ABSTRACT

Three Dimensional Topology Optimization with Orthotropic Material Orientation Design
for Additive Manufacturing Structures

Delin Jiang, M.S.M.E

Mentor: Douglas E. Smith, Ph.D.

Products produced with Additive Manufacturing often have anisotropic microstructures that form material layers are added during processing. Carbon fiber filled polymer deposited in beads with the Fused Filament Fabrication (FFF) process, for example, has been shown to have a highly anisotropic material response. This thesis considers the anisotropic behavior of FFF structures and presents a three dimensional topology optimization method that computes the best anisotropic material distribution and direction for minimum compliance of statically loaded AM structure. The optimization method is applied to single loaded problem, multiple loaded problem, and structures with regions of prescribed solid or void. Results show that printing the structures in different planes yields different minimum compliance, and that the lowest overall compliance is achieved when the applied load vector is parallel to the plane of material orientation. Furthermore, the preferred material orientation generally follows the outer contour of the dense material region for each layer.

Three Dimensional Topology Optimization with Orthotropic Material Orientation Design for
Additive Manufacturing Structures

by

Delin Jiang, B.S.M.E.

A Thesis

Approved by the Department of Mechanical Engineering

———————————————————————

Kenneth Van Treuren, PH.D., Chairperson

Submitted to the Graduate Faculty of
Baylor University in Partial Fulfillment of the
Requirements for the Degree
of

Master of Science in Mechanical Engineering

Approved by the the Thesis Committee

———————————————————————

Douglas E. Smith, Ph.D., Chairperson

———————————————————————

David A. Jack, Ph.D.

———————————————————————

Ronald B. Morgan, Ph.D.

Accepted by the Graduate School

August 2017

———————————————————————

J. Larry Lyon, Ph.D., Dean

TABLE OF CONTENTS

LIST OF FIGURES

vii

LIST OF TABLES

ACKNOWLEDGMENTS

I am extremely grateful for Dr. Douglas Smith for his guidance of my Master's work. Also, I would like to thank Dr. Douglas Smith, Dr. David Jack and Dr. Ronald Morgan for being my thesis committee and gave me very valuable advices to improve my understanding of my research. I would also like to thank for the other professors in Baylor University whom have taught me great lessons to improve my knowledge in engineering skills, as well as Baylor University for providing the funding and equipment to make this research possible. Furthermore, thanks to all the colleagues I have been interacted with in the lab; they have provide essential assistance to help me to finish this thesis work. Lastly, I would like to thank my family who has supported me in this academic journey; without their encouragement I could not have persevered the long hours of study.

DEDICATION

To my Grandpa, Yucai Cheng.

CHAPTER ONE

Introduction

1.1.   Fused Filament Fabrication and Topology Optimization

Fused filament fabrication (FFF) is one of the most popular Additive Manufacturing (AM) techniques in the market. This polymer extrution deposition technique gives FFF the advantage to produce parts with intricate geometry and save raw material at the same time. The market value of FFF has been growing exponentially [1], with increasingly more desktop 3D printers being sold annually for the past decade. A typical desktop 3D printer has 3 axes of motion; the extruder can move freely in a plane to deposit material onto a flat platform, and that same platform can be move up and down. In the printing process, a layer of material is deposited by the extruder, and thenafter the completion of the first layer, the platform moves down and the second layer is deposited on top of the first layer. Through this layer-by-layer printing process, a 3D object is made. While the thermoplastic-based feedstock used in FFF printing does not have sufficient mechanical properties for parts used in many industrial applications, the development of short carbon fiber filled (CFF) polymer feedstocks has shown promising results for producing parts with enhanced mechanical performance. In some cases the printed parts yielded higher specific strength than alloyed Aluminum [2]. More recently, the emergence of large scale 3D printing shows great prospect to be used for industrial manufacturing [3], such as printing car chassis, wind turbine blade molds and more.

For both small scale FFF and large scale polymer extrusion deposition 3D printing, the application of CFF polymer feedstock offers unique advantages since the fiber orientation can be defined based on the prescribed printing direction, making it possible to predict the mechanical properties of the parts before printing begins [2]. Unfortunately, computational approaches for determining the preferred bead direction that gives the stiffest or strongest parts have not been well addressed in the literature (see e.g., [4,5]). Topology optimization is a computational tool for computing the optimum material distribution of a structure for optimum properties. It has been applied in industry and in numerous research fields, including several AM techniques (see e.g. [6]). It is theorized in this thesis that it may also be a viable tool for designing the structure of FFF parts composed of carbon fiber filled polymer, if the method can incorporate anisotropic material properties in the optimization algorithm.

1.2.   *Brief Summary of Thesis Contents*

This thesis presents a topology optimization scheme that designs specifically for parts produced by additive manufacturing methods having anisotropic material behavior, by determining the optimum material distribution and orthotropic material orientations. It has six chapters in total. Chapter Two provides a background for the thesis research by discussing the development of FFF technology, the drawbacks of the thermoplastic polymer feedstocks, the potential of CFF polymer stocks with FFF and the expansion of FFF to large 3D printing. Chapter Two also presents several popular topology optimization methods and optimization algorithms are introduced. The preferred method, the Solid Isotropic Material Penalization (SIMP) then chosen, and its drawback as well as mitigation schemes are addressed. Finally, the research objective is stated.

In Chapter Three, the anisotropy of FFF processing is demonstrated through mechanical testing and material evaluations. Four different types of the thermoplastic material blended with short carbon fibers (CFs) are tested along with their neat thermoplastic counterparts. Test coupons are printed at different orientations, and tensile strength and tensile modulus are analyzed and compared. The details of how print orientation affects the tensile properties are discussed in detail. To measure how fiber length change before and after the extrusion, fiber length measurement on test coupons before and after the printing are analyzed. Furthermore, fractography is performed under the scanning electron microscope.

In Chapter Four, the discussion starts by presenting the general optimization mathematical problem statement. It is worthy to note that the problem of optimization can be involved in various academic fields (see e.g., [7]), but for all considerations they can be represented into one set of mathematical formula [7]. After the generali optimization problem is established, the Solid Isotropic Material Penalization (SIMP) method is formulated, and the extension of the SIMP method to accommodate fiber orientation is presented. The linear elastic static three dimensional Finite Element Analysis (FEA) is discussed in detail, including the derivation from the Strong to the Weak form, the selection of three dimensional isoparametric element and the choice of constitutive material model. Afterward, the design sensitivity with respect to each type of design variable is also derived. The chapter finishes the discussion by presenting the optimization solver developed by Matlab and explains how it can be integrated with extended SIMP method to solve for the minimum compliance as the objective function.

In Chapter Five, the design sensitivities obtained from analytical method are first verified by the finite difference method, an easy-to-implement numerical approximation. Then, several numerical examples are shown to demonstrate the capability of the extended SIMP method in three dimension to optimizing material and anisotropic material orientation distribution. In chapter six, we make some conclusion about some of the significant findings in our research, and we present several ideas about what can be done in the future based on this work.

1.3.   *Research Objective*

The goal of this thesis is to develop a three dimensional optimization approach specifically for orthotropic materials produced with the layer-by-layer FFF process. The approach pursued hereis based on the SIMP and CFAO methods, where the material orientation can only rotate in their respective printed layer. We are particularly interested in processing material with CFF polymer feedstocks; therefore, the constitutive matrix should include the material anisotropy where there is a preferred direction having the highest stiffness. An important assumption to make is that since the FFF printed parts are made through a layer-by-layer process, the orientation of the fibers are constrained in the print plane. This assumption is valid from the fact that fibers align along the print direction; therefore, it reduces the complexity of design problem.

The design tool will provide insight on the optimal topology and the best bead pattern of the printed part, therefore making the FFF process more efficient and competitive in the market. We will demonstrate the effectiveness of our method on structural problems that involve a cantilever beam in different print planes, a multiply loaded structure, an L-shape structure and a structure with prescribed outline. Before solving these problems, we

first show how the print direction affects the mechanical properties of the printed samples. As will be shown in Chapter 3, experimental results of the printed coupons at different print orientations are discussed. Several different kinds of thermoplastic materials blended with CFs are tested, along with their pure polymer counterparts. In the Chapter, some literature background regarding CFF FFF material testing and topology optimization are discussed.

CHAPTER TWO

Literature Review

Additive Manufacturing (AM) builds a part by adding material to the desired space on the platform, in a layer by layer fashion. As opposed to subtractive manufacturing, such as machining or CNC, AM saves raw material and can reduce production time and tooling cost. Furthermore, AM can be used to build very intricate geometry that is deemed impossible with conventional manufacturing techniques. The extra complexity and flexibility the AM technology attracts attention from many industries. The first Additive Manufacturing commercial machine was introduced in 1987, when a stereolithography system was invented to solidify liquid polymer using laser to form the parts [8]. From then, numerous techniques were commercialized. In general, AM can be categorized into seven processes: binder jetting, direct energy deposition, material extrusion, material jetting, powder bed fusion, sheet lamination and vat photopolymerization [9]. The market for AM technology has grown significantly over the past two decades. According to the 2016 Wohlers Report [1], the AM industry exceeded $5.1 billion in 2015, revealing a $1 billion increase for the second consecutive year. The report further shows more companies are adopting the AM technology. In 2015, 62 manufacturers sold industrial-graded AM systems, compared to 49 in 2014 which was twice as many as in 2011. Government sectors such as National Science Foundation and Department of Defense have provided significant funding to support research in AM; industries such as aerospace and automotive have also invested heavily in AM to improve their manufactured parts. The current trend reveals that

the AM technology will continue to expand and play a more prominent role in the future society.

The focus of this research is particularly the Fused Filament Fabrication (FFF) process, a popular AM technique. Throughout the last two decades, the market for FFF has been gradually expanding [1]. The CFF polymer feedstocks are expected to further push FFF technology to be more widely used in industry. In Section 2.1, we discuss the popularity, the market values and the drawbacks of FFF; also, the potential of blending short carbon fibers (CF) with polymer feedstock will be discussed. In Section 2.2 and Section 2.3, topology optimization is considered as a valuable tool for designing CFF FFF processed products where different design methods and auxiliary tools are also discussed. In Section 2.4 and Section 2.5, some historical studies regarding the application of topology optimization with AM are described using different material models in two dimension. In Section 2.6, we discuss the historical studies of topology optimization in three dimensions. Lastly, in Section 1.3 we give the research objective for this thesis.

2.1. *Fused Filament Fabrication with Carbon Fiber Filled Polymer*

2.1.1. *Fused Filament Fabrication*

FFF is an AM technique that takes the polymer feedstock in the form of a filament through a heated nozzle. The molten material is then deposited onto a platform moving relative to the nozzle to print a part through a layer-by-layer fashion. A schematic taken from [10] of the FFF process is shown in Figure 2.1. Note that multiple materials can be printed in the same part by utilizing multiple nozzles. The cost of the FFF machine varies, ranging from a couple hundred to tens of thousands of dollars depending on the build

precisions, build volume and other features. Therefore, it is very affordable for hobbyists who just want print objects for fun; it also attracts academic researchers who want to build models for experimental testing. Figure 2.2, taken from [1], shows the number of desktop 3D printers sold annually, from 2007 to 2015. Sales of the desktop 3D printers are shown to increase in an exponential meaner, and it is likely to continue for the next several years.
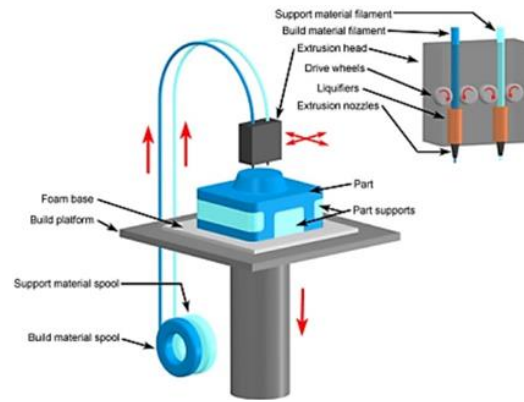
Figure 2.1. Fused filament fabrication Process [10]

Despite its advantages over conventional manufacturing process, FFF printed parts often have inferior mechanical properties (see e.g, [11,12]). The base materials used for printing are mostly thermoplastic polymers, which are typically weaker than metals [11,12]. In addition, print orientations lead to anisotropic material properties of the printed parts (see e.g., [11−14]), and the layer-by-layer print deposition process can produce voids in the printed samples, resulting in lower tensile strength than that found in the injection molded samples [11]. These inferior mechanical properties are illustrated in Figure 2.3. In order to print parts that give mechanical properties applicable for industrial applications and end use, the mechanical properties (tensile strength, tensile modulus and such) of the print materials need to be enhanced.

Source: Wohlers Report 2016

More than 278,000 desktop (under $5,000) 3D printers were sold worldwide last year
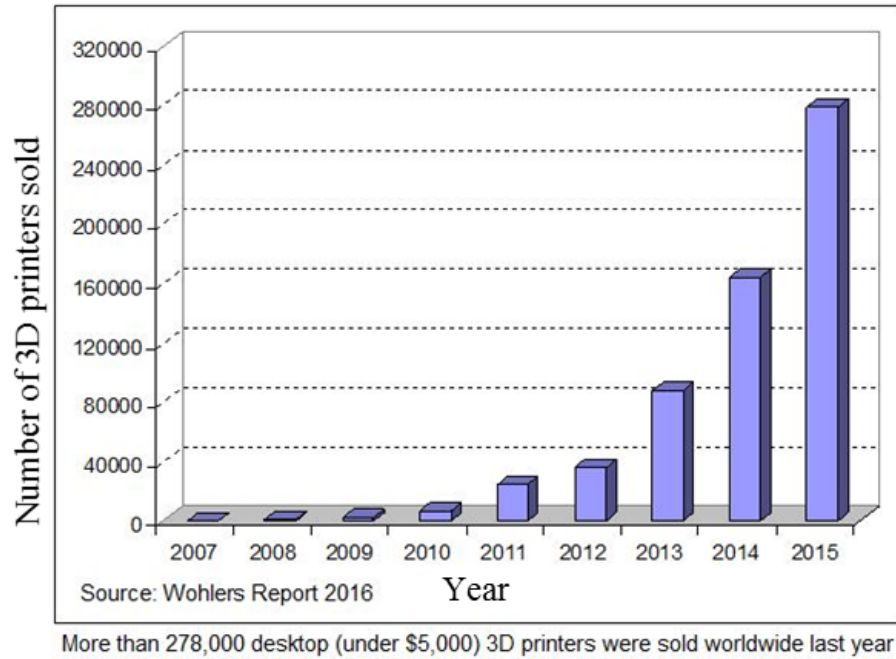
Figure 2.2. Volume of desktop 3D printers sold each year [1]

2.1.2. Carbon Fiber Filled Filaments

Many researchers have attempted to include various types of fillers within the polymer matrix to improve mechanical properties. These fillers include Titanium Dioxide [15], Jute Fibers [15], metal [16], glass fibers [17], thermotropic liquid crystalline polymer fibrils [18], vapor-grown carbon fiber [19] and graphene nanoplatelets [20]. Another viable candidate is short CFs [2,14,21]. CFs have superior tensile strength along the fiber direction as compared to thermoplastics that are typically used in FFF filament. It has a higher thermal conductivity and lower coefficient of thermal expansion (CTE) as compared to the thermoplastic polymers, therefore it has the potential to improve mechanical and thermal properties of FFF printed parts.

9
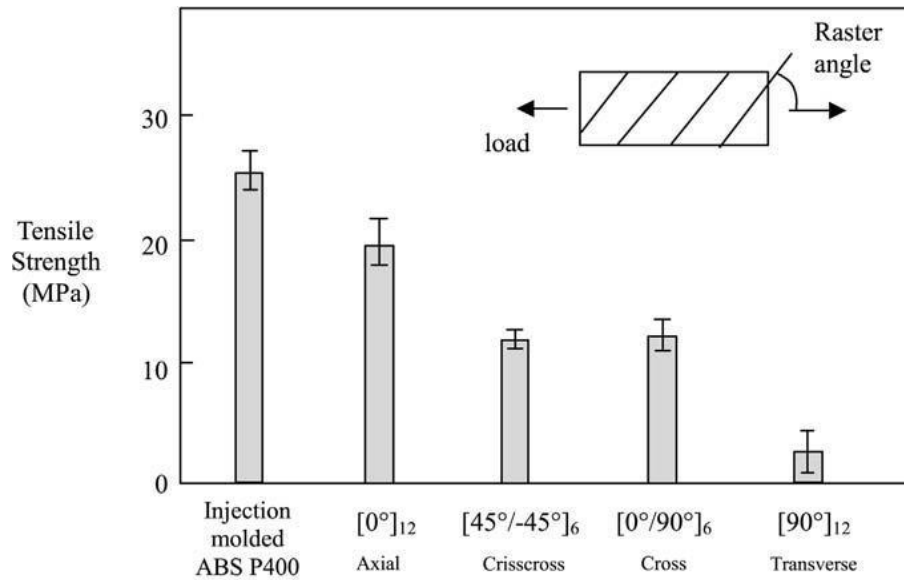
Figure 2.3. Anisotropic tensile properties using FFF process; all the FFF test sample have lower tensile strength than injection molded parts [11]

Studies have been conducted to blend short CF with Acrylonitrile Butadiene Styrene (ABS) for FFF applications. Love, et al. [22] showed CFF ABS polymer filament can significantly increases the strength and stiffness of FFF parts. In work presented by Love, et al. [15], CFF ABS dog bone samples had a tensile strength of 70.69MPa and a stiffness of 8.91GPa, comparing, respectively to 29.31MPa and 2.05GPa in the best scenarios for the pure ABS. They also demonstrated that the addition of CF decreased the distortion of the printed CFF ABS part which was due to an increase in thermal conductivity of 124%, as well as a decrease in CTE of 89% as compared to unfilled ABS, when the parameters are measured parallel to the deposition direction. Love, et al. [22] show two FFF printed bars. Figure 2.4 shows two FFF printed bars; the bottom bar is filled with short CF, and the top bar is made of pure ABS polymer. After the bars are cooled down, the deflection is very obvious for the unfilled ABS bar, whereas the CFF ABS bar has visually no deflection.

10

Figure 2.4. Deflection comparison for CFF and unfilled ABS FFF printed bars [22]

Ning, et al. [21] investigated the material properties of various CF weight fractions (from 0% to 15%) blended with ABS polymer matrix; an average fiber length of 150 um were supplied before the blending process with a common diameter of 7.2μm. They concluded that the effect of the CF increases the printed samples' tensile strength (by approximately +22%), Young's modulus (+31.6%) and bending strength (+11.8%), in the best cases. They also found that longer fiber length resulted in a higher tensile strength (about 7%) and modulus (about 20%), as expected. Furthermore, Tekinalp, et al. [2] conducted tensile tests with CFF ABS filament at various fiber weight fractions. The CFF ABS yielded an improved tensile strength and tensile modulus by as much as 115% and 700%, respectively. The reason why Tekinalp, et al. are able to get much better tensile improvement compared to the work done by Ning, et al. [23] is likely due to the longer fibers in the printed testing samples. Ning, et al. [23] used average fiber length of 150μm to prepare the testing samples, whereas there are many fibers that are longer than 400μm in the testing samples prepared by Tekinalp, et al. [2]. Figure 2.5 shows that as the fiber

content increased, the tensile strength and tensile modulus increased as well, even though the values are in general slightly lower than compression molded counterparts. They further discovered that the FFF process produced high fiber alignment along the print path. This implies the orientation of fibers in the printed part can be controlled within a part by specifying the deposition direction.
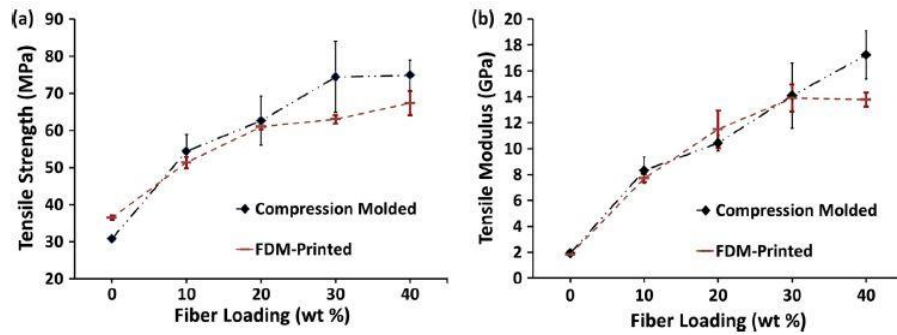
Figure 2.5. Tensile strength and tensile modulus at different fiber weight loading, for FFF processed and compression molded dog bone samples [2]

The aforementioned experimental results reveal the potential of CFF filament in FFF, and quite a few suppliers have commercialized this composite filament in the market. Several current vendors are 3DXTECH (Grand Rapids, West Michigan, USA), Triptech Plastics (Youngstown, Ohio, USA), Protoplant (Vancouver, Washington, USA), Colorfabb (Noorderpoort, Venlo, Netherlands) and Formfutura (Groenestraat, Nijmegenn, Netherlands). Unfortunately, little is known about the mechanical performance of parts produced with these commercially available CFF filaments. It is crucial to understand how well these filaments perform and what improvements are realized when CFs are added to the unfilled polymer. To the authors' knowledge, there are no publications investigating the mechanical properties of the commercially available CFF polymer filament. Therefore, Several CFF polymer filaments are purchased from one of the vendors and then the tested

for their mechanical properties in this research as described in Chapter Three. In addition to tensile tests, other important properties such as fiber length distribution, fiber weight fraction and fractographies will also be investigated.

2.1.3. *Large Scale 3D Printing*

Large scale 3D printing is a more recent innovation, and it aims to print parts having a large volume, using an FFF system. Perhaps the most prominent example is the Big Area Additive Manufacturing (BAAM) system [24] developed by Oak Ridge National Laboratory. The BAAM process is uses a mini-extruder mounted to the moving head of a precision gantry system. Its design concept is similar to that of a desktop 3D printer, except for a couple of key features. First, the size of the nozzle for BAAM is on the order of 0.2 inches, comparing to 0.4 to 0.6 mm diameter for a typical desktop 3D printer. Secondly, instead of using filament, CFF polymer pellets are supplied to feed the extruder system for printing, which eliminates the need for buying premade filament. BAAM extruders can deliver up to 100 lbs. of material per hour to the deposition platform, and the gantry system is capable of moving up to 200 inches per second with a position accuracy of 0.002 inches [25]. The system can drastically reduce manufacturing cost and energy input from many comparable manufacturing methods (see e.g. [25]). Figure 2.6 compares the energy input for different manufacturing processes, and the BAAM stands out as the second most efficient production technique. It has accomplished several remarkable achievements, including the 3D printed Shelby Cobra in 2015 [3]. The printed model is shown in Figure 2.7. Besides the BAAM system, other institutions have created similar large 3D printers and print objects with CFF polymer pellets [26]. The emergence of large scale 3D printing is gradually shifting the application of FFF from hobbyist and academic research use to

industrial application, and the desire for lighter and stronger products will continue to drive

this polymer composite manufacturing method to be more competitive in the market.
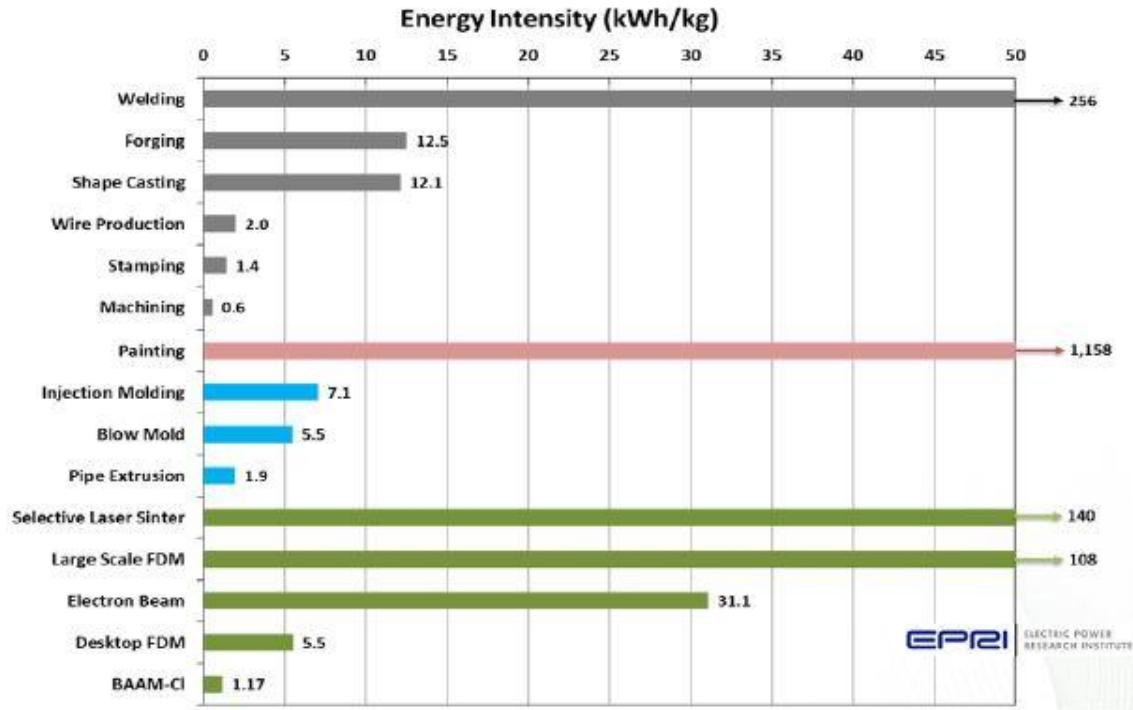
Figure 2.6. Manufacturing energy input required per kg for different manufacturing techniques [25]

For FFF processing, having a computational tool to design the optimal material

distribution of a printed part would greatly enhance the competitiveness of this AM

technique. Especially for industrial 3D printing, time of operation and raw material can be

further reduced, while not compromising the mechanical integrity of the final product.

Such a tool needs to take the print orientation from the FFF process into consideration, as

bead pattern direction greatly affects the mechanical properties of a printed part. The

degree of anisotropy is increased when CF is added to the base polymer material. One

viable method for taking advantage of the orthotropic nature of FFF parts is topology

optimization which can be used to generate the optimal topology and fiber orientation for the printed part.

Figure 2.7. 3D printed Shelby Cobra [3]

2.2.   Topology Optimization Methods

Topology optimization is a numerical approach that computes an optimal distribution of material given a design domain, an objective function, constraints and boundary conditions. It has been applied in the fields of aerospace [27], architecture [28] and automotive [29]. In structural optimization, topology optimization is commonly used to solve for the minimum compliance of a structure, given constraints on the amount of allowable material. There are numerous optimization schemes developed by the researchers, and a few of the popular methods are discussed in this section.

2.2.1. Homogenization Method

Bendsoe and Kikuchi [30] where among the first to study topology optimization and proposed the Homogenization Method for linear elastic structure design which is based

on adjusting the size of microscale voids in each cell. The optimization consists of the following steps: First, the design domain is determined and constrained by boundary conditions and design constraints. The design domain is then discretized into many finite elements, and each finite element in the discretized domain is then assigned homogenous and isotropic material properties. Secondly, the design space is discretized into unit cells, and each unit cell has a one or more voids. For each unit cell, the effective material property is calculated using the method of homogenization, a method that constitutes a numerical relationship between the solid region of the cell and the elastic properties of the cell. Thirdly, the parameters that define the shape and the orientation of the porous region in each cell are the design variables to be optimized by the selected algorithm. The objective of this method is to transform the original shape optimization problem (i.e., one where the design domain itself changes shape) into a sizing optimization problem (i.e., one where the design domain is fixed and element or material properties within the design domain are adjusted). As a result, the use of the finite element method within the optimization problem is greatly simplified as the need for remeshing is eliminated.

Suzuki and Kikuchi [31] applied the Homogenization method to several structural problems. To illustrate the method, the design domain, shown in Figure 2.8a, is discretized into multiple unit square cells. For each cell, one rectangular void is assigned with length variables a and b, shown in Figure 2.8b. Furthermore, the cell can rotate, so the stiffness tensor (material property) of each cell depends on the length variables, a and b, as well as the rotational angle θ. The numerical relationship is demonstrated in equation (2.1).

$$E_{ijkl}^{H} = E_{ijkl}^{H}(a, b, \theta) \tag{2.1}$$

16

This implies the number of design variables is three times the number of elements in the design domain, which is not desirable when the number of elements is large. The issue can be mitigated, in part, by using alternate types of voids, such as a circular shape defined by one parameter. This approach has been used to design cellular structures, which will be discussed in more detail below.
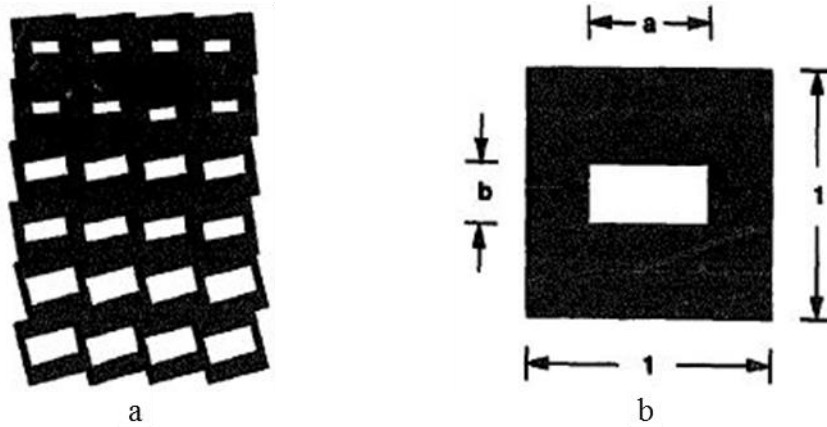
Figure 2.8. **a**: Design domain discretized with multiple cells; **b**: A single cell element with a rectangular void [31]

The Homogenization method has been further developed and applied in many papers. Unfortunately, the process of optimizing microscale voids can be cumbersome, especially in solving for cell orientation. In addition, the optimized void sizes can be so small that is not practical to fabricate, even with precise AM techniques. Therefore, the Homogenization method would not be considered for this research.

2.2.2. *Evolutionary Structural Optimization*

Inspired by the nature, Xie, et al. [32] proposed the Evolutionary Structural Optimization (ESO) method. The design process is simple. First, the design domain is discretized into elements. Second, finite element analysis (FEA) is performed to calculate

the stress distribution in the design domain where von Mises stress is often used to quantify the stress distribution. A stress threshold is calculated by multiplying a user-defined rejection ratio (RR) by the maximum stress in design domain, and any element with stress less than the threshold value is eliminated from the design domain. The FEA analysis and element elimination is repeated using the same RR until the steady state is reached. Afterward, an evolutionary ratio (ER), used to increase the value of RR, is then added to the RR and previous analysis and element elimination process is repeated again to reach steady state, until the final optimum criterion is reached. A typical convergence criteria is that stress values within all element within the design domain are within 25% of the maximum. Despite the simplicity of the method, it has been proven to be robust and has been applied to numerous structural problems. One issue with the ESO method is it only eliminates elements during the iteration. It is very possible that after the elimination of some elements, their neighboring elements experience high stresses and need the previously eliminated elements to return, which the ESO method prohibits.

An improvement on the ESO method is Bidirectional Evolutionary Structural Optimization (BESO) method, proposed by Querin, et al.[33]. The BESO algorithm searches in both directions, allowing not only removing the material with low stress but also adding extra material to the region with high stress. This method offers a more thorough search in the design domain and therefore is more likely to avoid local minimum solution. The ESO and BESO methods are simple to implement, but the stress based schemes do not take the directionality of the material property into account. Therefore, these approaches are not to be considered for the current AM orthotropic material optimization.

2.2.3. *Solid Isotropic Material Penalization*

Solid Isotropic Material Penalization (SIMP) method, also refers to as the "Power-Law Approach", is another popular structural topology optimization scheme. Compared to the Homogenization method, the SIMP method reduces the number of design variables in each element by assuming that the isotropic material properties are constant, each being multiplied by a single element density design variable raised to a predefined power [34–36]. The typical relationship used to define the material properties in the SIMP method is

$$E_{ijkl} = \rho^p E_{ijkl}^0 \tag{2.2}$$

Where ρ is the density variable from 0 to 1, and that E_{ijkl}^0 is the stiffness tensor for the particular element. The exponent p serves the role of penalization that tends to drive the density variable to either solid or void (1 or 0) during the optimization process. To demonstrate how the method works, Sigmund [37] presented the 99 line topology optimization code based on SIMP method. The problem aims to solve for the minimum compliance structure where the topology optimization problem is stated as

$$\min: c(\boldsymbol{\rho}) = \mathbf{U}(\boldsymbol{\rho})^T \mathbf{K}(\boldsymbol{\rho}) \mathbf{U}(\boldsymbol{\rho}) = \sum_{e=1}^{N} (\rho_e)^p \mathbf{u}_e^T \mathbf{k_0} \mathbf{u_e}$$

$$\text{Subject to: } \begin{cases} \dfrac{V(\boldsymbol{\rho})}{V_0} = f \\ \mathbf{K}(\boldsymbol{\rho})\mathbf{U}(\boldsymbol{\rho}) = \mathbf{F} \\ 0 < \rho_{min} \leq \boldsymbol{\rho} \leq 1 \end{cases} \tag{2.3}$$

The total compliance c is the objective function to be minimized, which is calculated by assembling all of the element level compliances together. **K**, **U** and **F** are the global stiffness matrix, displacement vector and applied load vector of the finite element model, respectively. The variables $\mathbf{k_0}$ and $\mathbf{u_e}$ respectively represent the stiffness matrix and displacement vector at the elemental level [37]. There are a total number of N elements,

and ρ_e is a design variable in one particular element. The variable p is the penalization power. Since the material property is modeled as isotropic, the element stiffness matrix becomes $\mathbf{k_e} = (\rho_e)^p \mathbf{k_0}$, where $\mathbf{k_0}$ is a constant matrix that is defined in terms of the mechanical properties (elastic modulus E and Poisson's ratio υ) of the underlying material in the usual manner [38]. The objective function is subjected to three constraints. First, it is constrained by a volume constraint. This volume constraint limits the amount of material to fill the space, and it is conveniently considered as the total design volume multiplied by a fraction value f, which ranges from 0 to 1. The second constraint, $\mathbf{KU} = \mathbf{F,}$ states that each optimization iteration has to obey the static equilibrium state. The last constraint limits the allowable value of each design variable.

There are numerous optimization algorithms that may be applied to solve Equation (2.3). These include algorithms such as Optimality Criteria (OC) [39], Sequential Linear Programming (SLP) [40], and Method of Moving Asymptotes (MMA) [41]. MMA is well suited to solve smooth and non-linear optimization problems. In each step of the iteration, MMA generates a convex sub-problem which is subsequently optimized. The generation of the sub-problems are strictly controlled by the "moving asymptotes", the design sensitivity and the iteration history. The asymptotes are updated every iteration to stabilize and speed up the convergence. In the case when the two asymptotes jump to negative and positive infinity respectively, the approximation function becomes identical to SLP. Therefore, the MMA and SLP essentially belong to the same family of algorithm. The OC method is another popular optimization algorithm, based on the condition of optimality. The algorithm takes the Lagrange Multipliers into the formulation and updates them iteratively to satisfy the constraints. One unique feature for the OC method is that it updates

each design variables independently, therefore making the method very efficient. In the

paper by Sigmund [37], the OC method is used where the design variables are updated as

$$\rho_e^{new} =$$

$$\begin{cases} if\ \rho_e B_e^\eta \leq \max(\rho_{min}, \rho_e - m): & \max(\rho_{min}, \rho_e - m) \\ if\ \max(\rho_{min}, \rho_e - m) < \rho_e B_e^\eta < \min(1, \rho_e + m): \rho_e B_e^\eta \\ if\ \min(1, \rho_e + m) \leq \rho_e B_e^\eta, & \min(1, \rho_e + m) \end{cases} \quad (2.4)$$

In the above, m is a positive move-limit, η is a numerical damping coefficient, and B_e is

found from Equation (2.5). A local optimum is found when $B_e = 1$, which satisfies the

Karush-Kuhn-Tucker optimality condition [7]. The update scheme adds material to where

$B_e > 1$ and removes material to where $B_e < 1$, as long as the density variable does not

violate the bounds. The derivative $\frac{\partial c}{\partial \rho_e}$ is the design sensitivity of the objective function at

the local elemental level, $\frac{\partial V}{\partial \rho_e}$ is the derivative of the volume with respect to each elemental

density variable, and λ is a Lagrange Multiplier which is adjusted in an inner iteration loop

in order to satisfy the volume constraint.

$$B_e = -\frac{\frac{\partial c}{\partial \rho_e}}{\lambda \frac{\partial V}{\partial \rho_e}} \quad (2.5)$$

Sigmund [37] demonstrated that the SIMP method could solve various structural

optimization problems including single load to multiple-load problems The program can

further solve geometry with prescribed elemental density values. The SIMP method has

been proven to be a robust iteration scheme, and it has been extended and modified to

handle problems with different physics and multiple materials. Its formulation can further

be modified to include anisotropic materials in the topology optimization, making it

applicable to AM processes like FFF. Hoglund, et al. [4,42] have demonstrated such work by integrating SIMP method with orthotropic material orientation to solve structural problem for FFF. Therefore, the SIMP method is chosen as the method in this research. The implementation of SIMP method is not without concern; a couple of major problems exist for the method involve checkerboarding of the resulting optimized material system and mesh-dependency. These two issues have been studied extensively and are discussed in the next section.

2.3. *Checkerboard, Mesh-dependency and Filtering*

Checkerboarding [43,44] in topology optimization refers to pattern that evolves in the design domain that appears in alternating solid-void manner such as that shown in Figure 2.9. Checkerboarding occurs in both the Homogenization and the SIMP methods. Diaz, et al. [43] suggested the reason of occurrence is because numerical analysis approximates that checkerboard pattern to yield higher stiffness than any other layout arrangement. However, such discrete pattern is impractical to be fabricated by any manufacturing method, including AM. Checkboard patterns can be mitigated by using higher order finite elements in the optimization [45,46], but this would add significant computational time if the number if elements stay the same.
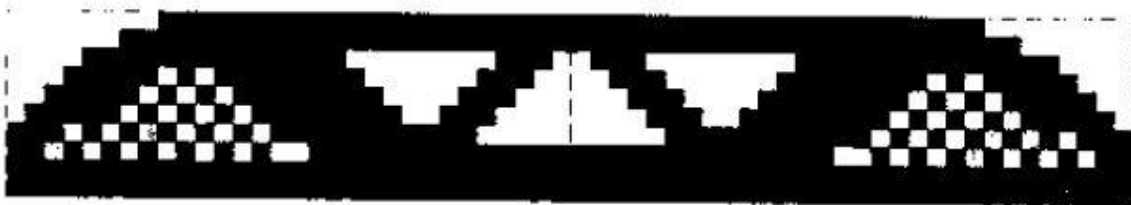
Figure 2.9. Optimized topology with checkerboard issue [44]

Mesh-dependency appears when the different qualitative solutions are obtained when larger or smaller elements are used in the finite element model. The issue is demonstrated in Figure 2.10, and it is due to the non-existence or the non-uniqueness of the solution of the given optimization problem. Perimeter control [47,48] and local gradient constraint [49] are the two possible schemes to obtain a mesh independent solution. Perimeter control introduces an upper bound for the perimeter of structure, therefore reducing black-white formation. However, different design problems can have different perimeter bounds, and it takes many tries to determine the proper value. Local gradient constraint prevents drastic density value jumps in gradient between local neighboring elements, but the scheme introduces two times the number of elements and more linear constraints into the problem for two dimensional problem, making it extremely computational intensive with fine mesh.
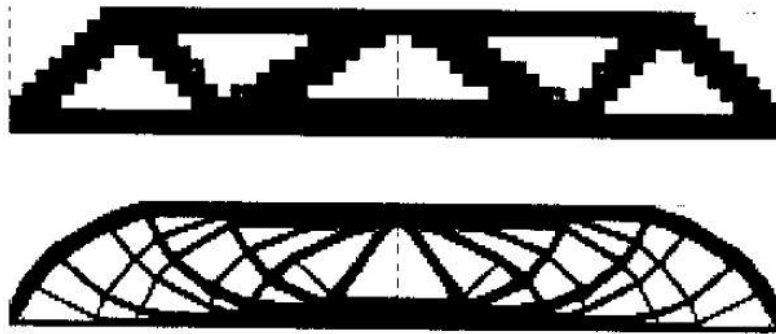
Figure 2.10. Optimized topology changes as the mesh becomes finer, from top (600 elements) to bottom (5400 elements) [44]

Sigmund [50] demonstrates that a filter technique that is based on image processing can mitigate both checkerboard and mesh-dependence. Discretized elements, like image pixels, can be filtered based on the surrounding elemental density values, and this modification leads to smoother transition among the elements. One big advantage of the

23

filter technique is that it does not impose any extra constraint to the problem, therefore the process is computational efficient. In addition, the implementation is relatively straight-forward, avoiding the need of complicated algorithms.

There are two common kinds of filters commonly used in topology optimization. The first type is the density filter, introduced by Bruns, et al. [51]. The elemental density value is filtered by taking the weighted average of itself and its neighboring density values, before performing a finite element analysis on the structure. The sensitivities are then calculated using the filtered densities. The numerical relationship between the surrounding elements and the filtered element can have various interpretations. One popular option is to use the linear decaying weighting function. This means the closer the element is to the filtered element, the more emphasis of its value will be imposed on that filtered element. Other weighting function such as Gaussian distribution [51] and constant weighting value [52] have been used. The other commonly used filter is the sensitivity filter [50,53]. In this approach, instead of modifying the density values, the design sensitivities are modified through weight average calculation similar to the density filter technique. Sigmund [52] points out that the sensitivity filter is a risky approach, as the modified sensitivities may not represent the actual descent direction of the function. However, numerous practical applications have shown it is very robust and efficient. Alternatively, design variables can be implicitly presented as solution of a Helmholtz type partial different equation (PDE) [54–56] with homogenous Neumann boundary conditions.

