Integrated Topology Optimization Design and Process Planning for Additive Manufacturing

by

Dylan J. Bender

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Applied Science in Mechanical Engineering

Ontario Tech University

Faculty of Engineering and Applied Science

Department of Automotive, Mechanical, and Manucaturing Engineering

Oshawa, Ontario, Canada

August 2019

THESIS EXAMINATION INFORMATION

Submitted by: **Dylan Bender**

Master of Applied Science in **Mechanical Engineering**

Thesis title: Integrated Topology Optimization Design and Process Planning for Additive Manufacturing

An oral defense of this thesis took place on August 5, 2019 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Martin Agelin-Chaab
Research Supervisor	Dr. Ahmad Barari
Examining Committee Member	Dr. Amirkianoosh Kiani
Thesis Examiner	Dr. Sayyed Ali Hosseini

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

Industry 4.0 demands that the systems and processes in today’s product design and manufacturing not just be automated, but to be robust and containing many feedback mechanisms which enables it to be self-correcting. The hypothetical upcoming Industry 5.0 promises on demand and personalized products which this thesis aims to take a step in the direction of. It is proposed that an integrated and optimized process for structural topology optimization and subsequent additive manufacturing is possible for automated design and manufacturing starting from its problem definition. An improvement on the benchmarked topology optimization methods is shown which allows the user control over the optimization’s convergence characteristics which is then further studied to find a robust set of optimization parameters. The resulting topology of the structure is then analyzed for its optimal printing orientation based on a custom-made algorithm which minimizes manufacturing costs. Furthermore, the structure is then sliced for instruction generation of layer-based manufacturing techniques in a novel fashion which also serves to provide feedback of the manufacturing process planning to the topology optimization design stage.

Keywords: topology optimization; additive manufacturing; process planning

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Dylan Bender

____ Dylan Bender

STATEMENT OF CONTRIBUTIONS

I performed the majority of the idea synthesis, development of algorithms, and writing of the following manuscripts.

Bender D., and Barari A., 2019, On Finding the Density-Gradient Vectors of a 3D Structure Resulting from Topology Optimization for Additive Manufacturing, Journal of Structural and Multidisciplinary Optimization – In Press

Bender D., and Barari A., 2019, Multi-Element Slicing Approach for Intelligent Additive Manufacturing Preprocessing – Proceedings of CSME 2019 – In Press

Bender D., and Barari A., 2019, Integrated Digital Design and Manufacturing for 3D Printing Technologies – Proceedings of IFAC IMS 2019 – In Press

Bender D., and Barari A., 2019, Direct Solid Element Slicing for Additive Manufacturing – Proceedings of ASME IDETC 2019 – In Press

Bender, D. and Barari, A., 2018, August. Overhanging Feature Analysis for the Additive Manufacturing of Topology Optimized Structures. In *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. V01AT02A044-V01AT02A044). American Society of Mechanical Engineers.

Bender, D. and Barari, A., 2018. Convergence Control for Topology Optimization. Proceedings of CSME 2018

DEDICATION

To my parents for their unconditional support and love. They may not have had the opportunities that I have been blessed with but they continually make sacrifices to ensure that my siblings and I have the opportunity to pursue a higher education, and our dreams.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisor Dr. Ahmad Barari for his continued support, guidance and general teachings for not only the duration of my masters, but also during my senior undergraduate year. Dr. Barari has consistently pushed me beyond what I thought I was capable of, and many of my achievements over the past few years can be directly attributed to him. I would also like to express my appreciation to the other members of my lab over that past few years for their continual and selfless will to help in my research. Namely I would like to mention Cody Berry, Hossein Gohari, Davin Jankovics, and Amirali Lalehpour for all of their various contributions to my research over the past few years.

The monetary support provided by parties such as the Faculty of Engineering and Applied Science, Nathan and Marvin Goldman of Durham Homes, NSERC, and both provincial and federal government agencies has been instrumental in allowing me attend and complete my graduate studies. Their contributions are very much appreciated.

