis, New York, 2. ed edition, 1999. ISBN 978-1-56032-712-7. OCLC: 633770931.
- S. Karakaya and O. Soykasap. Buckling optimization of laminated composite plates using genetic algorithm and generalized pattern search algorithm. *Structural and Multidisciplinary Optimization*, 39(5):477, Jan. 2009. ISSN 1615-1488. doi: 10.1007/s00158-008-0344-2. URL <https://doi.org/10.1007/s00158-008-0344-2>.
- C. Y. Kiyono, E. C. N. Silva, and J. N. Reddy. A novel fiber optimization method based on normal distribution function with continuously varying fiber path. *Composite Structures*, 160:503–515, Jan. 2017. ISSN 0263-8223. doi: 10.1016/j.compstruct.2016.10.064. URL <http://www.sciencedirect.com/science/article/pii/S0263822316314957>.
- S. R. Labanda. Benchmarking of optimization methods for topology optimization problems. page 24, 2014.
- B. Liu, R. Haftka, and P. Trompette. Maximization of buckling loads of composite panels using flexural lamination parameters. *Structural and Multidisciplinary Optimization*, 26(1):28–36, Jan 2004. ISSN 1615-1488. doi: 10.1007/s00158-003-0314-7. URL <https://doi.org/10.1007/s00158-003-0314-7>.
- J. Liu and H. Yu. Concurrent deposition path planning and structural topology optimization for additive manufacturing. *Rapid Prototyping Journal*, 23(5):930–942, Aug. 2017. ISSN 1355-2546. doi: 10.1108/RPJ-05-2016-0087. URL <http://www.emeraldinsight.com/doi/10.1108/RPJ-05-2016-0087>.
- D. H.-J. Lukaszewicz, C. Ward, and K. D. Potter. The engineering aspects of automated prepreg layup: History, present and future. *Composites Part B: Engineering*, 43(3): 997–1009, Apr. 2012. ISSN 13598368. doi: 10.1016/j.compositesb.2011.12.003. URL <https://linkinghub.elsevier.com/retrieve/pii/S1359836811005452>.
- J. H. Luo and H. C. Gea. Optimal orientation of orthotropic materials using an energy based method. *Structural optimization*, 15(3):230–236, June 1998. ISSN 1615-1488. doi: 10.1007/BF01203536. URL <https://doi.org/10.1007/BF01203536>.

- Y. Luo, W. Chen, S. Liu, Q. Li, and Y. Ma. A discrete-continuous parameterization (DCP) for concurrent optimization of structural topologies and continuous material orientations. *Composite Structures*, 236:111900, Mar. 2020. ISSN 0263-8223. doi: 10.1016/j.compstruct.2020.111900. URL <http://www.sciencedirect.com/science/article/pii/S0263822319331241>.
- Markforged. Composites Data Sheet, Sept. 2019. URL <http://static.markforged.com/downloads/composites-data-sheet.pdf>.
- R. Martins. A Short Course on Multidisciplinary Design Optimization, 2012. URL [http://adl.stanford.edu/aa222/Lecture\\_Notes\\_files/chapter3\\_gradient.pdf](http://adl.stanford.edu/aa222/Lecture_Notes_files/chapter3_gradient.pdf).
- R. Matsuzaki, M. Ueda, M. Namiki, T.-K. Jeong, H. Asahara, K. Horiguchi, T. Nakamura, A. Todoroki, and Y. Hirano. Three-dimensional printing of continuous-fiber composites by in-nozzle impregnation. *Scientific Reports*, 6(1):23058, Sept. 2016. ISSN 2045-2322. doi: 10.1038/srep23058. URL <http://www.nature.com/articles/srep23058>.
- S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, Nov. 1988. ISSN 00219991. doi: 10.1016/0021-9991(88)90002-2. URL <https://linkinghub.elsevier.com/retrieve/pii/0021999188900022>.
- M. Rakhshbahar and M. Sinapius. A Novel Approach: Combination of Automated Fiber Placement (AFP) and Additive Layer Manufacturing (ALM). page 9, 2018.