One common problem with both the density and sensitivity filters is that the optimized topology shows grey transition area along the edge of the members. To achieve a strict black-white solution, Pratt [57] introduced morphology operators called Erode and

Dilate operators. Depending on which one is used, the filtered element takes the minimum (Erode) or the maximum value (Dilate) of the neighboring elements. These operators can also be converted to continuous functions to suit for the gradient based optimization [58]. Furthermore, Guest, et al. [59] proposed a continuous Heaviside function to eliminate the grey transition boundary. In this case, any element with a density value larger than zero (100% void) turns to one (100% solid), and the result of the optimal solution is strictly a black-white solution. One caution to note is that the morphological operators and Heaviside functions described here are not volume preserving, which means the total volume of the design variables multiplied by their respective finite element volume before and after the filtering are not equal. Therefore, the volume constraint needs to be imposed on the filtered design variables. To establish a volume preserving technique for the morphological operators, the Erode and Dilate techniques can be combined [52], but this operation can add significant computational time on the filtering process.

Considering all the available filtering options available, we selected the sensitivity filter [44] for this research. The sensitivity filter requires a minimum amount of modification on the optimization code, and it has been used extensively by other researchers [4,42,60]. To apply the sensitivity filter to SIMP method, the modified sensitivity for each element compliance sensitivity is

$$\frac{\partial \hat{c}}{\partial \rho_e} = \frac{\sum_{i \in N_e} \omega(x_i) \rho_i \frac{\partial c}{\partial \rho_i}}{\rho_e \sum_{i \in N_e} \omega(x_i)} \tag{2.6}$$

where $\omega(x_i)$ is a linear weight function given as

$$\omega(x_i) = \max(r_{min} - \text{dist}(e, i), 0) \tag{2.7}$$

In the above, r_{min} is a distance that is used to identify the number of elements surrounding ρ_e that are to be considered for filtering. Outside the radius of r_{min}, the weight value equals to zero. Also, $dist(e, i)$ is the center distance of element ρ_e to ρ_i. From the linear weight function, the sensitivity value of $\frac{\partial c}{\partial \rho_e}$ itself has the highest influence on the filtered sensitivity $\frac{\partial \hat{c}}{\partial \rho_e}$, and the effect decreases linearly as $dist(e, i)$ increases. After selecting the appropriate material interpretation scheme, we will discuss several historical studies that involve topology optimization for additive manufacturing, where various material models are used.

2.4.   *Orthotropic Material Property with Additive Manufacturing and Topology Optimization*

Topology optimization has seen application in several additive manufacturing process that are used to realize the potential of AM. The homogenization method is a popular method among the researchers who are interested in variable cellular structures. For example, Sundararajan [61] applied the homogenization method with a smoothing scheme to optimize the MBB and a cantilever beam. The optimized shape is assembled using mesostructures through elements with square void such as that shown in Figure 2.11. The elastic constants of the element are correlated with the size of the void in the element. Sundararajan fabricated the optimized cantilever beam with selective laser sintering. The experimentally tested three point bend tests are compared with the numerical model displacement values, resulting in overall deviation less than 10%. A fabricated MBB beam under three point bend test from [52] is shown in Figure 2.12. The fabricated cantilever beam is tested under a three-point bending test, and the test is conducted in the elastic range of the fabricated structure.
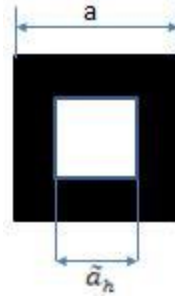
Figure 2.11. A mesostructure element [61]
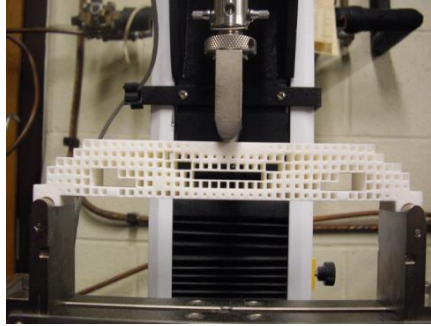
Figure 2.12. Selective laser sintering fabricated MBB tested with three point bend test [61]

Zhang, et al. [62] also employed the homogenization method to relate the constitutive material matrix to the microstructure characterizing parameter (MCP), using a second order polynomial function. The MCP in this case is the density variable in each element. After the homogenization, optimization is performed on the design domain. An example taken from [62] is shown in Figure 2.13a for a rectangular grid. The grid is discretized into square elements, and each element contains a design variable ρ_e. In addition, the design domain is covered by an array of lattice points, represented by the red dots. Each lattice point is located at the center of a hexagonal cell. The size of the circular void in each cell depends on the density value of the element. An example of the cellular structure taken from [62] constructed by the hexagonal cells is shown in Figure 2.13b.

27

Zhang, et al. [62] applied the SIMP method to obtain the optimum distribution of the density values over the design domain. The cellular structure is then constructed by assigning the lattice points the same values as their overlapping elemental density values. The model is then fabricated by stereolithography in Figure 2.14. The numerical analysis and experimental test further reveals that the optimized cellular structure yields a 37% higher flexural stiffness than non-optimized (uniform) cellular structure, for least 37%.

a   b
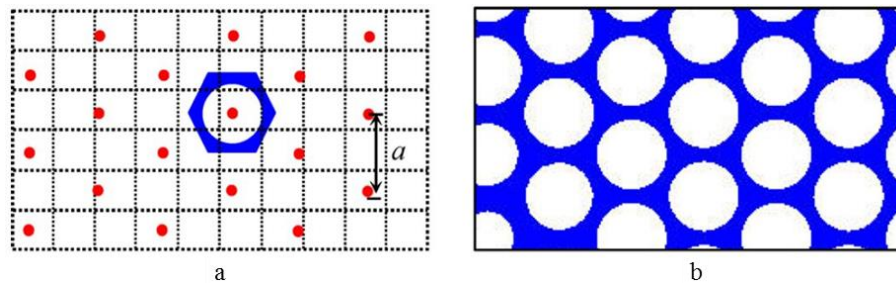
Figure 2.13: **a**: Discretized design domain covered by lattice points; b: An array of hexagonal cells with uniformly distributed circular voids [62]
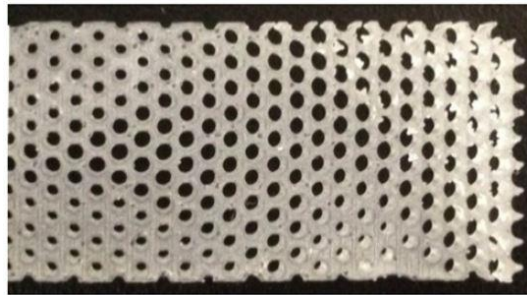
Figure 2.14. Variable cellular structure fabricated with SLA [62]

Burblies, et al. [63] performed a topology optimization on a bone structure based on the homogenization method. The problem first solves for the optimum distribution of the density with a volume constraint. The optimization process of the bone is shown in Figure 2.15, and the bottom picture shows the optimum density distribution. The density

values are then translated to corresponding pore sizes, and the multi-phase bone structure is shown in Figure 2.16.
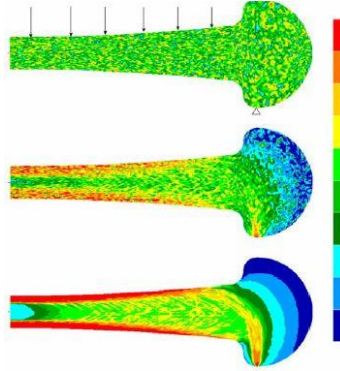
Figure 2.15. Optimization of density distribution in the bone. Blue color indicates low density value and red color indicates high density value [63]
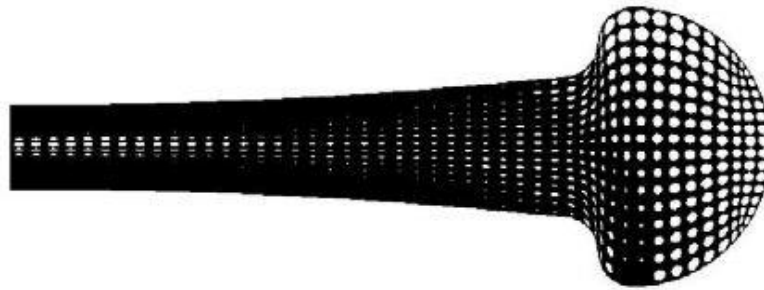
Figure 2.16. Final bone structure after mapping of the optimized density values [63]

Besides the application of the homogenization method, the SIMP method has also been applied to design cellular structures. Brackett, et al. [64] used the un-penalized SIMP method to solve a simply loaded cantilever beam , shown in Figure 2.17a. Again, the density values are mapped to a lattice structure, such as that shown in Figure 2.17c. Similarly, Rezaie, et al. [65] optimized a MBB beam using the SIMP method, and the model was reconstructed with cells filled with square holes. The part is fabricated with a desktop 3D printer. Rezaie's fabricated sample is shown in Figure 2.18. Furthermore,

Gaynor, et al. [66] designed a compliant mechanism with original, combinatory and multiphase SIMP approaches, and the designed force inverters were fabricated with multi-material Polyjet printing process. Two fabricated force inverters appear in Figure 2.19.
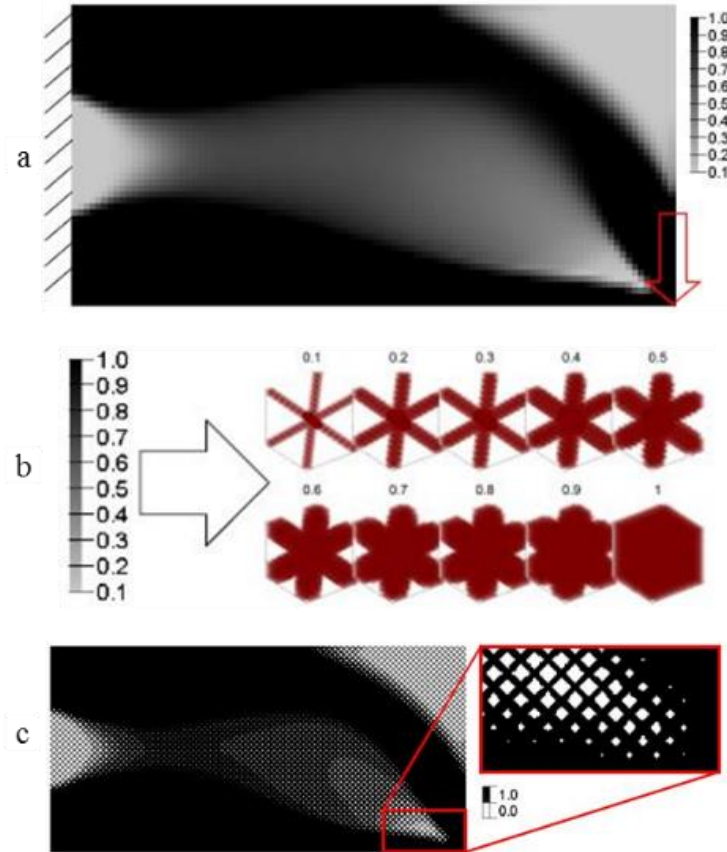
Figure 2.17. **a**: Un-penalized optimum density distribution for cantilever plate with SIMP method; **b**: relative density values to lattice structure; **c**: reconstruction of the lattice structure with corresponding density values [64]
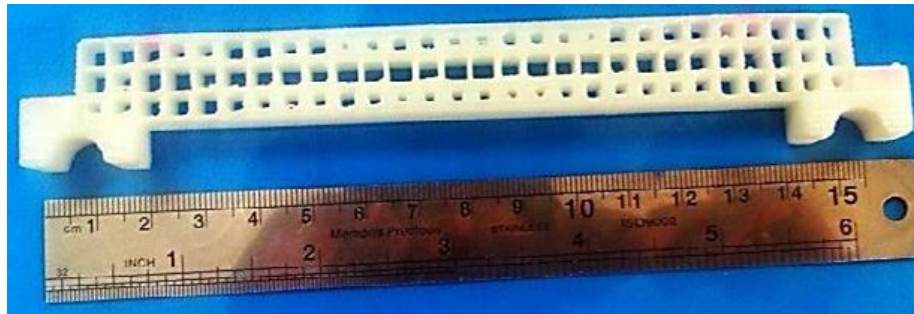
Figure 2.18. Optimized MBB beam fabricated with FFF machine [65]
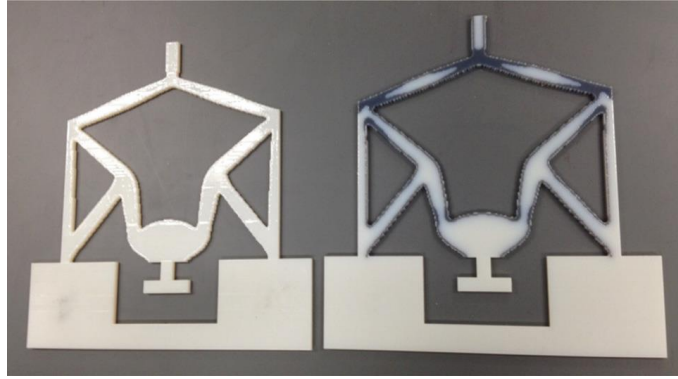
Figure 2.19. Two force invertors fabricated with Polyjet print; left part contains one material; right part contains two materials [66]

All the constitutive material models described in the works give above are isotropic. However, in FFF, many experimental tests have shown that the mechanical properties fabricated by FFF greatly depends on the print orientation [11–14]. Therefore, the directionality of the material property needs to be taken into account for the optimized topology to be valid. To accommodate anisotropic properties the SIMP material model may be modified to include orthotropic material model, which has been named the Solid Orthotropic Material Penalization (SOMP) method.

Lee, et al. [67] and Hoglund and Smith [42] studied compliance minimization by modifying the optimization code provided by Sigmund [37] to accommodate the orthotropic material matrix. They both concluded that by changing the ratios of stiffness along the orthogonal directions, the resultant optimal topologies are significantly different. An example taken from [42] of optimized MBB beams with different stiffness ratios in the x- and y- directions is shown in Figure 2.20. The differences of the optimized topology can be clearly observed. The differences of the optimized topology can be clearly distinguished in their papers. When the vertical stiffness is higher, the truss-like structure aligns more in the vertical direction, and vice versa when the horizontal stiffness is higher.

$E_x = E_y$

$10E_x = E_y$

$E_x = 10E_y$
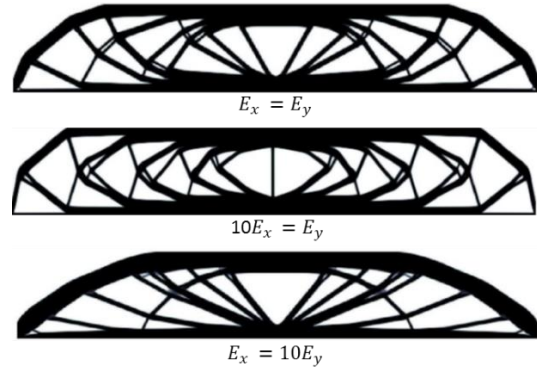
Figure 2.20. Optimized. MBB beams with SOMP method with different stiffness ratios [42]

Alamo and Silva [60] also applied the SOMP method in the field of biomechanics where the object was to understand the process of femoral bone tissue generation when a metallic implant is adopted. Organic structure such as human bone is sensitive to external environment, and its structure will change to best accommodate the outside stimulus. This process of adaption lends itself well to structural optimization. Also, the SOMP method is used here to include a more realistic constitutive model for the natural anisotropic material in the bone. Figure 2.21a taken from [60] shows the design domain and the boundary conditions, and Figure 2.21b shows the optimized density distribution.

a   b
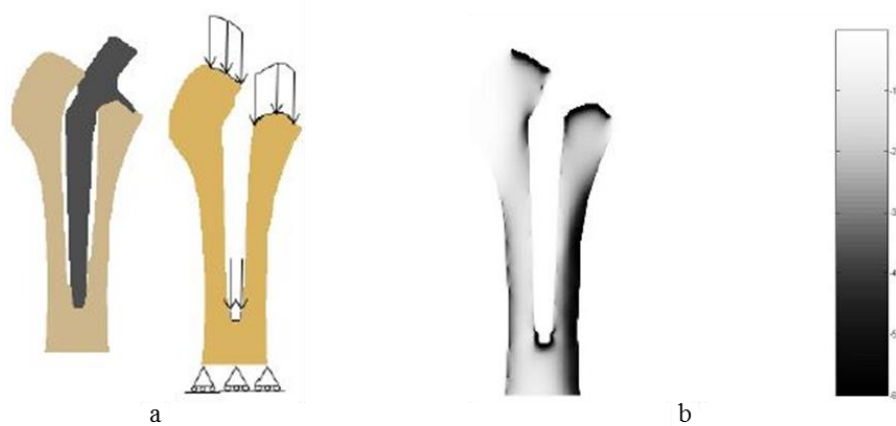
Figure 2.21. a: Design domain with along forces; **b**: optimized density distribution, with darker area indicating more material [60]

The SOMP method provides a means for incorporating material anisotropy into the topology optimization, however, the SOMP method is limited to design with predefined material orientation. In FFF, the nozzle has full range of freedom in the print plane. Therefore, the orientation of a deposited bead, and thus the primary direction of the carbon fibers within a bead, should also to be variable function of the design variables in the optimization. This requires that the optimization algorithm is capable of optimizing both the material distribution and material (or print) orientation. In the next section, material distribution and material orientation optimization will be discussed.

2.5.   *Continuous Fiber Angle Optimization*

Fiber orientation optimization becomes important as the application of fiber reinforced composites become popular, and the optimal layout design with a high specific strength material such as carbon fiber lamina is especially popular in the aerospace industry [68–73]. Some analytical approaches have been developed to address the orientation optimization problem. Gibiansky, et al. [74] and Suzuki, et al. [31] notes that the optimal orientation aligns with the major principal stress direction, in case for the "shear" weak type material; this is known as the stress based method. Pederson [75–77] proposed a strain based method that the optimum orientation can be calculated based on the principal strains. The stress and strain methods eliminate the need for iteration during the design process. However, Gea, et al. [78] identify that if the "repeated global minimum" (more than one set of design variables has the same minimum value) occurs, both methods would fail. Therefore, a new method is needed to improve the aforementioned limitations,.

One design method to account for full range of fiber orientation design is the Continuous Fiber Angle Optimization (CFAO). The CFAO has been used in designing the

optimum thickness and fiber orientation of orthotropic material [79,80], the maximum

buckling load of composite structure [81], as well as the optimum shape and material layout

of shell structure [82]. One concern for the CFAO is that because the angle variable has 2π

period, the problem is nonconvex. The non-convex complexity can potentially increase the

solving time significantly and care is needed to avoid over-constraining the design space.

The Free Material Optimization (FMO) method [83–87] is another effective

approach to optimize structure. The method relaxes the design space by treating the

stiffness tensor components at each point in the discretized domain as design variables, so

the material properties are varied point-by-point. Though the method provides more degree

of freedom in the design space, the optimized material properties may not represent the

actual material properties processed by FFF machine. Therefore, this method is not

practical for FFF process design.

In contrast to continuum variable design, Discrete Material Optimization (DMO) is

an alternative approach for determining material distributions. In DMO, the design

candidates are defined in advance; the candidates can be predefined sets of angle

orientations of the composite lamina or predefined sets of thicknesses of the material. The

optimization algorithm then selects the best option to optimize the objective function. The

discrete variables can be material thickness, material type and fiber orientation. Stegmann

and Lund [88] used DMO to optimize fiber angles in a discretized cantilever beam, shown

in Figure 2.22. Here, each discretized element composes a constitutive matrix that is the

sum of several matrices, each configured with different fiber orientation and multiplied by

a weight value. In the optimization process, the algorithm pushes every weight value to

zero except one in each element, and the fiber angle with the non-zero weight value is the

34

best option for the element. In DMO, the number of design variables equals to the number of discretized elements times the number of predefined candidates in each element, and some new parametrization schemes [89,90] are introduced to reduce the total number of design variables for large-scale design problem. Since the DMO method optimizes the problem with predefined sets of design candidates, it is not suitable for FFF processing since the fiber orientation can be adjusted continuously. Nonetheless, it is worthy to mention such method, as it is a well-established technique for optimizing composite laminate layout.
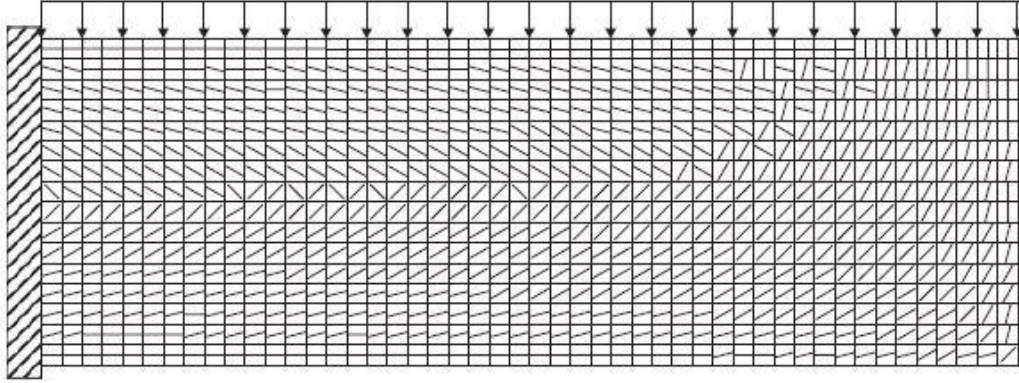
Figure 2.22. Optimal fiber distribution in cantilever beam with uniformly distributed top load; fiber angle are selected from [90,±75,±60,±45,±30,±15, 0] degrees. [88]

Nomura, et al. [5] proposed a general topology optimization method that is capable of simultaneously optimizing both fiber orientation and material distribution. The method can further cope with discrete angle sets. To avoid local minimum due to the periodic nature of fiber orientation, the orientation variables are represented by Cartesian components, along with relaxation of the orientation design space. Both the density values and Cartesian coordinates are regularized through Helmholtz PDE filter along with Heaviside functions. Furthermore, isoparametric functions are used to transform the regularized Cartesian

coordinates from a box bounds to circular bounds. Figure 2.23 taken from [5] shows the optimization results with isotropic and anisotropic material properties. To establish discrete angle sets for optimization, penalization is enforced on the even elemental numbers of the parametrized element, and this leads the optimization to prefer the unpenalized angle sets. One characteristic of using penalization for discrete angle sets design is that the number of options is changed by using different elements, while the design variables in each element stays the same.

Even though CFAO has its potential issue of local minimum, it is a very convenient tool for simultaneously optimizing topology and fiber orientation. Furthermore, the problem formulation can be very easy by integrating the orientation optimization with SIMP method. The continuum-based topology optimization solving both material distribution and fiber orientation has not attracted many researchers. Jia, et al. [91] implemented SIMP method with CFAO to design fiber reinforced composite for minimum compliance. The design sensitivities with respect to both density and fiber angle are calculated, and the optimization algorithm simultaneously solves for the optimal topology and fiber orientation. An example of solved cantilever geometry is shown in Figure 2.24. Due to the lack of filtering, some minor checkerboard pattern can be observed in the solution field. For the same type of problem, Setoodeh et al. [92] utilizes stress based approach with SIMP method to solve for minimum compliance, combined with cellular automata. They further extended the method to cope problem with multiple loads, though the fiber angle update scheme is based on random search style. The optimized cantilever beam and structure with multiple loads are illustrated in Figure 2.25.
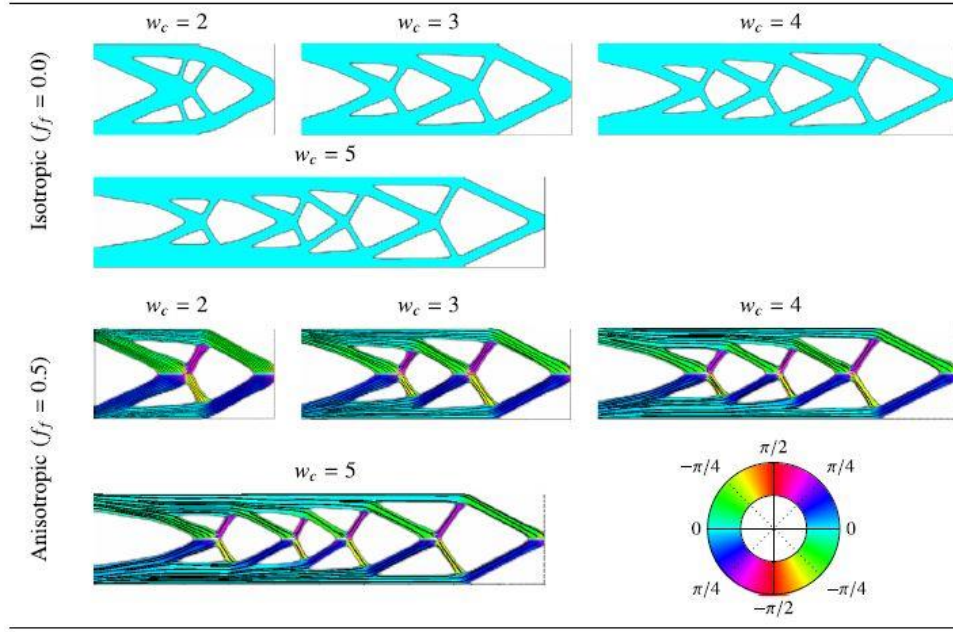
Figure 2.23. Optimization for short cantilever beam. Top row: optimization with isotropic material. Bottom row: optimization with anisotropic material (50% fiber volume fraction) [5]

Recently, Hoglund and Smith [4,93] proposed the CFAO method with SIMP scheme similar to [91] to optimize topology and fiber orientation, specifically for CFF FFF process. A sensitivity filter is implemented with respect to the density variables. The problem formulation is implemented into Matlab optimization solver *fmincon*. An optimized MBB beam is shown in Figure 2.26 where the optimal fibers align with axis of the truss members, as also can be perceived in the aforementioned research results. The optimized model can also be fabricated with desktop 3D printer by converting the model to a STL file using a Matlab code [94]. To show the effectiveness of CFAO with SIMP method, three point bending test was conducted with the fabricated MBB beams optimized from CFAO, and the results are compared to the printed MBB beams with fixed angle optimization [42]. They found that the CFAO beam is 29.9% stiffer than the vertical

optimized and printed beam; it is also 12.4% stiffer than the horizontally optimized and printed beam.
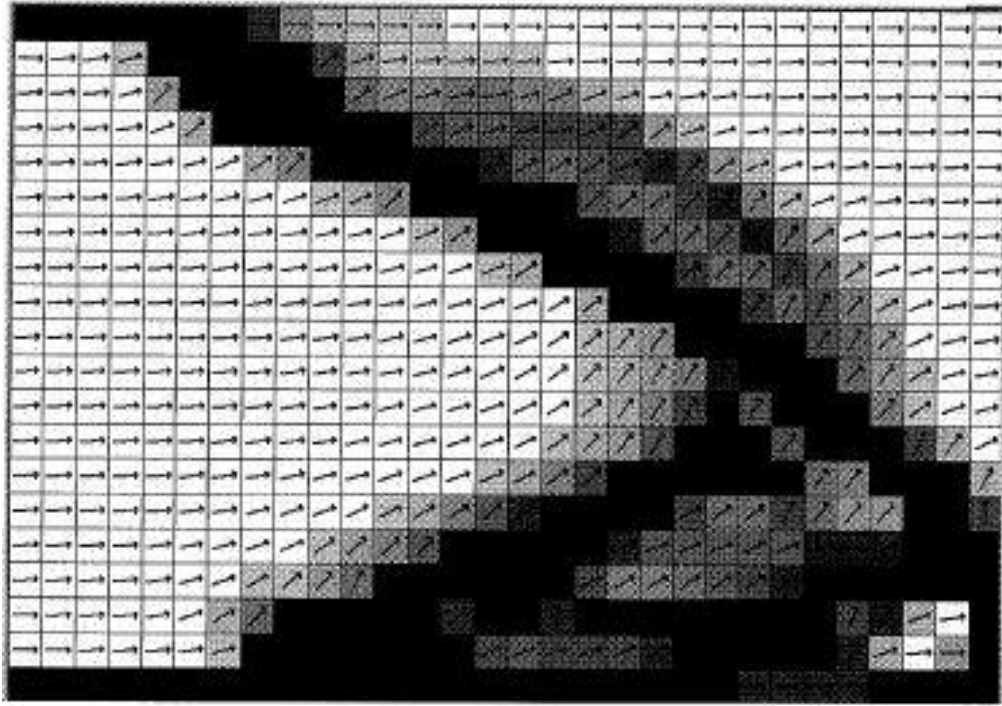
Figure 2.24. Optimized geometry and fiber orientation for a cantilever beam problem. Darker color represents higher distribution of material [91].

It has been shown that SIMP method combined with CFAO is a plausible tool to design for FFF processed parts, as the two dimensional optimization research demonstrated by Hoglund and Smith [4,93]. There are a few studies that can be further explored using this method. One of them is to design a full three dimensional model for optimal material distribution and material orientation, to simulate the layer-by-layer FFF process, where each the material orientation is constrained in its respective layer plane. To author's knowledge, this topic has yet to be addressed in the literature.
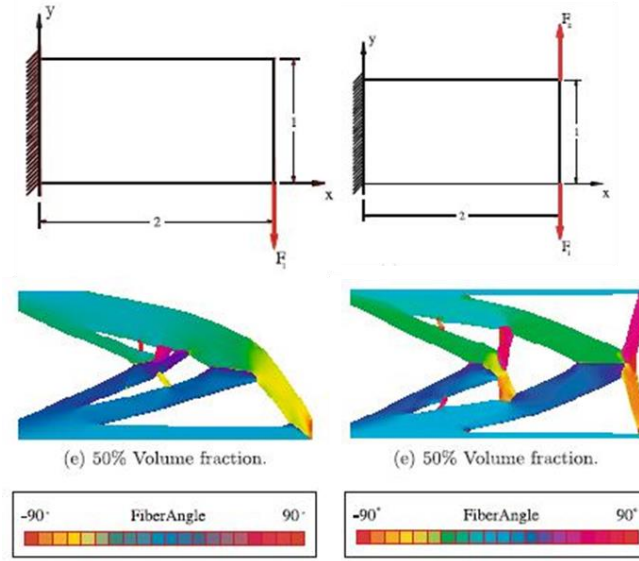
Figure 2.25. Optimization for topology and fiber orientation for single and multiple loaded structures. Design with cellular automata [92].
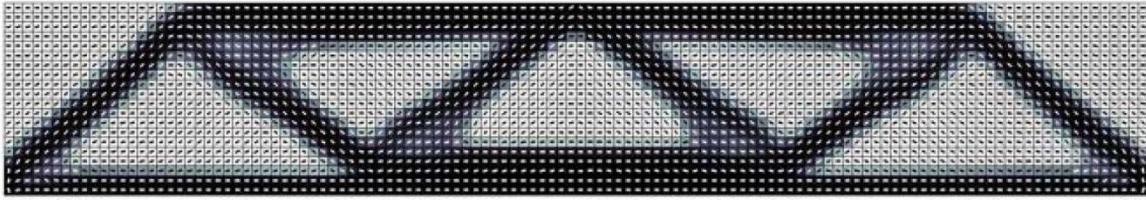
Figure 2.26. CFAO optimization with SIMP in Matlab for a MBB beam. 180 by 30 discretized elements [93].

2.6.   Three Dimensional Topology Optimization

Three dimensional topology optimization has been presented by Olhoff, et al. [95], Jacobsen, et al. [96] and Fernandes, et al. [97] who used three dimensional microstructures to design optimal structures. Diaz, et al. [98] presented a full relaxation scheme to solve for minimum compliance using eight node isoperimetric elements. Beckers [99] introduced a dual method to give a discrete (0 or 1) solution layout in three dimensions. Lastly, Langelaar recently [100] introduced a scheme to design for three dimensional self-

supporting structure for AM. The scheme can eliminate the need to add support structure during the fabrication process.

However, topology optimization studies in three dimensions using the SIMP method in AM are limited. The software OptiStruct [6] from Alair is a commercial code that is available for solving complex 3D structural optimization problems, but little has been published where Optistruct is applied to AM structures. Liu, et al. [101] proposed a three dimensional Matlab code for topology optimization based on SIMP method. The study is an extension of the 99 line Matlab optimization code provided by Sigmund [37]. The interpretation scheme is slightly changed according to the modification suggested by Andreassen, et al. [102], and the density filter [51] is adopted. Their Matlab code utilizes the OC algorithm [39] and can solve structure for minimum compliance, compliant mechanism and heat conduction. An optimized three dimensional cantilever bean is shown in Figure 2.27. The proposed algorithm can solve medium-size problem relatively quickly which is made possible since the stiffness matrix is the same for all the discretized elements. The Matlab based code for three dimensional SIMP topology optimization provides a starting point to design for FFF process in our work. In addition to the code written in Matlab, Zuo, et al. [103] provided a python code for three dimensional topology optimization, using BESO method [33] for compliance minimization.
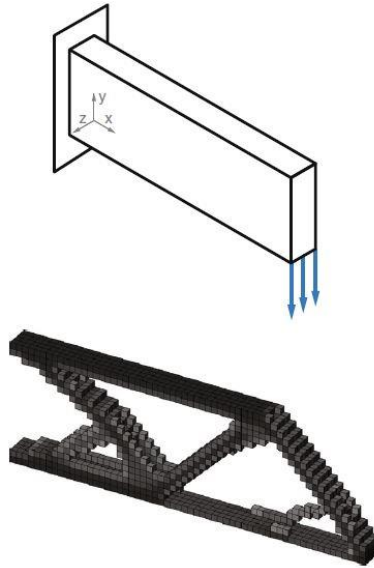
Figure 2.27. Optimized 3D cantilever beam [101]

CHAPTER THREE

Mechanical Behavior of carbon Fiber Composites in Fused Filament Fabrication

The chapter investigates CFF FFF parts produced on common desktop 3D printers, in this case Makerbot Replicator 2 and Replicator 2X, using commercially available filament. Tensile test samples fabricated with CFF polymer composite and unfilled polymer were printed and then tested following ASTM D3039M. Test bars were printed with FFF bead orientations aligned with the direction of the applied load at 0 degree, and also at 45 degrees, ±45 degrees, and normal to the loading axis at 90 degrees. The filament considered here was purchased from filament suppliers and included both CFF and unfilled PLA, ABS, PETG and Amphora. Additionally, CFF samples are evaluated for fiber length distribution (FLD) and fiber weight fraction, fracture surfaces are evaluated under SEM.

3.1.   Experimental Setup

3.1.1. Filament Processing

This study considers four different filament materials: Polyactic Acid (PLA), Acrylonitrile Butadiene Styrene (ABS), Polyethylene Terephthalate (PETG) and Amphora, a proprietary co-polymer produced by Eastman. PLA, ABS, and PETG filament were purchased from 3DXTECH (Grand Rapids, Michigan, USA), and Amphora was acquired from Triptech Plastics (Youngstown, Ohio, USA). Filament for each material was purchased as the unfilled polymer and the same with carbon fiber filler. In summary, testing was performed with commercially available filament made of PLA, CFF PLA, ABS, CFF

ABS, PETG, CFF PETG, Amphora, and CFF Amphora. Prior to use, each filament was stored in a sealed bag containing silica gel desiccant to avoid moisture absorption. A Makerbot Replicator 2X was used to print the ABS, PETG and Amphora filaments, and a Makerbot Replicator 2 was employed to print the PLA filaments. Bead directions of 0 degree, 45 degrees, ±45 degrees and 90 degrees with respect to the axis of the test bar were investigated, as shown in Figure *3.1*.
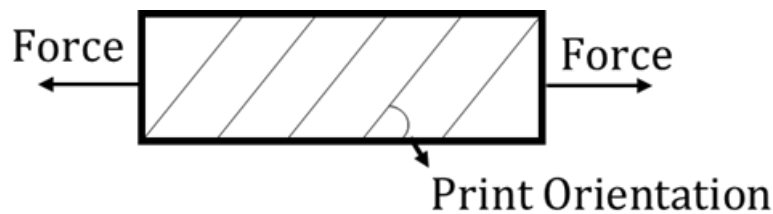
Figure 3.1. Definition of print orientation angle in the test coupon with respect to the direction of the tensile load

A nozzle diameter of 0.6mm made with hardened steel was used for printing all test samples. This choice of nozzle reduces clogging and wearing of the nozzle when carbon fiber filament is used. Printing process parameters appear in Table 3.1**.**

Table 3.1. Printer information, processing temperatures, layer infill information, printing surface, nozzle size and printing speed for each material filament.

Experimental Parameters	PLA CFF PLA	ABS CFF ABS	PETG CFF PETG	Amphora CFF Amphora
Printer	Makerbot Replicator 2	Makerbot Replicator 2X		
Extruder temperature	220˚ C	230˚ C	255˚C	250˚C
Heat bed temperature	Not required	110˚ C	80˚ C	70˚ C
layer infill	0.2mm height \ 100% infill			
Heat bed surface	Blue painter's tape	Kapton tape		
Nozzle	0.6mm hardened steel			
infill print speed (mm/s)	45 for base layer and outline 105 for the rest of the infills			45 for each layer

3.1.2. Testing and Measurement

The ASTM D638 procedure [104] (similar to ISO 527 [105]) was initially followed using a Type I dog bone test bar geometry. However, tensile tests at on samples with certain print orientations resulted in undesirable test data and premature failures at neck transition section of the dog-bone test sample, similar to issues described by Forster [106]. To resolve this issue, the ASTM 3039M procedure [107] designed specifically for polymer composites was adopted. The dimensions of our test coupons were 177.8mm×12.7mm×2.54mm as defined by ASTM 3039M. Machined 6061 Aluminum tapered tabs were bonded at the ends of the test samples with Loctite Super Glue Ultra Gel [108]. Five test samples were prepared for each filament and print orientation combination. The Test Resources, Inc (Shakopee, Minnesota, USA) 100 Series Single Column Test Machine [109] was employed for the tensile test. Samples were fitted with a 25mm extensometer [110] prior to testing for strain measurement. Tensile test parameters used in our study appear in Table 3.2.

Table 3.2. Testing parameters

Crosshead speed	2mm/min
Load cell capacity	4 kN
Sampling rate	25 Hz
Extensometer travel capacity	6.25mm

The average axial stress $\sigma = F/A$ in the test samples was calculated for each tensile test in the usual manner where F is the force measured by the load cell and A is the sample cross sectional area. The test bar area A used in the calculation was obtained by averaging values measured at three places (one towards each end and a third in the middle) in the gage section along each sample. The width of the rectangular test bar was measured with a digital caliper and the thickness was measured with a micrometer. The variation of the

44

three measured values at each dimension in all of the test coupons is very small; mostly at maximum of 0.001inches. No effort was made to account for surface roughness or voids that may have resulted during printing. Strain ε was measured directly from the extensometer. The modulus of elasticity $E = \Delta\sigma/\Delta\varepsilon$ was computed as the slope of a straight segment obtained by linear regression using data from $\varepsilon = 0.001$ to $\varepsilon = 0.003$, as recommended in the ASTM standard.

Following the tensile testing, fracture surface images were obtained using a JEOL JSM - 6610 LV Scanning Electron Microscope [111]. A TA-Q50 Series Thermogravimetric Analyzer [112] was employed to burn off the polymer resin for all CFF samples to measure the carbon fiber content and length distribution in each test sample. For the CFF materials, three specimens were extracted from CFF filament and 0 degree printed CFF test coupons. The customized burn-off procedure shown in Table 3.3 was developed to obtain carbon fiber weight percentage. For each sample, the specimen is heated at 10˚C per minute to reach 500˚C; then the temperature is set at that constant for an hour; After one hour, the test is finished and the data recording stops. During the burn-off test, nitrogen gas was purged into the chamber to prevent the fibers from oxidizing. We would expect the thermoplastic filament should burn off, while the fibers will not since the fibers have much higher breakdown temperature.

Table 3.3. TGA burn off procedure

step 1	3 samples / material
step 2	10˚C/min to 500˚C
step 3	Isothermal for 1 hour
step 4	test finished and cool down

The residual material from each CFF burn-off process was collected in sealed plastic tubes. Small particles with irregular shapes were identified under SEM inspection, indicating that all non-carbon fiber material did not completely burn off. To get the actual fiber weight fraction, the related unfilled polymer filament sample from each material was subjected to the same burn-off procedure, and its residual weight fraction was used as a reference to approximate the actual polymer weight fraction and fiber weight fraction in the CFF samples. To measure fiber length distribution, residuals from the burn-off procedure were distributed on copper tape for inspection and measurement under SEM. For each CFF filament and printed CFF test coupon, a minimum of 10 images were taken at 100X magnification. A custom Matlab (The Mathworks, Natick, Massachusetts, USA) program was written to calibrate the pixel width first based on the scale bar shown in Figure 3.2, and then to select the two ends of the fiber in a manual operation to compute fiber length. The images were taken under high vacuum at Secondary Electron Imaging (SEI) mode, with 12mm distance between the specimen and lens; the voltage is set at about 10kV. For the calibration bar shown in Figure 3.2, it occupies 100 pixel width of length, which means each pixel in the image has a width of 1µm. Since 100X magnification is applied to all the images, the uncertainty of the measurement from the pixel size or from the SEM machine itself should be same for all the length measurement. Approximately 1000 fibers were measured for each material type, and any fiber with clear two ends are picked out in the image. The weight average fiber length and weight average fiber aspect ratio were evaluated for each. Weight average values were calculated by dividing the data into a user-defined number of equally spaced intervals. The number of samples in each interval was then multiplied by the center value of its own interval. The sum of the values from each

interval was divided by the total number of fibers measured to obtain weight average fiber data.