My deep appreciation also goes out to the various teachers, mentors, and individuals that have selflessly taken time out of their lives to teach me new concepts, give me life lessons, and words of encouragement. I cannot possibly thank each and every one of them but I vow to selflessly pass on their lessons and kind-heartedness to anyone who may seek it. I am very fortunate to have met so many great people in my life.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 3

CHAPTER 5

CHAPTER 6

LIST OF FIGURES

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

LIST OF ABBREVIATIONS

AM	Additive Manufacturing
BESO	Bi-directional Evolutionary Structural Optimization
CNC	Computer Numerical Control
DLP	Digital Light Processing
DMLS	Direct Metal LASER Sintering
ER	Evolutionary Rate
ESO	Evolutionary Structural Optimization
GUI	Graphical User Interface
LASER	Light Amplification by Stimulated Emission of Radiation
LENS	LASER Engineered Net Shaping
PID	Proportional-Integral-Derivative
RSV	Required Support Volume
SIMP	Solid Isotropic Material with Penalization
SLA	Sterolithography
SLM	Selective LASER Melting
SLS	Selective LASER Sintering
STL	Standardized Tessellation Language
TO	Topology Optimization

Chapter 1. Introduction

1.1 Background and Motivation

Improving the component design and manufacturing processes of structural components is an ongoing effort by academia and industry alike for many important reasons such as limited resources (i.e. materials, energy, monetary, technical competence etc..), environmental impacts and even competition for technological innovation. It is for these reasons that light weight, low-cost, high performing, and consistent (for the purpose of interchangeability) structures is the goal of designers and manufacturers alike [1]. On top of these goals, customers of these designers and manufacturers are increasingly demanding their products to be personalized, and produced then delivered in as minimal amount of time as possible. The motivation of this thesis is to provide a process which tackles these outlined goals of the designers, the manufactures and the customers as well as proposing and implementing novel and custom techniques/algorithms to improve the final product and the design through manufacturing stages.

1.2 Scope of the Thesis

In order to achieve the objectives of this thesis, custom code and algorithms are proposed to take the structural problem definition of a user as an input, then automate the design and manufacturing stages while improving and optimizing many aspects of the process with novel approaches as I have introduced in [2].

The first step of the integrated process planning layout is the problem definition step. In this stage, the structural design problem is defined by the user typically via a GUI. In this interface, the user defines the “design domain” which is to say the spatial geometry in which the topology optimization may work within. This design domain is then meshed at the discretion of the user which decomposes the continuous design domain into a set of finite elements. The user would also define any boundary and loading conditions imparted on the meshed design domain. This meshed design domain and its boundary/loading conditions are then passed to the next step in the process.

The second step of the process is to perform structural optimization by means of Topology Optimization (TO) in order to achieve the best performing structure while satisfying design and manufacturing constraints. Two highly reviewed topology optimization techniques were implemented, namely the Bi-Directional Evolutionary Structural Optimization (BESO), and the Simple Isotropic Material with Penalization (SIMP) techniques. The two methods have their own individual advantages and disadvantages in the design stage and their theoretical background is explained in chapter 2 of this thesis. In the interest of increasing the robustness of the design stage, the BESO methodology was improved by converting the static optimization problem to one that is adaptive by means of implementing a Proportional-Integral-Derivative (PID) controller to the Evolutionary Rate (ER) [3]. This improvement allows for the control of various convergence characteristics which were previously uninvestigated and inherently, not utilized.

The third stage of the process is to determine the manufacturing parameters for the upcoming fourth stage (manufacturing). The goal of this stage is to minimize the cost and time to produce the part on the investigated layer-based additive manufacturing machines. This goal is carried out by using a novel formulation of the manufacturing costs which accounts for the costs of material, operating the machinery and post-processing of the work piece. This formulation is used to determine the optimal build orientation of the part within the additive manufacturing machine and uses a new technique to determine the requirements of additional supporting structural material required to fabricate the part [4, 5].

The fourth stage is to produce the set of machine instructions needed to manufacture the part. To do this, a novel approach to “slicing” the part is proposed which serves many purposes on top of just producing the slice contours to generate the machine instructions [6]. This proposed slicer removes the typical redundant step of converting the Finite Element representation of the design (produced by topology optimization) into the Standardized Tessellation Language (STL) representation (a reconstruction of a 3D structure’s surface by discrete planar triangles). By removing this design representation conversion step, the amount of computational time needed for the entire process is reduced and a reduction of geometrical errors is also achieved by eliminating the deviations of the STL representation from the TO optimized finite element based design. A third benefit of this slicer over the conventional STL slicer is that it enables the coupling of the manufacturing planning to the design stage to better design the structure in a way that accommodates the manufacturing constraints efficiently [7].

The fifth and final step of the proposed process is to finally manufacture the part. The slice contours produced in the previous stage may be used for any layer-based additive manufacturing technique but a Digital Light Processing (DLP) style Stereolithographic (SLA) 3D printer is used as an example. This printer was the result of an undergraduate capstone project titled “The Design and Development of a Layerless 3D Printing Mechanism” and chosen because developing the machine instructions based on the slice contours is relatively straight forward and custom control software for the machine has already been developed in house.