- N. Ranaivomiarana, F.-X. Irisarri, D. Bettebghor, and B. Desmorat. Concurrent optimization of material spatial distribution and material anisotropy repartition for two-dimensional structures. *Continuum Mechanics and Thermodynamics*, 31(1):133–146, Jan. 2019. ISSN 1432-0959. doi: 10.1007/s00161-018-0661-7. URL <https://doi.org/10.1007/s00161-018-0661-7>.
- J. A. Sethian. A Fast marching level Set Method for Monotonically Advancing Fronts. *Comp. Physics*, 118, pages 269–277, 1995. URL <https://math.berkeley.edu/~sethian/2006/Papers/sethian.fastmarching.pdf>.
- J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, June 1999. ISBN 978-0-521-64557-7. Google-Books-ID: ErpOoynE4dIC.
- O. Sigmund. On the usefulness of non-gradient approaches in topology optimization. *Structural and Multidisciplinary Optimization*, 43(5):589–596, May 2011. ISSN 1615-1488. doi: 10.1007/s00158-011-0638-7. URL <https://doi.org/10.1007/s00158-011-0638-7>.
- O. Sigmund and K. Maute. Topology optimization approaches: A comparative review. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, Dec. 2013.

- 
- ISSN 1615-147X, 1615-1488. doi: 10.1007/s00158-013-0978-6. URL <http://link.springer.com/10.1007/s00158-013-0978-6>.
- A. L. F. d. Silva, R. A. Salas, E. C. Nelli Silva, and J. N. Reddy. Topology optimization of fibers orientation in hyperelastic composite material. *Composite Structures*, 231, Jan. 2020. ISSN 0263-8223. doi: 10.1016/j.compstruct.2019.111488. URL <http://www.sciencedirect.com/science/article/pii/S0263822319306786>.
- A. Simchi and H. Pohl. Direct laser sintering of iron–graphite powder mixture. *Materials Science and Engineering: A*, 383(2):191–200, Oct. 2004. ISSN 09215093. doi: 10.1016/j.msea.2004.05.070. URL <https://linkinghub.elsevier.com/retrieve/pii/S0921509304007427>.
- J. Stegmann and E. Lund. Discrete material optimization of general composite shell structures. *International Journal for Numerical Methods in Engineering*, 62(14):2009–2027, Apr. 2005. ISSN 0029-5981, 1097-0207. doi: 10.1002/nme.1259. URL <http://doi.wiley.com/10.1002/nme.1259>.
- K. Svanberg. The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987. ISSN 1097-0207. doi: 10.1002/nme.1620240207. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620240207>.
- K. Svanberg. On the convexity and concavity of compliances. *Structural optimization*, 7(1):42–46, Feb. 1994. ISSN 1615-1488. doi: 10.1007/BF01742502. URL <https://doi.org/10.1007/BF01742502>.
- R. Sørensen and J. Kann. *Optimisation of Composite Structures Using Lamination Parameters in a Finite Element Application*. PhD thesis, Aalborg University, 2011. URL [https://projekter.aau.dk/projekter/files/52669014/MatersThesisDMS10\\_29a\\_2011.pdf](https://projekter.aau.dk/projekter/files/52669014/MatersThesisDMS10_29a_2011.pdf).
- B. Tatting and Z. Gurdal. Design and manufacture of elastically tailored tow placed plates. Jan. 2002.
- S. W. Tsai and E. M. Wu. A General Theory of Strength for Anisotropic Materials. page 23, Nov. 1970.
- S. R. S. Varadhan. On the behavior of the fundamental solution of the heat equation with variable coefficients. *Communications on Pure and Applied Mathematics*, 20(2):431–455, 1967. ISSN 1097-0312. doi: 10.1002/cpa.3160200210. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160200210>.
- Q. Xia and T. Shi. Optimization of composite structures with continuous spatial variation of fiber angle through Shepard interpolation. *Composite Structures*, 182, Sept. 2017. doi: 10.1016/j.compstruct.2017.09.052.