A schematic of the complete experimental process is shown in Figure 3.3. There are three tests conducted in this study: Test 1) The first one is the tensile test, Test 2) the second one is the fracture surface analysis, and Test 3) the third one is the fiber length distribution. The tests are visually described and follow in the order indicated by the direction of the arrows.
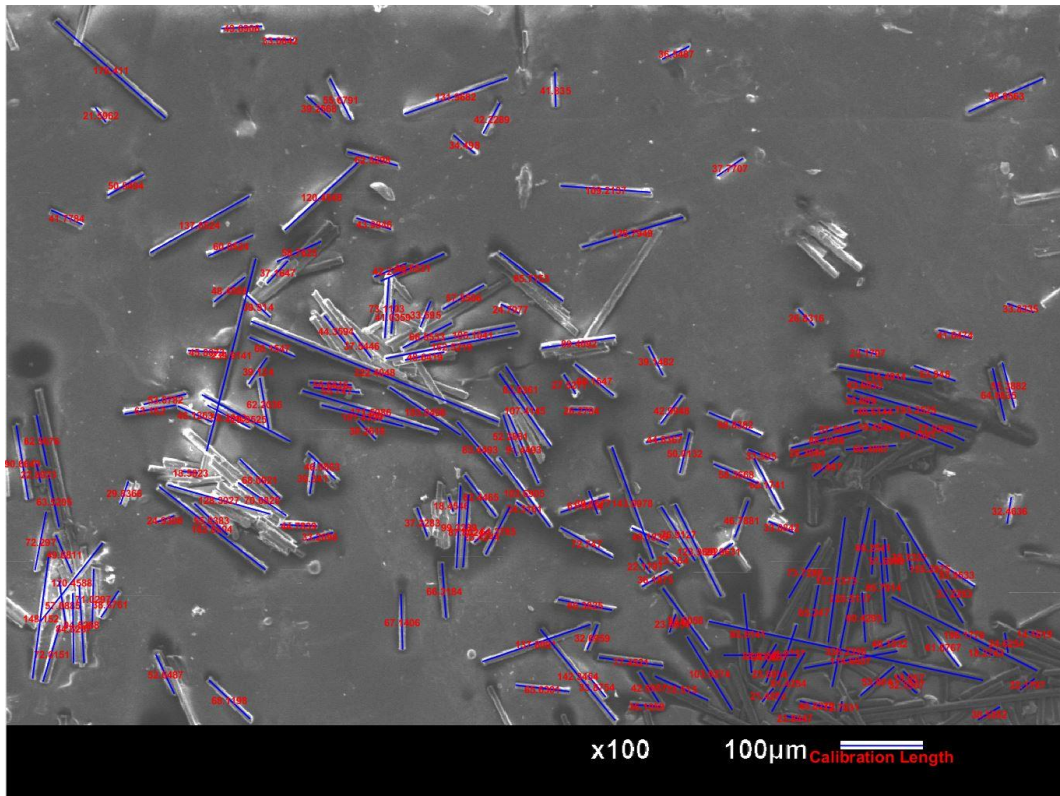
Figure 3.2. Fiber Length Measurement image
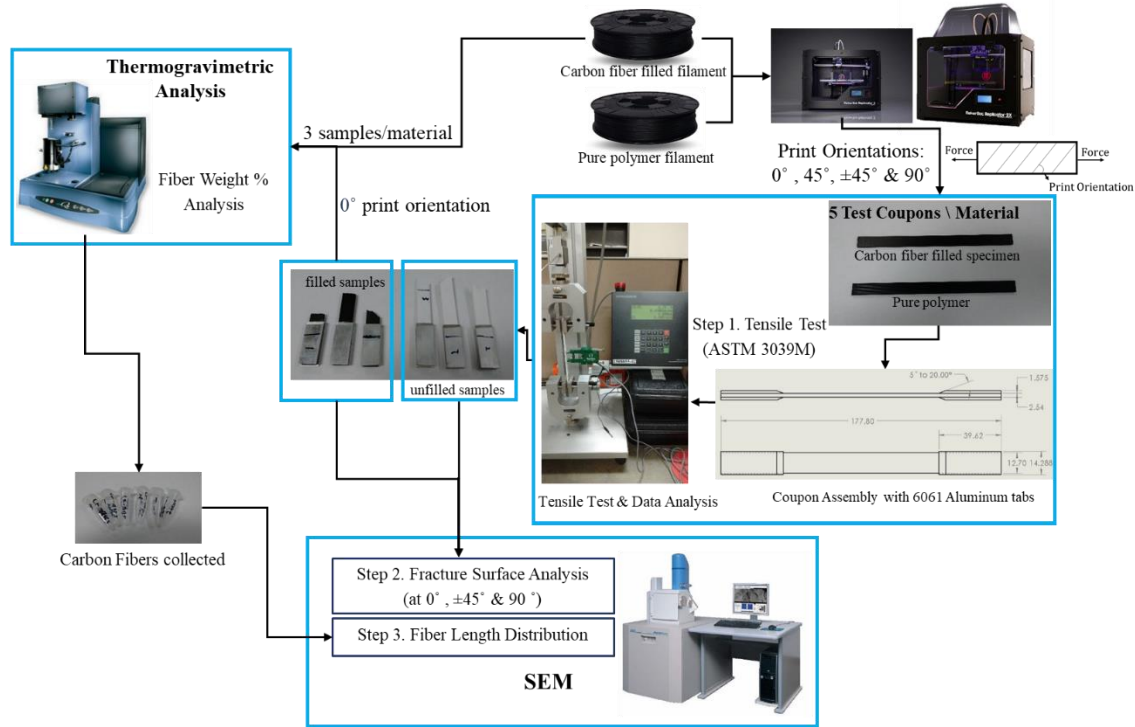
Figure 3.3. Experimental procedure schematic

3.2. *Results and Discussions*

3.2.1. *CFF ABS vs. ABS*

The measured fiber weight fraction is 16.8% for the CFF ABS filament, and the weight average aspect ratio for CFF ABS testing sample is 6.5. Tensile test curves of applied stress versus strain at different print orientation appear in

Figure *3.4*. Each line is a representative curve for the tensile test of the kind of material printed at that direction. The high initial slope of CFF ABS coupons indicates higher modulus when compared to the unfilled samples, however, CFF ABS coupons exhibit lower ductility than the unfilled ABS coupons. The trend follows for all the samples with four print orientations. Note that 'alt 45' indicates samples with alternating beads at ±45 degrees.
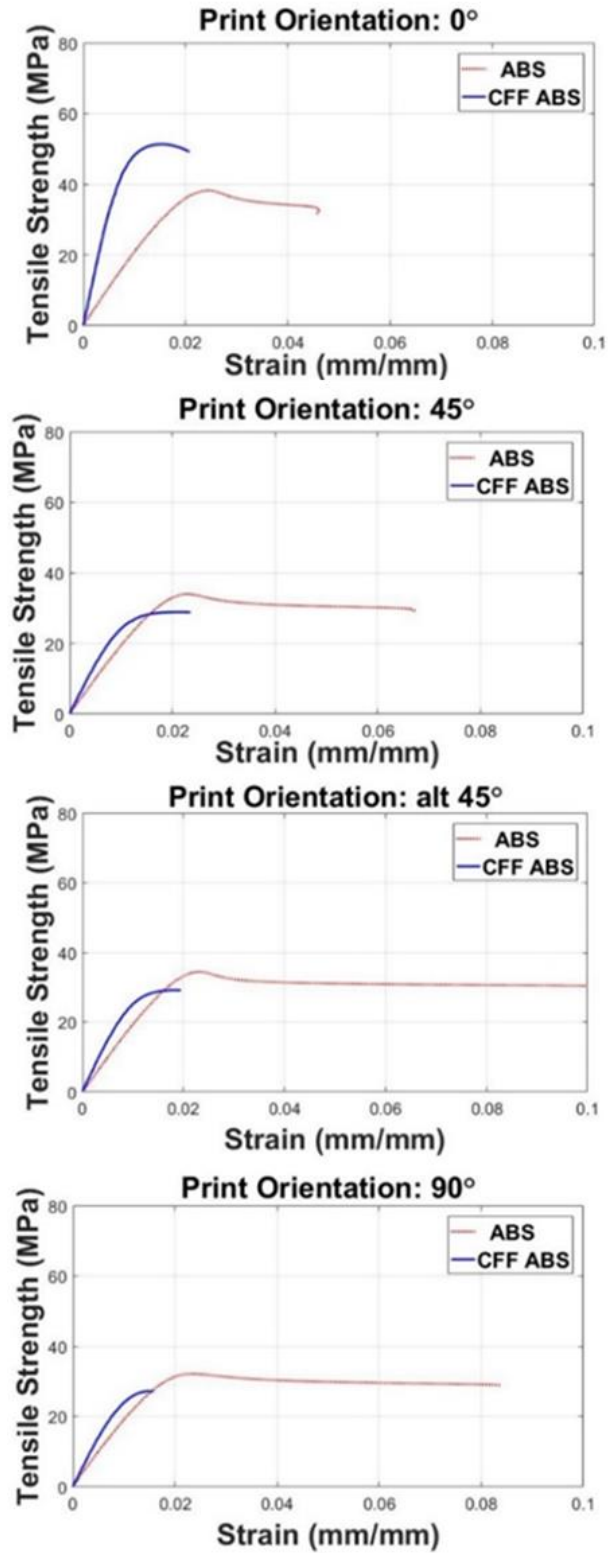
Figure 3.4. Typical test curves, ABS vs CFF ABS

The statistical average tensile strength and tensile modulus for unfilled ABS and CFF ABS test bars fabricated with the FFF process described above appear in Figure 3.5 and Figure 3.6, respectively. Error bars represent ±1 standard deviation over five samples. CFF test coupons at 0 degree print orientation are seen to have tensile strength and tensile modulus that is 33.2% and 212.2%, respectively, above their unfilled counterpart. However, the CFF ABS coupons exhibit inferior performance in tensile strength at other print orientations. Tensile strength was found to decrease by 13.5%, 15.9% and 15.7% at 45 degrees, ±45 degrees, and 90 degrees, respectively, when CF is added. Alternatively, the tensile modulus shows a small improvement with CF, increasing by 44.0%, 45.8% and 47.1% at 45 degrees, ±45 degrees, and 90 degrees, respectively.
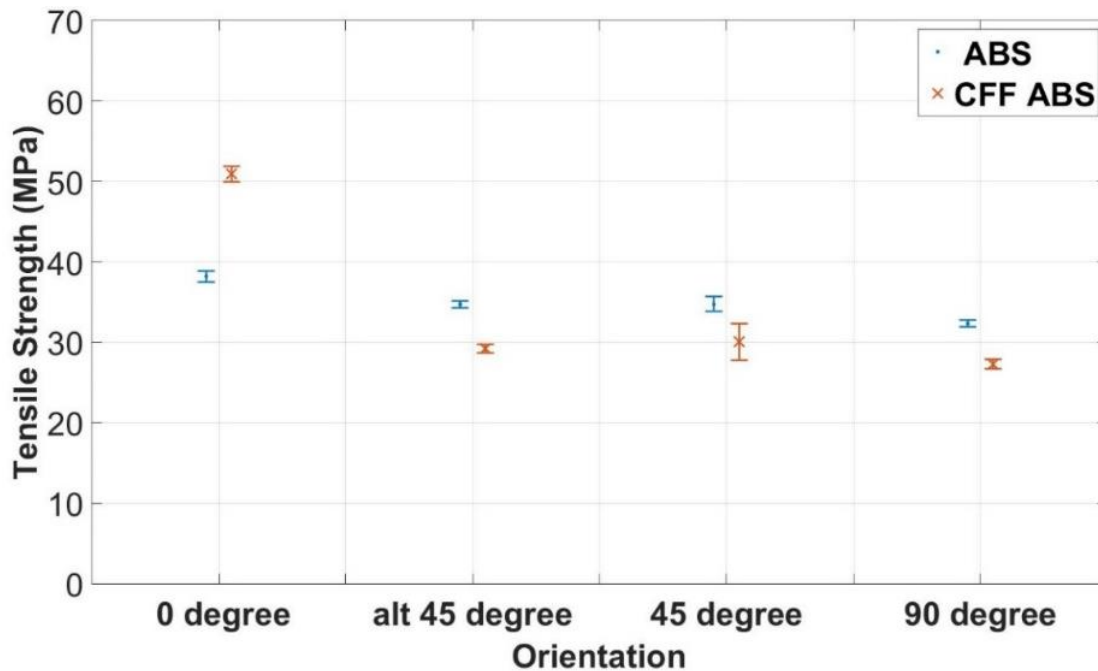
Figure 3.5. Tensile Strength ABS vs CFF ABS
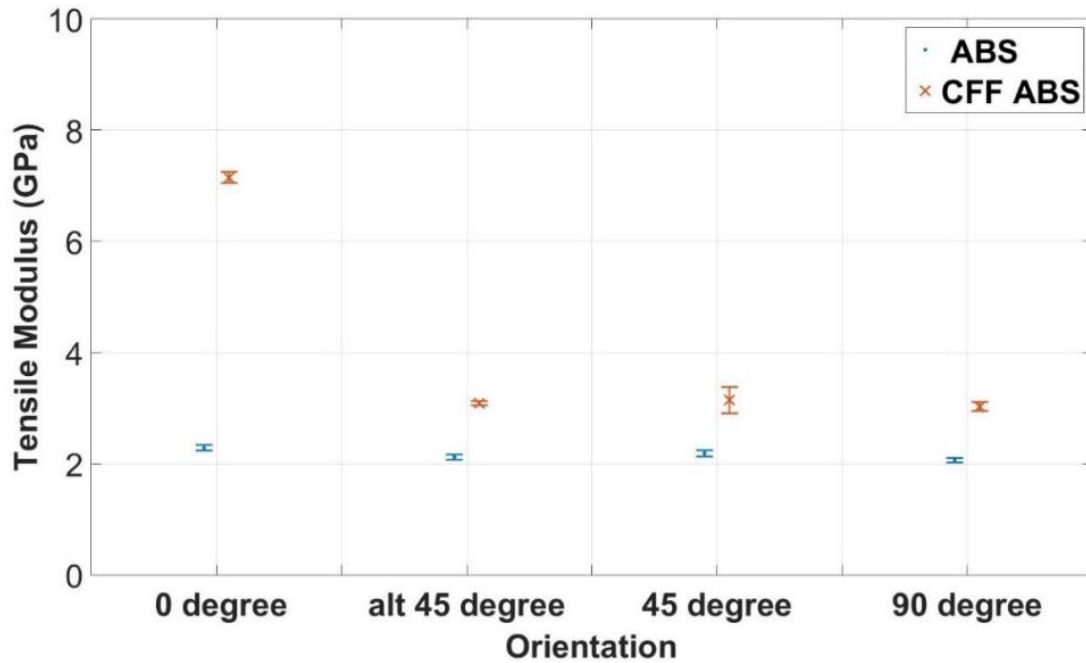
Figure 3.6. Tensile Modulus ABS vs CFF ABS

3.2.2. *CFF PLA vs. PLA*

The measured fiber weight fraction is 12.6% for the CFF PLA filament, and the weight average aspect ratio for CFF PLA testing sample is 8.0. Figure 3.7 shows tensile test curves of applied stress versus strain for unfilled PLA and CFF PLA. Note that CFF PLA is the least ductile material among all the test materials in this study. Testing of CFF PLA required that tabs with smaller tapered angle for 0 degree and 90 degree print orientations to avoid premature sample failure at the edge of the tabs. PLA coupons overall show slightly higher elongation than CFF PLA test coupons.

51
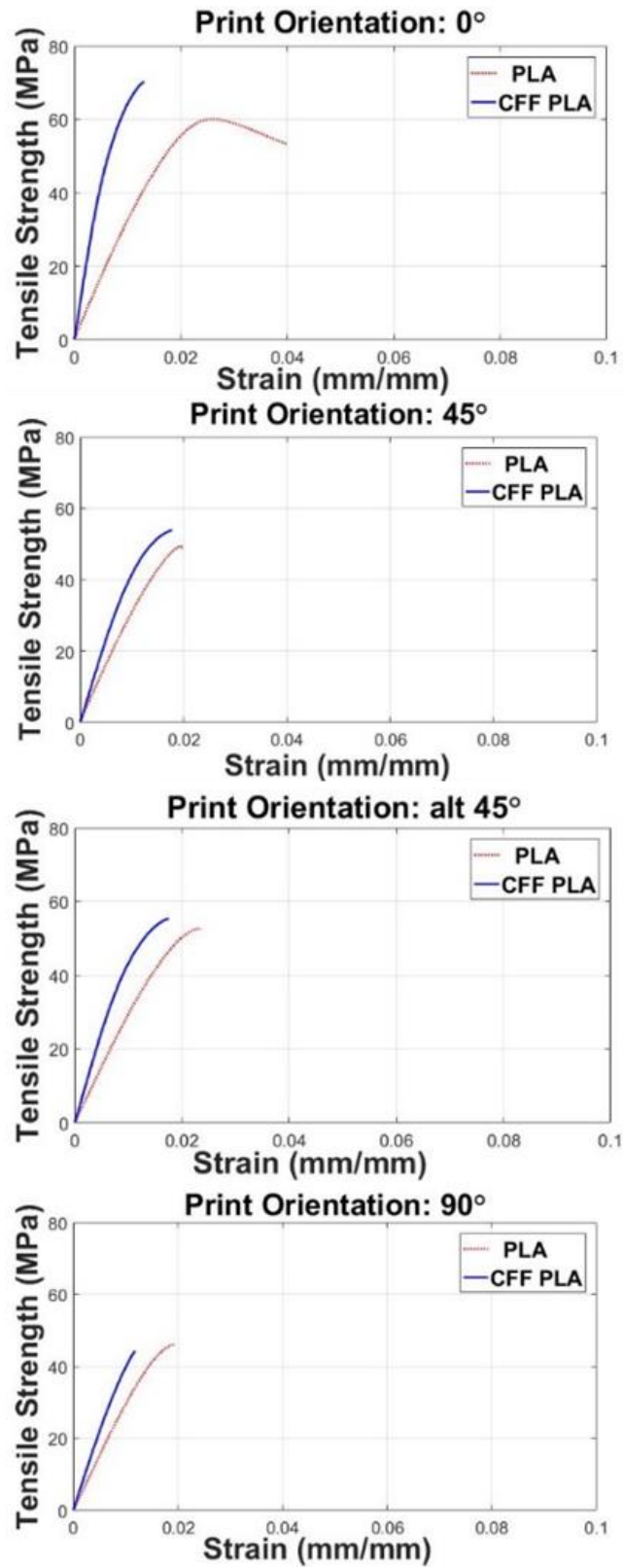
Figure 3.7. Typical tensile curves, PLA vs CFF PLA

A summary of PLA and CFF PLA tensile test results appear in Figure 3.8 and Figure 3.9. Note that tensile strength and tensile modulus is found to increase by 14.0% and 162.9%, respectively, for CFF PLA, at 0 degree print direction once CF is added. At 45 degrees, ±45 degrees, and 90 degrees print orientations, tensile strength improves by 2.2%, 4.6% and -4.0%, respectively, and tensile modulus increase by 50.7%, 59.5% and 44.5%, respectively.
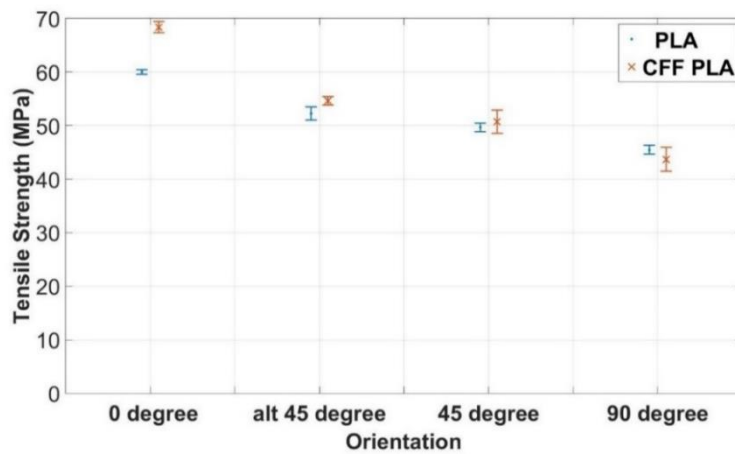
Figure 3.8. Tensile Strength PLA vs CFF PLA
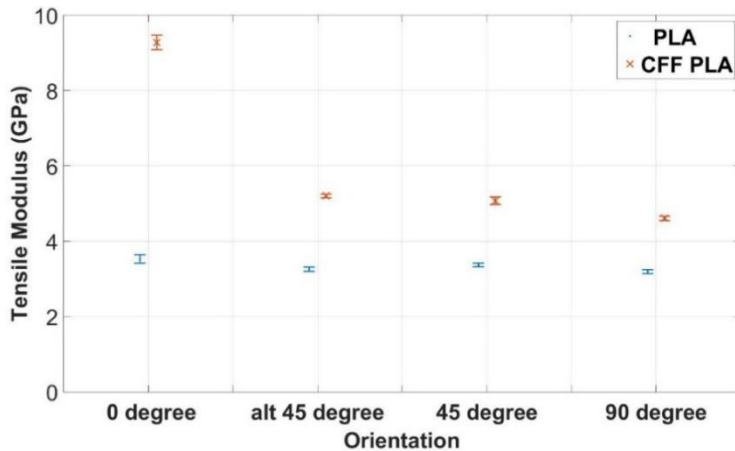
Figure 3.9. Tensile Modulus PLA vs CFF PLA

3.2.3. *CFF PETG vs. PETG*

The measured fiber weight fraction is 17.7% for the CFF PETG filament, and the weight average aspect ratio for CFF PETG testing sample is 9.4. The tensile test results for applied stress versus strain for PETG based filament appear in Figure 3.10. Note that unfilled PETG curves exhibit a decrease in strain during the tensile test. This decrease in the measured strain resulted from necking that occurred outside the gage section. The resulting extension in the necking region outside of the gage section caused the remainder of the test coupon to contract under the reduced load. This should not cause a problem with the tensile strength measurement since the maximum stress is recorded before the necking occurs. PETG typically has a relatively high ductility, and the test coupon at all print orientations did not fail before the test machine reached its data recording limit. This phenomenon made it difficult to capture the stress-strain behavior at high strain.
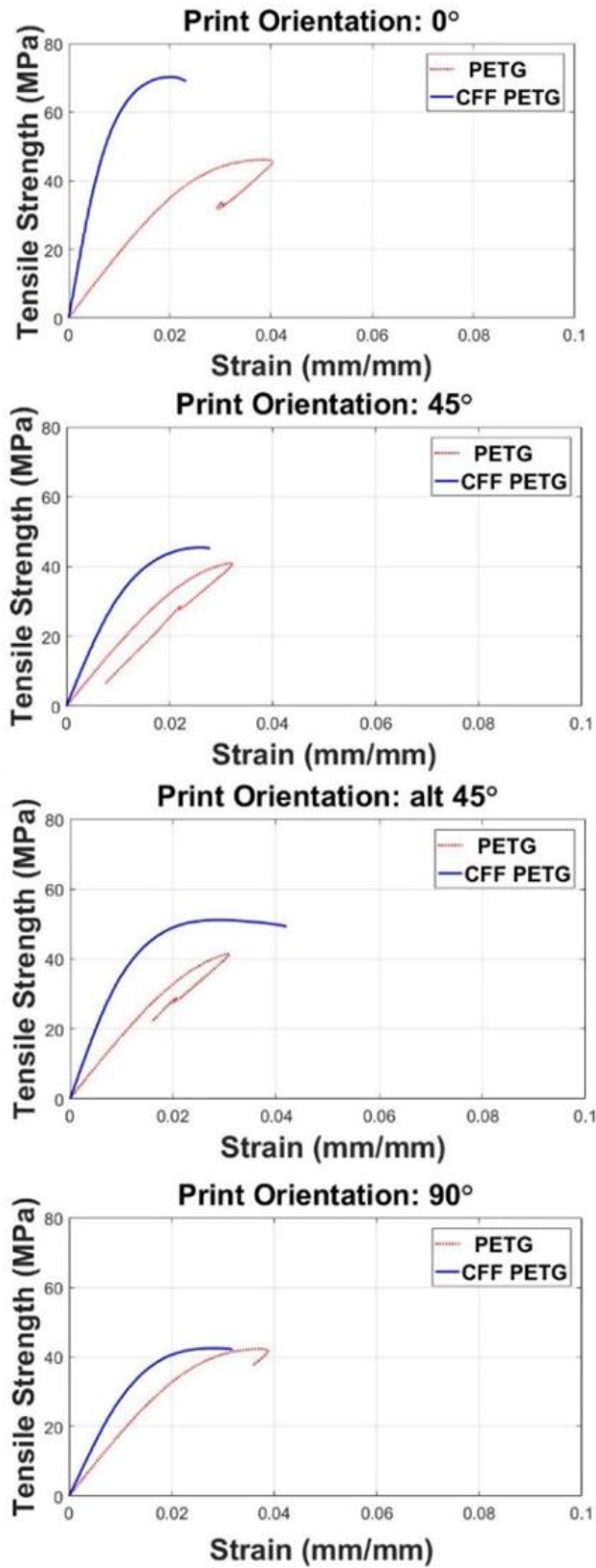
Figure 3.10. Typical tensile test curves, PETG vs CFF PETG

A summary of the tensile test results for PETG and CFF PETG appear in Figure 3.11 and Figure 3.12. The tensile strength and tensile modulus is found to increase by 48.2% and 313.2%, respectively, at 0 degree print direction. At 45 degrees, ±45 degrees, and 90 degrees print orientations, tensile strength improves by 7.9%, 23.2% and 2.9%, respectively, and tensile modulus increase by 83.7%, 121.5% and 68.8%, respectively.
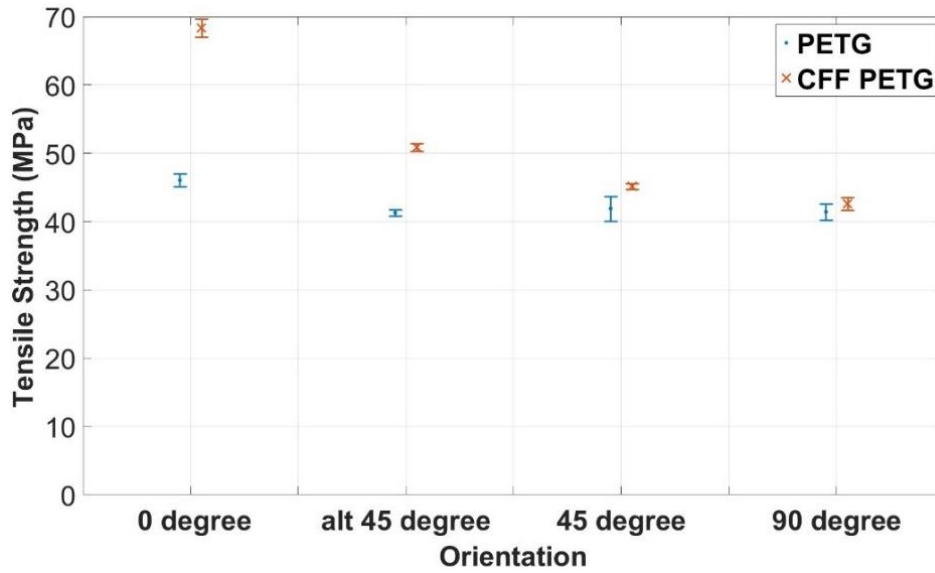
Figure 3.11. Tensile Strength PETG vs CFF PETG
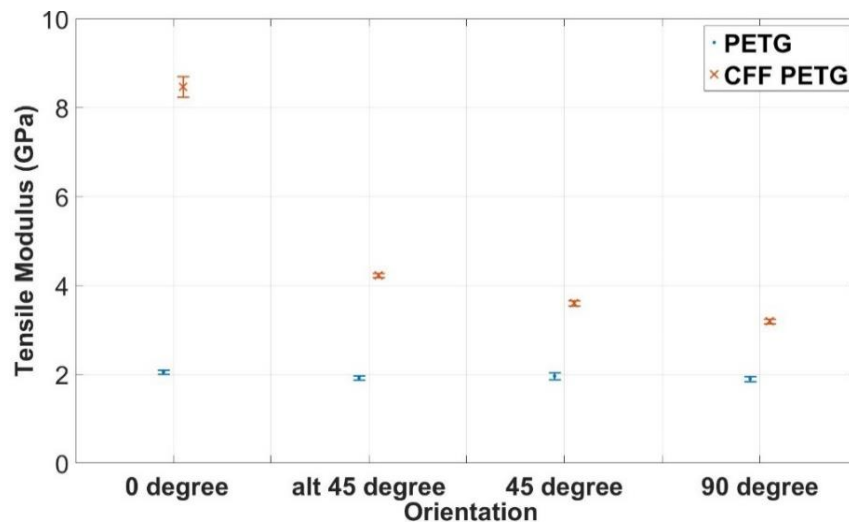
Figure 3.12. Tensile Modulus PETG vs CFF PETG

3.2.4. CFF Amphora vs. Amphora

The measured fiber weight fraction is 11.0% for the CFF Amphora filament, and the weight average aspect ratio for CFF Amphora testing sample is 5.8. Tensile test applied stress versus strain curves of for Amphora based filament appear in Figure 3.13. Note that measured results at 0 degree and 45 degree print orientation in Figure 3.13 also exhibit the strain reduction described for PETG tensile test curves. Amphora also exhibits high ductility, and the test coupon at 0 degree and 45 degrees print orientation did not fail.
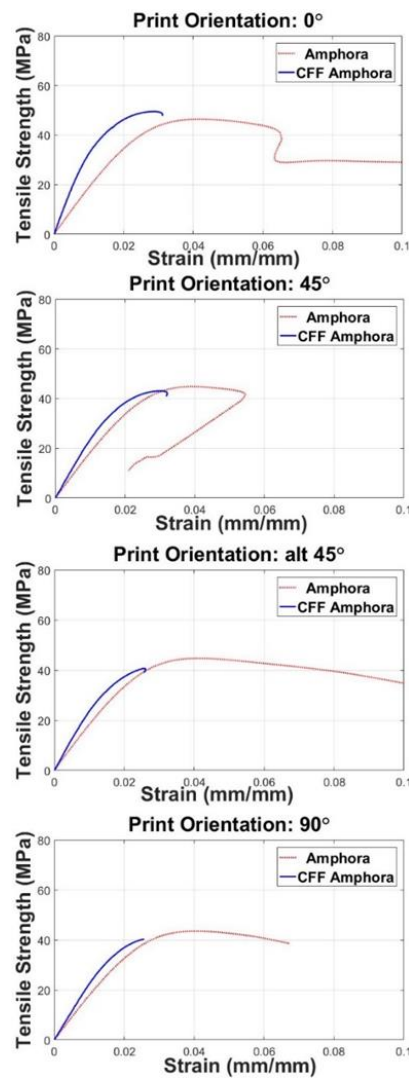
Figure 3.13. Typical tensile curves, Amphora vs CFF Amphora

A summary of tensile test results for Amphora and CFF Amphora appear in Figure 3.14 and Figure 3.15. The tensile strength (defined here as the maximum stress applied to the test sample during the test) and tensile modulus is shown to increase by 5.1% and 95.5%, respectively, at 0 degree print direction. At 45 degrees, ±45 degrees, and 90 degrees print orientations, tensile strength decreased by 6.2%, 5.4% and 8.4% for CFF Amphora, respectively. Results from the same experiments show that tensile modulus increases by 29.8%, 34.2% and 28.9%, respectively, when CF was used in the Amphora polymer.
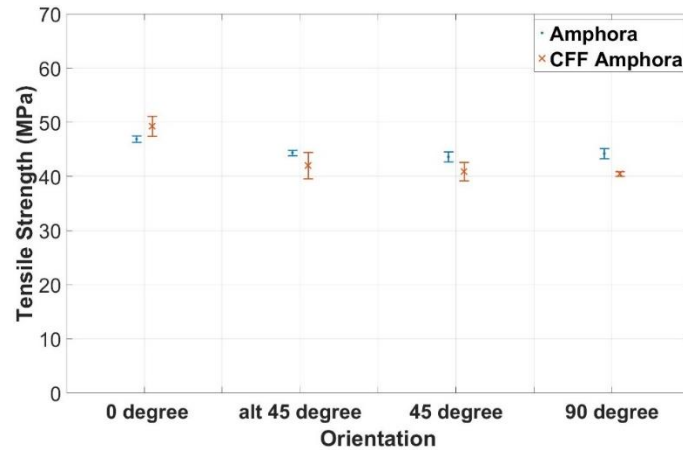
Figure 3.14. Tensile Strength Amphora vs CFF Amphora
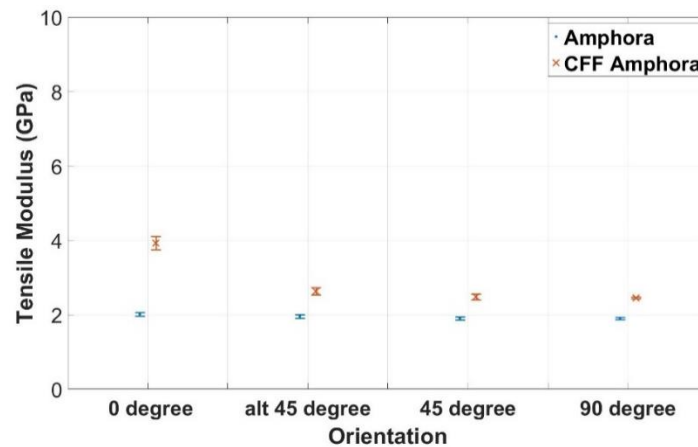
Figure 3.15. Tensile Modulus Amphora vs CFF Amphora

3.2.5. *Effect of Carbon Fiber on Tensile Properties*

The addition of carbon fibers significantly increases tensile modulus at 0 degree print orientation for PLA, ABS, PETG and Amphora test coupons. In general, the addition of carbon fiber improves tensile modulus of the test coupons for all print orientations considered here. When the print orientation aligns with the axis of the test bar, mechanical properties show the most improvement with the addition of carbon fiber. In this case, the high tensile strength of the carbon fiber greatly contributes to the increase in the mechanical properties of the CFF polymer test coupons.

CFF PLA at 0 degree print orientation have the greatest strength and modulus values, and CFF PETG at 0 degree print orientation yields the greatest strength and modulus improvement. In general, CFF test samples at the 0 degree print orientation gives the greatest strength and modulus improvement relative to their unfilled counterparts, for all the test materials. CFF polymer test samples at 90 degrees print orientation gives the least improvement, for all the test materials. Lastly, the addition of CF reduces the ductility of the printed samples compared to their unfilled test samples, dropping over 50% in most cases.

3.3. *Fractography*

Figure 3.16 to Figure 3.19 show SEM images typical of the fracture surfaces for each material considered in this study. The CFF fractured coupons were subjected to high vacuum chamber with SEM mode; the height between the sample mount and lens was at around 45mm, and voltage was set at around 5kV. For the unfilled fractured coupons, the machine setting was the same except the chamber was in low vacuum at Backscattered Electron Shadow (BES) mode). In each figure, the first column represents the unfilled

fractured surfaces, and second column represents CFF fractured surfaces. From the first to third row in order, fractured images at 0 degree, ±45 degrees and 90 degrees print orientations are shown. Features of interest are labeled in the SEM images with lower case letters.
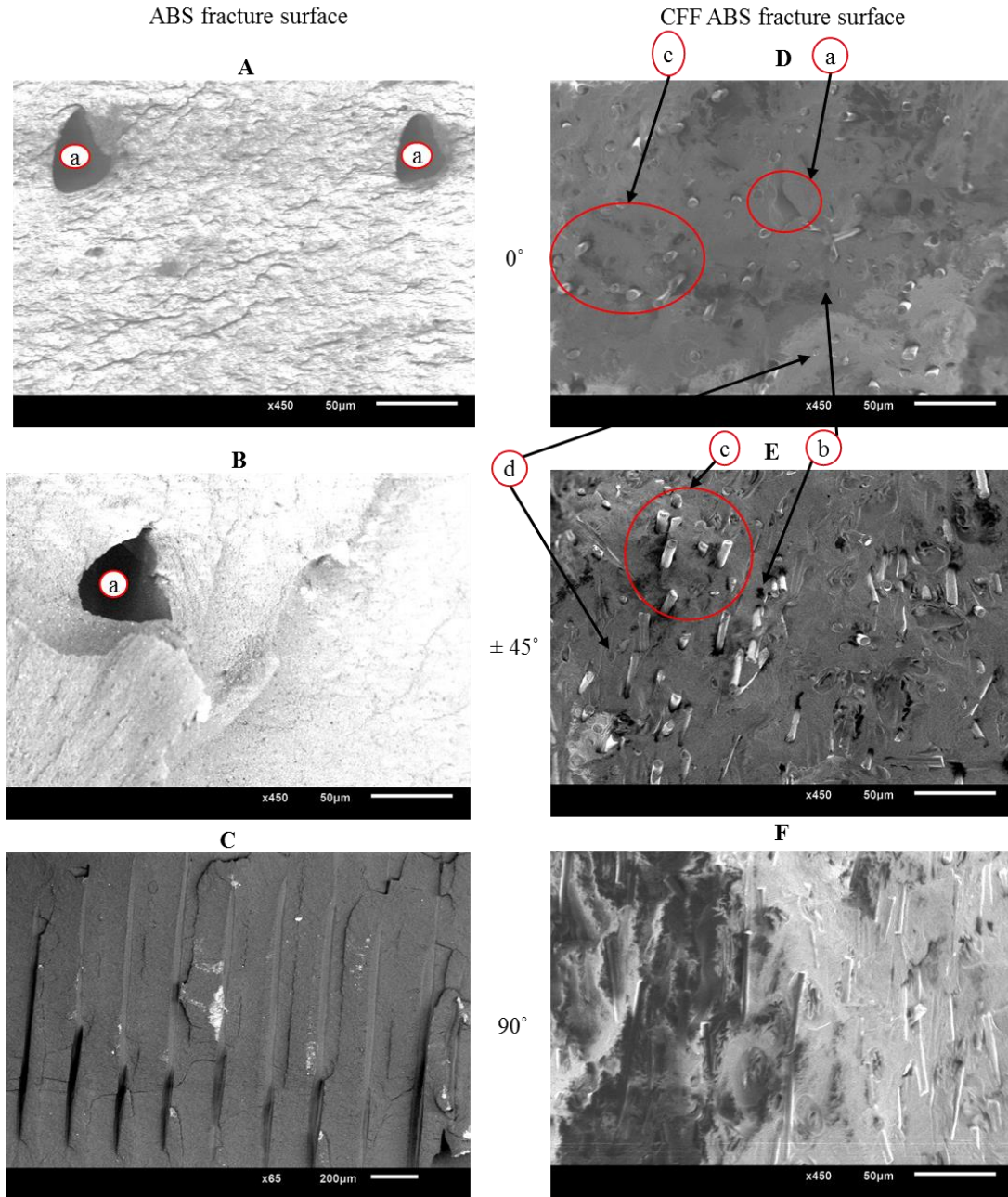
Figure 3.16. SEM Images for ABS and CFF ABS Fracture Surfaces

Type (a) voids appearing in Figure 16 that formed between print layers are easily identified inside the unfilled polymer test coupons at the 0 and ±45 degrees print orientations. The remainder of the fracture surface appeared to be free of voids. Type (a) voids are less prominent in CFF test coupon surfaces. This is likely a result of the CFF polymer being more thermally conductive than the unfilled polymer, thus altering the flow of heat and improving the fusion between layers during the deposition process [2]. Other defects, however, are more pronounced. For example, images D and E in the Figure 3.16 to Figure 3.19 have numerous small type (b) circular voids which appear to be sites where fibers pulled out of the polymer during the tensile test. There are also fibers exposed on the fracture surface, identified as type (c) feature. More interestingly, there seems to be pore enlargement around the type (c) feature at several sites. In addition, a type (d) pore has circular dented shape, and which appears to have occurred when the fiber pulled away from the polymer. Considering image F in Figure 3.16 to Figure 3.19, a large number of fibers are exposed on the fracture surfaces. Instead of sharing the tensile force between the CFF and the polymer matrix, the weak interfacial bond appears to have failed resulting in tensile strength that is lower than that of the unfilled polymer test coupon. This is confirmed by the tensile test results in Figure 3.5, Figure 3.8, Figure 3.11 and Figure 3.14. Finally, note that the fibers align well with the print orientation in each image, further agreeing with the observation made by Tekinalp, et al. [2].