1.3 Outline of Thesis

This thesis is divided into 7 chapters. Chapter (1) describes the motivation for the work, the scope of the work, as well as introduces the developed techniques and algorithms proposed in this thesis. Chapter (2) discusses the theoretical background needed to understand the methodologies presented in the following chapter. This includes the background and review of topology optimization, additive manufacturing including, and a discussion about their respective challenges and opportunities. Chapter (3) implements a PID controller which achieves not before realized convergence control of the topology optimization algorithm. Chapter (4) describes a novel approach for determining overhanging surface build angles by means of density gradient analysis which is implemented in the next chapter. Chapter (5) discusses a new approach to minimize the manufacturing cost by means of part orientation optimization based on the analysis of the previous chapter and Chapter (6) presents a finite element based slicing approach for layer-based additive manufacturing machines. Chapter (7) summarizes the improvements to the process proposed in this thesis and discusses avenues of future works based of the various chapters and then finally concludes this thesis.

Chapter 2. Background

2.1 Topology Optimization

It is often the goal of an engineer to make the best or most effective use of a situation or resources for a given design, and such is the concept of optimization. Topology refers to the study of geometric properties and spatial relations and thus Topology Optimization (TO) is the study concerning optimal spatial order and connectivity for the case of discrete designs such as trusses, or is considered the study of determining the best locations and geometries of cavities within a continuous design domain [1]. The results of TO are often organic in appearance which in this context is to say they have an uncanny resemblance to topologies found in nature which allows designers/engineers to create novel, and highly efficient conceptual ideations. One example of an organic-like of topology is shown below in Fig. 2.1 as an example of a topology optimized bridge type structure.

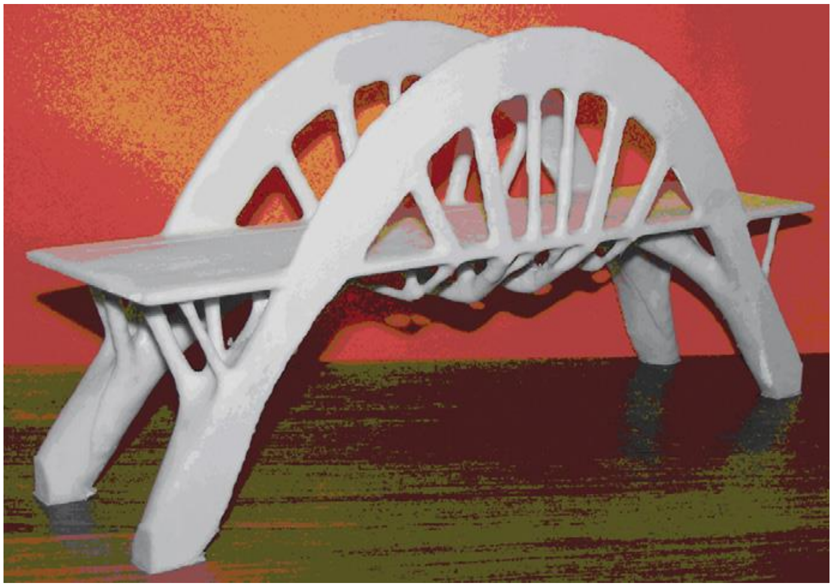


Figure 2.1: Results of a stiffness-based topology optimized bridge-like structure with an efficient, and organic-like topology [8]

Topology optimization is most often studied within the context of structural optimization as in the case of this thesis but can be applied to a wide variety of engineering applications. One non–structural application of topology optimization includes the design of an air-cooled heat sink [9], another being the design of piezoelectric structures [10], another is heat or electrical conductivity [11] and even the design of functionally graded materials is possible [12]. Some notable implementations of topology optimization within the context of structural optimization including the architectural design of buildings [13, 14], bridge type structure, [15] ,design of a small satellite [16], an automotive backrest frame [17], an engine bracket [18], aircraft landing gear [19], or a tracked vehicle road arm [20] to name just a few. For the remainder of this thesis, topology optimization will be referred to within the context of structural topology optimization.

Many methods to perform topology optimization exist [1, 21-23], most of which are based off of Finite Element Analysis (FEA); namely, the Homogenization Method [24, 25], the Evolutionary Structural Optimization (ESO) [26, 27], Bi-Directional Evolutionary Structural Optimization (BESO) [28, 29], Solid Isotropic Material with Penalization (SIMP) [30, 31], and the Level Set method [32, 33]. This thesis deals with the BESO and SIMP methods which have a similar approach to the structural TO problem.