PLA fracture surface

CFF PLA fracture surface
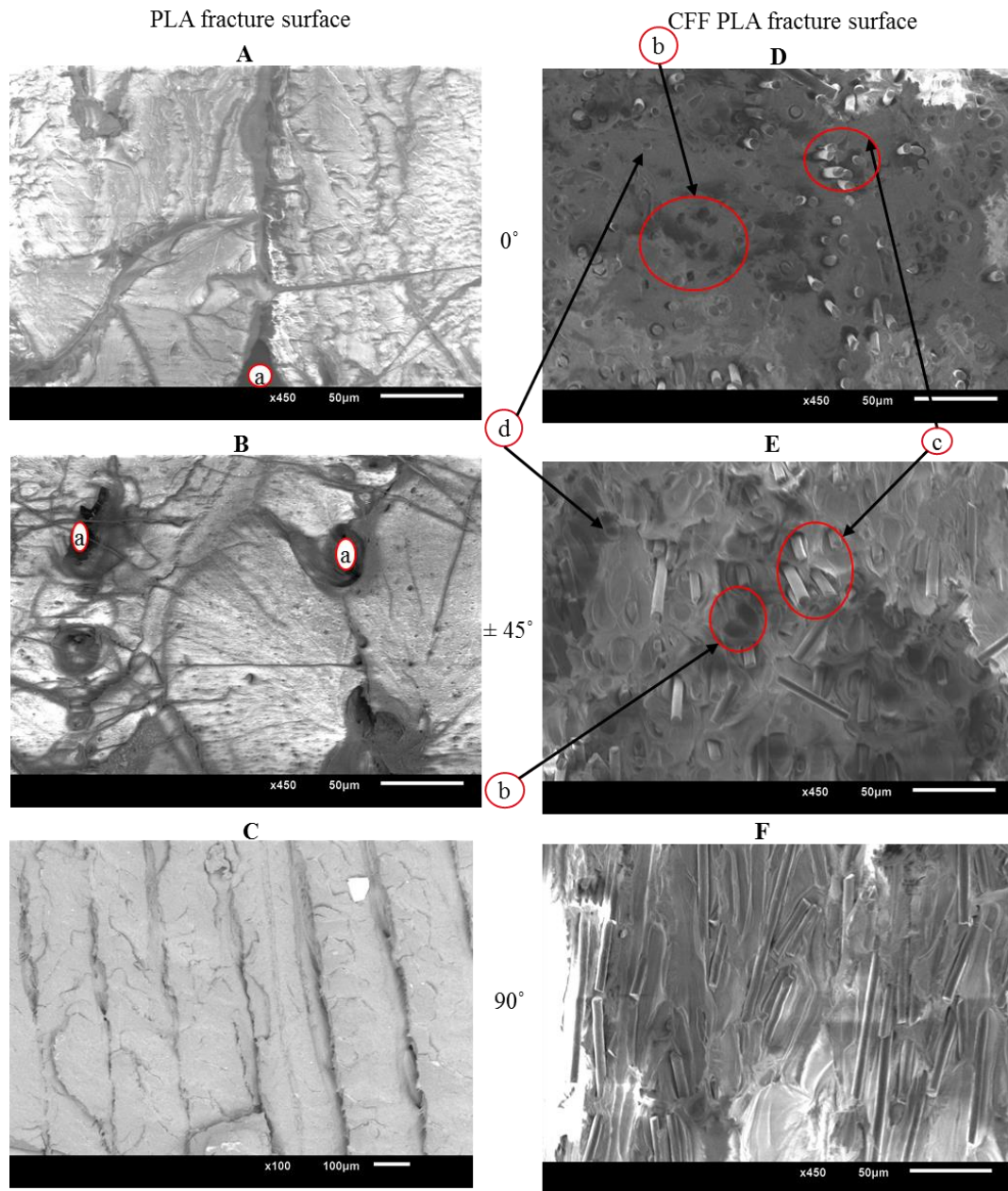
Figure 3.17. SEM Images for PLA and CFF PLA fracture surfaces

PETG fracture surface   CFF PETG fracture surface

A   **D**

Test coupons did not
rupture before the machine
reached data recording limit   0°
and stopped testing

B   **E**

± 45°

C   **F**

90°
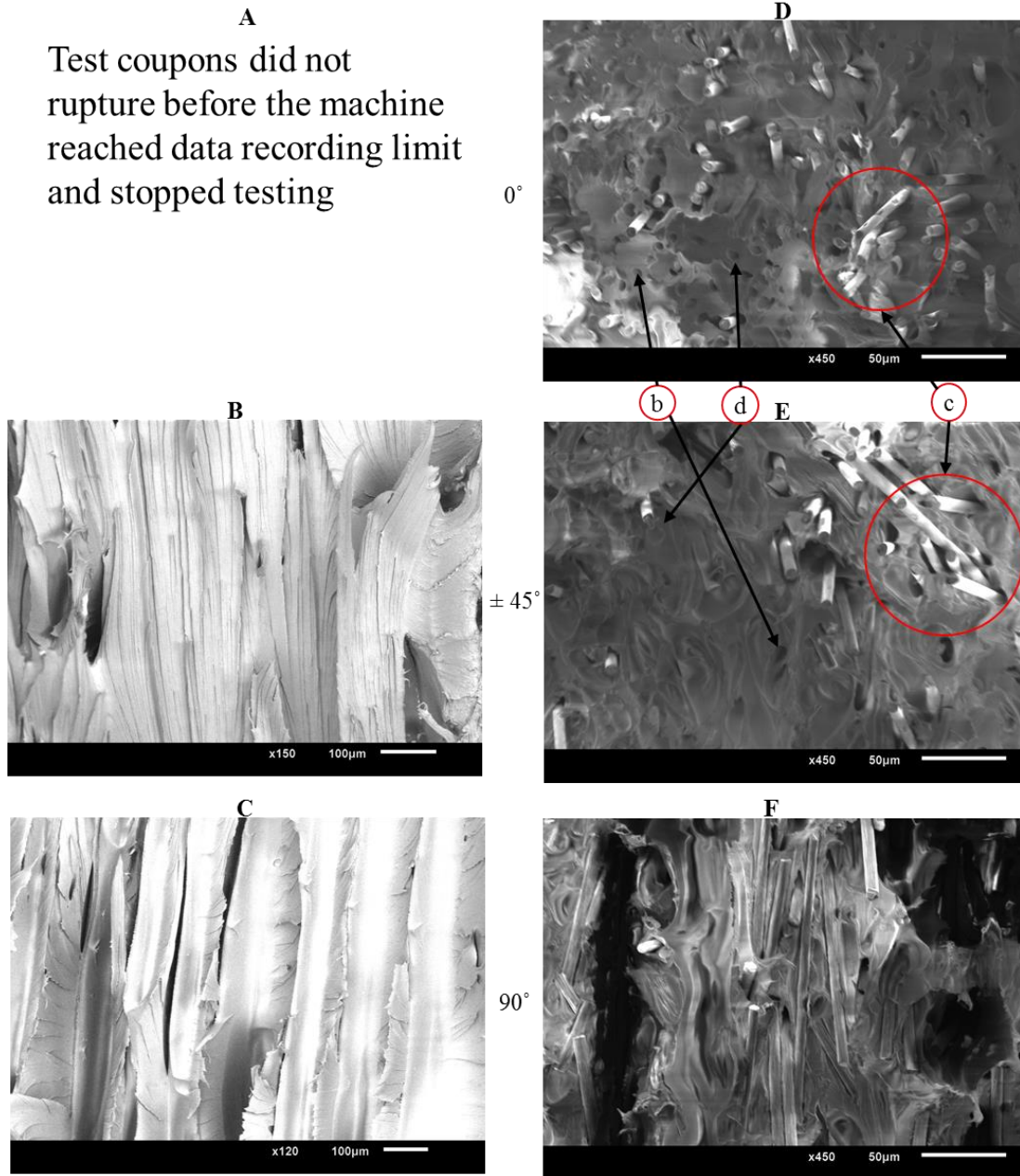
Figure 3.18. SEM Images for PETG and CFF PETG fracture surfaces

63
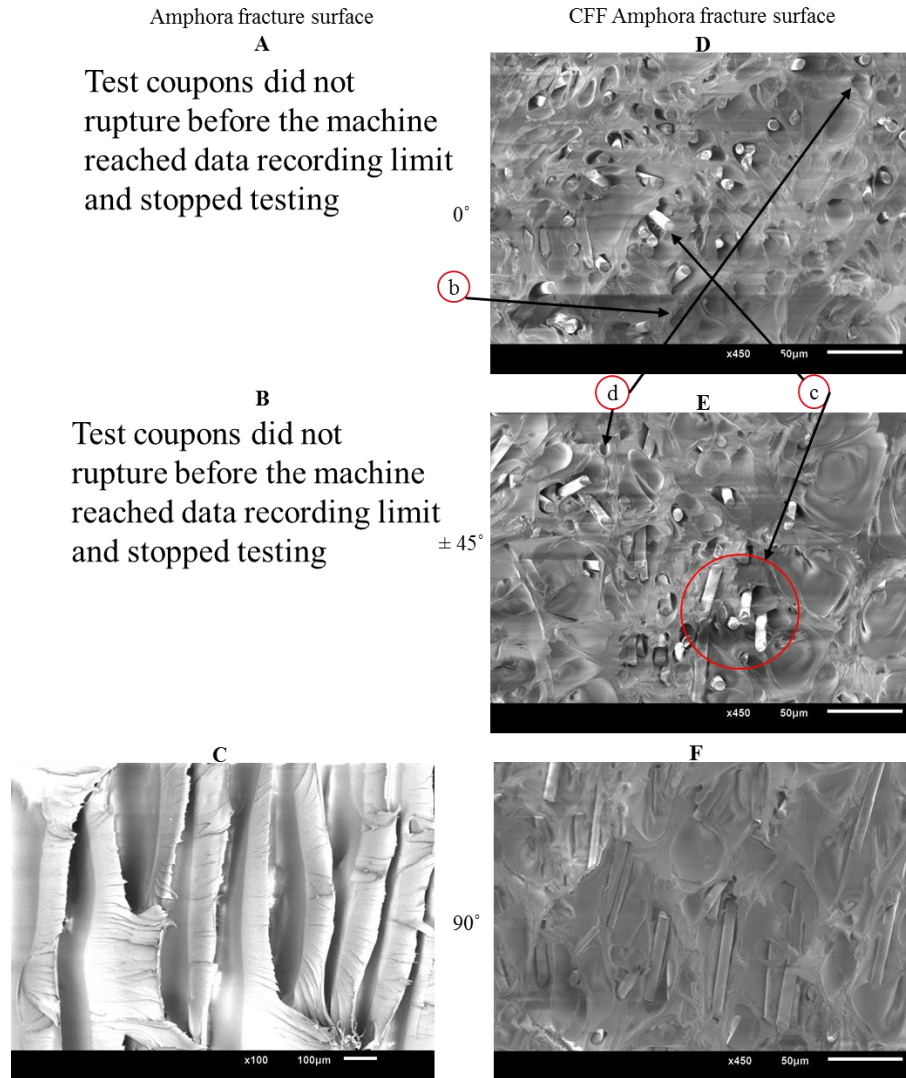
Figure 3.19. SEM Images for Amphora and CFF Amphora fracture surfaces

3.4. *Fiber Weight Fractions and Fiber Length Distribution*

Carbon fiber weight fraction for each CFF polymer material tested in this study appears in Table 3.4. FLD comparisons before and after the print appear in Figure 3.20, and weight average fiber length, weight average fiber aspect ratio and number of fibers evaluated for each CFF polymer material are shown in Table 3.5. For material purchased from 3DXTECH, fiber diameter measured on fifteen individual fibers yielded an average value of 7.77μm. Five measurements of fiber diameter on CFF Amphora filament resulted

in an average fiber diameter of 7.68μm. We found that the CFF Amphora had shorter fibers than the other CFF materials. This could likely be the cause of CFF Amphora exhibiting less improvement over Amphora than that seen in the other CFF materials.

Results appearing in Figure 3.20 show that FLD for printed CFF samples shift downward slightly as compared to the CFF filament FLD, suggesting that the printing process caused a slight reduction in fiber length overall. Furthermore, the differences between weight average aspect ratios before and after the print are less than 1.5. These results show that the fiber length remains mostly unaffected during deposition as the CFF filament travels throughout the nozzle.

Table 3.4. Carbon fiber weight fraction for each CFF polymer material

CFF ABS filament	16.8%
CFF PLA filament	12.6%
CFF PETG filament	17.7%
CFF Amphora filament	11.0%
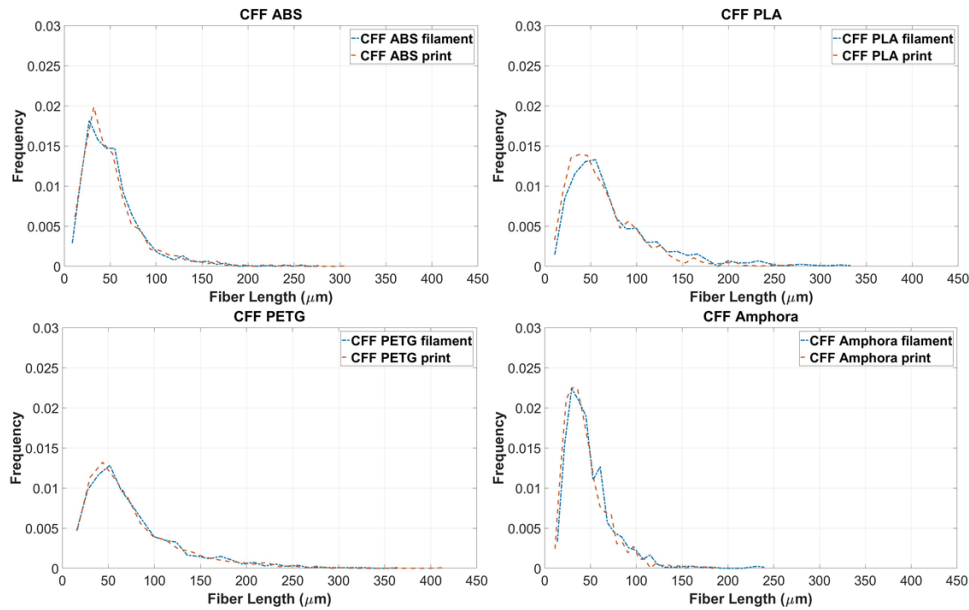
Figure 3.20. Fiber length distributions plots for each CFF material filament

Table 3.5. Fiber length and aspect ratio information for each CFF material

Results	CFF ABS filament	CFF ABS print	CFF PLA filament	CFF PLA print	CFF PETG filament	CFF PETG print	CFF Amphora filament	CFF Amphora print
weight average fiber length	51.4 μm	50.8 μm	72.8 μm	62.3 μm	74.9 μm	73.1 μm	48.5 μm	44.9 μm
weight average aspect ratio	6.6	6.5	9.4	8.0	9.6	9.4	6.3	5.8
# of fibers evaluated	1391	1391	1064	1064	1123	1123	1015	1015

This chapter investigated the tensile properties and fiber length distributions for four commercially available CFF filaments, and results were compared against their unfilled polymer counterparts. Fracture surfaces of filled and unfilled samples were inspected under the SEM. The addition of carbon fiber to the polymer was shown to increase the modulus of elasticity for all materials and all print orientations considered here, where the greatest improvement of 313.2% occurred at 0 degree print orientation of PETG. PETG printed at 0 degree also showed the highest increase in tensile strength of 48.2%. All samples printed at 0 degree showed an increase in tensile strength with the CFF PETG having the greatest overall tensile property improvement among the four selected CFF filaments. Results with print orientations other than 0 showed much less improvement with the addition of CFF, with ABS and Amphora showing a decrease in tensile strength for all print orientations other than zero degree. The FFF process results in small fiber breakage when processing CFF filled filament; however, all the samples have weight average of fiber length less than 100μm. Finally, improving interfacial bonding strength between fiber and polymer matrix is needed to fully realize the potential of CFF FFF parts. Particularly from the tensile test, it is obvious that the fiber orientation is crucial to the

mechanical properties of the printed the parts, which justifies the need for a design tools

that takes the fiber orientation into account. In the next chapter, the formulation of the three

dimensional optimization will be discussed in detail, based on SIMP method and CFAO.

CHAPTER FOUR

Three Dimensional Topology Optimization Scheme with SIMP Method and CFAO

In this chapter, we discuss the topology optimization process developed here for structures produced by the FFF process. In Section 4.1, a general form of the design optimization problem is presented which is followed in Section 4.2 with the traditional SIMP method. The SIMP method is extended in Section 4.3 to include the design variables of material orientation. Next, the FEA procedure for linear elastic static analysis is derived in Section 4.4. The type of finite element and the integration of the elemental stiffness matrix are presented in Section 4.5. Section 4.6 considers the constitutive material model we use in the topology optimization, and Section 0 derives the design sensitivities of the compliance objective function with respect to the density and material orientation variables. The optimization algorithm and a flow chart of the optimization process is discussed in Section 4.8. Lastly, fiber orientation modelling is briefly described in Section 4.9. The correlation between the fiber orientation state and predicted elastic constants are also included.

4.1.   General Formulation of Design Optimization

Design optimization involves problems across wide areas of discipline and enjoys application in many industries. It is applied to the fields of computer science, operational research, transportation, telecommunication and more. Despite the diversity of the fields that are involved, it is common to write the general design optimization problem statement in a standard form as

$$Minimize\ f(x) = f(x_1, x_2, \dots, x_n) \tag{4.1}$$

$$subject\ to: h_j(x) = h_j(x_1, x_2, \dots, x_n) = 0;\ j = 1\ to\ p \tag{4.2}$$

$$and\ g_j(x) = g_j(x_1, x_2, \dots, x_n) \le 0;\ i = 1\ to\ m \tag{4.3}$$

$$x_{iL} < x_i < x_{iU} \tag{4.4}$$

Where $f(x)$ is the objective function to be solved, $h_j(x)$ represents the equality constraints, and $g_j(x)$ represents the inequality constraint. The design is parameterized by n design variables which are form the n-dimensional design variable vector x which are adjusted in the optimization solution to solve Equations (4.1) – (4.4). While the form given in Equation (4.1) designates a minimization problem, we note that maximization may be performed by multiply the objective function $f(x)$ by -1. Equation (4.2) defines the p equality constraints $h_j(x)$ in the design problem. It is important to keep in mind that the number of independent equality constraints cannot exceed the number of design variables. Otherwise, the design problem becomes overdetermined.. Also, in the case when $p = n$, the solution is simply the roots of the equality constraints; therefore, no optimization is necessary. Equation (4.3) defines the m inequality constraints $g_j(x)$. There are no restrictions on how many inequality constraints can be imposed. All of the design variables have upper or lower or both bounds that may be imposed for design considerations. The topology optimization problem used in this thesis is in the form of the standard design optimization model given in Equations (4.1) – (4.4). In the next section we focus on the SIMP method and its extension with CFAO to design AM structures produced with anisotropic materials such as those obtained in the FFF process with CF polymer composites. In the next section, the problem formulation for SIMP method is introduced again, and the objective and the constraint functions are to be derived in detail.

Sigmund [37] developed a 99 line Matlab code based on the SIMP method, which has since been extended in several papers to address more complex topology optimization problems. The three dimensional topology optimization code built for the research in this thesis is based on the SIMP formulation for structural optimization. Therefore, we begin our discussion of topology optimization with the SIMP formulation. Equations (4.5). and (4.6). state the objective function and design constraints, respectively, which has become the standard SIMP formulation.

$$\min: c(\boldsymbol{\rho}) = \mathbf{U}(\boldsymbol{\rho})^\mathbf{T}\mathbf{F} = \mathbf{U}(\boldsymbol{\rho})^\mathbf{T}\mathbf{K}(\boldsymbol{\rho})\mathbf{U}(\boldsymbol{\rho}) = \sum_{e=1}^{N}(\rho_e)^p u_e^T k_0 u_e \qquad (4.5)$$

$$\text{Subject to: } \begin{cases} \dfrac{V(\boldsymbol{\rho})}{V_0} = f \\ \mathbf{K}(\boldsymbol{\rho})\mathbf{U}(\boldsymbol{\rho}) = \mathbf{F} \\ 0 < \rho_{min} \leq \boldsymbol{\rho} \leq 1 \end{cases} \qquad (4.6)$$

The objective function $c(\boldsymbol{\rho})$ is the compliance [113] which depends on the density design variable vector $\boldsymbol{\rho}$. $\mathbf{K}$, $\mathbf{U}$ and $\mathbf{F}$ are the global stiffness matrix, displacement vector and applied load vector of the finite element model, respectively. The variables $\mathbf{k_0}$ and $\mathbf{u_e}$ respectively represent the stiffness matrix and displacement vector at the elemental level. Each design variable ρ_e is bounded by a lower and upper bounds of 0 and 1, respectively. Furthermore, two equality constraints have to be satisfied by the optimization. The first equality constraint limits the maximum amount of volume can be filled with material with respect to the design domain, and the second constraint is the static equilibrium condition which can be achieved by finite element analysis.

In structural engineering, minimizing the compliance is equivalent to maximizing the stiffness of a structure where the stiffness depends on both the material and the

geometry of the structure. The compliance is a property that measures the degree of deformation of a structure for a given loading condition [113]. For example, consider a simple cantilever beam is loaded with a transverse point force F on its free end. The resulting tip displacement δ in the direction of F may be used to define the stiffness as

$$k = \frac{F}{\delta} \tag{4.7}$$

In many circumstances, it is very desirable to design products with high stiffness, providing reliability and safety for the users. The compliance for this simple problem is calculated as the potential energy $c = \delta F = k\delta^2$. The compliance function provides a means for including the potential energy from all of the applied loads, making it desirable for use in topology optimization.

The volume equality constraint restricts how much material is allowed to occupy the design domain. It is possible to replace equality form in Equation (4.6) with the inequality constraint

$$\frac{V(\boldsymbol{\rho})}{V_0} \leq f \tag{4.8}$$

since it would not affect the result of the topology optimization. This is because more material can always give higher stiffness for the structure, therefore the optimization algorithm will push the available volume fraction to the limit, resulting inan active constraint, which is the same as equality constraint in Equation (4.6).

The lower and upper bounds (ρ_{min} and 1, respectively) imposed on the components of the density design variable vector $\boldsymbol{\rho}$ correspond to void and solid. Lastly, $\mathbf{K}(\boldsymbol{\rho})\mathbf{U}(\boldsymbol{\rho}) = \mathbf{F}$ in Equation (4.6) is to enforce static equilibrium which is implemented through the finite element analysis.

4.3. Extension of SIMP Method with CFAO

To optimize the material orientation in the SIMP method, additional material orientation variables are introduced by including an orientation angle into each element. The resulting modified SIMP are very much the same as the original SIMP formulation, except by introducing an additional sets of material orientation variables $\boldsymbol{\theta}$.

$$\min: c(\boldsymbol{\rho}, \boldsymbol{\theta}) = \mathbf{U}(\boldsymbol{\rho}, \boldsymbol{\theta})^{\mathrm{T}}\mathbf{F} = \mathbf{U}(\boldsymbol{\rho}, \boldsymbol{\theta})^{\mathrm{T}}\mathbf{K}(\boldsymbol{\rho}, \boldsymbol{\theta})\mathbf{U}(\boldsymbol{\rho}, \boldsymbol{\theta})$$
$$= \sum_{e=1}^{N} (\rho_e)^{\mathrm{p}}\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_e(\theta_e)\mathbf{u}_e \tag{4.9}$$

$$\text{Subject to: } \begin{cases} \dfrac{V(\boldsymbol{\rho})}{V_0} = f \\ \mathbf{K}(\boldsymbol{\rho}, \boldsymbol{\theta})\mathbf{U}(\boldsymbol{\rho}, \boldsymbol{\theta}) = \mathbf{F} \\ 0 < \rho_{\min} \le \boldsymbol{\rho} \le 1 \\ -2\pi \le \boldsymbol{\theta} \le 2\pi \end{cases} \tag{4.10}$$

Again, $\mathbf{K}$, $\mathbf{U}$ and $\mathbf{F}$ are the global stiffness matrix, displacement vector and applied load vector of the finite element model, respectively. The variables $\mathbf{k}_e$ and $\mathbf{u}_e$ respectively represent the stiffness matrix and displacement vector at the elemental level, with a total number of N finite elements. P is the penalization power. Notice in Equation (4.9), the elemental stiffness matrix is no longer a constant matrix. Instead, it depends on the fiber angle variable θ_e given as

$$\mathbf{k}_e = \mathbf{k}_e(\rho_e, \theta_e) = (\rho_e)^{\mathrm{p}}\mathbf{k}_e(\theta_e) \tag{4.11}$$

In three dimensions, it would seem necessary to use 2 variables in each element to describe the directionality of the material. However, in this study, only one material orientation variable is needed for each element, due to the assumption made that fibers rotate only in the print plane. Each material orientation variable θ_e are bounded between -2π and 2π, as shown in Equation (4.10). where the wide range allows the angle to rotate into its optimal

position, avoiding a potential local minimum due to limitations placed on the problem by these side constraints. Since we have replaced the isotropic material of SIMP with an orthotropic material model, two material constants are needed for matrix construction, namely the density ρ and angle θ. This modification on the elemental stiffness matrices potentially adds significant computational time for the optimization, since each of the elemental stiffness has to be evaluated independently. For the purpose of this research, the penalization power is set to 3, as it is commonly used in the literature [37,101,102]. The lower bound of density variable is set to $\rho_{min} = 10^{-6}$. In the next section, the finite element analysis used to define $\mathbf{KU} = \mathbf{F}$ in this research is derived.

4.4.   *Finite Element Analysis in Three Dimension*

To evaluate the compliance function in the minimization problem of Equation (4.9), a Finite Element Analysis (FEA) is performed which is based on the theory of linear elasticity. FEA is a popular method used to compute an approximate solution of the partial differential equations (PDE) defined over the domain of interest. To solve the PDE analytically in many cases can be impossible. Alternatively, an approximate solution can be computed on the domain by discretizing it into elements, connected by the nodes where an approximate solution variables are solved at the nodes. In this approach, the solution is interpolated over the element using shape functions. The theory of elasticity forms a vector field problem, which is based on four assumptions [114]: First, the deformation of the structure is small; second, the behavior of the material is linear; third, the dynamic effect is neglected; and fourth, there is no gaps or overlaps during the deformation of the solid. To derive the finite element form for linear elastic structure domain, the general Strong Form for the static problem is introduced first as

$$
\begin{aligned}
(a) \quad & \nabla_S^T \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} \text{ on } \Omega \\
(b) \quad & \boldsymbol{\sigma} = \mathbf{C} \nabla_S \mathbf{u} \\
(c) \quad & \boldsymbol{\tau} \mathbf{n} = \bar{\mathbf{t}} \text{ on } \Gamma_t \\
(d) \quad & \vec{\mathbf{u}} = \vec{\bar{\mathbf{u}}} \text{ on } \Gamma_u
\end{aligned}
\tag{4.12}
$$

where σ is the stress matrix component from the second order stress tensor $\boldsymbol{\tau}$, and ∇_S is the symmetric gradient matrix operator [114]. The vectors $\mathbf{b}$ and $\mathbf{t}$ are the body force and traction force vectors, respectively, and $\mathbf{C}$ is the constitutive material matrix, and $\mathbf{u}$ is the displacement vector. Lastly, $\mathbf{n}$ is the unit normal with respect to the boundary surface. Equation (4.12) a is the equilibrium condition, stating the sum of the forces in the solid at any point must vanish. The superscript T indicates transposition. The stress matrix $\boldsymbol{\sigma}$ occurs due to the deformation of the body, and the vector $\mathbf{b}$ represents the body force, which can be used to model loads such as gravity or magnetic force. Equation (4.12)b contains two relationships. The first one is the implicit strain-displacement relations, between the strain matrix $\boldsymbol{\epsilon}$ and the displacement matrix $\mathbf{u}$ given as the kinematic equation

$$
\boldsymbol{\epsilon} = \nabla_s \mathbf{u}
\tag{4.13}
$$

The Constitutive Equation

$$
\boldsymbol{\sigma} = \mathbf{C} \boldsymbol{\epsilon}
\tag{4.14}
$$

describes the relationship between stress and strain matrix. The constitutive matrix $\mathbf{C}$ contains the material elastic constants, such as elastic modulus, shear modulus and Poisson's ratio. Two interesting properties for the constitutive matrix $\mathbf{C}$ is that it is always symmetric and positive-definite.

Equation (4.12)c describes the prescribed external tractions $\bar{\mathbf{t}}$ on the boundaries of the domain. This condition is also referred as natural boundary (Γ_t) condition. Equation (4.12)d represents a different kind of boundary condition, known as the prescribed

boundary (Γ_u) condition. Here, the prescribed displacements are imposed on the boundaries. There are a couple of important rules on defining boundary conditions. First, the displacement and traction cannot be both imposed on the same portion of the boundary; which is stated as

$$\Gamma_t \cap \Gamma_u = 0 \tag{4.15}$$

Second, on any portion of the boundary, either prescribed or natural boundaries must be prescribed, given as

$$\Gamma_t \cup \Gamma_u = \Gamma \tag{4.16}$$

To transform the Strong Form in Equation (4.12) to the Weak Form, a weight function $\mathbf{w}$ is introduced. One requirement for the admissible weight function is that it has to vanish on the domain boundary where $\mathbf{u}$ is prescribed, written as

$$\mathbf{w} = \mathbf{0} \text{ on } \Gamma_u \tag{4.17}$$

In the method of weighted residuals, the weight function w is multiplied by Equation (4.12)a and (4.12)c and then both products are individually integrated over the domain Ω. Applying Green's theorem and after some algebraic manipulation, the Weak Form is obtained as

$$\int_\Omega (\nabla_s \mathbf{w})^\mathrm{T} \mathbf{C} \nabla_s \mathbf{u} \mathrm{d}\Omega = \int_{\Gamma_t} \mathbf{w}^\mathrm{T} \bar{\mathbf{t}} \mathrm{d}\Gamma + \int_\Omega \mathbf{w}^\mathrm{T} \mathbf{b} \mathrm{d}\Omega \tag{4.18}$$

At this point, the dimension of the domain has not been specified so that the aforementioned derivation from Strong to Weak Form can be applied to either two or three dimensions. The next step is to discretize the domain with finite elements. Since this thesis involves structural optimization in three dimensions, the corresponding displacement matrix $\mathbf{d}$ is defined as

$$\mathbf{d} = \begin{bmatrix} u_{x1} \ u_{y1} \ u_{z1} \ u_{x2} \ u_{y2} \ u_{z2} \ \cdots \ u_{xn_{np}} \ u_{yn_{np}} \ u_{zn_{np}} \end{bmatrix}^T \tag{4.19}$$

where n_{np} stands for the number of nodes in the finite element model. The shape function matrix $\mathbf{N^e}$ is defined for each element as

$$\mathbf{N^e} = \begin{bmatrix} N_1^e & 0 & 0 & N_2^e & 0 & 0 & \cdots & N_{n_{en}}^e & 0 & 0 \\ 0 & N_1^e & 0 & 0 & N_2^e & 0 & \cdots & 0 & N_{n_{en}}^e & 0 \\ 0 & 0 & N_1^e & 0 & 0 & N_2^e & \cdots & 0 & 0 & N_{n_{en}}^e \end{bmatrix} \tag{4.20}$$

where $N_{n_{en}}^e$ represents the shape function for the corresponding node in the element, where its purpose is to approximate a smooth solution field in each element. The subscript n_{en} defines the number of node in each element. The resulting elemental displacement matrix and weight function matrix are, respectively, approximated by the shape function on each element domain Ω_e as

$$\begin{aligned} \mathbf{u}(x,y,z) &\approx \mathbf{u^e}(x,y,z) = \mathbf{N^e}(x,y,z)\mathbf{d^e} \\ \mathbf{w}^T(x,y,z) &\approx \mathbf{w^{eT}}(x,y,z) = \mathbf{w^{eT}}\mathbf{N^e}(x,y,z)^T \end{aligned} \tag{4.21}$$

where $\mathbf{d^e}$ and $\mathbf{w^e}$ represent the displacement and weight function matrices at the element level, respectively. Following discretization, Equation (4.18) is evaluated by summing all the elemental integrals as

$$\sum_{e=1}^{nel} \left\{ \int_{\Omega^e} (\mathbf{\nabla_s w^e})^T \mathbf{C^e} \mathbf{\nabla_s u^e} d\Omega - \int_{\Gamma_t^e} \mathbf{w^{eT}}\mathbf{\bar{t}}d\Gamma - \int_{\Omega^e} \mathbf{w^{eT}}\mathbf{b}d\Omega \right\} = 0 \tag{4.22}$$

In the above, nel is the number of finite elements that is used for discretization, and $\mathbf{C^e}$ is the material property matrix in the corresponding element. The approximation of strain matrix in (4.21) can be written in terms of the nodal displacements using Equation (4.13) as

$$\epsilon = \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \epsilon_{yz} \\ \epsilon_{xz} \\ \epsilon_{xy} \end{bmatrix} \approx \epsilon^{e} = \nabla_{s}u^{e} = \nabla_{s}N^{e}d^{e} = B^{e}d^{e} \tag{4.23}$$

where the matrix B^{e} is the strain-displacement matrix. The B^{e} matrix is also used to obtain derivatives of the weight function as

$$(\nabla_{s}w^{e})^{T} = (B^{e}w^{e})^{T} = w^{eT}B^{eT} \tag{4.24}$$

Recalling the symmetric gradient operator ∇_{s} in three dimension is written as

$$\nabla_{s}^{T} = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 & 0 & 0 & \dfrac{\partial}{\partial z} & \dfrac{\partial}{\partial y} \\ 0 & \dfrac{\partial}{\partial y} & 0 & \dfrac{\partial}{\partial z} & 0 & \dfrac{\partial}{\partial x} \\ 0 & 0 & \dfrac{\partial}{\partial z} & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} & 0 \end{bmatrix} \tag{4.25}$$

the strain-displacement matrix is therefore written as

$$B^{e} = \nabla_{s}N^{e} =$$

$$\begin{bmatrix} \dfrac{\partial N_1^e}{\partial x} & 0 & 0 & \dfrac{\partial N_2^e}{\partial x} & 0 & 0 & \dots & \dfrac{\partial N_{n_{en}}^e}{\partial x} & 0 & 0 \\[2ex] 0 & \dfrac{\partial N_1^e}{\partial y} & 0 & 0 & \dfrac{\partial N_2^e}{\partial y} & 0 & \dots & 0 & \dfrac{\partial N_{n_{en}}^e}{\partial y} & 0 \\[2ex] 0 & 0 & \dfrac{\partial N_1^e}{\partial z} & 0 & 0 & \dfrac{\partial N_2^e}{\partial z} & \dots & 0 & 0 & \dfrac{\partial N_{n_{en}}^e}{\partial z} \\[2ex] 0 & \dfrac{\partial N_1^e}{\partial z} & \dfrac{\partial N_1^e}{\partial y} & 0 & \dfrac{\partial N_2^e}{\partial z} & \dfrac{\partial N_2^e}{\partial y} & \dots & 0 & \dfrac{\partial N_{n_{en}}^e}{\partial z} & \dfrac{\partial N_{n_{en}}^e}{\partial y} \\[2ex] \dfrac{\partial N_1^e}{\partial z} & 0 & \dfrac{\partial N_1^e}{\partial x} & \dfrac{\partial N_2^e}{\partial z} & 0 & \dfrac{\partial N_2^e}{\partial x} & \dots & \dfrac{\partial N_{n_{en}}^e}{\partial z} & 0 & \dfrac{\partial N_{n_{en}}^e}{\partial x} \\[2ex] \dfrac{\partial N_1^e}{\partial y} & \dfrac{\partial N_1^e}{\partial x} & 0 & \dfrac{\partial N_2^e}{\partial y} & \dfrac{\partial N_2^e}{\partial x} & 0 & \dots & \dfrac{\partial N_{n_{en}}^e}{\partial y} & \dfrac{\partial N_{n_{en}}^e}{\partial x} & 0 \end{bmatrix} \tag{4.26}$$

Equations (4.21), (4.23) and (4.24) are substituted into Equation (4.22), and then following some algebraic manipulation, we obtain

$$KU = F \tag{4.27}$$

Note that Equation (4.27) is one of the design constraints in SIMP method in Equation (4.10). The global stiffens matrix $\mathbf{K}$ and global force vector $\mathbf{F}$ are assembled through summation of the elemental matrices, respectively written as

$$\mathbf{k_e} = \mathbf{K^e} = \int_{\Omega^e} \mathbf{B^{eT}C^eB^e} d\Omega \tag{4.28}$$

and

$$\mathbf{F^e} = \int_{\Omega^e} \mathbf{N^{eT}b} d\Omega + \int_{\Gamma_t^e} \mathbf{N^{eT}\bar{t}} d\Gamma \tag{4.29}$$

To solve for the unknown nodal values and reaction forces, Equation (4.27) is partitioned as

$$\begin{bmatrix} \mathbf{K_{FF}} & \mathbf{K_{EF}^T} \\ \mathbf{K_{EF}} & \mathbf{K_{EE}} \end{bmatrix} \begin{bmatrix} \mathbf{U_F} \\ \mathbf{U_E} \end{bmatrix} = \begin{bmatrix} \mathbf{F_F} \\ \mathbf{F_E} \end{bmatrix} \tag{4.30}$$

where subscript $\mathbf{E}$ represents the nodes that are prescribed an essential boundary condition, and subscript $\mathbf{F}$ represents the free degrees of freedom. The unknown displacement vector $\mathbf{U_F}$ is solved first, and then the reaction forces $\mathbf{F_E}$ are solved using the second row of the matrix equation.

4.5. Three Dimensional Isoparametric Element

Isoparametric elements are widely used in FEA primarily since they are capable of representing irregular shaped domains and curved boundaries. In an isoparametric element, the spatial location within the element is mapped to a master or reference element where integrations over the element are more easily performed. To integrate terms in the weak form over the reference domain, the numerical integration method of Gauss Quadrature is commonly employed. Gauss Quadrature evaluates the integrand at predefined Gauss points in the master element [115] to perform the integration. For the three dimensional FEA in

this research, the 8-node hexahedral isoparametric element is selected for domain

discretization. The geometry of the element is shown in

Figure 4.1, where each coordinate that defines the master element takes on values

from -1 to 1, which is the domain of integration in Gauss Quadrature. Typical shape

functions for the element are given as

$$
\mathbf{N^e} = \frac{1}{8}
\begin{Bmatrix}
(1 - \xi)(1 - \eta)(1 - \zeta) \\
(1 + \xi)(1 - \eta)(1 - \zeta) \\
(1 + \xi)(1 + \eta)(1 - \zeta) \\
(1 - \xi)(1 + \eta)(1 - \zeta) \\
(1 - \xi)(1 - \eta)(1 + \zeta) \\
(1 + \xi)(1 - \eta)(1 + \zeta) \\
(1 + \xi)(1 + \eta)(1 + \zeta) \\
(1 - \xi)(1 + \eta)(1 + \zeta)
\end{Bmatrix}
\tag{4.31}
$$

The shape functions interpolate nodal values within the element and thus provide an

approximate solution field over the master elemental domain. Shape functions must satisfy

the partition of unity requirement given as

$$
\sum_{i}^{n} \mathbf{N_i^e}(\mathbf{\Xi}) = 1 \,,
\tag{4.32}
$$

And

$$
\mathbf{N_i^e}(\mathbf{\Xi_j}) = \delta_{ij} =
\begin{cases}
1 \text{ if } i = j \\
0 \text{ if } i \neq j
\end{cases}
\tag{4.33}
$$

which states that the sum of the shape function values at any point in the domain has to

equal to one, and the shape function value at their corresponding node equals to one and

zero at any other nodes. The vector $\mathbf{\Xi}$ in Equation (4.32) and (4.33) contains the coordinate

vectors such that $\mathbf{\Xi} = \{\xi, \eta, \zeta\}$.
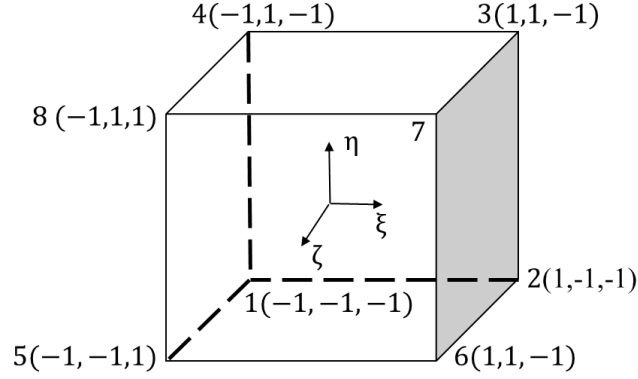
Figure 4.1. 8-node isoparametric element, with elemental node numbers and master coordinates axes

To integrate the elemental stiffness matrix $\mathbf{k_e}$, we approximate the analytical form using Gauss Quadrature integration as

$$\mathbf{k_e}(\theta_e) = \iiint\limits_{-1}^{1} \mathbf{B}^{eT}\mathbf{C}^{e\prime}(\theta_e)\mathbf{B}^e d\Omega$$

$$\approx \sum_{i=1}^{n_{gp}} \sum_{j=1}^{n_{gp}} \sum_{k=1}^{n_{gp}} \left(W_i W_j W_k \mathbf{B}^e\left(\xi_i, \eta_j, \zeta_k\right)^T \mathbf{C}^{e\prime}(\theta_e)\mathbf{B}^e\left(\xi_i, \eta_j, \zeta_k\right)\left|\mathbf{J}\left(\xi_i, \eta_j, \zeta_k\right)\right| \right)$$

(4.34)

where W_i, W_j and W_k are the Gauss Quadrature weights, one for each coordinate direction. Since the stiffness matrix is integrated in three dimensions, we need a Gauss weight for each coordinate i, j, k to integrate over the master element domain. The value of the Gauss weight depends on the number of integration point n_{gp} that is used (Note that we have assumed the same number of Gauss points in all three directions). The matrix $\mathbf{B}^e$ is the strain-displacement matrix and $\mathbf{J}$ is the Jacobian matrix for the transformation from the reference element to the actual element. The dimensions of the matrices in the above equations are as follows: $\mathbf{B}^e$ is 6x24, $\mathbf{C}^e$ is 6x6, and the elemental stiffens matrix $\mathbf{k_e}$ is 24x24. Table 4.1 shows the Gauss point coordinates and weight values for various numbers

of Gauss quadrature points. The research performed here uses $n_{gp} = 2$ point Gauss quadrature in all three master element coordinate directions.