The first step is the structural problem definition step where a designer/engineer will define the design domain and its boundary conditions, loads and constraints. The user will provide this information, usually through a graphical user interface. This design domain of the structure that the user has defined is then meshed into a finite element representation of the model to numerically solve for the stresses or strains of each of the finite elements. A design update is then made based on the chosen TO method and the new structure is analyzed again using numerical finite element methods. This process continues for tens or sometimes hundreds of iterations until an optimal design has been formed. This process is visualized in Fig 2.2. It is the structural design updating scheme that discerns the BESO and SIMP methods and are described in section 2.1.1 and section 2.1.2 respectively.

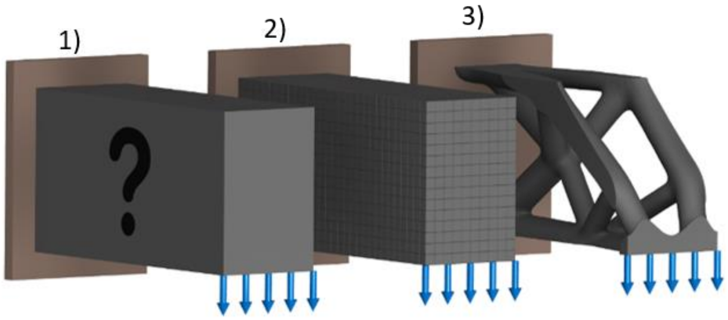


Figure 2.2: The process of structural design by means of topology optimization 1) Define the design problem, 2) Discretize the design domain, 3) Optimize the structure's topology

2.1.1 Bi-Directional Evolutionary Structural Optimization

As previously stated, the first step of the structural design process by means of topology optimization is to first define the design domain of the problem. Often a simple geometry is used for the design domain such as a rectangular prism in order to simplify the setup. The design domain is then discretized into a mesh of finite elements (cuboid, tetrahedral, prism, wedge, etc…) for the purpose of numerically solving the unknown forces and displacements within the structure by using finite element methods [34]. Often in research applications, cuboid-type elements (8 nodes, 6 faces, 12 edges) are solely implemented for the sake of simplicity and the number of elements comprising the design domain is limited by the available computational power. An example of this type of mesh is seen in Fig 2.3 which comprises of 8 cuboid-type elements and a total of 30 nodes. The user will apply a set of boundary conditions, loads and displacements to the nodes of this elemental mesh.

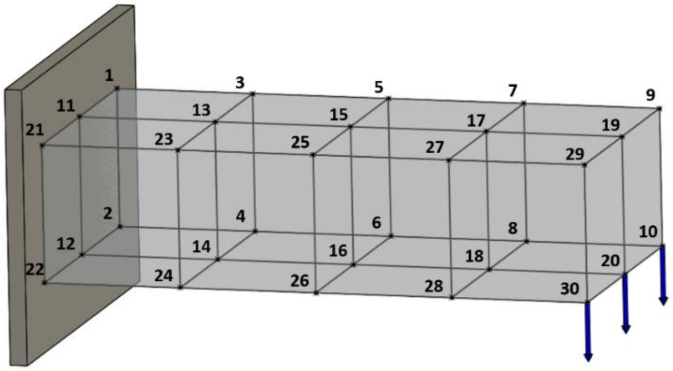


Figure 2.3: Global node IDs of a structure composed of 8 elements and 30 nodes

The design of the structure is defined by determining whether or not an element (x_i) should be solid (1) or void (0). In the case of the original Evolutionary Structural Optimization (ESO) algorithm, the void elements are removed from the design domain altogether and have no effect on the FEA. For the case of the implemented BESO algorithm, it is realized that elements may be removed prematurely so the concept of removing a void element from the design domain is replaced with the concept of making the void elements a soft element with very low density effectively reducing its Young's Modulus [36, 37]. This allows for the sensitivity of the void elements relative to the global structural stiffness to be calculated, which in turn potentially allows for the re-addition of void elements back into the design domain from one iteration to the next. To conceptualize this, the density of a void element takes on the value of small value rather than a value of 0 (e.g: $x_{Void} = 0.0001$). The young's modulus for a void element (E_{Void}) is a product of the Young's modulus of a solid element (E_{Solid}) and the elements density as shown in equation 2.1. It is worth noting here that the SIMP algorithm described in the next section differs from the BESO algorithm by linearly interpolated the density of an element between solid (1) and void (0) allowing for the young's modulus to also be interpolated.

$$E_{Void} = E_{Solid} x_{Void}^{p} \tag{2.1}$$

Where p is the penalty exponent and is used in the material interpolation scheme to achieve a nearly solid-void design [31, 38, 39]. This penalty exponent is useful in the SIMP based TO method to help steer the evolution to a nearly solid-void design (free of intermediate density elements).

Finite element methods are implemented to analyze the performance of the structure at each iteration. From this analysis, the unknown values for the nodal force vector, $\boldsymbol{f}$ and the nodal displacement vector, $\boldsymbol{u}$ are determined and used to calculate the overall performance of the structure. The stiffness of the overall structure gives a measure for the structure's performance and its inverse value, known as the mean stiffness compliance, C

is used as the objective function of the topology optimization. The mean compliance of the structure is calculated as:

$$C = \frac{1}{2}\boldsymbol{f}^T\boldsymbol{u} \tag{2.2}$$

The mean compliance of the structure is the objective function for the topology optimization process. The goal is to achieve a structure with maximum stiffness (minimal mean compliance) for a given volume of material by determining whether each element of the design domain should be a solid or void. Other structural performance measures have been suggested such as displacement or local stress based measures [40, 41], stiffness optimization with multiple materials [38, 42], periodic structures (i.e. honeycomb sandwich plates) [43, 44], design dependent gravitational loading [45-47], compliant mechanisms [48], or to maximize the structure's natural frequency [49-51]. This thesis is solely concerned with stiffness and volumetric-based structural topology optimization because it's the simplest form and the developed techniques apply to all of the forms. The constraints on structural topology optimization are as follows:

$$K\boldsymbol{u} = \boldsymbol{f} \tag{2.3}$$

$$V^* - \sum_{i=1}^{N} V_i x_i = 0 \tag{2.4}$$

$$x_i = x_{min} \; or \; 1 \tag{2.5}$$

Where V^* is the prescribed target volume, V_i and x_i are the volume and density of element i respectively, and K is the global stiffness matrix for the structure calculated as:

$$K = \sum_i x_i^p K_i^0 \tag{2.6}$$

Where K_i^0 is the elemental stiffness matrix of the solid element.

To solve the optimization problem, a sensitivity-based analysis is performed on the dual formulation of the optimization problem [52]. In the BESO procedure, the sensitivity number of an element (α_i) is interpreted to represent the change of the mean compliance or total strain energy of the structure that is equal to the elemental strain energy. This is defined in equation 2.7 and 2.8 and used to determine the relative ranking of the elements in comparison to each other for updating the design from iteration to iteration. Essentially, the more structurally efficient an element is to the design, the more likely it is to be included in the design of the next iteration and vice versa.

$$\frac{\partial C}{\partial x_i} = -\frac{1}{2} p x_i^{p-1} u_i^T K_i^0 u_i \tag{2.7}$$

$$\alpha_i = -\frac{1}{2}\frac{\partial C}{\partial x_i} = \begin{cases} {}^1/_2\, u_i^T K_i^0 u_i \quad when\ x_i = 1 \\ {}^1/_2\, p x_{min}^{p-1} u_i^T K_i^0 u_i \quad when\ x_i = x_{min} \end{cases} \tag{2.8}$$

There exists three fundamental issues with the topology optimization method described so far, namely the problems of “checker boarding”, “mesh dependency”, and “instability of the evolutionary process”. Well studied solutions to solve these issues can be implemented to solve these problems but their explanation is delayed until the end of this section. It worth noting that it is as this point in the TO process that those solutions would be implemented, however, the fundamental function of the TO process is not dependent on their implementation, and that is why their discussion is skipped at this point.

After implementing the solutions to the three aforementioned TO problems, the next step of the process is to updated the design of the structure in preparation of the next iteration of the evolutionary-based procedure. The BESO process typically starts with every element in the design domain as a solid element and then slowly converges to its target volume (typically $V^* = 50\%$ in 2D examples, and $V^* = 10\% - 15\%$ in 3D examples) by removing a net small amount (more elements are removed than are added) of elements each iteration; an initial guess can sometimes help the solution to converge [8]. Before the design can be updated, the target volume to the subsequent iteration needs to be calculated based on the predefined Evolutionary Rate (ER) of the BESO algorithm. A smaller value of ER will result in an evolution that make smaller changes between iterations and thus takes longer to converge than a larger value of ER. Using a smaller value of ER has the advantage of a more stable evolution and typically will result in a lower compliance at the expense of increased computational time. Common values of ER range from 0.5% to 5%. The evolution of the structure's volume is expressed in equation 2.9 where k represents the current iteration of the topology optimization and k+1 represents the next iteration.