The Jacobian matrix $\mathbf{J}(\xi_i, \eta_j, \zeta_k)$ in Equation (4.35) is evaluated as

$$
[\mathbf{J}] =
\begin{bmatrix}
\dfrac{\partial \mathbf{x}}{\partial \xi} & \dfrac{\partial \mathbf{y}}{\partial \xi} & \dfrac{\partial \mathbf{z}}{\partial \xi} \\[6pt]
\dfrac{\partial \mathbf{x}}{\partial \eta} & \dfrac{\partial \mathbf{y}}{\partial \eta} & \dfrac{\partial \mathbf{z}}{\partial \eta} \\[6pt]
\dfrac{\partial \mathbf{x}}{\partial \zeta} & \dfrac{\partial \mathbf{y}}{\partial \zeta} & \dfrac{\partial \mathbf{z}}{\partial \zeta}
\end{bmatrix}
$$

$$
=
\begin{bmatrix}
\dfrac{\partial N_1^e}{\partial \xi} & \dfrac{\partial N_2^e}{\partial \xi} & \dfrac{\partial N_3^e}{\partial \xi} & \dfrac{\partial N_4^e}{\partial \xi} & \dfrac{\partial N_5^e}{\partial \xi} & \dfrac{\partial N_6^e}{\partial \xi} & \dfrac{\partial N_7^e}{\partial \xi} & \dfrac{\partial N_8^e}{\partial \xi} \\[6pt]
\dfrac{\partial N_1^e}{\partial \eta} & \dfrac{\partial N_2^e}{\partial \eta} & \dfrac{\partial N_3^e}{\partial \eta} & \dfrac{\partial N_4^e}{\partial \eta} & \dfrac{\partial N_5^e}{\partial \eta} & \dfrac{\partial N_6^e}{\partial \eta} & \dfrac{\partial N_7^e}{\partial \eta} & \dfrac{\partial N_8^e}{\partial \eta} \\[6pt]
\dfrac{\partial N_1^e}{\partial \zeta} & \dfrac{\partial N_2^e}{\partial \zeta} & \dfrac{\partial N_3^e}{\partial \zeta} & \dfrac{\partial N_4^e}{\partial \zeta} & \dfrac{\partial N_5^e}{\partial \zeta} & \dfrac{\partial N_6^e}{\partial \zeta} & \dfrac{\partial N_7^e}{\partial \zeta} & \dfrac{\partial N_8^e}{\partial \zeta}
\end{bmatrix}
\begin{bmatrix}
x_1^e & y_1^e & z_1^e \\
x_2^e & y_2^e & z_2^e \\
x_3^e & y_3^e & z_3^e \\
x_4^e & y_4^e & z_4^e \\
x_5^e & y_5^e & z_5^e \\
x_6^e & y_6^e & z_6^e \\
x_7^e & y_7^e & z_7^e \\
x_8^e & y_8^e & z_8^e
\end{bmatrix}
\tag{4.35}
$$

The first matrix in Equation (4.35) represent the derivatives of the physical coordinates with respect to the master element coordinates. In the final expression in Equation (4.35), the first matrix contains the derivatives of the shape function with respect to the master elemental coordinates, and the matrix on the right side contains the coordinates of the element nodes in three dimensions.

Table 4.1. Weight values and evaluation points for different number of Gauss Quadrature

Number of Gauss Points	W_i	Evaluation base points
1	2	0
2	1	$\dfrac{1}{\sqrt{3}}, -\dfrac{1}{\sqrt{3}}$
3	$\dfrac{5}{9}, \dfrac{8}{9}, \dfrac{5}{9}$	$\dfrac{\sqrt{3}}{5}, 0, -\dfrac{\sqrt{3}}{5}$

4.6.   Constitutive Material Model

To evaluate the stiffness matrix in Equation (4.34), we must compute the material property matrix $\mathbf{C}^{e\prime}$ which includes the constitutive model for the material in the structure. The matrix $\mathbf{C}^{e\prime}$ is the constitutive material matrix after the rotation, and this second order tensor contains the original constitutive material tensor and the transformational matrix. Because of the rotation of the material orientation in each finite element, a matrix transformation is needed to calculate the material properties with respect to the global coordinate.

Here we only consider a single rotation θ about the z-axis when defining material properties. To derive the transformed material model $\mathbf{C}^{e\prime}$, a coordinate transformation is defined that relates the three dimensional Cartesian system (xyz) to the coordinate system $(x\prime y\prime z\prime)$, offset counterclockwise by an angle value θ about the z axis as shown in Figure 4.2. As shown, the coordinate location P positioned in the (xy) (or $(x\prime y\prime)$) plane, with coordinates (u, v, w) or $(u\prime v\prime w\prime)$ which are related through

$$\begin{bmatrix} u\prime \\ v\prime \\ w\prime \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{4.36}$$

or similarly

$$\begin{bmatrix} x\prime \\ y\prime \\ z\prime \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4.37}$$
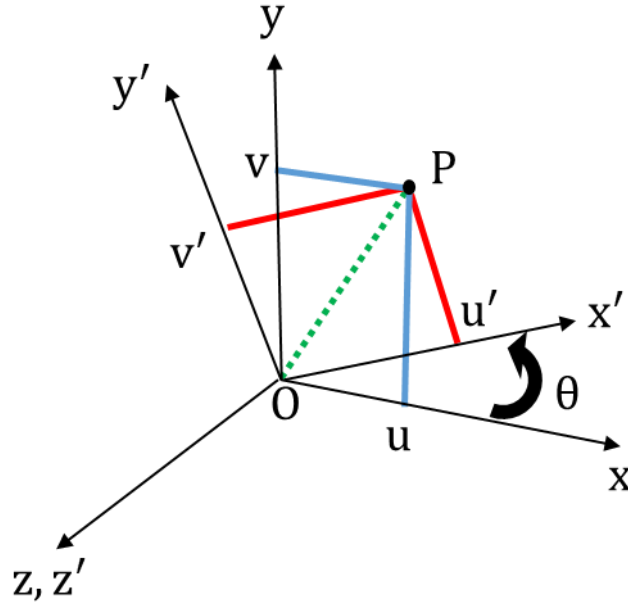
Figure 4.2. Three dimensional coordinate transformation according to z axis

In three dimensions, there are six components of strains in both the original and rotated coordinate systems, three normal strains and three shear strains each. The strains of vector OP in the original coordinate system are.

$$\epsilon_x = \frac{\partial u}{\partial x}, \epsilon_y = \frac{\partial v}{\partial y}, \epsilon_z = \frac{\partial w}{\partial z}$$

$$\gamma_{yz} = 2\epsilon_{yz} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}, \gamma_{xz} = 2\epsilon_{xz} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}, \gamma_{xy} = 2\epsilon_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$$

(4.38)

and similarly, for rotated coordinate system we write

$$\epsilon'_x = \frac{\partial u'}{\partial x'}, \epsilon'_y = \frac{\partial v'}{\partial y'}, \epsilon'_z = \frac{\partial w'}{\partial z'}$$

$$\gamma'_{yz} = 2\epsilon'_{yz} = \frac{\partial v'}{\partial z'} + \frac{\partial w'}{\partial y'}, \gamma'_{xz} = 2\epsilon'_{xz} = \frac{\partial w'}{\partial x'} + \frac{\partial u'}{\partial z'}, \gamma'_{xy} = 2\epsilon'_{xy} = \frac{\partial u'}{\partial y'} + \frac{\partial v'}{\partial x'}$$

(4.39)

When transforming the shear strain, the tensor strain ϵ_{ij} or ϵ'_{ij} is used instead of the engineering shear strain γ_{ij} or γ'_{ij}. To write the strains in xyz in terms of the strains in x'y'z', the chain rule is used to establish the relationship. For example, for the normal strain along the x-coordinate written in terms of the $x'y'z'$ axes is

$$\epsilon_x = \frac{\partial u}{\partial x'}\frac{\partial x'}{\partial x} + \frac{\partial u}{\partial y'}\frac{\partial y'}{\partial x} + \frac{\partial u}{\partial z'}\frac{\partial z'}{\partial x} \tag{4.40}$$

Rewriting Equation (4.36) gives

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} \tag{4.41}$$

where it follows that

$$\left. \begin{aligned} \frac{\partial u}{\partial i'} &= \frac{\partial u'}{\partial i'}\cos(\theta) - \frac{\partial v'}{\partial i'}\sin(\theta) \\ \frac{\partial v}{\partial i'} &= \frac{\partial u'}{\partial i'}\sin(\theta) + \frac{\partial v'}{\partial i'}\cos(\theta) \\ \frac{\partial w}{\partial i'} &= \frac{\partial w'}{\partial i'} \end{aligned} \right\} \quad i = x, y, z \tag{4.42}$$

Furthermore, from Equation (4.37), the derivative of the rotated coordinate axes with respect to the original coordinate axes can be written as,

$$\begin{aligned} \frac{\partial x'}{\partial x} &= \cos(\theta) & \frac{\partial x'}{\partial y} &= \sin(\theta) & \frac{\partial x'}{\partial z} &= 0 \\ \frac{\partial y'}{\partial x} &= -\sin(\theta) & \frac{\partial y'}{\partial y} &= \cos(\theta) & \frac{\partial y'}{\partial z} &= 0 \\ \frac{\partial z'}{\partial x} &= 0 & \frac{\partial z'}{\partial y} &= 0 & \frac{\partial z'}{\partial z} &= 1 \end{aligned} \tag{4.43}$$

From Equation (4.39), (4.42) and (4.43), Equation (4.40) becomes

$$\epsilon_x = \cos(\theta)^2\epsilon'_x - 2\cos(\theta)\sin(\theta)\epsilon_{xy} + \sin(\theta)^2\epsilon_y \tag{4.44}$$

Expression for ϵ_y, ϵ_z, ϵ_{yz}, ϵ_{xz}, and ϵ_{xy} can be derived in a similar manner, where the resulting strain transformation is

$$\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \varepsilon_{yz} \\ \varepsilon_{xz} \\ \varepsilon_{xy} \end{bmatrix} = \begin{bmatrix} \cos(\theta)^2 & \sin(\theta)^2 & 0 & 0 & 0 & -2\sin(\theta)\cos(\theta) \\ \sin(\theta)^2 & \cos(\theta)^2 & 0 & 0 & 0 & 2\sin(\theta)\cos(\theta) \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 0 & -\sin(\theta) & \cos(\theta) & 0 \\ \cos(\theta)\sin(\theta) & -\cos(\theta)\sin(\theta) & 0 & 0 & 0 & \cos(\theta)^2 - \sin(\theta)^2 \end{bmatrix} \begin{bmatrix} \varepsilon'_x \\ \varepsilon'_y \\ \varepsilon'_z \\ \varepsilon'_{yz} \\ \varepsilon'_{xz} \\ \varepsilon'_{xy} \end{bmatrix} \tag{4.45}$$

Here we the transformational matrix $\mathbf{T}(\theta)$ such that it's inverse is given as

$$\mathbf{T^{-1}}(\theta) =$$

$$\begin{bmatrix} \cos(\theta)^2 & \sin(\theta)^2 & 0 & 0 & 0 & -2\sin(\theta)\cos(\theta) \\ \sin(\theta)^2 & \cos(\theta)^2 & 0 & 0 & 0 & 2\sin(\theta)\cos(\theta) \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 0 & -\sin(\theta) & \cos(\theta) & 0 \\ \cos(\theta)\sin(\theta) & -\cos(\theta)\sin(\theta) & 0 & 0 & 0 & \cos(\theta)^2 - \sin(\theta)^2 \end{bmatrix} \quad (4.46)$$

to give $\boldsymbol{\epsilon} = \mathbf{T^{-1}}(\boldsymbol{\theta})\boldsymbol{\epsilon'}.$

Once the transformation matrix $\mathbf{T}(\theta)$ is defined, the stress-strain relationship in the rotated coordinate system can evaluated from

$$\begin{bmatrix} \sigma'_x \\ \sigma'_y \\ \sigma'_z \\ \sigma'_{yz} \\ \sigma'_{xz} \\ \sigma'_{xy} \end{bmatrix} = C \begin{bmatrix} \epsilon'_x \\ \epsilon'_y \\ \epsilon'_z \\ 2\epsilon'_{yz} \\ 2\epsilon'_{xz} \\ 2\epsilon'_{xy} \end{bmatrix} = C \begin{bmatrix} \epsilon'_x \\ \epsilon'_y \\ \epsilon'_z \\ \gamma'_{yz} \\ \gamma'_{xz} \\ \gamma'_{xy} \end{bmatrix} = \quad (4.47)$$

To accommodate the engineering shear strains for transformation, a matrix $\mathbf{R}$ containing diagonal terms is employed which is defined as

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \quad (4.48)$$

Applying the transformational matrix and matrix $\mathbf{R}$ into Equation (4.48), we obtain,

$$\mathbf{T}(\theta) \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{bmatrix} = \mathbf{CRT}(\theta) \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \epsilon_{yz} \\ \epsilon_{xz} \\ \epsilon_{xy} \end{bmatrix} \quad (4.49)$$

Pre-multiply Equation 4.48 by $\mathbf{T}^{-1}(\theta)$ and substituting $\mathbf{R}^{-1}$ as needed, Equation (4.49) is

becomes

$$
\begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{bmatrix} = \mathbf{T}^{-1}(\theta)\mathbf{C}\mathbf{R}\mathbf{T}(\theta)\mathbf{R}^{-1} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{bmatrix} \tag{4.50}
$$

Furthermore, it can be shown that

$$
\mathbf{T}^{-T}(\theta) = \mathbf{R}\mathbf{T}(\theta)\mathbf{R}^{-1} \tag{4.51}
$$

Eventually, the stress and strain vectors in the original coordinate system can be written as

shown in Equation (4.52).

$$
\begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{bmatrix} = \mathbf{T}^{-1}(\theta)\mathbf{C}\mathbf{T}^{-T}(\theta) \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{bmatrix} \rightarrow \boldsymbol{\sigma} = \mathbf{C}'(\theta)\boldsymbol{\epsilon} \tag{4.52}
$$

The matrix $\mathbf{C}'(\theta)$ is therefore the transformed constitutive material matrix that is used in

Equation (4.34), in terms of the elemental material equations.

The last piece of information that is needed to execute the FEA in this thesis is the

material elastic constants for defining the original constitutive material matrix $\mathbf{C}$. Here we

model each discretized element as a unidirectional fiber composite with a transverse

isotropic property [116]. Therefore, five material constants are needed, namely E_x (elastic

stiffness along the fiber axis), E_y (transverse elastic stiffness of the fiber), G_{xy} (shear

modulus of the fiber in the x-y plane)), υ_{xy} (Poisson's ratio of the fiber in the x-y plane)

and υ_{yz} (Poisson's ratio of the fiber in the y-z plane) [117] which compose the constitutive

material matrix $\mathbf{C}$ written as

$$\mathbf{C} = \begin{bmatrix} \dfrac{1}{E_x} & -\dfrac{\upsilon_{xy}}{E_x} & -\dfrac{\upsilon_{xy}}{E_x} & 0 & 0 & 0 \\[2mm] -\dfrac{\upsilon_{xy}}{E_x} & \dfrac{1}{E_y} & -\dfrac{\upsilon_{yz}}{E_y} & 0 & 0 & 0 \\[2mm] -\dfrac{\upsilon_{xy}}{E_x} & -\dfrac{\upsilon_{yz}}{E_y} & \dfrac{1}{E_y} & 0 & 0 & 0 \\[2mm] 0 & 0 & 0 & \dfrac{2(1+\upsilon_{yz})}{E_y} & 0 & 0 \\[2mm] 0 & 0 & 0 & 0 & \dfrac{1}{G_{xy}} & 0 \\[2mm] 0 & 0 & 0 & 0 & 0 & \dfrac{1}{G_{xy}} \end{bmatrix}^{-1} \tag{4.53}$$

4.7.   *Design Sensitivity and Filtering*

Design sensitivity analysis is required for the solution of the topology optimization problem. Design sensitivities quantify the relationship between changes in the design variables and changes in the objective and constraint functions. In this research, sensitivities with respect to both the elemental densities and material orientation angle design variables are needed for the gradient-based optimization algorithm. There are various methods employed for obtaining design sensitivities which are considered in the text to follow. Sensitivity filtering techniques discussed in Chapter Two is also discussed below.

A common method for evaluating design sensitivities is the Finite Difference method which is easily derived from a Taylor series expansion [115]. Considering a small perturbation $\Delta \mathbf{x}_i$ in a function $F(x)$, the derivative $\nabla_i F(\mathbf{x})$ may be approximated using the first-order Taylor Expansion

$$F(\mathbf{x} + \Delta \mathbf{x}_i) = F(\mathbf{x}) + \nabla_i F(\mathbf{x}) * \Delta \mathbf{x}_i + o(\Delta x_i^2) \tag{4.54}$$

where $\nabla_i F(\mathbf{x})$ is the i-th component of the gradient $\nabla F(\mathbf{x})$. The term $o(\Delta x_i^2)$ is the truncation error of approximation where the square superscript indicates by taking half of

the original step Δx_i, the error will be approximately quartered. The vector $\Delta \mathbf{x}_i$ is defined in terms of a perturbation Δx_i of the i-th component written as

$$\Delta \mathbf{x}_i = [0,0,0, \dots ,,0,0, \Delta x_i, 0,0, \dots ,0,0,0,] \tag{4.55}$$

There are many finite difference schemes that can be used where it is common to use the forward finite difference, which it is obtained by rearranging Equation (4.45) as

$$\nabla_i F(\mathbf{x}) = \frac{F(\mathbf{x} + \Delta \mathbf{x}_i) - F(\mathbf{x})}{\Delta x_i} + o(\Delta x_i) \tag{4.56}$$

When the Taylor series is expanded with a backward perturbation, Equation (4.57) then becomes

$$F(\mathbf{x} - \Delta \mathbf{x}_i) = F(\mathbf{x}) - \nabla_i F(\mathbf{x}) * \Delta \mathbf{x}_i + o(\Delta x_i^2) \tag{4.57}$$

which yields the backward finite difference,

$$\nabla_i F(\mathbf{x}) = \frac{F(\mathbf{x}) - F(\mathbf{x} - \Delta \mathbf{x}_i)}{\Delta x_i} + o(\Delta x_i) \tag{4.58}$$

There is also the central finite difference formula, which is derived by averaging the backward and forward finite difference formulas yielding

$$\nabla_i F(\mathbf{x}) = \frac{F(\mathbf{x} + \Delta \mathbf{x}_i) - F(\mathbf{x} - \Delta \mathbf{x}_i)}{2\Delta x_i} + o(\Delta x_i^2) \tag{4.59}$$

Note that $o(\Delta x_i)$ in each Equation (4.56) and (4.58) represent the order of accuracy if the step size is changed. The finite difference method is relatively easy to implement; however, errors are introduced depending on the size of the perturbation $\Delta \mathbf{x}_i$. More importantly, finite difference method requires an additional FEA solution for each design variable which can be computationally prohibitive in case of large number of design variables. Therefore, faster and more accurate method will be adopted using analytical derivation.

There are two analytical methods developed to calculate the design sensitivities which are commonly implemented with the finite element method: the Direct

Differentiation Method and the Adjoint Variable Method [118]. The Direct Differention method calculates the sensitivity by directly taking the derivative of the finite element equations with respect to the function's implicit and explicit design variables. The Adjoint Variable method employs a Lagrange multiplier to solve for the design sensitivity. Comparing the two methods, if the number of design variables is greater than the number of performance measures (i.e., objective function plus implicit constraints), the Adjoint Variable Method is preferred, and vice versa. For this research, there is only one implicit objective function, no implicit constraints, and we expect many design variables, so the Adjoint Variable Method is justified for our design sensitivities.

To derive the design sensitivity with respect to the density design variable, the material orientation variable is assumed to be constant. We consider the objective function defined in Equation (4.60) defined in terms of the design variable vector $\boldsymbol{\phi} = (\boldsymbol{\rho}, \boldsymbol{\theta})$ where $\boldsymbol{\rho}$ and $\boldsymbol{\theta}$ are as defined in Equation (4.9) above. Both $\bar{c}(\boldsymbol{\rho})$ and $c(\mathbf{U}(\boldsymbol{\phi}), \boldsymbol{\phi})$ are equivalent expressions for the objective function. The latter expression recognizes the implicit dependence of c on the design variable through the displacement vector. In the Adjoint Variable Method, the objective function is added to the product of an arbitrary Lagrange Multiplier $\boldsymbol{\lambda}$ and the static equilibrium equation to obtain the augmented function $\bar{c}$ as

$$\bar{c}(\boldsymbol{\phi}) = c(\mathbf{U}(\boldsymbol{\phi}), \boldsymbol{\phi}) - \boldsymbol{\lambda}(\boldsymbol{\phi}) * (\mathbf{K}(\boldsymbol{\phi})\mathbf{U}(\boldsymbol{\phi}) - \mathbf{F}(\boldsymbol{\phi})) \qquad (4.60)$$

Taking the derivative of Equation (4.60) with respect to the density design variable ρ_e yields

$$\frac{D\bar{c}}{D\rho_e} = \frac{\partial c}{\partial \rho_e} + \frac{\partial c}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial \rho_e} - \frac{D\boldsymbol{\lambda}}{D\rho_e} * (\mathbf{KU} - \mathbf{F}) - \boldsymbol{\lambda} * (\frac{D\mathbf{K}}{D\rho_e}\mathbf{U} + \mathbf{K}\frac{D\mathbf{U}}{D\rho_e} - \frac{D\mathbf{F}}{D\rho_e}) \qquad (4.61)$$

where the arguments have been omitted for conciseness. Since the static equilibrium is

assumed to be satisfied, the derivative term $\frac{D\lambda}{D\rho_e}$ may be eliminated by first rearranging

Equation (4.61) as

$$\frac{D\bar{c}}{D\rho_e} = \frac{\partial c}{\partial \rho_e} + \frac{\partial \mathbf{U}}{\partial \rho_e}\left(\frac{\partial c}{\partial \mathbf{U}} - \mathbf{K}\lambda\right) - \lambda * \left(\frac{D\mathbf{K}}{D\rho_e}\mathbf{U} - \frac{D\mathbf{F}}{D\rho_e}\right) \tag{4.62}$$

Since $\boldsymbol{\lambda}$ is arbitrary, it can be defined to eliminate the term $\frac{\partial \mathbf{U}}{\partial \rho_e}$ by setting its coefficient to

zero to obtain

$$\frac{\partial c}{\partial \mathbf{U}} - \mathbf{K}\lambda = 0 \tag{4.63}$$

The compliance objective function given above can be expressed in terms of the global

displacement $\mathbf{U}$ and global force vector $\mathbf{F}$ as

$$c = \mathbf{U}^\mathbf{T}\mathbf{F} \tag{4.64}$$

The Adjoint Variable evaluation process is summarized as

$$\frac{\partial c}{\partial \mathbf{U}} - \mathbf{K}\lambda = 0 \rightarrow \frac{\partial \mathbf{UF}}{\partial \mathbf{U}} - \mathbf{K}\lambda = 0 \rightarrow \mathbf{F} = \mathbf{K}\lambda \rightarrow \lambda = \mathbf{U} \tag{4.65}$$

where we have assumed that the load F is not a function of design. In the above, the

Lagrange Multiplier vector is simply the global displacement vector $\mathbf{U}$ for the compliance

objective function. Substituting $\boldsymbol{\lambda} = \mathbf{U}$ into Equation (4.62) yields the desired design

sensitivity. Note that since there is no explicit relationship between the objective function

and a density variable ($\frac{\partial c}{\partial \rho_e} = 0$), as well as between global force vector and density

variable ($\frac{D\mathbf{F}}{D\rho_e} = 0$), the design sensitivity of the compliance with respect to a density design

variable becomes,

$$\frac{D\bar{c}}{D\rho_e} = -\mathbf{U}^\mathrm{T}\frac{D\mathbf{K}}{D\rho_e}\mathbf{U} \tag{4.66}$$

The foregoing expression can be conveniently translated into the FEA procedure. In the discretized domain, the derivative of the elemental compliance and stiffness with respect to the all the density variables are zero, except the single ρ_e that is associated with the element e. Therefore, using the expression of elemental stiffness in Equation (4.66), the design sensitivity of compliance c with respect to the density variable ρ_e is evaluated at the elemental level as

$$\frac{\partial c}{\partial \rho_e} = -p(\rho_e)^{p-1} \mathbf{u_e^T k_e}(\theta) \mathbf{u_e} \tag{4.67}$$

The design sensitivity for material orientation design variables can also be obtained with Adjoint Variable Method at the element level. Similar to Equation (4.67), the compliance sensitivity with respect to material orientation variable θ_e is given as

$$\frac{\partial c}{\partial \theta_e} = -(\rho_e)^p \mathbf{u_e^T} \frac{\partial \mathbf{k_e}(\theta_e)}{\partial \theta_e} \mathbf{u_e} \tag{4.68}$$

Unlike density design variables, the elemental stiffness matrix $\mathbf{k_e}$ depends on the material orientation variable θ_e Recalling Equations (4.34) and (4.52), the elemental stiffness matrix in Equation (4.6) is written for Gauss Quadrature evaluations as

$$\mathbf{k_e}(\theta_e) \approx \sum_{i=1}^{n_{gp}} \sum_{j=1}^{n_{gp}} \sum_{k=1}^{n_{gp}} \left(W_i W_j W_k \mathbf{B^e}(\xi_i, \eta_j, \zeta_k)^{\mathrm{T}} \mathbf{T}^{-1}(\theta_e) \mathbf{C} \mathbf{T}^{-\mathrm{T}}(\theta) \mathbf{B^e}(\xi_i, \eta_j, \zeta_k) |\mathbf{J}(\xi_i, \eta_j, \zeta_k)| \right) \tag{4.69}$$

where we note the dependence of $\mathbf{T}(\theta_e)$ on θ_e. Taking the derivative of Equation (4.69) with respect to material orientation variable in the usual manner yields

$$\begin{aligned}
\frac{\partial \mathbf{k_e}(\theta)}{\partial \theta_e} = \sum_{i=1}^{n_{gp}} \sum_{j=1}^{n_{gp}} \sum_{k=1}^{n_{gp}} &\left\{ W_i W_j W_k \mathbf{B}(\xi_i, \eta_j, \zeta_k)^{\mathrm{T}} \left(\frac{\partial \mathbf{T}(\theta_e)^{-1}}{\partial \theta_e} \mathbf{C} \mathbf{T}(\theta_e)^{-\mathrm{T}} \right. \right. \\
&\left. \left. + \mathbf{T}(\theta_e)^{-1} \mathbf{C} \frac{\partial \mathbf{T}(\theta_e)^{-\mathrm{T}}}{\partial \theta_e} \right) \mathbf{B}(\xi_i, \eta_j, \zeta_k) |\mathbf{J}(\xi_i, \eta_j, \zeta_k)| \right\}
\end{aligned} \tag{4.70}$$

By combining Equations (4.68) and (4.70), the design sensitivity for a material orientation design variable is

$$
\frac{\partial c}{\partial \theta_e} = \rho_e^p \mathbf{u}_e^T \left\{ \sum_{i=1}^{n_{gp}} \sum_{j=1}^{n_{gp}} \sum_{k=1}^{n_{gp}} \left\{ W_i W_j W_k \mathbf{B}\big(\xi_i, \eta_j, \zeta_k\big)^T \Big(\frac{\partial \mathbf{T}(\theta_e)^{-1}}{\partial \theta_e} \mathbf{C}\mathbf{T}(\theta_e)^{-T} \right. \right.
$$
$$
\left. \left. + \mathbf{T}(\theta_e)^{-1}\mathbf{C}\frac{\partial \mathbf{T}(\theta_e)^{-T}}{\partial \theta_e}\Big)\mathbf{B}\big(\xi_i, \eta_j, \zeta_k\big)\big|\mathbf{J}\big(\xi_i, \eta_j, \zeta_k\big)\big| \right\} \right\} \mathbf{u}_e
$$

(4.71)

A common issue with the SIMP method is the checker board effect resulting from a numerical issue in the topology optimization solution process [44]. To eliminate checker board patterns, design sensitivities of the objective function taken with respect to density variable are filtered using the linear weight average of neighboring element values as described in Equation (2.6) and (2.7), which is rewritten here as

$$
\frac{\partial \hat{c}}{\partial \rho_e} = \frac{\sum_{i \in N_e} \omega(x_i)\rho_i \frac{\partial c}{\partial \rho_i}}{\rho_e \sum_{i \in N_e} \omega(x_i)}, \qquad \omega(x_i) = r_{min} - \text{dist}(e, i)
$$

(4.72)

The variable $\text{dist}(e, i)$ is the center distance between the filtered element and the surrounding element. The value of r_{min} is set to 1.5 for all the computational examples in this thesis where we use a regular grid of cube element having sides with a length of 1 unit. The higher the value of r_{min}, the more surrounding elements are involved in the filtering process, and vice versa. As long as the radius us bigger than one in our work, the filtering process will take in effect. Otherwise, a radius value of less than one is equivalent to no filtering at all, and checkerboarding is likely to occur.

4.8.   *Optimization Solver*

To solve the compliance minimization problem in Equation (4.8) in this thesis, the Matlab optimization solver *fmincon* [119] is applied. The Matlab function *fmincon* is a

nonlinear, constraint-based solver which may be used to solve the general optimization problem stated in Equation (4.73). The design variable vector $\mathbf{x}$ contains parameters in the FE model that can be changed to obtain a better design as defined by the objective function and constraints. The constraints $c(\mathbf{x}) \leq 0$ and $ceq(\mathbf{x}) = 0$ represent the nonlinear inequality and equality constraints, respectively. In addition, linear inequality and equality constraints written respectively as $\mathbf{A} \cdot \mathbf{x} \leq b$ and $\mathbf{Aeq} \cdot \mathbf{x} = beq$ may be specified in the optimization problem. The general nonlinear constrained problem solved with *fmincon* may be written as

$$\min_{\mathbf{x}} f(\mathbf{x}) \begin{cases} c(\mathbf{x}) \leq 0 \\ ceq(\mathbf{x}) = 0 \\ \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b} \\ \mathbf{Aeq} \cdot \mathbf{x} = \mathbf{beq} \\ \mathbf{x_{lb}} \leq \mathbf{x} \leq \mathbf{x_{ub}} \end{cases} \tag{4.73}$$

where $f(\mathbf{x})$ is the objective function to be minimized. The vectors $\mathbf{x_{lb}}$ and $\mathbf{x_{ub}}$ are the lower and upper bounds for the design variables, respectively.

The syntax for running `fmincon` in Matlab is

$$[x,fval]=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options) \tag{4.74}$$

where the output vector `x` of length n (n being the number of design variables) is the design variable vector that is the solution to the optimization problem in Equation (4.73). The output `fval` is optimized objective value. There are more outputs available for the fmincon solver, and the readers can visit the website [119] for more information. In this research, the first n/2 components of the design variable vector `x` contains the density variables, and the remainder contains the material angle variables. The first input entry `fun` is a Matlab function or function M-file that evaluates the objective function and its

93

design sensitivities with respect to all of the design variables. The second input vector `x0`

contains the initial guesses for the design variables, and it is the same dimension as the

array `x`. The third input `A` and fourth input `b` define the linear inequality constraints. These

two inputs are replaced with empty brackets when there are no linear inequality constraints

such as in the optimization problems solved in this research.

The next two inputs `Aeq`, `beq` define the linear equality constraints, which we use

for volume constraint in Equation (4.10). These volume constraints are defined with

components of the vector `Aeq` taking values of 1 for the n/2 components and 0 for the

remainder of the array; `beq` is the volume constraint scalar that is user-defined fraction

ranges from 0 to 1. `lb` and `ub` are the lower and upper bound vectors for each design

variable, respectively. The entry identified as nonlcon is also user-defined Matlab function

or function M-file that calculates four outputs: the nonlinear inequality vector **c**, the

nonlinear equality vector **ceq**, the gradients of the nonlinear inequality vector **Gc**, and the

gradients of the nonlinear equality vector **Geq**. The nonlcon willl be used for two of the

examples we are demonstrating.

Finally, the `options` variable allows the user to specify parameters for the

optimization solver in order to replace default values. Example parameters include the type

of method for optimization, the tolerance for convergence criteria, the option to plot the

iteration history, the option to include design sensitivities and more.

Note that `fmincon` has several optimization algorithms available to the user, and

the *interior-point* method is selected for the purpose of this research. The *interior-point*

method solves the constrained nonlinear problem by introducing extra barrier functions

into the objective function, and in each iteration the algorithm uses either the direct step,

based on Karush-Kuhn-Tucker condition, or the conjugate gradient step, based on a radius of trust region, to solve for the optimum. For more specific information, the reader can visit the Matlab website [119].

Here we summarize the topology optimization process that includes orthotropic materials for modeling parts produced with the FFF process. For the discussion to follow, refer to Figure 4.3 which shows a flow chart of the optimization process. First, the design domain and boundary conditions are defined, and the domain is discretized into three dimensional hexagonal finite elements. Second, the FEA procedure is performed to calculate the displacement vector of nodal displacements in the x-, y-, and z-directions. Third, the objective function compliance and its design sensitivities are evaluated on an element-by-element basis. Fourth, the design sensitivity with respect to the density design variables are filtered using a linear weight average function. Fifth, the objective function and the design sensitivities serve as input the optimization solver *fmincon* which determines a new set of design variable values. If the relative changes of design variables and objective function values between two iterations are smaller than 0.1%, the algorithm stops the iteration process and the outputs the solution; otherwise, the iteration continues until the convergence criteria is met. The computer used for the example optimizations in the next chapter has a i7-4790 CPU processor at 3.60GHz, with a RAM of 32.0GB.

During the iteration process, the elements that have density value larger than 0.5 are plotted to illustrate the evolution of the topology, and the corresponding material orientation is also plotted inside each element to show their orientation states. Darker color of the element indicates that that element has a higher density. In addition, the computational time, the number of iterations and the computed compliance are recorded.
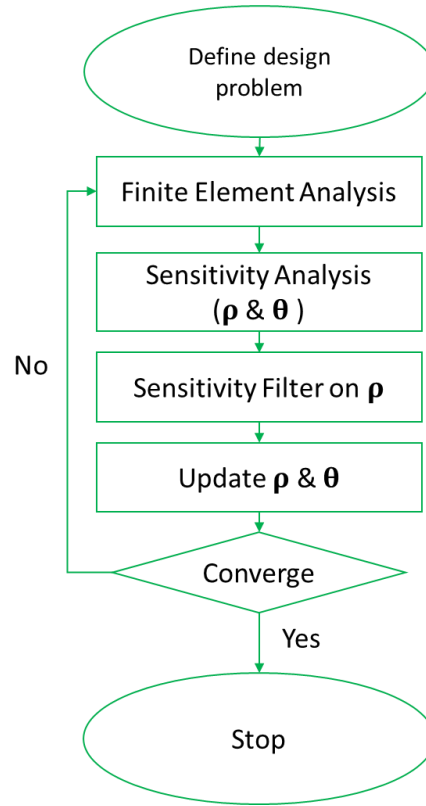
Figure 4.3. Flow chart demonstrating the process of minimizing the compliance

4.9.   Fiber Orientation and Elastic Constants calculation

In this section, the relationship between fiber orientation and material elastic constants are presented. The intent here is not to develop a fiber orientation modeling approach, or to provide a comprehensive review of the topic. Instead, this section simply gives a brief background on how fiber orientation is modeled in a flow field, such as the extrusion flow in the FFF process, and a procedure that has been used to obtain realistic elastic constants for an extrudate formed during the FFF process It is important to note here that Heller, et al. [120,121] provides a detailed study of FFF nozzle flow, extrudate swell, and the resulting fiber orientation. This section only to gives a brief overview of Heller's work and is limited to material related to this thesis, namely, the calculation of elastic constants in an FFF bead which we use for optimization. When a CFF composite is being

96

extruded out of the nozzle, the CFs will rotate due to their interaction with the polymer, the other fibers and the boundary wall of the nozzle. It is very important to understand the fibers' motion in the polymer during the extrusion process. As we have shown in the tensile test from Chapter Three, the orientation beads, and thus that of the CFs, greatly affect the mechanical properties of the FFF printed structure.

Fiber orientation modeling has been studied extensively over the last three decades due to its important application in injection modeling, compression modeling and other extrusion process of fiber filled composites. The work by Jeffery [122] on periodic motion of a single ellipsoidal fiber suspended within a viscous, incompressible shear flow is the basis for most fiber orientation work. The motion of a single fiber can be described as [122]

$$\frac{D\mathbf{p}}{Dt} = \mathbf{\Omega} \cdot \mathbf{p} + \lambda(\mathbf{\Gamma} \cdot \mathbf{p} - \mathbf{\Gamma} : \mathbf{ppp}) \tag{4.75}$$

where $\mathbf{p}$ is the unit vector along the axis of an ellipsoid as, and $\mathbf{\Omega}$ is the vorticity tensor given as

$$\mathbf{\Omega} = \frac{1}{2}[(\nabla\mathbf{v}) - (\nabla\mathbf{v})^{\mathrm{T}}] \tag{4.76}$$

and $\mathbf{\Gamma}$ is the rate of deformation tensor

$$\mathbf{\Gamma} = \frac{1}{2}[(\nabla\mathbf{v}) + (\nabla\mathbf{v})^{\mathrm{T}}] \tag{4.77}$$

In the above, $\nabla\mathbf{v}$ represents the velocity gradient in the viscous fluid and λ is defined as

$$\lambda = \frac{r_e^2 - 1}{r_e^2 + 1} \tag{4.78}$$

Here r_e is the equivalent or hydrodynamic aspect ratio [123] of the fiber. Equation (4.74) is applicable to only one fiber, rendering it ineffective for modeling groups of interacting fibers.

To determine the fiber orientation of multiple fibers and fiber-fiber interaction in a fluid, Advani and Tucker [124] proposed the fiber orientation tensor evolution equation

$$\frac{D\mathbf{A}}{Dt} = -(\mathbf{\Omega} \cdot \mathbf{A} - \mathbf{A} \cdot \mathbf{\Omega}) + \lambda(\mathbf{\Gamma} \cdot \mathbf{A} + \mathbf{A} \cdot \mathbf{\Gamma} - 2\mathbb{A}:\mathbf{\Gamma}) + \mathbf{D_r} \qquad (4.79)$$

where $\mathbf{D_r}$ is the isotropic rotary diffusivity (IRD)

$$\mathbf{D_r} = 2C_I\dot{\gamma}(\mathbf{I} - 3\mathbf{A}) \qquad (4.80)$$

which is written as a function of the empirically derived interaction coefficient C_I, the scalar magnitude of the rate of deformation tensor $\dot{\gamma}$, and the identity tensor $\mathbf{I}$.

In the above, $\mathbf{A}$ and $\mathbb{A}$ are the second and fourth order orientation tensors, respectively, given as [122]

$$\mathbf{A} = \oint_s P_iP_j\psi(\mathbf{p})d\mathbf{p} \qquad (4.81)$$

$$\mathbb{A} = \oint_s P_iP_jP_kP_l\psi(\mathbf{p})d\mathbf{p} \qquad (4.82)$$

Here the region s represents the surface of the unit sphere and P_i, P_j, P_k and P_l each represents the component of the unit vector $\mathbf{p}$, and $\psi(\mathbf{p})$ stands for the probability density function for fiber orientation. Also, in order to solve for Equation (4.78), $\mathbb{A}_{ijkl}$ which is a higher order tensor has to be known in advance. The closures [125–127] are then needed to approximate this higher order tensor using the second order tensor $\mathbf{A}$.

Heller, et al. [121] used the Equation (4.78) and Orthotropic Fitted Closure [127] to model fiber orientation in a typical extrusion nozzle for desktop 3D printer, as shown in Figure 4.4. The materials they used were based on 15% carbon fiber by volume, with $E_f = 240GPa$, $v_f = 0.2$, $E_m = 2GPa$ and $v_m = 0.4$. The aspect ratio of the fiber is 15. Since the nozzle is in an axisymmetric shape, their work was simplified by using 2D axisymmetric

boundary conditions for nozzle. Ten streamlines were used to model the fiber orientation from the inlet to outlet of the nozzle. Figure 4.5 shows the values of fiber orientation tensor A_{33}, which indicates the degree of alignment along the nozzle axis, in the extrudate just outside the nozzle exit. The value of z in Figure 4.5 is the radial coordinate measured from the center of the extrudate flow.
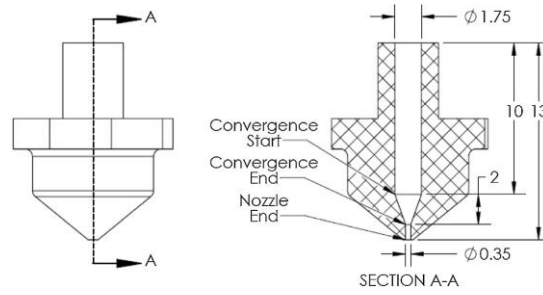
Figure 4.4. Cross section of FFF nozzle, units in diameter, taken from Heller, et al. [121]
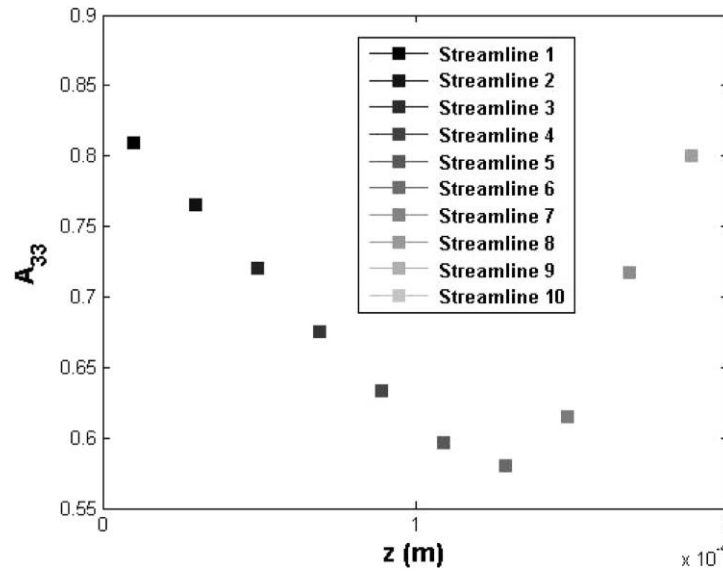
Figure 4.5. Steady state fiber alignment values for ten streamlines after nozzle exit, taken from Heller, et al. [121]

Based on the steady state fiber orientations in Figure 4.5, Heller, et al. [120] were able to back out the average elastic constants using the orientation homogenization method

99

introduced by Advani and Tucker [124] and was derive by Jack and Smith [128]. Each ten

fiber orientation is being treated as a unidirectional short fiber composite. Each composite

is homogenized using a constant mean strain or constant mean stress; and the average

elastic constants are simply the combined values of ten unidirectional short fiber

composites using trapezoidal rule. Tandon-Weng [129] is used to calculate the associated

coefficients for the orientation homogenization method. The average elastic constants

Heller, et al. [119] solved were taken to construct the traverse isotropic material matrix

described in Section 4.6. The elastic constants of the polymer composites below the

extrudate swell region from Heller, et al. [119] that we use in the simulations below appear

in Table 4.2. Comparing to the experimental work on tensile modulus in Chapter 3, which

finds that the tensile modulus at 0 degree is higher than 90 degree print orientations for all

four CFF polymer composites. This difference can also be observed in Table 4.2. The value

of E_x is about twice as high as E_y, which causes the anisotropy.