$$V_{k+1} = \begin{cases} \max(V_k(1 - ER), V^*) \ \ when\, V_k > \ V^* \\ \min(V_k(1 + ER), V^*) \ \ when\, V_k \ \le \ V^* \end{cases} \qquad (k = 1, 2, 3, \ldots) \qquad (2.9)$$

The next step is to finally update the design of the structure. This is first done by ordering the elements according to their sensitivity numbers from highest to lowest. The following criteria is then used to determine whether a solid element should be flipped into a void element or vice versa:

A solid element, i of the design domain on the k^{th} iteration ($x_i^k = 1$) is to be removed from the design in the subsequent iteration ($x_i^{k+1} = \ x_{min}$) if:

$$\alpha_i \ \le \ \alpha_{del}^{th} \qquad (2.10)$$

A void element, i of the design domain on the k^{th} iteration ($x_i^k = x_{min}$) is to be added to the design in the subsequent iteration ($x_i^{k+1} = \ 1$) if:

$$\alpha_i > \alpha_{add}^{th} \tag{2.11}$$

Where α_{del}^{th} is the threshold sensitivity number for removing elements and α_{add}^{th} is the threshold sensitivity number for adding elements back into the design of the structure. The two threshold sensitivity numbers are determined by these three steps:

i) Let $\alpha_{del}^{th} = \alpha_{add}^{th}$ which are given values determined from V_{k+1} previously calculated. In example, if there are 1000 sorted elements in the design domain and V_{k+1} requires that 75% of the domain should be solid in the next iteration, then the threshold sensitivity numbers would be given a value of the sensitivity number of the 750th element in the list.

ii) Calculate the volume addition ratio (AR) which is defined to be the ratio of the number of added elements to the total number of elements of the design domain. If $AR \leq AR_{max}$ then skip step 3 (where AR_{max} is the prescribed maximum volume addition ratio), otherwise, recalculate the two threshold sensitivity numbers in step 3

iii) Determine α_{add}^{th} by first sorting just the void elements from highest to lowest according to their sensitivity numbers. Then multiply AR_{max} by the total number of elements to determine how many elements should be added to the design. The elements to be added are at the top of the sorted list of void elements and the sensitivity number of the element just below the last one to be added it the value chosen for α_{add}^{th}. The remove threshold sensitivity number, α_{del}^{th} is then determined so that the removed volume is equal to ($V_k - V_{k+1}$ + the volume of added elements).

The cycle of finite element analysis and element removal/addition repeats until two criteria have been met. First, the objective volume (V^*) needs to have been reached, then if the commonly used convergence criteria described in equation (2.12) has been satisfied, the topology optimization is terminated. This convergence criteria determines if the change in the objective function is smaller than the allowable convergence tolerance (τ) as well as averaging out the compliance value (objective) of the last $2N$ iterations. Combined, the two-convergence criterion effectively ensure that the optimization has reached its target volume and that it has reached a steady state by have been given enough time for the convergence to settle. Note that a typical value of N = 5 is chosen which implies that the change in the objective function over the past 10 iterations is acceptably small, and typical value of $\tau = 1.0\%$ is often used

$$\frac{\left|\sum_{i=1}^{N} c_{k-1+1} - \sum_{i=1}^{N} c_{k-N-i+1}\right|}{\left|\sum_{i=1}^{N} c_{k-1+1}\right|} \leq \tau \tag{2.11}$$

The entire process of BESO based topology optimization which has been described since the beginning of this section can be overviewed in the BESO method flowchart of figure 2.5. The exception is the filtering and averaging of sensitivity numbers which is described in section 2.1.2. A sample history of a BESO based topology optimization of a cantilevered beam is presented in figure 2.4. The code provided in the appendix of chapter 4 of [1] is implemented with nelx = 120, nely = 60, volfrac = 0.5, er = 0.02, and rmin = 3.5.

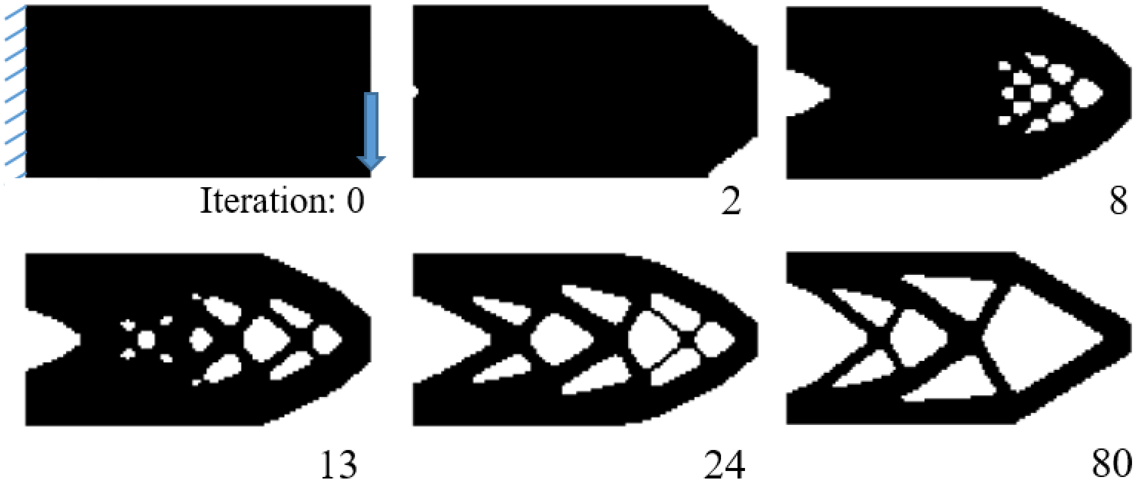


Figure 2.4: Sample evolutionary history of a BESO algorithm.

2.1.2 Improving characteristics of the Topology Optimization Process

As previously mentioned, there exists four fundamental issues with the topology optimization which hinder its stability and applicability. These issues are referred to as 1) the checker boarding problem [53-55] 2) mesh dependency [1, 22, 56] 3) convergence instability [8] and 4) convergence to a local optimum [57]. Issues number 1) and 2) are solved using element filtering while issue number 3) is solved by averaging the sensitivity numbers with its historical information and 4) is hindered by the continuation method.

2.1.2.1 Filtering Scheme

The issue of checker boarding is that it causes difficulty in interpreting and manufacturing the results of the topology optimization. Alternating patterns of solid and voids elements are produced because the introduction of more holes without changing the structural volume will generally increase the efficiency of a given design. The issue of mesh-dependency refers to the problem of obtaining different resulting designs from using