Table 4.2. Elastic Properties from the extrudate swell region [119]. Units in Pa for the modulus constants

E_x	E_y	G_{xy}	υ_{xy}	υyz
7.34	3.43	1.39	0.42	0.47

CHAPTER FIVE

Design Sensitivity Validation and Computational Examples

In this chapter, we show how the topology optimization approach in Chapter 4 can be applied to models having different print directions, various loading cases, and user defined geometry constraints. In Section 5.1, the design sensitivities of objective function are verified with the finite difference method. In Section 5.2, a simple cantilever beam example is optimized, and two different print directions are studied and compared to explore the influence of print orientation on optimal material layout. In Section 5.3, a multiple load case is optimized and in Section 5.4, an L-shape geometry is optimized where a nonlinear constraint file is applied to prescribe elements with minimum allowable density values. In Section 5.5, a topology optimization is presented which considers that the outline of the structure is defined in advance, and the infill pattern is the primary interest of the study to minimize the compliance.

5.1.   *Design Sensitivity Validation*

In this section we use the forward finite difference method to validate the design sensitivities computed with the Adjoint Variable Method described in Chapter 4. Recalling from Section 0, the design sensitivities with respect to the density and material orientation variables using forward finite difference method are respectively written as

$$
\begin{aligned}
\frac{\partial c}{\partial \rho_i} &= \frac{c(\boldsymbol{\rho} + \boldsymbol{\Delta\rho}_i) - c(\boldsymbol{\rho})}{\Delta\rho_i} \\
\frac{\partial c}{\partial \theta_i} &= \frac{c(\boldsymbol{\theta} + \boldsymbol{\Delta\theta}_i) - c(\boldsymbol{\theta})}{\Delta\theta_i}
\end{aligned}
\tag{0.1}
$$

Recall that Equations (4.67) and (4.71) are the analytical sensitivities using Adjoint Variable Method. It is important to note that the weight filter is not and should not be used for sensitivity validation, as it would distort the original sensitivities. For the material constants, Heller, et al. [121] presented a computational method for predicting material orientation in a FFF nozzle, modeled with Newtonian fluid and isotropic fibers as described in Section 4.7 above. Based on the fiber orientation state, they were able to calculate the elastic constants of a short fiber polymer composite extruded from a FFF nozzle, which are needed for this research. The elastic constants of the polymer composites in Table 4.2 are used for sensitivity validation and optimization.

The model used for sensitivity validation is the three-dimensional cantilever beam shown in Figure 5.1. The displacements in the three Cartesian directions are fixed at the base, and one single load is applied in the negative y-direction at the bottom-center of the tip. A finite element model with elements as uniform cubes was defined as 5 element in x by 3 element sin y by 2 elements in z to be used for sensitivity analysis validation, and the elemental arrangement is shown in Figure 5.2. Finite difference perturbation steps of 10^{-4}, 10^{-6}, 10^{-8} and 10^{-10} are used for both density and orientation design variables in the finite difference derivative evaluations. The design sensitivity verification runs were performed for density and material orientation variables at 0.4 and 0 radians, respectively. In these simulations, the x-y coordinate plane is treated as the plane of layer extrusion.
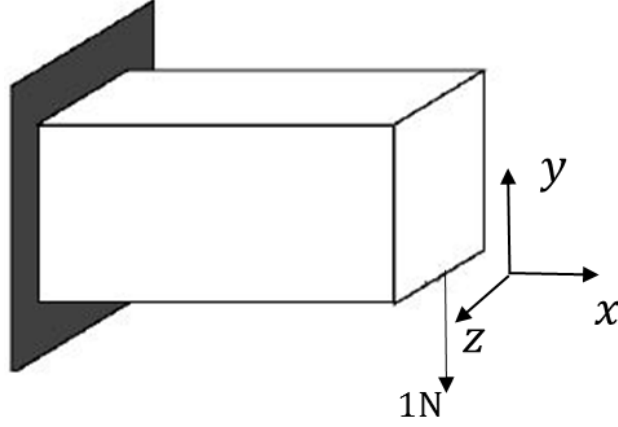
Figure 5.1. Cantilever beam with one single load

Figure 5.2. Elemental assignment for the cantilever beam as viewed looking at the front face of the domain; the top and bottom grids indicate the back and front layer of the domain, respectively.

Table 5.1 displays the largest absolute relative difference between the sensitivities computed with the Adjoint Variable Method and the finite difference method. The absolute relative difference is calculated as below

$$|\Delta| = \frac{\frac{\partial c}{\partial \phi_{\text{analytical}}} - \frac{\partial c}{\partial \phi_{\text{finite difference}}}}{\frac{\partial c}{\partial \phi_{\text{analytical}}}} \qquad (0.2)$$

Where $\phi = (\rho, \theta)$. The absolute relative difference yields the smallest values for both density and material orientation variables at $\Delta x = 10^{-6}$. At $\Delta x = 10^{-8}$ and $\Delta x = 10^{-10}$ where x is used here to represent either a density or orientation design variable. Large errors in this study occur due to the round-off error for very small perturbation sizes. Furthermore, for larger perturbations of $\Delta x = 10^{-4}$, the absolute relative differences are higher than for the case with $\Delta x = 10^{-6}$. The increased absolute relative differences occurring due to the truncation error of when a larger perturbation step is employed. Overall, absolute relative differences are on the order of 10^{-5} or less, demonstrating that the analytical sensitivities derived from the Adjoint Variable Method are correct.

Table 5.1. The largest absolute relative differences between the sensitivities using Adjoint method and finite difference method, using different perturbation steps.

Perturbation size	$\Delta x = 10^{-4}$	$\Delta x = 10^{-6}$	$\Delta x = 10^{-8}$	$\Delta x = 10^{-10}$
For ρ_e	$2.048e^{-4}$	$2.084e^{-6}$	$2.972e^{-5}$	$4.578e^{-3}$
For θ_e	$8.090e^{-3}$	$6.657e^{-5}$	$4.302e^{-3}$	$2.155e^{-1}$

5.2.   *Cantilever Beam Printed in Different Planes*

The first example considered here that demonstrates the material distribution and orientation topology optimization in three dimensions is a cantilever beam with a unit load applied to its tip, as shown in Figure 5.3. The optimization is performed assuming the print plane is defined as the x-y Cartesian plane Here we consider printing on each of three different print plane orientations. The topology optimization is first performed assuming that deposition is conducted from the back to the front face (labeled Case 1), then from the bottom to the top face (labeled Case 2), and finally from the left to right face (labeled Case

3). A finite element model having 20x10x6, 20x6x10 and 10x6x20 elements are used for Case 1, Case 2 and Case 3, respectively; this constitutes a total of 2400 design variables for the optimization problem. A volume fraction of $\frac{V(\rho)}{V_0} = 0.4$ in Equation (4.6) is assigned as the limit on the volume constraint, and the initial densities and material orientations are assigned 0.4 and 0 radians, respectively.
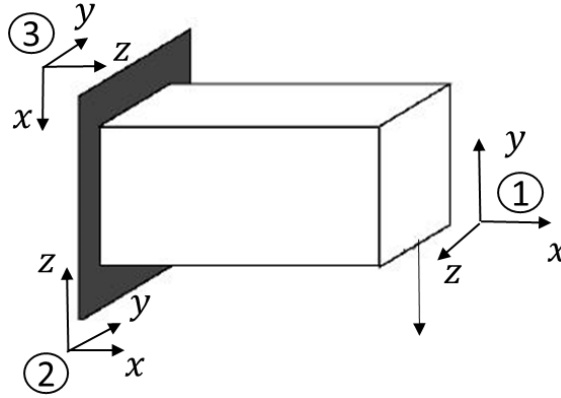
Figure 5.3. Cantilever beam with point load, printed in three different cases

Figure 5.4 to Figure 5.12 show the optimized topology and the material orientation for each of the 3 print plane directions described above, presented as isometric, top and front views. Figure 5.4, Figure 5.7 and Figure 5.10 show the isometric perspectives of the optimization result for Case 1, Case 2, and Case 3, respectively. The optimal structures all form a shape similar to that of an I beam regardless of the print orientation. In all cases, material is distributed to the wider sides of the I-beam shape, as indicated by the darker color. This extra material gives the I beam the needed stiffness to support the bending load. In addition to the 3D plots, for each loading case we plot the density and material orientation for each layer of the structure in the view that is orthogonal to the print plane to better illustrate the material distribution and material orientation of the structure.
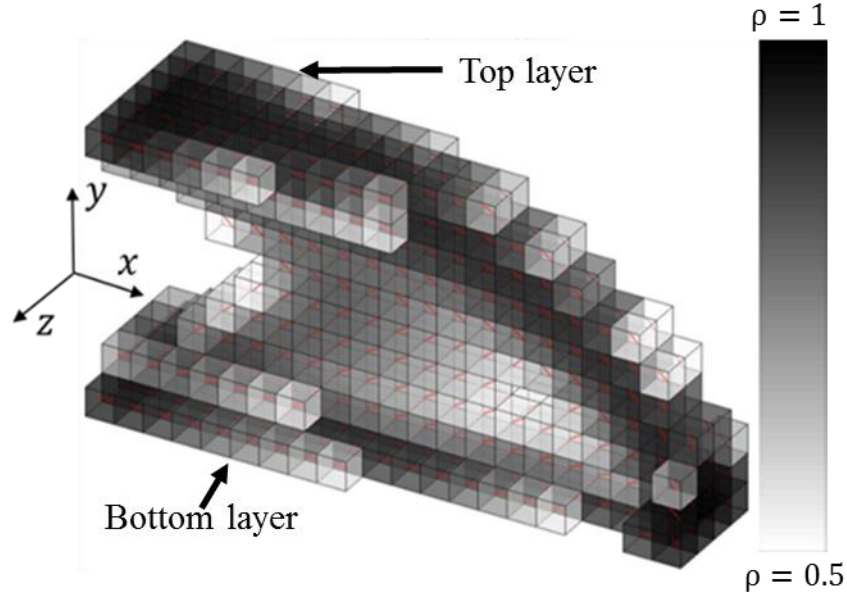
Figure 5.4. Optimal element density values for Case 1 (isometric view). Note that element with $\rho_e \leq 0.5$ are not shown. Darker color means more material.

Figure 5.5. Topology Result for Case 1 (top view). Note that element with $\rho_e \leq 0.5$ are not shown. Darker color means more material.

Figure 5.6 shows the each layer of their material distribution and material orientation. In Figure 5.5, the material orientation point to the right direction where the force is loaded. The optimal result is symmetric about the beam's mid plane which is expected since the load is applied at the center of the right bottom edge, as shown in Figure 5.6, By the comparing the density and orientation values of layer 1 to lay 6, layer 2 to layer 5 and layer 3 to layer 4, the layer pairs look visually identical, which confirms our

hypothesis. The results in Figure 5.6 show that the optimal material orientation follows the

outer contour of the structure for each layer where the dense material is distributed, which

is very similar to results given by Hoglund and Smith [4] and Nomura, et al. [5].
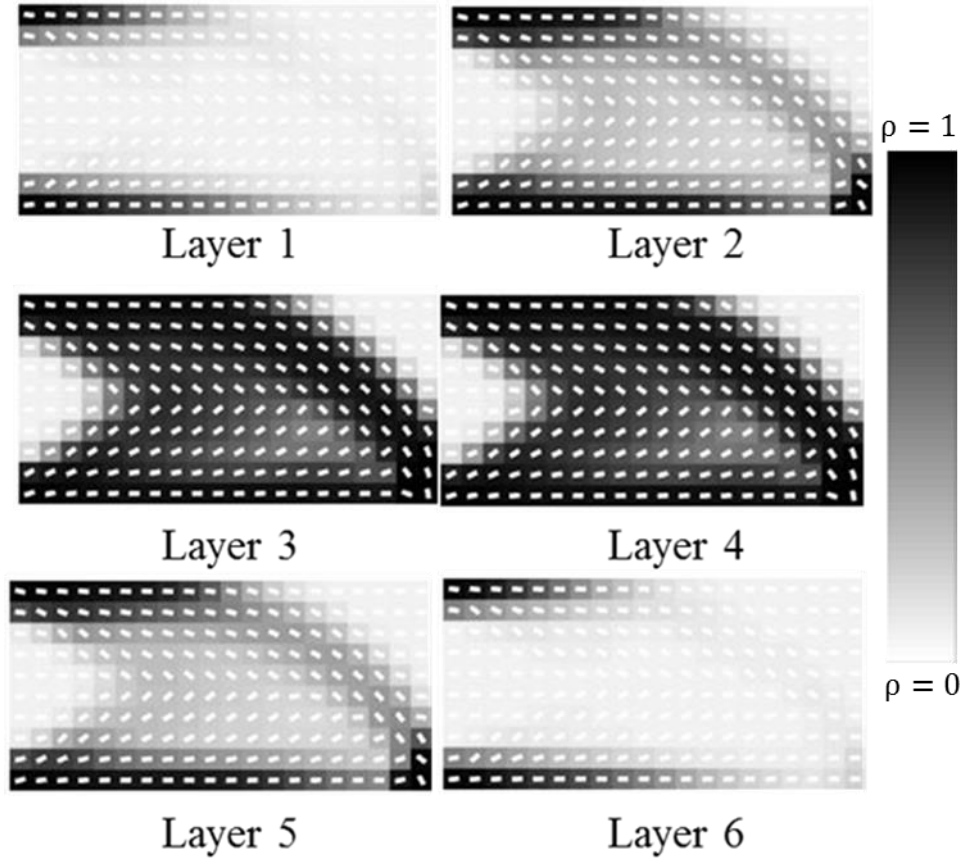
Figure 5.6. Optimized material distribution and material orientation, layer-by-layer plots for Case 1. Layer 1 is corresponds to the back layer, and layer 6 corresponds to the front layer. Darker color means more material.

Figure 5.8 shows layer-by-layer plot for the material distribution and material

orientation. Note that the direction of material orientation tends to point toward the applied

load. which is as expected in order to support the moment carried by the beam.
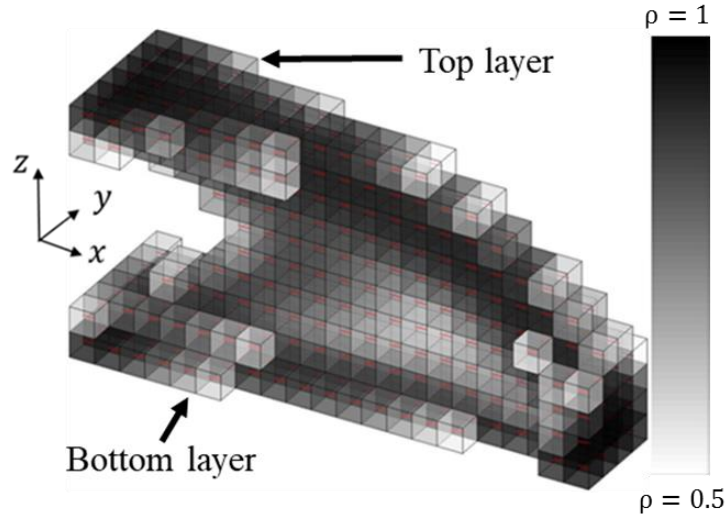
Figure 5.7. Optimal element density values for Case 2 (isometric view). Note that element with $\rho_e \leq 0.5$ are not shown. Darker color means more material.

Figure 5.8. Optimized material distribution and material orientation, layer-by-layer plots for Case 2. Layer 1 is corresponds to the bottom layer, and layer 10 corresponds to the top layer. Darker color means more material.

ρ = 1

ρ = 0.5
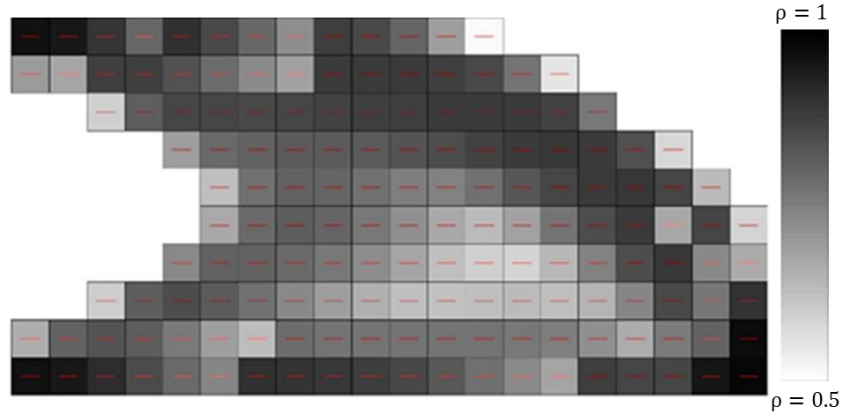
Figure 5.9. Topology Result for Case 2 (front view). Note that element with $\rho_e \leq 0.5$ are not shown. Darker color means more material.

Top layer

ρ = 1

ρ = 0.5

Bottom layer
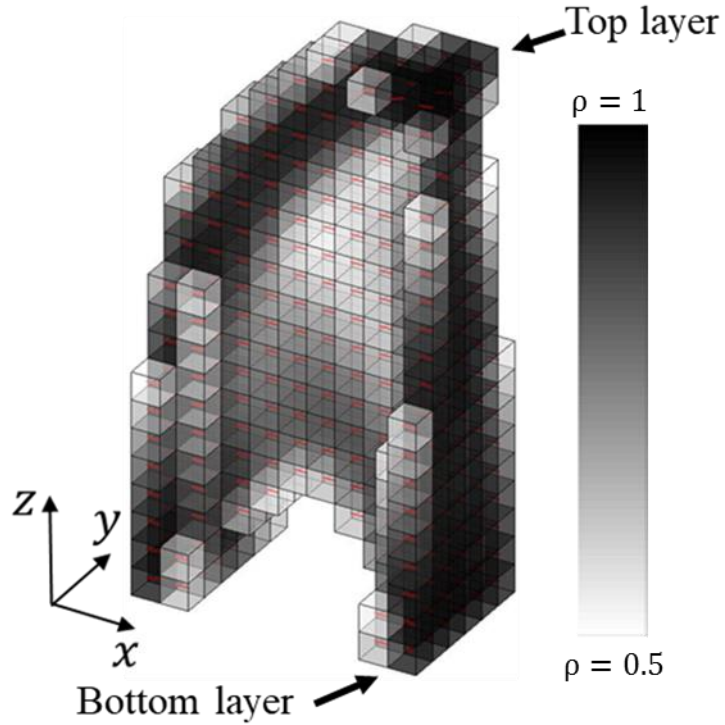
Figure 5.10. Optimal element density values for Case 3 (isometric view). Note that element with $\rho_e \leq 0.5$ are not shown. Darker color means more material.

Figure 5.12 shows the material distribution and material orientation at each layer,

with layer 1 at the bottom to layer 20 at the top of the optimized structure. Fiber orientation

109

symmetry exists in each layer by comparing the first three layers counting form the top to

the three layers counting from the bottom.
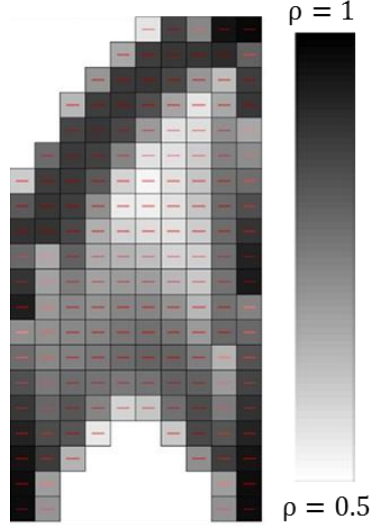
Figure 5.11. Topology Result for Case 3 (front view). Note that element with $\rho_e \leq 0.5$ are not shown. Darker color means more material.
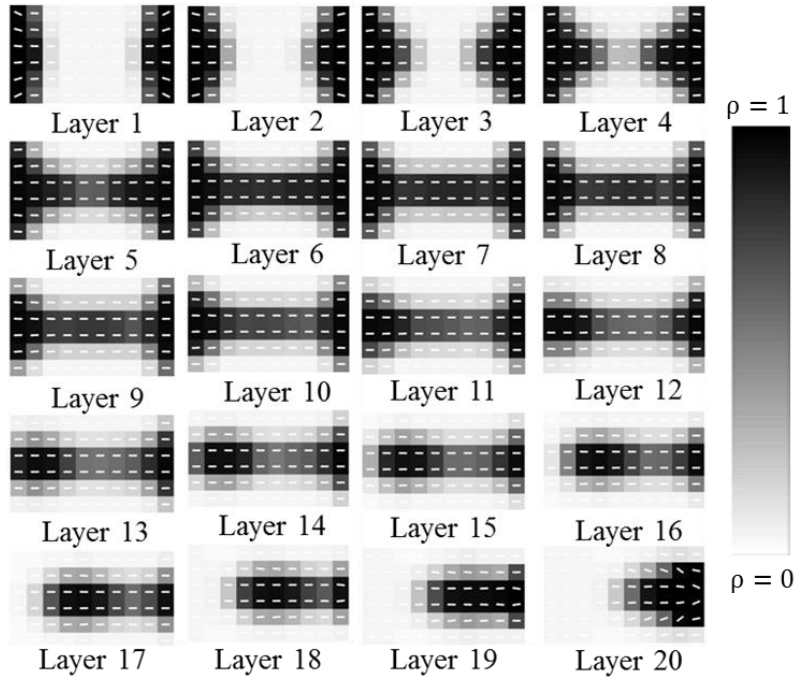
Figure 5.12. Optimized material distribution and material orientation, layer-by-layer plots for Case 3. Layer 1 is corresponds to the bottom layer, and layer 20 corresponds to the top layer. Darker color means more material.

5.2.1. Computational Result Comparison

Table 5.2 compares the CPU time, the number of iterations and the compliance for the three topology optimization cases presented above. The amount of CPU time required to solve Case 1 was 48.8% and 70.8% more than Case 2 and Case 3, respectively. Similarly, Case 1 required24 and 31 more optimization iterations than Cases 2 and 3, respectively. Case 1 yields 23% and 63% lower optimal compliance than Case 2 and Case 3, respectively. In Case 1, orientation design variables allow the preferred material direction to rotate in a plane that has a larger effect on the structure's compliance However, for Case 2 and Case 3, the plane of material rotation is normal to the direction of the force resulting in an optimization that yields less improvement in terms of the compliance. These results show that the relationship between the applied loads and plane of material orientation has a significant effect on the outcome of the topology optimization.

Table 5.2 Topology result comparisons between Case 1 and Case 2.

Case study	CPU time (sec)	Iterations	Compliance (N*m)
Case 1	246.3	78	3.48
Case 2	165.5	54	4.28
Case 3	144.2	47	5.66

5.3.   Multiple Load Cases – Symmetric Loading

In this section, we demonstrate the capability of our orthotropic topology optimization approach on AM structures designed to support multiple load cases, such as that shown in Figure 5.13. In this example, a unit force is applied individually to the upper right edges and the lower right edge as shown in Figure 5.3.1. To accommodate multiple load cases in the topology optimization, the compliance is modified as,

$$c(\boldsymbol{\rho}, \boldsymbol{\theta}) = \sum_{l=1}^{M} \mathbf{U}(\boldsymbol{\rho})_l^T \mathbf{F}_l = \sum_{l=1}^{M} \mathbf{U}(\boldsymbol{\rho})_l^T \mathbf{K}(\boldsymbol{\rho})_l \mathbf{U}(\boldsymbol{\rho})_l$$
$$= \sum_{l=1}^{M} \sum_{e=1}^{N} (\rho_{le})^p u_{le}^T k_{le}(\theta_{le}) u_{le}$$

(0.3)

where M is the number of load cases.

In our example, a finite element model with 30 elements by 30 elements by 4 elements forms the design domain, giving a total of 7200 design variables for the problem. A volume fraction of $\frac{V(\rho)}{V_0} = 0.5$ is assigned to the volume design constraint in Equation (4.6), with the initial density values of 0.5 and material orientations of $\theta_i = 0$, $i = 1 \dots 3600$.
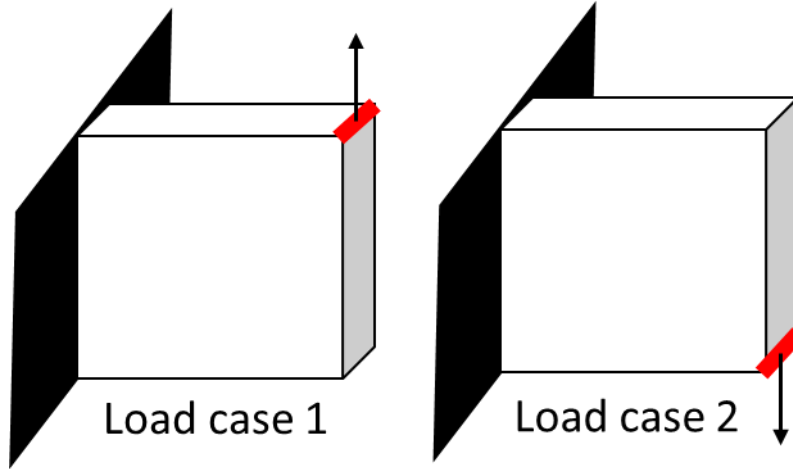
Figure 5.13. Multiple loaded structure; the upper and lower right edges are imposed with a unit force at each node (30x30x4 elements). The red lines mean a unit force is applied to each node at the upper and lower right edges.

The isometric view of the optimized topology with corresponding material orientations is shown in Figure 5.14. Again, to illustrate the layer-by-layer material distribution and fiber orientation in the print plane, Figure 5.15 shows the plots of the optimized design variables for the four layers, from layer 1 of back to layer 4 of the front.

Because the forces are at each node on the right-top and right-bottom edges, we would expect the optimized layers in the view of the print plane to be much the same, and this is very much true by observing all four layers.
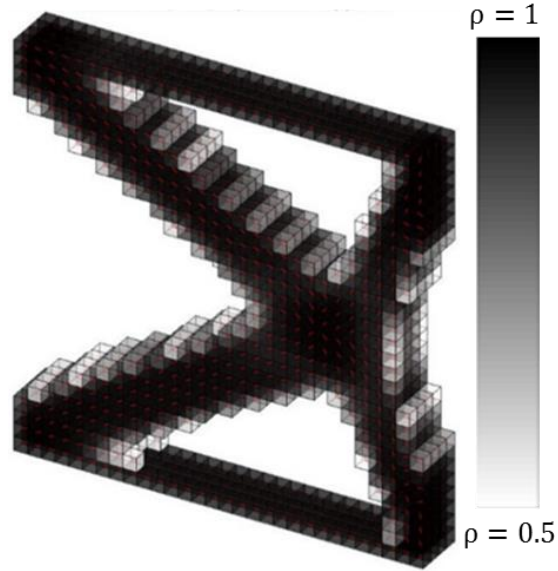
Figure 5.14. Optimized topology for multiple loaded case (isometric view). Note that element with $\rho_e \leq 0.5$ are not shown. Darker color means more material.
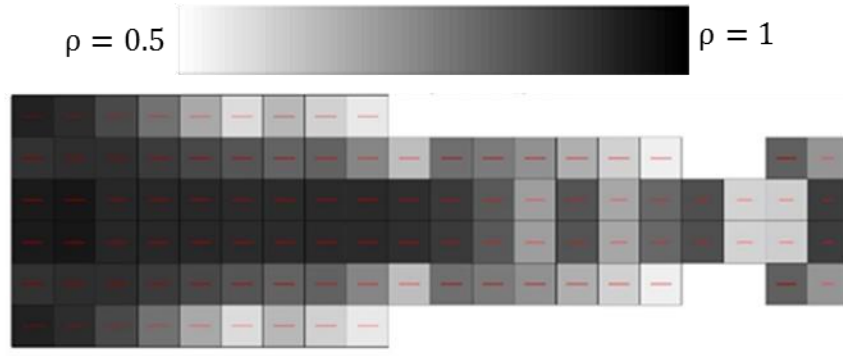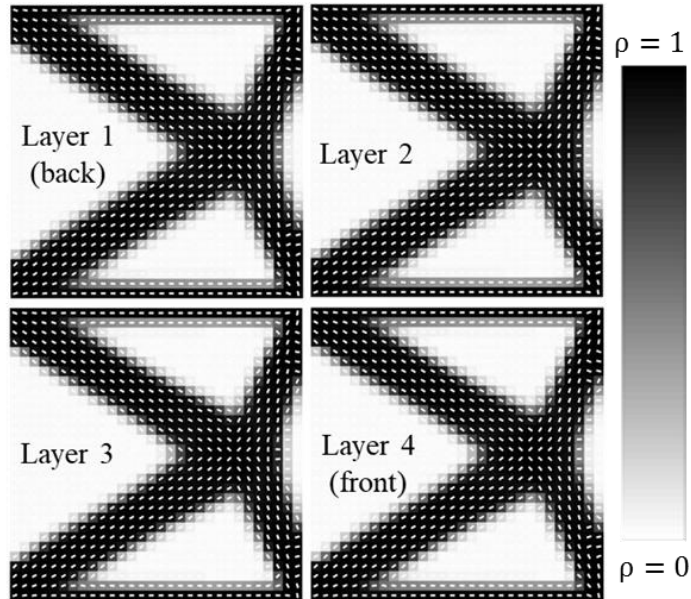

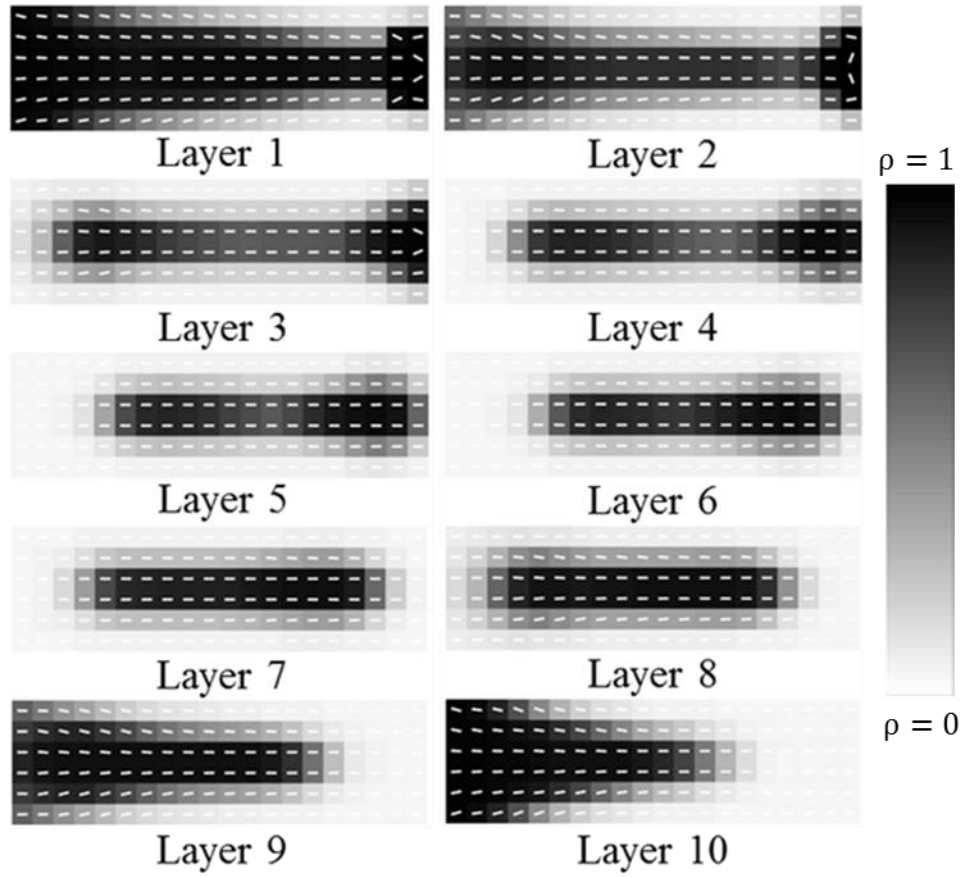Figure 5.15. Optimized material distribution and material orientation for multiple loaded case, layer-layer-plots. Layer 1 is corresponds to the back layer, and layer 4 corresponds to the front layer. Darker color means more material.

Truss structures are formed and fibers align along the direction of the truss members as can be seen in the optimal topology appearing in Figure 5.5. Since there are more design variables involved, the amount of iteration and computation time increased as well. The computation time for the optimization was 3606sec, which included 143 optimization iterations. The optimized compliance decreased from 334.3Nm in the initial design to 49.5 Nm in the final optimization (a decrease of 575.4%). Fibers in general follow the outline of the topology.

5.4.   L-shape Problem

In this section, an L-shape structure is studied, as shown in Figure 5.16. To simulate the L-shape, a rectangular 3D design domain is defined first and certain elements in the domain are forced to be at the lower limits of the density values. A finite element model composed of 20 elements by 20 elements by 8 elements is used for the design domain, yielding a total of 6400 design variables in the topology optimization. Here, the nonlinear constraint file `nonlcon` can be used to constrain a selected set of element have density design variable values at the lower bound of the allowable range, i.e., 1×10^{-6}. Nonlinear equality constraint vector **ceq** and the corresponding gradient matrix **Geq** are defined to set the desired constraint. In the input command `options`, two structural parameters are turned on; GradConstris is selected to include the gradient matrix **Geq** in the optimization process, and MaxFunctionEvaluations is selected to allow infinite numbers of evaluation. The upper face in Figure 5.16 of the L-shape is fixed in all three directions, and a unit load is imposed on the right face. A volume fraction of $\frac{V(\rho)}{V_0} = 0.35$ is assigned to the volume

constraint in Equation (4.6), with the initial density values of 0.35 and initial material orientations set at 0 radians.
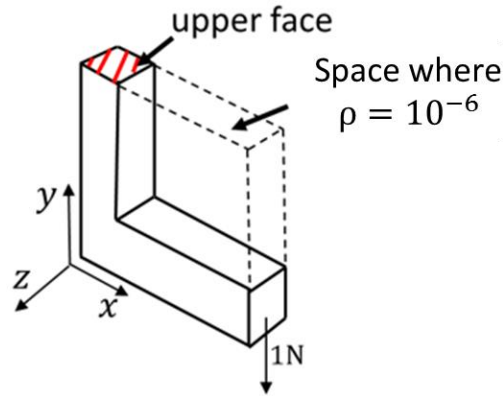
Figure 5.16. L-shape structure (20 elements by 20 elements by 8 elements). The upper right square enclosed with dotted lines indicate that for the associated elements, $\rho_e = 10^{-6}$

The CPU time required to compute the optimized solution was 1320.7 sec, which included 50 optimization iterations. The compliance decreased from 58.9Nm in the initial design to 15.5Nm in the optimal design, a decrease of 280%. Figure 5.17 shows the isometric view of the optimized topology. Darker color in an element indicates a higher value of density. Note that the location where the force is applied has relatively higher values of material density. Also, the inner surface in Figure 5.17 where the structure turns 90 degrees has density values approaching unity; this is due to the structure experiences bending stress, and more material is needed near the surface of the inside corner region. Figure 5.18 shows the layer-by-layer plots of the optimized structure, where the material orientation states can be identified. In the lower region of the topology in Figure 5.17, the fibers tend to follow the outline of the structure. In the upper region of the structure in Figure 5.17, the fiber do not follow the outline of the structure; instead, the fibers lay in a horizontal manner to provide highest bending resistance from the load.

Figure 5.17. Optimized topology for L-shape structure (isometric view). Note that element with $\rho_e \leq 0.5$ are not shown. Darker color means more material.

In Figure 5.18, the material orientation in each layer generally align toward the right side in the top half part of the L-shape until the structure turns 90 degrees. In addition, material orientation in the lower half of the structure follow the outer contour of the region where the material is dense (shown in black). Since the load is applied at the center node in z-direction, the optimal topology exhibits symmetry about the structures x-y mid-plane. Figure 5.18 also shows that denser material is distributed to the region where the structure takes the 90-degree turn, as well as the outer surfaces at the top half.
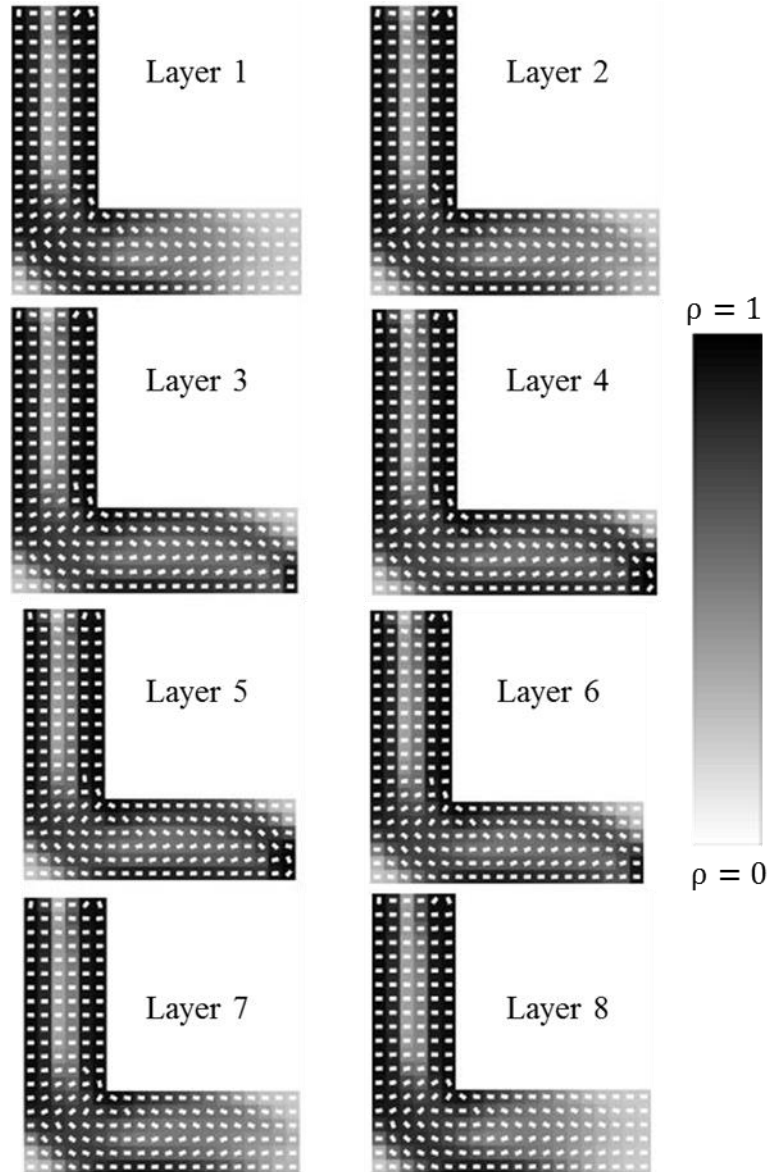
Figure 5.18. Optimized topology for L-shape structure, layer-by-layer plots of material distribution and material orientation. Layer 1 is corresponds to the back layer, and layer 8 corresponds to the front layer. Darker color means more material.

5.5. *Topology Optimization of Infill Pattern*

In some cases, the outline of the structure is defined in advance leaving only the infill pattern to be determined. Instead of optimizing a structure with certain elements of void space, as studied in the previous case study of L-shape structure, here density values of selected elements are constrained to be at the upper limit of the density variable value

range. In this study the nonlinear constraint file nonlcon is utilized again; GradConstr is turned on to include gradient matrix **Geq,** and MaxFunctionEvaluations is selected to allowable infinite function valuations. The cantilever beam problem from Section 5.2 is considered again here for the topology optimization design, as shown in Figure 5.19. However, in this case each node at the lower edge of the right side is prescribed with a unit force. The red line means a unit force is applied to each node at the lower right edge. Furthermore, the elements at the outer surfaces except at the front and back faces are prescribed with density values of 0.6. It would be more reasonable to use value of 1 here since it completely means solid. However, we want to allow more free density values to optimize the inner structure. An elemental set of 30 elements by 16 elements by 8 elements is defined as the design space, which translates to a total of 7680 design variables, with a volume constraint of $\frac{V(\rho)}{V_0} = 0.5$. The initial density values are 0.5 and the initial fibers orientations are 0 radians.
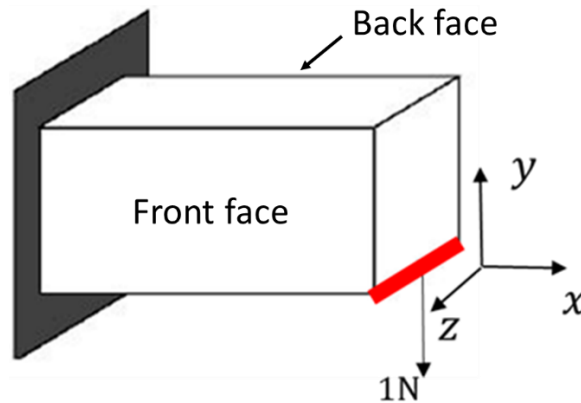
Figure 5.19. Cantilever beam problem; in this case the red line means a unit force is applied to each node at the lower right edge, and the outside surface of the beam is prescribed with elemental values.

The optimization finished in 1587.9 sec, 53 iterations and improved from 425.1 Nm in the initial design to 160.7 Nm in the final design (decrease of 164.5%), an decrease in compliance of. Figure 5.20 shows the isometric view of the optimized topology, and the prescribed elemental sets at the outer surfaces can be clearly identified.
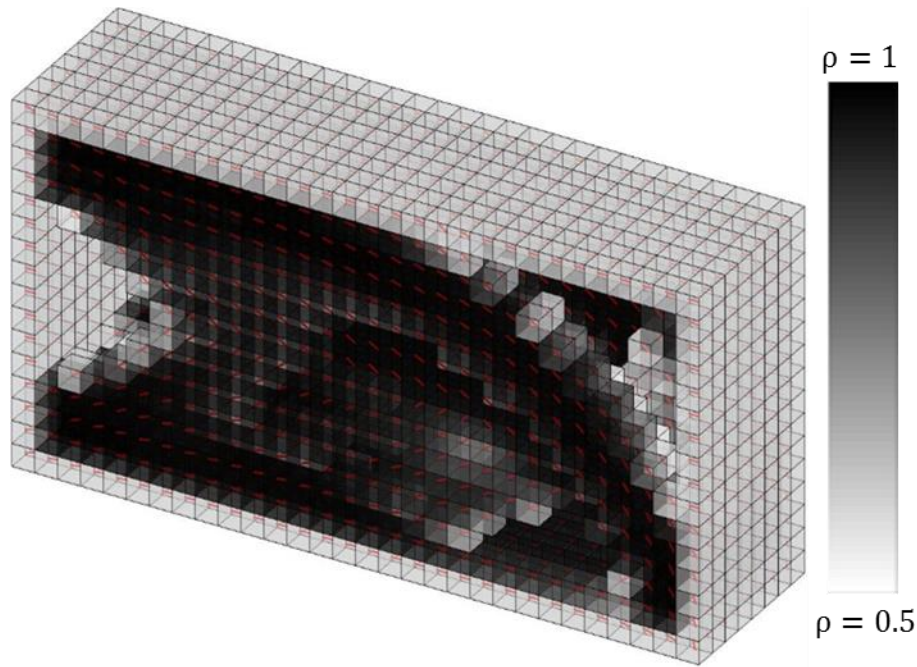
Figure 5.20. Optimized topology of the outline cantilever beam (isometric view). Note that element with $\rho_e \leq 0.5$ are not shown. Darker color means more material.

To better identify the infill region, Figure 5.21 shows the layer-by-layer material distribution and material orientation optimal result. Note that there material is distributed more towards the back and front layers as evidenced by the higher density values in these elements. Layers nearer to the structure's mid-plane provide little material support as indicated by the low density values in layers 3, 4, 5 and 6 which have densities approaching zero. Note that the infill pattern in Figure 5.21 is similar to the optimized results for the beam example from Section 5.2 that appear in Figure 5.6. Lastly, the symmetry of optimal

layout is evident by comparing layer 1 to layer 8, layer 2 to layer 7, layer 3 to layer 6 and

layer 4 to layer 5.
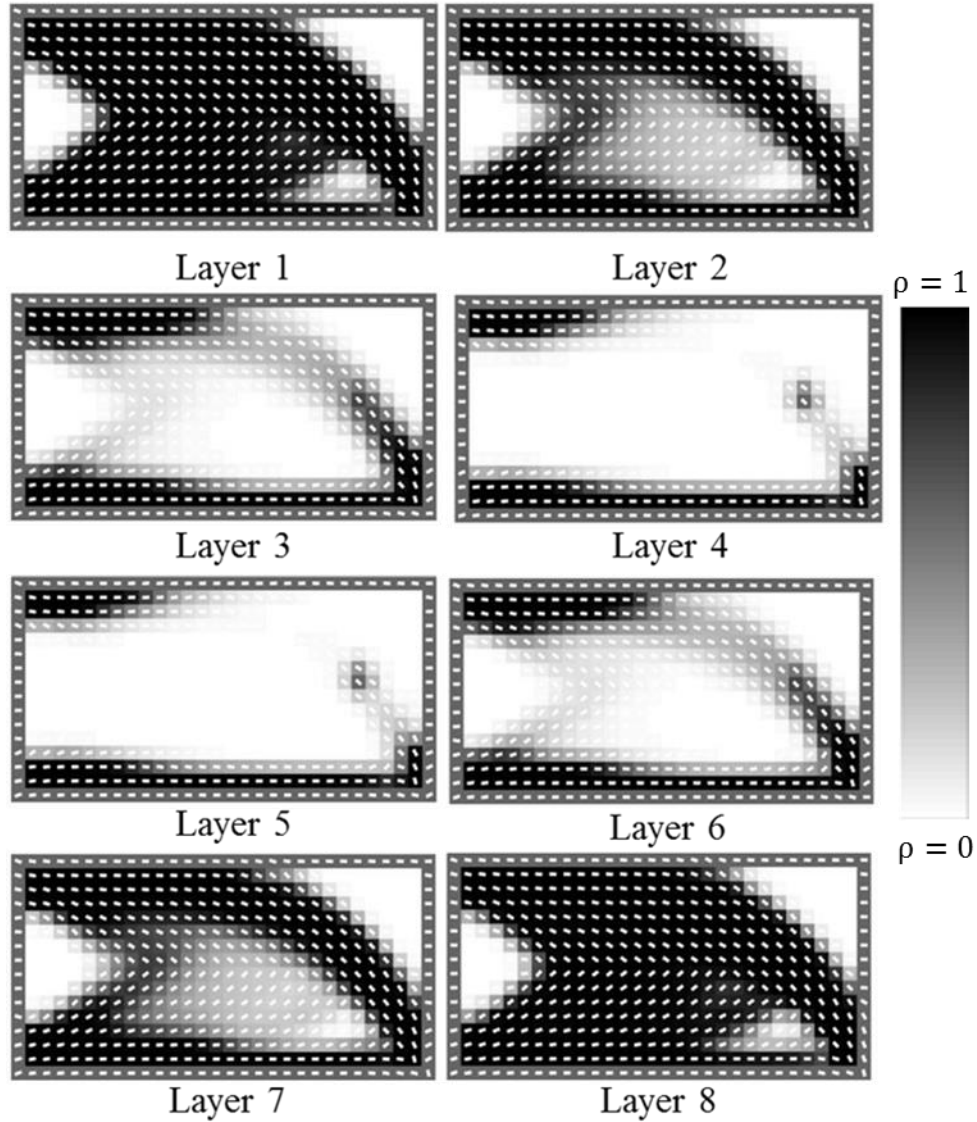
Figure 5.21. Optimized material distribution and material orientation, layer-by-layer plots of the outline cantilever beam. Layer 1 is corresponds to the back layer, and layer 8 corresponds to the front layer. Darker color means more material.

5.6.   *Multiple Load Cases – Non-Symmetric Loading*

The final example optimization is for a beam structure with multiple non-symmetric

load cases. In this example, three separate load cases are imposed on the structure as shown

120

in Figure *5.22*. An elemental set of 20 elements by 10 elements by10 elements is applied to the design, which translates to a total of 4000 design variables. A volume constraint of $\frac{V(\rho)}{V_0} = 0.4$ is imposed to the design domain, with initial density values of 0.4 and the initial fibers orientations of 0 radians.
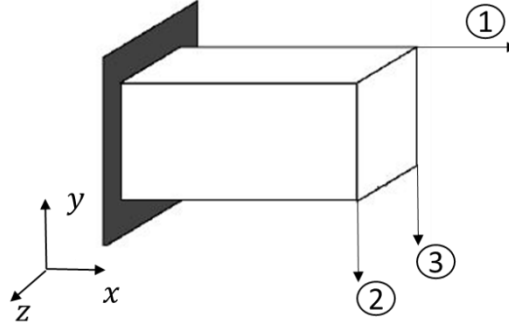
Figure 5.22. Multi-load cases scenarios with three loads

Figure 5.23 shows the result of the topology optimization in an isometric view. The resulting high density elements tend to form a box-like structure as apparent in the figure.. Figure 5.24 shows the layer-by-layer plots from the back (layer 1) to the front (layer 10). It can be seen that elements in the back of the structure has higher density values than element towards the front, which is likely due to the loads being applied more towards the back of the structure. Density distributions towards the front of the structure tends to have a similar pattern as that seen in the simply loaded cantilever beam as shown in Figure 5.6 in Section 5.2. From layer 1 to layer 7 as shown in Figure 5.24, the material orientations in the first row of the design domain align with force at the top. Material orientations in layers 1 to layers 10 have similar pattern as the Case 1 studied in Section 5.2. The topology optimization problem required 122 iterations and 922.8 CPU seconds to converge. The optimization reduced the compliance by 1051.4% from 118.6Nm in the initial design to 10.3 Nm in the optimal design.
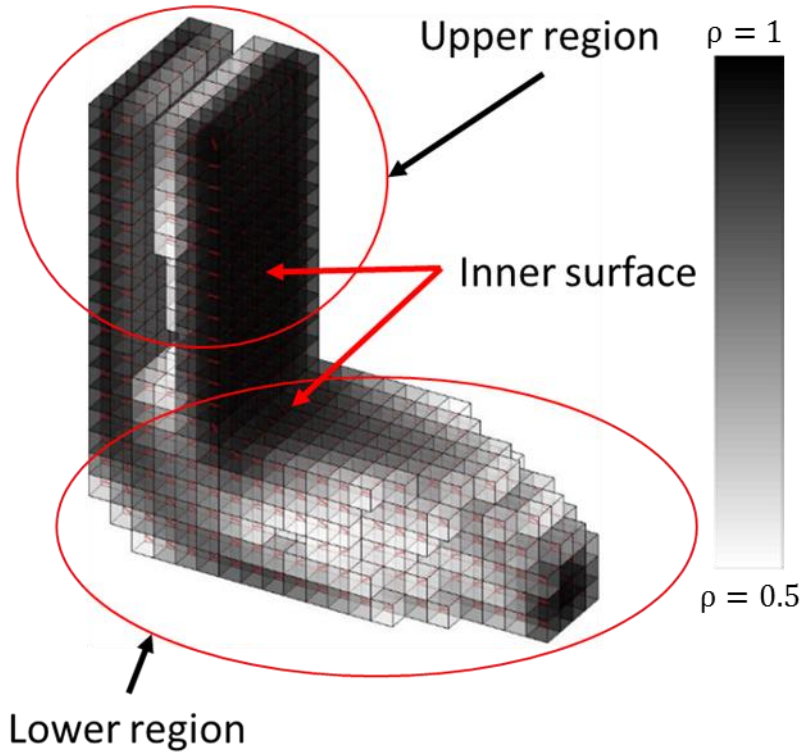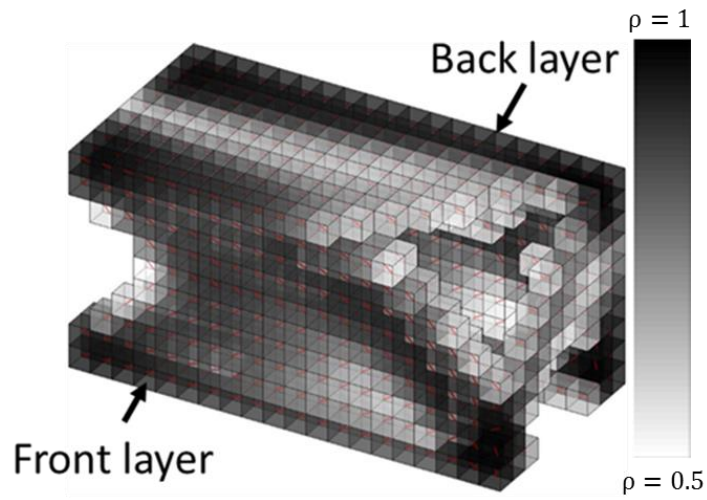
Figure 5.23. Optimized structure (isometric view). Note that element with $\rho_e \leq 0.5$ are not shown. Darker color means more material.
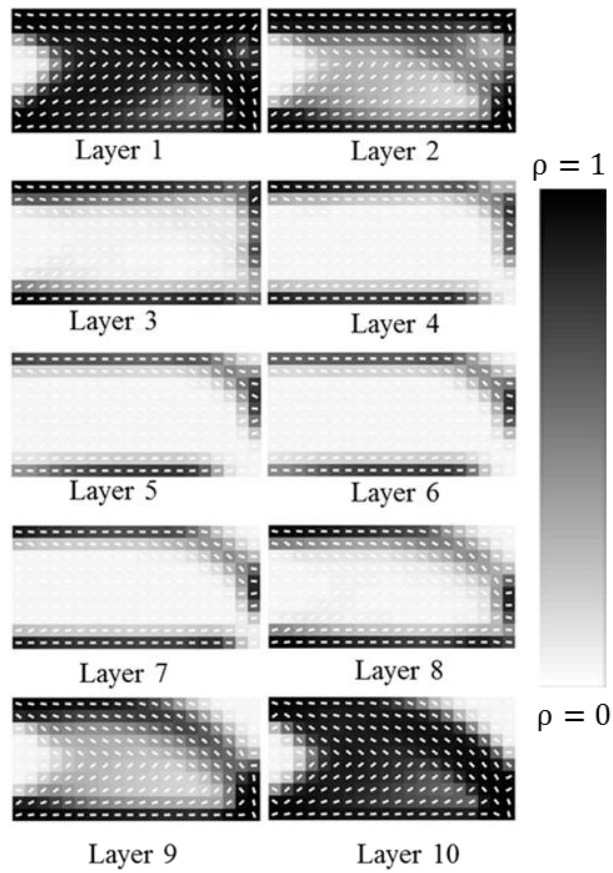
Figure 5.24. Optimized material distribution and material orientation, layer-by-layer plots for non-symmetric load cases. Layer 1 is corresponds to the back layer, and layer 10 corresponds to the front layer.

CHAPTER SIX

Conclusion and Future Work

In this chapter some final remarks will be addressed regarding this research. In the conclusion section, discussion of several significant findings from the research from both material testing and topology optimization will be made. Furthermore, several future studies based on the present optimization scheme will be discussed.

6.1.   Conclusion

This thesis consists of two major parts. First, CFF polymers composites are tested using tensile machine to evaluate their tensile properties and a comparison was made with unfilled counter parts. Four different types of polymer materials are selected for printing test coupons according ASTM 3039 standard [107], at four different print orientations. FLD measurements were also conducted to compare the FLD before and after the extrusion process. Lastly, fractured samples were observed under the SEM and failure of the test samples were analyzed based on these findings. Important findings that result from the CFF testing include:

- 0 degree print orientation results in the highest mechanical property improvement compared to pure polymer filament of the same material for all the CFF polymer composites.

- CFF PETG at 0 degree print orientation yields the greatest strength and modulus improvement compared to the unfilled PETG among all CFF samples tested,

resulting in 313.2% increase in tensile modulus and 48.2% increase in tensile strength.

- CFF polymer test samples at 90 degrees print orientation gives the least improvement, for all the test materials. Many circular voids and other irregular shaped voids were observed at the 0, 45 and ±45 degrees print orientations. For CFF fractured test coupons at the 90 degrees print orientation, many fibers were exposed on the surface.

- The FFF deposition process appeared to cause a slight reduction in fiber length. However, the weight average fiber length aspect ratio differences are less than 1.5 before and after the print for all the tested materials.

In the project for material distribution and material orientation optimization to design for FFF process, we have successfully implemented the modified SIMP with CFAO and Matlab *fmincon* optimization tool to solve 3D optimization problem. It is important here to note that most of the codes are written in-house, except the optimization algorithm and the linear sensitivity filter. The objective value is the compliance of structure that needs to be minimized. It is calculated using Finite Element Analysis with linear eight node isoparametric element; the *fmincon* solver requires gradients to be provided for the direction search, and sensitivities of the density and the fiber orientation variables are calculated using Adjoint Variable Method for each element. The sensitivities were further verified with the Finite Difference method. Furthermore, a linear weighted sensitivity filter is used on the density variables to prevent the checkerboarding of the material distribution. Some of the important points we can make for this project are:

- The optimization scheme can solve single loaded structures, multiple loaded structures, and structures with regions of solid or void that are prescribed.

- For the problem involved in printing at multiple planes, the case where fiber rotation plane aligns with the force direction gives the lowest compliance; Furthermore, printing the structure at different plane direction gives very different compliance result, which is important for designers to consider the potential loading scenarios.

- Fiber orientation generally follow the outer contour of the dense material region for each layer.

- For problem with high number of design variables such as the studies shown from Section 5.3 to Section 5.5, it can take up to an hour to solve the problem. The efficiency of the code can be improved, and we make some comments in the next section for some potential future work.

6.2.   *Future Work*

Based on the material testing work and 3D topology optimization code that we have developed, there are several recommendations can be stated to further improve researchers' understanding of FFF process. Here we would make some following recommendations:

For material testing of CFF polymer composites with FFF process:

- Fiber orientation in the printed parts are really important. To be able acquire empirical data for fiber orientation would further help verifying the simulation work in fiber orientation that had been done in the research community.

- More experimental testing can be carried out on other CFF polymer filaments to expand the date set. At the same time, mathematical modeling can also be

developed to predict the mechanical properties based on different parameters such as fiber weight fraction.

For topology optimization with FFF process:

- To improve the efficiency of the optimization algorithm, it would be very desirable to design a program specifically for material distribution and fiber orientation optimization. This main goal is speed up the convergence of the solving time. The extra benefit for having a custom optimization algorithm is that the user can implement different filter to get more discrete material distribution and therefore better optimized objective value. Sigmund [52] provides a good summary of different kinds of filters.

- The elastic constants obtained from Heller, et al. [120] are based on the study of fibers flowing in Newtonian flow. Since most polymers are non-Newtonian, more accurate data are needed.

- The experimental work is essential to verify the optimization results. To do this, a custom code that can translate the material distribution and the material orientation into a STL file and can subsequently be printed by 3D printer would be very helpful to achieve this goal.

- Cellular structure is a popular topic of study as discussed in the literature review. Even though the present code works in three dimension, it can easily be modified to do optimization in 2D. The author successfully duplicated work done by Zhang,et al.[62]. To expand his work into FFF, the constitutive material model needs to be modified into orthotropic states to take the material orientation into

126

account. However, the material orientation variables would not be design variables since we would only print the part in a fixed angle of orientation.

It is the author's hope that this thesis work would help the 3D printing community to better understand the process of FFF. FFF is an exciting technology and much work is needed to be done before we prefect the printing process to achieve another industries revolution.

APPENDIX

APPENDIX

Matlab Codes for Topology Optimization

```matlab
function top_CFAO(nx,ny,nz,rho0,theta0,p,r)
% This file initiate the optimization
close all; clc; format compact;
global nelx nely nelz penal rmin history video c0 Hs H passive_ele phi
global H1 H_01
%
history.iter = [];
video = VideoWriter(['need_change','.avi']);
video.FrameRate = 5; open(video);
%
nelx = nx; nely = ny; nelz = nz; penal = p; rmin = r; c0 = rho0;
% precalculate Hs for sesitivity filter
nele = nelx*nely*nelz;
iH = ones(nele*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for k1 = 1:nelz
    for i1 = 1:nelx
        for j1 = 1:nely
            e1 = (k1-1)*nelx*nely + (i1-1)*nely+j1;
            for k2 = max(k1-(ceil(rmin)-1),1):min(k1+(ceil(rmin)-
1),nelz)
                for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-
1),nelx)
                    for j2 = max(j1-(ceil(rmin)-
1),1):min(j1+(ceil(rmin)-1),nely)
                        e2 = (k2-1)*nelx*nely + (i2-1)*nely+j2;
                        k = k+1;
                        iH(k) = e1;
                        jH(k) = e2;
                        sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-
j2)^2+(k1-k2)^2));
                        H_1(k) = 1;
                    end
                end
            end
        end
    end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);
% for mean sensitivity
H_01 = sparse(iH,jH,H_1);
H1 = sum(H_01,2);
% option for fmincon
option = optimoptions('fmincon','Algorithm','interior-point',...
```

```
        'GradObj','on',...
        'TolX',1E-3,...
        'TolFun',1E-3,...
        'PlotFcns',@optimplotfval,...
        'OutputFcn',@outfun);
%       'GradConstr','on',...
%       'MaxFunctionEvaluations',inf);
%
rho0 = rho0*ones(nely,nelx,nelz); theta0 = theta0*ones(nely,nelx,nelz);
%% passive elements (L shape)
% for i = 1:nelx
%     for j = 1:nely
%         if i > floor(nelx/3);
%             if j < ceil(nely*2/3)+1
%                 passive(j,i) = 1;
%             else
%                 passive(j,i) = 0;
%             end
%         else
%             passive(j,i) = 0;
%         end
%     end
% end
% passive = repmat(passive,1,1,nelz);
% rho0(find(passive)) = 1e-6; passive_ele = find(passive);
%% passive elements (catilever edge forces in y-direc)
% passive = zeros(nely,nelx,nelz);
% passive(:,1,:) = 1; passive(1,:,:) = 1; passive(:,end,:) = 1;
passive(end,:,:) = 1;
% rho0(find(passive)) = 0.6; passive_ele = find(passive);
%%
% define phi (rotation about y axis)
offset = 0;
a = -offset/180*pi; b = offset/180*pi;
phi = a + (b-a)*rand(nele,1); phi = reshape(phi,nely,nelx,nelz); %
%
x0 = [rho0(:);theta0(:)];
lb = [1E-6*ones(length(rho0(:)),1);-pi*2*ones(length(theta0(:)),1)];
ub = [ones(length(rho0(:)),1);pi*2*ones(length(theta0(:)),1)];
%
% equality constraint
Aeq = [ones(1,length(rho0(:))) zeros(1,length(theta0(:)))]; beq =
nelx*nely*nelz*c0;
% fmincon % add @top_nonlcon (only for passive element)
x = fmincon('top_obj',x0,[],[],Aeq,beq,lb,ub,[],option);
close(video);
plot_layer(x)
end




function [F,dF] = top_obj(x)
% This file calls the file to the objective function and its
sensitivities and
% filtering
global ieqn nelx nely penal D Hs H nelz phi
```

```matlab
% define design variables
rho = x(1:length(x)/2); theta = x((length(x)/2+1):end);
rho = reshape(rho,nely,nelx,nelz); theta =
reshape(theta,nely,nelx,nelz);


% read input file
D = fea_input(nelx,nely,nelz); D.p = penal;
% initialize equation numbers for dofs and global matrices
ieqn = initialize(D);
% FE analysis
[utot,ptot] = proc0(D,rho,theta,phi);
% calculate c & dc
[F,dcdrho,dcdtheta] = Fcalc(utot,ptot,D,rho,theta,phi);
% filter
dcdrho = check(rho,dcdrho,Hs,H,1);
% design sensitivity
dF = [dcdrho(:);dcdtheta(:)];
end
%%%%%%%%% MESH-INDEPENDENCY FILTER for density
%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [dFn]=check(rho,dcdrho,Hs,H,flag)
global  nelx nely nelz
global H1 H_01
switch flag
    case 1
        % original sensitivity filter
        dcn(:) = H*(rho(:).*dcdrho(:))./Hs./rho(:);
    case 2
        % modified sensitivity filter
        dcn(:) = H*(rho(:).*dcdrho(:))./(H*rho(:));
    case 3
        % mean sensitivity filter
        dcn(:) = H_01*dcdrho(:)./(H1);
end
dFn = reshape(dcn,nely,nelx,nelz);
end


function D = fea_input(nelx,nely,nelz)
% This file determines which load case to optimize and create the
element
% and nodal coordinate sets
global ielem iprops iforce force idisp disp X
% read number of nodes, elements, applied forces and prescribed
displacements, and element type
D.nnode = (nelx+1)*(nely+1)*(nelz+1);      % no. of nodes in model
D.nel = nelx*nely*nelz;                     % no. of elements in model
D.etype = 3;                                % element type (only one per
model)
D.nelx = nelx;
D.nely = nely;
D.nelz = nelz;
%
% D.ndv = D.nel;                            % number of design
variables
%
```

```
% set properties related to element properties
switch D.etype
case 3
    D.elname = 'elemental_3D';
    D.ndof = 3; D.nenode = 8;
end
%% nodal coordinates mesh (standard rectangular)
[xloc,yloc,zloc] = meshgrid(0:nelx,nely:-1:0,0:nelz);
X = [xloc(:),yloc(:),zloc(:)]; node_mat =
reshape((1:D.nnode)',(nely+1),(nelx+1),(nelz+1));
%% element data (standard rectangular)
ielem =  zeros(D.nel,D.nenode); icont = 0;
for k = 1:nelz
    for i = 1:nelx
        for j = 1:nely
            icont = icont + 1;
            ielem(icont,1) = (nely+1)*(i-1) + j + 1 + (k-
1)*(nelx+1)*(nely+1);
            ielem(icont,2) = (nely+1)*i     + j + 1 + (k-
1)*(nelx+1)*(nely+1);
            ielem(icont,3) = (nely+1)*i     + j     + (k-
1)*(nelx+1)*(nely+1);
            ielem(icont,4) = (nely+1)*(i-1) + j     + (k-
1)*(nelx+1)*(nely+1);
            ielem(icont,5) = (nely+1)*(i-1) + j + 1 +
(k)*(nelx+1)*(nely+1);
            ielem(icont,6) = (nely+1)*i     + j + 1 +
(k)*(nelx+1)*(nely+1);
            ielem(icont,7) = (nely+1)*i     + j     +
(k)*(nelx+1)*(nely+1);
            ielem(icont,8) = (nely+1)*(i-1) + j     +
(k)*(nelx+1)*(nely+1);

        end
    end
end
%
iprops = ones(length(ielem),1);
%% L shape mesh (force in y-direc)
% D.loadcase = 1;                          % no. of loadcases
% unode = node_mat(1,1:floor(nelx/3),:);
% fnode = node_mat(end-1,end,floor(nelz/2)+1);
% D.nforce = length(fnode);                % no. of applied forces
per loadcase
% D.ndisp = length(unode(:))*3;
% % read applied loads
% iforce = [fnode(:), 2*ones(length(fnode),1)];
% force = -ones(1,D.nforce);
% %
% % read applied displacements
% idisp = [repmat(unode(:),3,1)
[ones(length(unode(:)),1);2*ones(length(unode(:)),1);3*ones(length(unod
e(:)),1)]];
% disp = zeros(1,D.ndisp);
%% cantilever beam (force in y-direc) edge forces
% D.loadcase = 1;                          % no. of loadcases
```

```
% D.nforce = nelz+1;                           % no. of applied forces
per loadcase
% D.ndisp = 3*(nely+1)*(nelz+1);
% % read applied loads
% fnode = node_mat(end,end,:);
% iforce = [fnode(:), 2*ones(length(fnode),1)];
% force = -ones(1,D.nforce)*1;
% %
% % read applied displacements
% unode = node_mat(:,1,:);
% idisp = [repmat(unode(:),3,1)
[ones(length(unode(:)),1);2*ones(length(unode(:)),1);3*ones(length(unod
e(:)),1)]];
% disp = zeros(1,D.ndisp);
%% cantilever beam (force in y-direc) point force
D.loadcase = 1;                                % no. of loadcases
D.nforce = 1;                                  % no. of applied forces per
loadcase
D.ndisp = 3*(nely+1)*(nelz+1);
% read applied loads
fnode = node_mat(end,end,floor(nelz/2)+1);
iforce = [fnode(:), 2*ones(length(fnode),1)];
force = -ones(1,D.nforce);
%
% read applied displacements
unode = node_mat(:,1,:);
idisp = [repmat(unode(:),3,1)
[ones(length(unode(:)),1);2*ones(length(unode(:)),1);3*ones(length(unod
e(:)),1)]];
 disp = zeros(1,D.ndisp);
%% cantilever beam (force in z-direc) edge forces
% D.loadcase = 1;                              % no. of loadcases
% D.nforce = nely+1;                           % no. of applied forces
per loadcase
% D.ndisp = 3*(nely+1)*(nelz+1);
% % read applied loads
% fnode = node_mat(:,end,1);
% iforce = [fnode(:), 3*ones(length(fnode),1)];
% force = -ones(1,D.nforce);
% %
% % read applied displacements
% unode = node_mat(:,1,:);
% idisp = [repmat(unode(:),3,1)
[ones(length(unode(:)),1);2*ones(length(unode(:)),1);3*ones(length(unod
e(:)),1)]];
%  disp = zeros(1,D.ndisp);
%% cantilever beam (force in z-direc) point force
% D.loadcase = 1;                              % no. of loadcases
% D.nforce = 1;                                % no. of applied forces
per loadcase
% D.ndisp = 3*(nely+1)*(nelz+1);
% % read applied loads
% fnode = node_mat(floor(nely/2)+1,end,1);
% iforce = [fnode(:), 3*ones(length(fnode),1)];
% force = -ones(1,D.nforce);
% %
% % read applied displacements
```

```
% unode = node_mat(:,1,:);
% idisp = [repmat(unode(:),3,1)
[ones(length(unode(:)),1);2*ones(length(unode(:)),1);3*ones(length(unod
e(:)),1)]];
%  disp = zeros(1,D.ndisp);
%% cantilever beam (force in x-direc) point force
% D.loadcase = 1;                                      % no. of loadcases
% D.nforce = 1;                                        % no. of applied forces
per loadcase
% D.ndisp = 3*(nelx+1)*(nely+1);
% % read applied loads
% fnode = node_mat(floor(nely/2)+1,end,end);
% iforce = [fnode(:), 1*ones(length(fnode),1)];
% force = ones(1,D.nforce);
% %
% % read applied displacements
% unode = node_mat(:,:,1);
% idisp = [repmat(unode(:),3,1)
[ones(length(unode(:)),1);2*ones(length(unode(:)),1);3*ones(length(unod
e(:)),1)]];
%  disp = zeros(1,D.ndisp);
%% MBB point load (force in y direc)
% D.loadcase = 1;                                      % no. of loadcases
% D.nforce = 1;                                        % no. of applied forces
per loadcase
% D.ndisp = 2*3 + 2;
% % read applied loads
% fnode = [node_mat(1,floor(nelx/2)+1,floor(nelz/2)+1)];
% iforce = [fnode(:), 2*ones(length(fnode),1)];
% force = -ones(1,D.nforce);
% %
% % read applied displacements
% unode =
[node_mat(end,1,1),node_mat(end,1,end),node_mat(end,end,1),node_mat(end
,end,end)]';
% idisp = [[unode([1 2]);repmat(unode([3 4]),3,1)],
[2*ones(2,1);ones(2,1);2*ones(2,1);3*ones(2,1)]];
% disp = zeros(1,D.ndisp);
%% MBB point load (force in z direc)
% D.loadcase = 1;                                      % no. of loadcases
% D.nforce = 1;                                        % no. of applied forces
per loadcase
% D.ndisp = 2*3 + 2;
% % read applied loads
% fnode = node_mat(floor(nely/2)+1,floor(nelx/2)+1,end);
% iforce = [fnode(:), 3*ones(length(fnode),1)];
% force = -ones(1,D.nforce);
% %
% % read applied displacements
% unode =
[node_mat(1,1,1),node_mat(end,1,1),node_mat(1,end,1),node_mat(end,end,1
)]';
% idisp = [[unode([1 2]);repmat(unode([3 4]),3,1)],
[3*ones(2,1);ones(2,1);2*ones(2,1);3*ones(2,1)]];
% disp = zeros(1,D.ndisp);
%% multiple loads
% % read applied loads
```

```matlab
% D.loadcase = 2;                              % no. of loadcases
% D.nforce = (nelz+1);                         % no. of applied forces
per loadcase
% D.ndisp = 3*(nely+1)*(nelz+1);               % no. of prescribed disp
% %
% fnode1 = node_mat(end,end,:);  fnode2 = node_mat(1,end,:);
% iforce = [[fnode1(:);fnode2(:)], 2*ones(D.nforce*D.loadcase,1)];
% force = [-ones(1,length(fnode1(:))),ones(1,length(fnode1(:)))];
% %
% % read applied displacements
% unode = node_mat(:,1,:);
% idisp = [repmat(unode(:),3,1)
[ones(length(unode(:)),1);2*ones(length(unode(:)),1);3*ones(length(unod
e(:)),1)]];
% disp = zeros(1,D.ndisp);
%%
% plot the nodes & elements
% figure;
% set(gcf,'Name','ISO display','NumberTitle','off');
% for i = 1:D.nel
%     elemnodes = ielem(i,:);
%     xx = X(elemnodes,1); yy = X(elemnodes,2); zz = X(elemnodes,3);
%     v = [xx,yy,zz];
%     % flip coordinates
% %     v(:,[2,3]) = v(:,[3,2]); v(:,2) = -v(:,2);
%     %
%     face = [1 2 3 4; 5 6 7 8; 4 3 7 8; 1 2 6 5; 1 5 8 4; 2 6 7 3];
%     patch('Faces',face,'Vertices',v,'FaceColor','none');
%     text(mean(v(:,1)),mean(v(:,2)),mean(v(:,3)),num2str(i));
%     hold on;
% end
% axis equal; view([30,30]); xlabel('\bfx'); ylabel('\bfy');
zlabel('\bfz');
%
% figure;
% for i = 1:length(X)
%     plot3(X(i,1),X(i,2),X(i,3),'x');
%     text(X(i,1),X(i,2),X(i,3),num2str(i));
%     hold on;
% end
% axis equal; %
set(gca,'XTickLabel',[],'YTickLabel',[],'ZTickLabel',[]);
% view([30,30]); xlabel('x'); ylabel('y'); zlabel('z'); keyboard


function [F,dcdrho,dcdtheta] = Fcalc(utot,ptot,D,rho,theta,phi)
% This file calculate the objective function and its sensitivities
global ielem
%
F = 0; dcdrho = zeros(D.nely,D.nelx,D.nelz); dcdtheta =
zeros(D.nely,D.nelx,D.nelz);
%
k = 0;
for m = 1:D.nelz
    for i = 1:D.nelx
        for j = 1:D.nely
```

```
            k = k+1;
            elemnodes = ielem(k,:);
            % get element dof number array
            ieldof = get_ieldof(elemnodes,D);
            [c_e,dcdrho_e,dcdtheta_e] =
feval(D.elname,rho(j,i,m),theta(j,i,m),phi(j,i,m),elemnodes,D,1,utot(ie
ldof,:));
            % objective function value
            F = F + c_e;
            % derivative w.r.t density
            dcdrho(j,i,m) = dcdrho(j,i,m) + dcdrho_e;
            % derivative w.r.t theta for gradient check
            dcdtheta(j,i,m) = dcdtheta(j,i,m) + dcdtheta_e;
        end
    end
end


function [eldat_F,eldat_K,eldat_KK] =
elemental_3D(eprop,theta,phi,lnodes,D,icode,de)
% elemental file
global X
xloc = [X(lnodes,1) X(lnodes,2) X(lnodes,3)];
p = D.p;
% define element properties
Ex = 7.34;
Ey = 3.43;
Gxy = 1.39;
vxy = 0.42;
vyz = 0.47;
%% Transverse Isotropic
SS=[     1/Ex,    -vxy/Ex,    -vxy/Ex,              0,        0,      0;...
      -vxy/Ex,       1/Ey,    -vyz/Ey,              0,        0,      0;...
      -vxy/Ex,    -vyz/Ey,       1/Ey,              0,        0,      0;...
            0,          0,          0,   2*(1+vyz)/Ey,        0,      0;...
            0,          0,          0,              0,    1/Gxy,      0;...
            0,          0,          0,              0,        0, 1/Gxy];
C = inv(SS);
%% formulate transformative (also its derivative) matrices
% subscript represents the axis which the coordinate is rotating about
% transofrmation matrix T
% inverse T
Tinv = [cos(theta)^2*cos(phi)^2         sin(theta)^2
cos(theta)^2*sin(phi)^2         -2*cos(theta)*sin(theta)*sin(phi)
cos(theta)^2*sin(2*phi)                  -
2*cos(theta)*cos(phi)*sin(theta);
        cos(phi)^2*sin(theta)^2         cos(theta)^2
sin(theta)^2*sin(phi)^2         sin(2*theta)*sin(phi)
sin(theta)^2*sin(2*phi)                   cos(phi)*sin(2*theta);
        sin(phi)^2                      0
cos(phi)^2                              0                              -
2*cos(phi)*sin(phi)                      0;
        -cos(phi)*sin(theta)*sin(phi)   0
cos(phi)*sin(theta)*sin(phi)    cos(theta)*cos(phi)
cos(2*phi)*sin(theta)                   -cos(theta)*sin(phi);
```

```
        -cos(theta)*cos(phi)*sin(phi)      0
cos(theta)*cos(phi)*sin(phi)      -cos(phi)*sin(theta)
cos(theta)*cos(2*phi)                         sin(theta)*sin(phi);
        cos(theta)*cos(phi)^2*sin(theta) -cos(theta)*sin(theta)
cos(theta)*sin(theta)*sin(phi)^2 cos(2*theta)*sin(phi)
2*cos(theta)*cos(phi)*sin(theta)*sin(phi) cos(2*theta)*cos(phi)];
% inverse & transpose T
Tinvtransp = Tinv';
% derivative of inverse T
DTinv = [-2*cos(phi)^2*cos(theta)*sin(theta),
2*cos(theta)*sin(theta),      -2*cos(theta)*sin(phi)^2*sin(theta),
2*sin(phi)*sin(theta)^2-2*cos(theta)^2*sin(phi), -
2*sin(2*phi)*cos(theta)*sin(theta),
2*cos(phi)*sin(theta)^2 - 2*cos(phi)*cos(theta)^2;
        2*cos(phi)^2*cos(theta)*sin(theta),                -
2*cos(theta)*sin(theta),      2*cos(theta)*sin(phi)^2*sin(theta),
2*cos(2*theta)*sin(phi),
2*sin(2*phi)*cos(theta)*sin(theta),
2*cos(2*theta)*cos(phi);
        0,                                                  0,
0,                                                 0,
0,                                                             0;
        -cos(phi)*cos(theta)*sin(phi),                      0,
cos(phi)*cos(theta)*sin(phi),                 -cos(phi)*sin(theta),
cos(2*phi)*cos(theta),
sin(phi)*sin(theta);
        cos(phi)*sin(phi)*sin(theta),                       0,
-cos(phi)*sin(phi)*sin(theta),                -cos(phi)*cos(theta),
-cos(2*phi)*sin(theta),
cos(theta)*sin(phi);
        cos(phi)^2*cos(theta)^2-cos(phi)^2*sin(theta)^2, sin(theta)^2-
cos(theta)^2,    cos(theta)^2*sin(phi)^2-sin(phi)^2*sin(theta)^2, -
2*sin(2*theta)*sin(phi),
2*cos(phi)*cos(theta)^2*sin(phi)-2*cos(phi)*sin(phi)*sin(theta)^2, -
2*sin(2*theta)*cos(phi)];
% double derivative of inverse T
% derivative of inverse & transpose T
DTinvtransp = DTinv';
% double derivative of inverse & transpose T
%% zero element matrices
switch icode
    case -1
        eldat_F = zeros(D.nenode*D.ndof,1);
        eldat_K = zeros(D.nenode*D.ndof);
    case 1
        eldat_F = 0;
        eldat_K = 0;
        eldat_KK = 0;
    case 2
        eldat_F = 0;
        eldat_K = 0;
end
%% evaluate element matrices
% jac = eye(2);     % identity matrix
% jinv = eye(2);    % inv of identity is identity
%
% define weighting factors and gauss points
```

```matlab
wij = [2 0 0;1 1 0;5/9 8/9 5/9]';
xij = [0 0 0;-1/sqrt(3) 1/sqrt(3) 0;-sqrt(3/5) 0 sqrt(3/5)]';
%
ngpt = 2;
for ii = 1:ngpt;
    for jj = 1:ngpt;
        for kk = 1:ngpt
            w = wij(ii,ngpt)*wij(jj,ngpt)*wij(kk,ngpt);
            xi = xij(ii,ngpt); eta = xij(jj,ngpt); zeta = xij(kk,ngpt);
            % define matrix dN
            dN = 1/8*[-(eta-1)*(zeta-1),(eta-1)*(zeta-1),-
(eta+1)*(zeta-1),(eta+1)*(zeta-1),(eta-1)*(zeta+1),-(eta-
1)*(zeta+1),(eta+1)*(zeta+1),-(eta+1)*(zeta+1);
                      -(xi-1)*(zeta-1),(xi+1)*(zeta-1),-(xi+1)*(zeta-
1),(xi-1)*(zeta-1),(xi-1)*(zeta+1),-(xi+1)*(zeta+1),(xi+1)*(zeta+1),-
(xi-1)*(zeta+1);
                      -(eta-1)*(xi-1),(eta-1)*(xi+1),-
(eta+1)*(xi+1),(eta+1)*(xi-1),(eta-1)*(xi-1),-(eta-
1)*(xi+1),(eta+1)*(xi+1),-(eta+1)*(xi-1)];
            % jacobian, its inverse & determinant
            J = dN*xloc;
            detj = det(J);
            % define strain displacement matrix
            B = J\dN;
            B = [B(1,1) 0       0      B(1,2) 0       0      B(1,3) 0
0      B(1,4) 0       0      B(1,5) 0        0      B(1,6) 0       0
B(1,7) 0       0      B(1,8) 0       0;
                 0      B(2,1) 0       0      B(2,2) 0       0
B(2,3) 0       0      B(2,4) 0       0      B(2,5) 0        0      B(2,6)
0      0      B(2,7) 0       0      B(2,8) 0;
                 0      0      B(3,1) 0       0      B(3,2) 0       0
B(3,3) 0       0      B(3,4) 0       0      B(3,5) 0        0      B(3,6)
0      0      B(3,7) 0       0      B(3,8);
                 0      B(3,1) B(2,1) 0      B(3,2) B(2,2) 0
B(3,3) B(2,3) 0      B(3,4) B(2,4) 0       B(3,5) B(2,5) 0      B(3,6)
B(2,6) 0      B(3,7) B(2,7) 0      B(3,8) B(2,8);
                 B(3,1) 0      B(1,1) B(3,2) 0      B(1,2) B(3,3) 0
B(1,3) B(3,4) 0      B(1,4) B(3,5) 0       B(1,5) B(3,6) 0      B(1,6)
B(3,7) 0      B(1,7) B(3,8) 0      B(1,8);
                 B(2,1) B(1,1) 0      B(2,2) B(1,2) 0      B(2,3)
B(1,3) 0      B(2,4) B(1,4) 0      B(2,5) B(1,5) 0      B(2,6) B(1,6)
0      B(2,7) B(1,7) 0      B(2,8) B(1,8) 0];
            %
            ke = B'*Tinv*C*Tinvtransp*B; ke = (ke + ke')/2;
            switch icode
                case -1
                    % compute elemental stiffness matrix
                    eldat_K = eldat_K + eprop^p*ke*detj*w;
                case 1
                    % comopute c at element level
                    eldat_F = eldat_F +
trace(eprop^p*de'*ke*de*detj*w);
                    % compute dc at element level w.r.t density
                    eldat_K = eldat_K - trace(p*eprop^(p-
1)*de'*ke*de*detj*w);
                    % 1st deriative w.r.t theta
```

138

```matlab
                          eldat_KK = eldat_KK -
trace(eprop^p*de'*(B'*DTinv*C*Tinvtransp*B + ...

B'*Tinv*C*DTinvtransp*B)*de*detj*w);
              end
          end
      end
end


function [c,ceq,Gc,Geq] = top_nonlcon(x)
% This file deals only with the passive element
global passive_ele
rhop = x(1:length(x)/2);
c = [];
ceq = [rhop(passive_ele) - 0.6];
if nargout > 2
    Gc = [];
    Geq =
sparse([passive_ele]',[1:length(passive_ele)],ones(1,length(passive_ele
)),length(x),length(ceq));
end


function stop = outfun(x, optimValues, state)
stop = false;
% this file mainly calls the plot command the design variable during
the iteration
global  history nelx nely nelz
%
switch state
    case 'init'
        tic
    case 'iter'
        % plot the design variables
        t = toc; plot_outplot(x,t,optimValues,1);
        % display
        disp([' Iter:'   sprintf(' %4i ',optimValues.iteration)...
              ' Obj:' sprintf(' %6.4f ',optimValues.fval) ...
              ' Vol:'    sprintf(' %6.4f
',sum(sum(sum(x(1:length(x)/2))))/(nelx*nely*nelz)) ...
              'Step:' sprintf(' %6.4f ',optimValues.stepsize) ...
              'Time Elapsed:' sprintf(' %6.4f ',t)]);
    case 'done'
        % record the iteration
        history.iter = [history.iter; optimValues.iteration];
        %
        t = toc; plot_outplot(x,t,optimValues)
end


function ieldof = get_ieldof(lnodes,D)
% this mfile gets element equation numbers from global equation number
% array
nenode = D.nenode;  % no. of nodes per element
```

```
ndof = D.ndof;        % no. of dofs per node
%
jj = 0;
for j1 = 1:D.nenode;
    for j2 = 1:D.ndof;
        jj = jj + 1;
        ieldof(jj) = lnodes(j1)*D.ndof + j2 - D.ndof;
    end
end
end


function plot_outplot(x,t,optimresult,a)
% this file does the plot job
global ielem video X nelx nely nelz phi
%
rho = x(1:length(x)/2); theta = x(length(x)/2+1:end); phi0 = phi(:);
XX = X;
%
figure(2); ielem1 = ielem((find(rho>.5)),:);
%
xx = XX(:,1); yy = XX(:,2); zz = XX(:,3);
%
theta1 = theta(find(rho>.5),:); phi1 = phi0(find(rho>.5),:);
% define the fiber segmentss
center_xx = mean(xx(ielem1),2); center_yy = mean(yy(ielem1),2);
center_zz = mean(zz(ielem1),2);
tip1x = center_xx + 0.25.*cos(theta1).*cos(phi1); tip1y = center_yy +
0.25*sin(theta1); tip1z = center_zz - 0.25.*cos(theta1).*sin(phi1);
tip2x = center_xx - 0.25.*cos(theta1).*cos(phi1); tip2y = center_yy -
0.25*sin(theta1); tip2z = center_zz + 0.25.*cos(theta1).*sin(phi1);
% cooridnate transformation
[row,col] = size(ielem1);
if nely >= nelz
    xxx = [center_xx;tip1x;tip2x]; yyy = [center_yy;tip1y;tip2y]; zzz =
[center_zz;tip1z;tip2z];
    XXX = [xxx,yyy,zzz]; XXX(:,[2,3]) = XXX(:,[3,2]); XXX(:,2) = -
XXX(:,2);
    xxx = reshape(XXX(:,1),row,3); yyy = reshape(XXX(:,2),row,3); zzz =
reshape(XXX(:,3),row,3);
    center_xx = xxx(:,1); tip1x = xxx(:,2); tip2x = xxx(:,3);
    center_yy = yyy(:,1); tip1y = yyy(:,2); tip2y = yyy(:,3);
    center_zz = zzz(:,1); tip1z = zzz(:,2); tip2z = zzz(:,3);
end
% plot fiber orientaiton
fiber_x = [tip2x';center_xx';tip1x']; fiber_y =
[tip2y';center_yy';tip1y']; fiber_z = [tip2z';center_zz';tip1z'];
plot3(fiber_x,fiber_y,fiber_z,'-r','linewidth',2);
title(['time: ',num2str(t),' sec;',' iter:
',num2str(optimresult.iteration),';  compilance:
',num2str(optimresult.fval),'  Mnd(%): ',num2str(sum(4*rho(:).*(1-
rho(:)))/length(rho)*100)]);
hold on;
% cooridnate transformation
if nely >= nelz
    XX(:,[2,3]) = XX(:,[3,2]); XX(:,2) = -XX(:,2);
```

```
end
% plot density
rho1 = repmat(rho(find(rho>.5),:),6,1);
faces1 = [ielem1(:,[1 2 3 4]);
    ielem1(:,[5 6 7 8]);
    ielem1(:,[4 3 7 8]);
    ielem1(:,[1 2 6 5]);
    ielem1(:,[1 5 8 4]);
    ielem1(:,[2 6 7 3])];
% plot the 3D elements with material distribution
patch('Faces',faces1,'Vertices',XX,'FaceVertexCData',1-
rho1,'FaceColor','flat','FaceAlpha',.5);
axis equal; axis tight; axis off; colormap gray; hold off;
view([30 30])
drawnow;
%
if nargin > 3
frame = getframe(figure(2)); writeVideo(video,frame);
end


function [utot,ptot] = proc0(D,rho,theta,phi)
% This file does FEA matrix solving
global ielem iprops iforce force ieqn
global Kff Kfe Kee Kffi Kfei Keei Kffj Kfej Keej Uf Ue Ff Fe
%
rho = rho(:); theta = theta(:); phi = phi(:);
% compute and assemble element stiffness matrix and element load vector
for i = 1:D.nel;
    elemnodes = ielem(i,:);
    % get element dof number array
    ieldof = get_ieldof(elemnodes,D);
    % cacluate element stiffness matrix and load vector
    [eldat_F,eldat_K] =
feval(D.elname,rho(i),theta(i),phi(i),elemnodes,D,-1);
    % assemble element stiffness matrix and load vector
    assemkp_sparse(eldat_F,eldat_K,ieqn(ieldof),D);
end
%
% apply nodal forces
k = 0;
for j = 1:D.loadcase
    for i = 1:D.nforce;
        k = k + 1;
        num = iforce(k,1)*D.ndof + iforce(k,2) - D.ndof;
        Ff(ieqn(num),j) = Ff(ieqn(num),j) + force(k);
    end
end
% define sparse matrices
Kff = sparse(Kffi,Kffj,Kff,D.nnode*D.ndof-D.ndisp,D.nnode*D.ndof-
D.ndisp);
Kee = sparse(Keei,Keej,Kee,D.ndisp,D.ndisp);
Kfe = sparse(Kfei,Kfej,Kfe,D.nnode*D.ndof-D.ndisp,D.ndisp);
%
% solve system of equations
Uf = Kff\(Ff - Kfe*Ue);
```

```
% compute nodal reactions
Fe = Kee*Ue + Kfe'*Uf;
%
% complete global displacement and force vector
%
for j = 1:D.loadcase
    for i = 1:D.nnode*D.ndof;
        if (ieqn(i) < 0);
            utot(i,j) = Ue(-ieqn(i),j);
            ptot(i,j) = Fe(-ieqn(i),j);
        else
            utot(i,j) = Uf(ieqn(i),j);
            ptot(i,j) = Ff(ieqn(i),j);
        end
    end
end
end


function ieqn = initialize(D)
%
global idisp disp ielem
global Kff Kfe Kee Kffi Kfei Keei Kffj Kfej Keej Uf Ue Ff Fe Kindex
%
nnode = D.nnode;    % no. of nodes in model
ndof = D.ndof;      % no. of dofs per node
%
%set total degrees of freedom
ndoftot = D.nnode*D.ndof;
%
% initialize ieqn
ieqn = zeros(1,ndoftot);
%
% identify fixed degrees of freedom
iessential = 0;
for i = 1:D.ndisp;
    iessential = iessential + 1;
    num = idisp(i,1)*D.ndof + idisp(i,2) - D.ndof;
    ieqn(num) = -iessential;
end
%
% identify free degrees of freedom
ifree = 0;
for i = 1:ndoftot;
    if ieqn(i) == 0;
        ifree = ifree + 1;
        ieqn(i) = ifree;
    end
end
%
% determine number of element free and fixed nodes
ndofff = 0; ndofee = 0; ndoffe = 0;
for i = 1:D.nel
    jj = 0;
    %idof = zeros(D.nenode*D.ndof,1);
    for j1 = 1:D.nenode;
```

```
        for j2 = 1:D.ndof;
            jj = jj + 1;
            numj = ielem(i,j1)*D.ndof + j2 - D.ndof;
            %idof(jj,1) = numj;
            ieleqn(jj) = ieqn(numj);
        end
    end
    ndofff = ndofff + (sum(ieleqn>0))^2;
    ndofee = ndofee + (sum(ieleqn<0))^2;
    ndoffe = ndoffe + (sum(ieleqn>0)*sum(ieleqn<0));
end
%
% initialize global matrices;
Kindex = [0 0 0];
Kffi = zeros(ndofff,1); Kffj = zeros(ndofff,1); Kff = zeros(ndofff,1);
Keei = zeros(ndofee,1); Keej = zeros(ndofee,1); Kee = zeros(ndofee,1);
Kfei = zeros(ndoffe,1); Kfej = zeros(ndoffe,1); Kfe = zeros(ndoffe,1);
% Kff = zeros(D.nnode*D.ndof-D.ndisp);
% Kfe = zeros(D.nnode*D.ndof-D.ndisp,D.ndisp);
% Kee = zeros(D.ndisp,D.ndisp);
Uf = zeros(D.nnode*D.ndof-D.ndisp,D.loadcase);
Ue = repmat(disp',1,D.loadcase);
Ff = zeros(D.nnode*D.ndof-D.ndisp,D.loadcase);
Fe = zeros(D.ndisp,D.loadcase);
end




function plot_layer(x)
% this file plots layer-by-layer plot in the print plane
global ielem X nelx nely nelz
%
ielem1 = ielem(1:nelx*nely,1:4);
%
rho = x(1:length(x)/2); theta = x(length(x)/2+1:end);
rho = reshape(rho,nely,nelx,nelz); theta =
reshape(theta,nely,nelx,nelz);
% ielem1 = ielem((find(rho>.5)),:); rho1 = rho(find(rho>.5),:); theta1
= theta(find(rho>.5),:);
%
XX = X; xx = XX(:,1); yy = XX(:,2); zz = XX(:,3);

keyboard
figure;
for i = 1:nelz % change view direciton
    subplot(nelz,1,i)
    rho_plot = 1-rho(:,:,i); rho_plot = rho_plot(:);
    theta_plot = theta(:,:,i); theta_plot = theta_plot(:);
    % plot density
    patch(xx(ielem1)',yy(ielem1)',rho_plot,'edgecolor','none');
    colormap gray; caxis([0 1]); axis equal; axis tight; axis off; hold
on;
    % plot fiber angle
    center_xx = mean(xx(ielem1),2); center_yy = mean(yy(ielem1),2);
    tip1x = center_xx + 0.25.*cos(theta_plot); tip1y = center_yy +
0.25*sin(theta_plot);
```

```
    tip2x = center_xx - 0.25.*cos(theta_plot); tip2y = center_yy -
0.25*sin(theta_plot);
    fiber_x = [tip2x';center_xx';tip1x']; fiber_y =
[tip2y';center_yy';tip1y'];
    plot(fiber_x,fiber_y,'-w','linewidth',2); hold on
end
% save the solution variables into a file
%   save('20x10x6_optimized_data.dat','x','-ascii')




function assemkp_sparse(eldat_F,eldat_K,ieleqn,D)
% This function assembles the global load vector (case 1) and
% global stiffness matrices (case 2) from the element data in
% eldat and ieleqn. Free and essential dofs are accounted for
% via the sign on m and n.
%
global Kff Kffi Kffj Kee Keei Keej Kfe Kfei Kfej Ff Kindex
%
neldoftot = D.ndof*D.nenode;    % total no. of dofs per element
free = find(ieleqn>0);          % element free dofs
essential = find(ieleqn<0);     % element essential dofs
%
% assembly global arrays
% assemble load vector
% Ff(ieleqn(free)) = Ff(ieleqn(free))+eldat_F(free);
%
% assemble stiffness matrix
% define index variables
Lff = length(free);
Lee = length(essential);
Lfe = (neldoftot - Lff - Lee)/2;
eldati = repmat(ieleqn',1,neldoftot);
eldatj = repmat(ieleqn,neldoftot,1);
% extract free-free submatrix components
Eff = reshape(eldat_K(free,free),Lff^2,1);
Effi = reshape(eldati(free,free),Lff^2,1);
Effj = reshape(eldatj(free,free),Lff^2,1);
% extract essential-essential submatrix components
Eee = reshape(eldat_K(essential,essential),Lee^2,1);
Eeei = -reshape(eldati(essential,essential),Lee^2,1);
Eeej = -reshape(eldatj(essential,essential),Lee^2,1);
% extract assemble free-essential submatrix components
Efe = reshape(eldat_K(free,essential),Lff*Lee,1);
Efei = abs(reshape(eldati(free,essential),Lff*Lee,1));
Efej = abs(reshape(eldatj(free,essential),Lff*Lee,1));
% assemble components and indices into sparse form
Kff(Kindex(1)+1:(Kindex(1)+Lff^2)) = Eff;
Kffi(Kindex(1)+1:(Kindex(1)+Lff^2)) = Effi;
Kffj(Kindex(1)+1:(Kindex(1)+Lff^2)) = Effj;
Kee(Kindex(2)+1:(Kindex(2)+Lee^2)) = Eee;
Keei(Kindex(2)+1:(Kindex(2)+Lee^2)) = Eeei;
Keej(Kindex(2)+1:(Kindex(2)+Lee^2)) = Eeej;
Kfe(Kindex(3)+1:(Kindex(3)+Lff*Lee)) = Efe;
Kfei(Kindex(3)+1:(Kindex(3)+Lff*Lee)) = Efei;
Kfej(Kindex(3)+1:(Kindex(3)+Lff*Lee)) = Efej;
```

```
% update Kindex
Kindex = Kindex + [Lff^2 Lee^2 Lff*Lee];
end
```

BIBLIOGRAPHY

[1]   Wohlers, T., 2016, *Wohlers Report 2016*, Wohlers Associates, Inc.

[2]   Tekinalp, H. L., Kunc, V., Velez-Garcia, G. M., Duty, C. E., Love, L. J., Naskar, A. K., Blue, C. A., and Ozcan, S., 2014, "Highly Oriented Carbon Fiber–polymer Composites via Additive Manufacturing," Compos. Sci. Technol., **105**, pp. 144–150.

[3]   "Shelby Cobra" [Online]. Available: http://web.ornl.gov/sci/manufacturing/shelby/. [Accessed: 20-Jan-2017].

[4]   Hoglund, R., and Smith, D. E., "CONTINUOUS FIBER ANGLE TOPOLOGY OPTIMIZATION FOR POLYMER FUSED FILAMENT FABRICATION."

[5]   Nomura, T., Dede, E. M., Lee, J., Yamasaki, S., Matsumori, T., Kawamoto, A., and Kikuchi, N., 2015, "General Topology Optimization Method with Continuous and Discrete Orientation Design Using Isoparametric Projection," Int. J. Numer. Methods Eng., **101**(8), pp. 571–605.

[6]   "Structural Analysis Solver for Linear & Nonlinear | Altair OptiStruct" [Online]. Available: http://www.altairhyperworks.com/product/OptiStruct. [Accessed: 04-May-2017].

[7]   "Introduction to Optimum Design - 3rd Edition" [Online]. Available: https://www.elsevier.com/books/introduction-to-optimum-design/arora/978-0-12-381375-6. [Accessed: 01-Feb-2017].

[8]   Wohlers, T., and Gornet, T., 2011, "History of Additive Manufacturing," Wohlers Rep. Addit. Manuf. 3D Print. State Ind. Annu. Worldw. Prog. Rep.

[9]   2012, "ASTM F2792 - 12a Standard Terminology for Additive Manufacturing Technologies."

[10]   2009, "Fused Deposition Modeling (FDM)," Kyle Stetz Rapid Prototyp. Study.

[11]   Sung- Hoon Ahn, Michael Montero, Dan Odell, Shad Roundy, and Paul K. Wright, 2002, "Anisotropic Material Properties of Fused Deposition Modeling ABS," Rapid Prototyp. J., **8**(4), pp. 248–257.

[12]   Ziemian, C., Sharma, M., and Ziemi, S., 2012, "Anisotropic Mechanical Properties of ABS Parts Fabricated by Fused Deposition Modelling," *Mechanical Engineering*, M. Gokcek, ed., InTech.

[13] Schöppner, V., and KTP, K. P., "MECHANICAL PROPERTIES OF FUSED DEPOSITION MODELING PARTS MANUFACTURED WITH ULTEM* 9085."

[14] Jiang, D., and Smith, D. E., "MECHANICAL BEHAVIOR OF CARBON FIBER COMPOSITES PRODUCED WITH FUSED FILAMENT FABRICATION."

[15] Perez, A. R. T., Roberson, D. A., and Wicker, R. B., 2014, "Fracture Surface Analysis of 3D-Printed Tensile Specimens of Novel ABS-Based Materials," J. Fail. Anal. Prev., **14**(3), pp. 343–353.

[16] Hwang, S., Reyes, E. I., Moon, K., Rumpf, R. C., and Kim, N. S., 2014, "Thermo-Mechanical Characterization of Metal/Polymer Composite Filaments and Printing Parameter Study for Fused Deposition Modeling in the 3D Printing Process," J. Electron. Mater., **44**(3), pp. 771–777.

[17] Zhong, W., Li, F., Zhang, Z., Song, L., and Li, Z., 2001, "Short Fiber Reinforced Composites for Fused Deposition Modeling," Mater. Sci. Eng. A, **301**(2), pp. 125–130.

[18] Gray, R. W., Baird, D. G., and Bøhn, J. H., 1998, "Thermoplastic Composites Reinforced with Long Fiber Thermotropic Liquid Crystalline Polymers for Fused Deposition Modeling," Polym. Compos., **19**(4), pp. 383–394.

[19] Shofner, M. L., Lozano, K., Rodríguez-Macías, F. J., and Barrera, E. V., 2003, "Nanofiber-Reinforced Polymers Prepared by Fused Deposition Modeling," J. Appl. Polym. Sci., **89**(11), pp. 3081–3090.

[20] Dul, S., Fambri, L., and Pegoretti, A., 2016, "Fused Deposition Modelling with ABS–graphene Nanocomposites," Compos. Part Appl. Sci. Manuf., **85**, pp. 181–191.

[21] Ning, F., Cong, W., Qiu, J., Wei, J., and Wang, S., 2015, "Additive Manufacturing of Carbon Fiber Reinforced Thermoplastic Composites Using Fused Deposition Modeling," Compos. Part B Eng., **80**, pp. 369–378.

[22] Love, L. J., Kunc, V., Rios, O., Duty, C. E., Elliott, A. M., Post, B. K., Smith, R. J., and Blue, C. A., 2014, "The Importance of Carbon Fiber to Polymer Additive Manufacturing," J. Mater. Res., **29**(17), pp. 1893–1898.

[23] Ning, F., Cong, W., Qiu, J., Wei, J., and Wang, S., 2015, "Additive Manufacturing of Carbon Fiber Reinforced Thermoplastic Composites Using Fused Deposition Modeling," Compos. Part B Eng., **80**, pp. 369–378.

[24] "BAAM Specifications," Cincinnati Inc. [Online]. Available: http://www.e-ci.com/baam-specifications/. [Accessed: 02-Dec-2016].

[25] Post, B., Lloyd, P. D., Lindahl, J., Lind, R. F., Love, L. J., and Kunc, V., 2016, *The Economics of Big Area Addtiive Manufacturing*, Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States). Manufacturing Demonstration Facility (MDF).

[26] Spinnie, N. K. 1991-, 2016, "Large Scale Fused Deposition Modeling : The Effect of Process Parameters on Bead Geometry.," Thesis.

[27] Zhu, J.-H., Zhang, W.-H., and Xia, L., 2016, "Topology Optimization in Aircraft and Aerospace Structures Design," Arch. Comput. Methods Eng., **23**(4), pp. 595–622.

[28] Beghini, L. L., Beghini, A., Katz, N., Baker, W. F., and Paulino, G. H., 2014, "Connecting Architecture and Engineering through Structural Topology Optimization," Eng. Struct., **59**, pp. 716–726.

[29] Wang, L., Basu, P. K., and Leiva, J. P., 2004, "Automobile Body Reinforcement by Finite Element Optimization," Finite Elem. Anal. Des., **40**(8), pp. 879–893.

[30] Bendsøe, M. P., and Kikuchi, N., 1988, "Generating Optimal Topologies in Structural Design Using a Homogenization Method," Comput. Methods Appl. Mech. Eng., **71**(2), pp. 197–224.

[31] Suzuki, K., and Kikuchi, N., 1991, "A Homogenization Method for Shape and Topology Optimization," Comput. Methods Appl. Mech. Eng., **93**(3), pp. 291–318.

[32] Xie, Y. M., and Steven, G. P., 1993, "A Simple Evolutionary Procedure for Structural Optimization," Comput. Struct., **49**(5), pp. 885–896.

[33] Querin, O. m., Steven, G. p., and Xie, Y. m., 1998, "Evolutionary Structural Optimisation (ESO) Using a Bidirectional Algorithm," Eng. Comput., **15**(8), pp. 1031–1048.

[34] Mlejnek, H. P., 1992, "Some Aspects of the Genesis of Structures," Struct. Optim., **5**(1-2), pp. 64–69.

[35] Bendsøe, M. P., 1989, "Optimal Shape Design as a Material Distribution Problem," Struct. Optim., **1**(4), pp. 193–202.

[36] Zhou, M., and Rozvany, G. I. N., 1991, "The COC Algorithm, Part II: Topological, Geometrical and Generalized Shape Optimization," Comput. Methods Appl. Mech. Eng., **89**(1–3), pp. 309–336.

[37] Sigmund, O., 2001, "A 99 Line Topology Optimization Code Written in Matlab," Struct. Multidiscip. Optim., **21**(2), pp. 120–127.

[38] Reddy, J., 2005, *An Introduction to the Finite Element Method*, McGraw-Hill Education.

[39] *Topology Optimization - Theory, Methods, and | Martin Philip Bendsoe | Springer.*

[40] Pedersen, P., 1980, *The Integrated Approach of FEM-SLP for Solving Problems of Optimal Design*, Department of Solid Mechanics, Technical University of Denmark.

[41] Svanberg, K., 1987, "The Method of Moving Asymptotes—a New Method for Structural Optimization," Int. J. Numer. Methods Eng., **24**(2), pp. 359–373.

[42] Hoglund, R., and Smith, D. E., 2015, "Non-Isotropic Material Distribution Topology Optimization for Fused Deposition Modeling Products," *Proceeding of the 2015 Solid Freeform Fabrication Symposium, Austin, TX, Pg*.

[43] Díaz, A., and Sigmund, O., 1995, "Checkerboard Patterns in Layout Optimization," Struct. Optim., **10**(1), pp. 40–45.

[44] Sigmund, O., and Petersson, J., 1998, "Numerical Instabilities in Topology Optimization: A Survey on Procedures Dealing with Checkerboards, Mesh-Dependencies and Local Minima," Struct. Optim., **16**(1), pp. 68–75.

[45] Díaz, A., and Sigmund, O., 1995, "Checkerboard Patterns in Layout Optimization," Struct. Optim., **10**(1), pp. 40–45.

[46] "Stability of Finite Element Model for Distributed Parameter Optimization and Topology Design," ResearchGate.

[47] Haber, R. B., Jog, C. S., and Bendsøe, M. P., 1996, "A New Approach to Variable-Topology Shape Design Using a Constraint on Perimeter," Struct. Optim., **11**(1-2), pp. 1–12.

[48] Ambrosio, L., and Buttazzo, G., 1993, "An Optimal Design Problem with Perimeter Penalization," Calc. Var. Partial Differ. Equ., **1**(1), pp. 55–69.

[49] Petersson, J., and Sigmund, O., 1998, "Slope Constrained Topology Optimization," Int. J. Numer. Methods Eng., **41**(8), pp. 1417–1434.

[50] Sigmund, O., 1994, *Design of Materials Structures Using Topology Optimization*, Department of Solid Mechanics, Technical University of Denmark.

[51] Bruns, T. E., and Tortorelli, D. A., 2001, "Topology Optimization of Non-Linear Elastic Structures and Compliant Mechanisms," Comput. Methods Appl. Mech. Eng., **190**(26–27), pp. 3443–3459.

[52] Sigmund, O., 2007, "Morphology-Based Black and White Filters for Topology Optimization," Struct. Multidiscip. Optim., **33**(4-5), pp. 401–424.

[53] Sigmund, O., 1997, "On the Design of Compliant Mechanisms Using Topology Optimization," Mech. Struct. Mach., **25**(4), pp. 493–524.

[54] Lazarov, B. S., and Sigmund, O., 2011, "Filters in Topology Optimization Based on Helmholtz-Type Differential Equations," Int. J. Numer. Methods Eng., **86**(6), pp. 765–781.

[55] Lazarov, B. S., and Sigmund, O., 2009, "Sensitivity Filters In Topology Optimisation As A Solution To Helmholtz Type Differential Equation," WCSMO-8 8th World Congr. Struct. Multidiscip. Optim. - 8 2009 LNEC Lisbon Port.

[56] Kawamoto, A., Matsumori, T., Yamasaki, S., Nomura, T., Kondoh, T., and Nishiwaki, S., 2011, "Heaviside Projection Based Topology Optimization by a PDE-Filtered Scalar Function," Struct. Multidiscip. Optim., **44**(1), pp. 19–24.

[57] Pratt, W. K., 1991, *Digital Image Processing (2Nd Ed.)*, John Wiley & Sons, Inc., New York, NY, USA.

[58] KREISSELMEIER, G., and STEINHAUSER, R., 1983, "Application of Vector Performance Optimization to a Robust Control Loop Design for a Fighter Aircraft," Int. J. Control, **37**(2), pp. 251–284.

[59] Guest, J. K., Prévost, J. H., and Belytschko, T., 2004, "Achieving Minimum Length Scale in Topology Optimization Using Nodal Design Variables and Projection Functions," Int. J. Numer. Methods Eng., **61**(2), pp. 238–254.

[60] Silva Junior, F. I. da, and Álamo, J. R., 2012, "Adapting the Simp Model for Topology Optimization of Biomechanical Structures," Pan American Congress of Applied Mechanics.

[61] Sundararajan, V. G., 2010, "Topology Optimization for Additive Manufacturing of Customized Meso-Structures Using Homogenization and Parametric Smoothing Functions," thesis.

[62] Zhang, P., Toman, J., Yu, Y., Biyikli, E., Kirca, M., Chmielus, M., and To, A. C., 2015, "Efficient Design-Optimization of Variable-Density Hexagonal Cellular Structure by Additive Manufacturing: Theory and Validation," J. Manuf. Sci. Eng., **137**(2), pp. 021004–021004–8.

[63] Burblies, A., Busse, M., Paulino, G. H., Pindera, M., Dodds, R. H., Rochinha, F. A., Dave, E., and Chen, L., 2008, "Computer Based Porosity Design by Multi Phase Topology Optimization," AIP Conf. Proc., **973**(1), pp. 285–290.

[64] Brackett, D., Ashcroft, I., and Hague, R., 2011, "Topology Optimization for Additive Manufacturing," *Proceedings of the Solid Freeform Fabrication Symposium, Austin, TX*, pp. 348–362.

[65] Rezaie, R., Badrossamay, M., Ghaie, A., and Moosavi, H., 2013, "Topology Optimization for Fused Deposition Modeling Process," Procedia CIRP, **6**, pp. 521–526.

[66] Gaynor, A. T., Meisel, N. A., Williams, C. B., and Guest, J. K., 2014, "Multiple-Material Topology Optimization of Compliant Mechanisms Created Via PolyJet Three-Dimensional Printing," J. Manuf. Sci. Eng., **136**(6), pp. 061015–061015–10.

[67] Lee, D.-K., Hoan, D.-Q., Tuan, L.-A., Lee, K.-H., and Lee, J.-H., "Topology Optimization of Laminated Composite Plates with Isotropic and Orthotropic Material."

[68] Harris, C. E., Starnes, J. H., and Shuart, M. J., 2002, "Design and Manufacturing of Aerospace Composite Structures, State-of-the-Art Assessment," J. Aircr., **39**(4), pp. 545–560.

[69] SCHMIT, L. A., and FLEURY, C., 1980, "Discrete-Continuous Variable Structural Synthesis Using Dual Methods," AIAA J., **18**(12), pp. 1515–1524.

[70] OLSEN, G. R., and VANDERPLAATS, G. N., 1989, "Method for Nonlinear Optimization with Discrete Design Variables," AIAA J., **27**(11), pp. 1584–1589.

[71] RICHE, R. L., and HAFTKA, R. T., 1993, "Optimization of Laminate Stacking Sequence for Buckling Load Maximization by Genetic Algorithm," AIAA J., **31**(5), pp. 951–956.

[72] HAFTKA, R. T., and WALSH, J. L., 1992, "Stacking-Sequence Optimization for Buckling of Laminated Plates by Integer Programming," AIAA J., **30**(3), pp. 814–819.

[73] Vanderplaats, G. N., 1982, "Structural Optimization-Past, Present, and Future," AIAA J., **20**(7), pp. 992–1000.

[74] Gibiansky, L. V., and Cherkaev, A. V., 1997, "Microstructures of Composites of Extremal Rigidity and Exact Bounds on the Associated Energy Density," *Topics in the Mathematical Modelling of Composite Materials*, A. Cherkaev, and R. Kohn, eds., Birkhäuser Boston, pp. 273–317.

[75] Pedersen, P., 1989, "On Optimal Orientation of Orthotropic Materials," Struct. Optim., **1**(2), pp. 101–106.

[76] Pedersen, P., 1990, "Bounds on Elastic Energy in Solids of Orthotropic Materials," Struct. Optim., **2**(1), pp. 55–63.

[77] Pedersen, P., 1991, "On Thickness and Orientational Design with Orthotropic Materials," Struct. Optim., **3**(2), pp. 69–78.

[78] Gea, H. C., and Luo, J. H., 2004, "On the Stress-Based and Strain-Based Methods for Predicting Optimal Orientation of Orthotropic Materials," Struct. Multidiscip. Optim., **26**(3-4), pp. 229–234.

[79] Bruyneel, M., and Fleury, C., 2002, "Composite Structures Optimization Using Sequential Convex Programming," Adv. Eng. Softw., **33**(7–10), pp. 697–711.

[80] Pedersen, P., 2004, "Examples of Density, Orientation, and Shape-Optimal 2D-Design for Stiffness And/or Strength with Orthotropic Materials," Struct. Multidiscip. Optim., **26**(1-2), pp. 37–49.

[81] Lindgaard, E., and Lund, E., 2011, "Optimization Formulations for the Maximum Nonlinear Buckling Load of Composite Structures," Struct. Multidiscip. Optim., **43**(5), pp. 631–646.

[82] Ansola, R., Canales, J., Tarrago, J. A., and Rasmussen, J., 2002, "On Simultaneous Shape and Material Layout Optimization of Shell Structures," Struct. Multidiscip. Optim., **24**(3), pp. 175–184.

[83] Ben-Tal, A., Kovara, M., Nemirovski, A., and Zowe, J., 2000, "Free Material Design via Semidefinite Programming: The Multiload Case with Contact Conditions," SIAM Rev., **42**(4), pp. 695–715.

[84] Zowe, J., Kočvara, M., and Bendsøe, M. P., 1997, "Free Material Optimization via Mathematical Programming," Math. Program., **79**(1-3), pp. 445–466.

[85] Haslinger, J., Kočvara, M., Leugering, G., and Stingl, M., 2010, "Multidisciplinary Free Material Optimization," SIAM J. Appl. Math., **70**(7), pp. 2709–2728.

[86] Kočvara, M., and Zowe, J., 2002, "Free Material Optimization: An Overview," *Trends in Industrial and Applied Mathematics*, A.H. Siddiqi, and M. Kočvara, eds., Springer US, pp. 181–215.

[87] Kočvara, M., Stingl, M., and Zowe, J., 2008, "Free Material Optimization: Recent Progress," Optimization, **57**(1), pp. 79–100.

[88] Stegmann, J., and Lund, E., 2005, "Discrete Material Optimization of General Composite Shell Structures," Int. J. Numer. Methods Eng., **62**(14), pp. 2009–2027.

[89] Gao, T., Zhang, W., and Duysinx, P., 2012, "A Bi-Value Coding Parameterization Scheme for the Discrete Optimal Orientation Design of the Composite Laminate," Int. J. Numer. Methods Eng., **91**(1), pp. 98–114.

[90] Bruyneel, M., 2011, "SFP—a New Parameterization Based on Shape Functions for Optimal Material Selection: Application to Conventional Composite Plies," Struct. Multidiscip. Optim., **43**(1), pp. 17–27.

[91] Jia, H. P., Jiang, C. D., Li, G. P., Mu, R. Q., Liu, B., and Jiang, C. B., 2008, "Topology Optimization of Orthotropic Material Structure," Mater. Sci. Forum, **575-578**, pp. 978–989.

[92] Setoodeh, S., Abdalla, M. M., and Gürdal, Z., 2005, "Combined Topology and Fiber Path Design of Composite Layers Using Cellular Automata," Struct. Multidiscip. Optim., **30**(6), pp. 413–421.

[93] Hoglund, R. M. 1992-, 2016, "An Anisotropic Topology Optimization Method for Carbon Fiber-Reinforced Fused Filament Fabrication.," Thesis.

[94] "Download-Archive," Top3d.

[95] Olhoff, N., Rønholt, E., and Scheel, J., 1998, "Topology Optimization of Three-Dimensional Structures Using Optimum Microstructures," Struct. Optim., **16**(1), pp. 1–18.

[96] Jacobsen, J. B., Olhoff, N., and Rønholt, E., 1998, "Generalized Shape Optimization of Three-Dimensional Structures Using Materials with Optimum Microstructures," Mech. Mater., **28**(1–4), pp. 207–225.

[97] Fernandes, P., Guedes, J. M., and Rodrigues, H., 1999, "Topology Optimization of Three-Dimensional Linear Elastic Structures with a Constraint on 'perimeter,'" Comput. Struct., **73**(6), pp. 583–594.

[98] Díaz, A., and Lipton, R., 1997, "Optimal Material Layout for 3D Elastic Structures," Struct. Optim., **13**(1), pp. 60–64.

[99] Beckers, M., 1999, "Topology Optimization Using a Dual Method with Discrete Variables," Struct. Optim., **17**(1), pp. 14–24.

[100] Langelaar, M., 2016, "Topology Optimization of 3D Self-Supporting Structures for Additive Manufacturing," Addit. Manuf., **12, Part A**, pp. 60–70.

[101] Liu, K., and Tovar, A., 2014, "An Efficient 3D Topology Optimization Code Written in Matlab," Struct. Multidiscip. Optim., **50**(6), pp. 1175–1196.

[102] Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., and Sigmund, O., 2011, "Efficient Topology Optimization in MATLAB Using 88 Lines of Code," Struct. Multidiscip. Optim., **43**(1), pp. 1–16.

[103] Zuo, Z. H., and Xie, Y. M., 2015, "A Simple and Compact Python Code for Complex 3D Topology Optimization," Adv. Eng. Softw., **85**, pp. 1–11.

[104] 2014, *ASTM D638-14 Standard Test Method for Tensile Properties of Plastics*, ASTM International, West Conshohocken, PA,.

[105] "ISO 527-1:2012 - Plastics -- Determination of Tensile Properties -- Part 1: General Principles" [Online]. Available: https://www.iso.org/standard/56045.html. [Accessed: 04-May-2017].

[106]   Forster, A. M., 2015, "Materials Testing Standards for Additive Manufacturing of Polymer Materials: State of the Art and Standards Applicability," ResearchGate, pp. 67–123.

[107]   2014, *ASTM D3039/D3039M-14 Standard Test Method for Tensile Properties of Polymer Matrix Composite Materials*, ASTM International, West Conshohocken, PA,.

[108]   "Super Glue ULTRA Gel Control from Loctite Adhesives" [Online]. Available: http://www.loctiteproducts.com/p/sg_ug_cntrl/overview/Loctite-Super-Glue-ULTRA-Gel-Control.htm. [Accessed: 21-Aug-2016].

[109]   Lachica, L., "100 Family Single Column Electromechanical Universal Test Machine," TestResources Test Mach. Grips Fixt. [Online]. Available: http://www.testresources.net/test-machines/100-family. [Accessed: 12-Jul-2016].

[110]   "Epsilon Tech 3542 Axial Extensometers" [Online]. Available: http://www.epsilontech.com/3542.htm. [Accessed: 21-Jun-2016].

[111]   "JSM-6610 Series Scanning Electron Microscope | Products | JEOL Ltd." [Online]. Available: http://www.jeol.co.jp/en/products/detail/JSM-6610series.html. [Accessed: 21-Aug-2016].

[112]   "Q50" [Online]. Available: http://www.tainstruments.com.tw/product.aspx?siteid=6&id=21&n=1. [Accessed: 21-Aug-2016].

[113]   Shigley, J. E., 2011, *Shigley's Mechanical Engineering Design*, Tata McGraw-Hill Education.

[114]   Noble, B. &, "A First Course in Finite Elements / Edition 1," Barnes Noble [Online]. Available: http://www.barnesandnoble.com/p/first-course-in-finite-elements-jacob-fish/1101201872/2685404381061. [Accessed: 02-Feb-2017].

[115]   Chapra, S., *Applied Numerical Methods W/MATLAB*, McGraw-Hill Higher Education.

[116]   Barbero, E. J., 2010, *Introduction to Composite Materials Design, Second Edition*, CRC Press.

[117]   Kollár, L. P., and Springer, G. S., 2003, *Mechanics of Composite Structures*, Cambridge University Press.

[118]   Tortorelli, D. A., and Michaleris, P., 1994, "Design Sensitivity Analysis: Overview and Review," Inverse Probl. Eng., **1**(1), pp. 71–105.

[119] "Find Minimum of Constrained Nonlinear Multivariable Function - MATLAB Fmincon" [Online]. Available: https://www.mathworks.com/help/optim/ug/fmincon.html. [Accessed: 17-Dec-2016].

[120] Heller, B. P. 1989-, 2015, "Effects of Nozzle Geometry and Extrudate Swell on Fiber Orientation in Fused Deposition Modeling Nozzle Flow.," Thesis.

[121] Heller, B. P., Smith, D. E., and Jack, D. A., 2016, "Effects of Extrudate Swell and Nozzle Geometry on Fiber Orientation in Fused Filament Fabrication Nozzle Flow," Addit. Manuf., **12, Part B**, pp. 252–264.

[122] Jeffery, G. B., 1922, "The Motion of Ellipsoidal Particles Immersed in a Viscous Fluid," Proc. R. Soc. Lond. Math. Phys. Eng. Sci., **102**(715), pp. 161–179.

[123] Zhang, D., 2013, "Flow-Induced Micro- and Nano-Fiber Suspensions in Short-Fiber Reinforced Composite Materials Processing," Thesis, University of Missouri--Columbia.

[124] Advani, S. G., 1987, "The Use of Tensors to Describe and Predict Fiber Orientation in Short Fiber Composites," J. Rheol., **31**, p. 751.

[125] Montgomery-Smith, S., Jack, D., and Smith, D. E., 2011, "The Fast Exact Closure for Jeffery's Equation with Diffusion," J. Non-Newton. Fluid Mech., **166**(7–8), pp. 343–353.

[126] Chung, S. T., and Kwon, T. H., 1996, "Coupled Analysis of Injection Molding Filling and Fiber Orientation, Including in-Plane Velocity Gradient Effect," Polym. Compos., **17**(6), pp. 859–872.

[127] 1995, "Orthotropic Closure Approximations for Flow- induced Fiber Orientation," J. Rheol., **39**(6), pp. 1095–1122.

[128] Jack, D. A., and Smith, D. E., 2007, "The Effect of Fibre Orientation Closure Approximations on Mechanical Property Predictions," Compos. Part Appl. Sci. Manuf., **38**(3), pp. 975–982.

[129] Tandon, G. P., and Weng, G. J., 1984, "The Effect of Aspect Ratio of Inclusions on the Elastic Properties of Unidirectionally Aligned Composites," Polym. Compos., **5**(4), pp. 327–333.