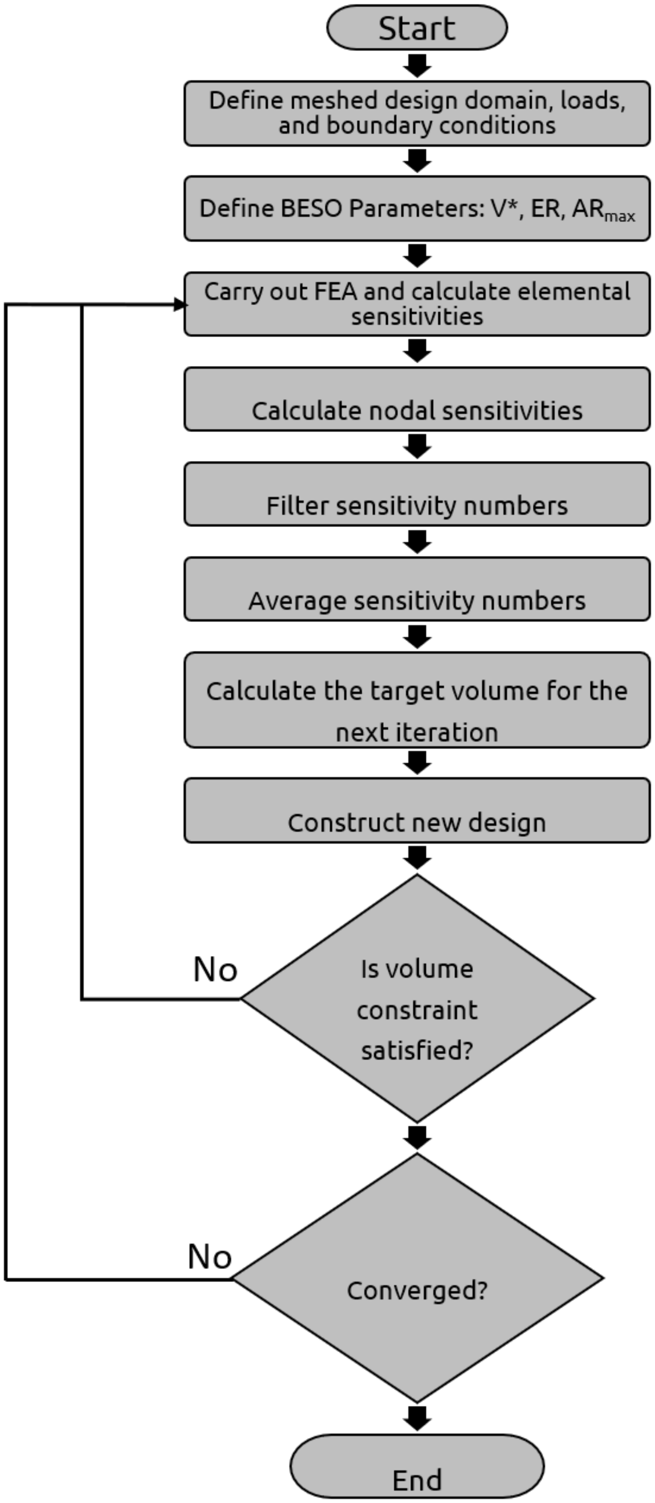

Start
Define meshed design domain, loads, and boundary conditions
Define BESO Parameters: V*, ER, AR$_{max}$
Carry out FEA and calculate elemental sensitivities
Calculate nodal sensitivities
Filter sensitivity numbers
Average sensitivity numbers
Calculate the target volume for the next iteration
Construct new design
No
Is volume constraint satisfied?
No
Converged?
End

Figure 2.5: Flowchart of the BESO method

different resolution of a finite element mesh. Ideally, increasing the resolution of the mesh (increasing the number of finite elements) should increase the definition of the structures boundary but instead, the resulting topology will have an increased number of members and those members will be of smaller sizes. These two issues are pictured in figure 2.6 for a 2D case example of a bridge-like design problem. Picture b) demonstrates a topology that is easy to interpret and easy to manufacture that is the result of applying the filtering. Picture c) demonstrates a resulting topology that has the checker boarding issue and picture d) demonstrates a completely different topology resulting from refining the original mesh.

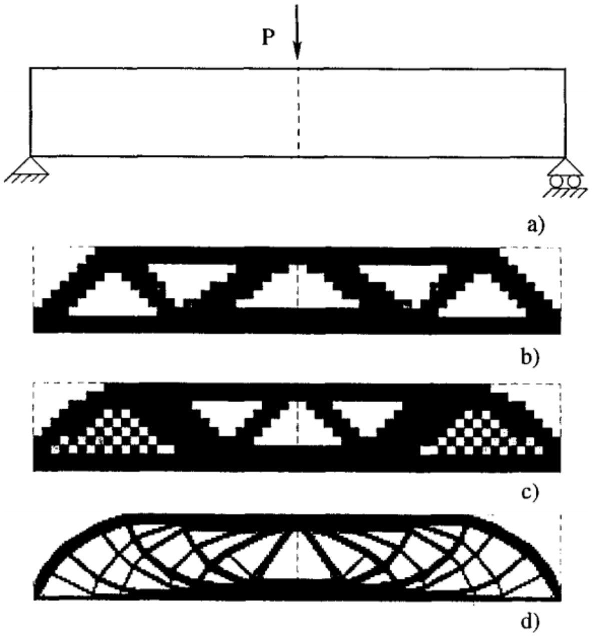


Figure 2.6: a) Definition of a bridge-like structural design problem b) easy to interpret/manufacture result c) topology suffering from checker boarding and d) the issue of a different resulting topology after mesh refinement [56]

The filtering scheme is used to solve the issue of checker boarding and mesh dependency. The first step in its procedure is to calculate the nodal sensitivity numbers (α^n) from the already determined elemental sensitivity numbers (α^e) by using a distance-based weighting scheme. The distance between a given node j, and an element i, is denoted as as r_{ij} and is used to calculate the weight of a node with M number of elements that are directly connected to it. The nodal sensitivity numbers are calculated as:

$$\alpha_j^n = \sum_{i=1}^{M} \omega_i \alpha_i^e \tag{2.12}$$

Where the distance based weight factor is defined as:

$$\omega_i = \frac{1}{M-1}\left(1 - \frac{r_{ij}}{\sum_{i=1}^{M} r_{ij}}\right) \tag{2.13}$$

Once the nodal sensitivity numbers have been calculated, they are used to determine the smoothed elemental sensitivity numbers. This is achieved by projecting the nodal sensitivity numbers onto the design domain using another filter scheme. This filter has a length scale (r_{min}) that does not change value with mesh refinement. The length scale is used to determine which nodes are within a certain distance of a particular element. The number of elements that lie within a circle of influence (2D) or a sphere of influence (3D) with radius r_{min} is stored as the value k. For a given element, and its influencing nodal sensitivity numbers, its improved elemental sensitivity number is calculated as:

$$\alpha_i = \frac{\sum_{j=1}^{k} \omega\left(r_{ij}\right)\alpha_j^n}{\sum_{j=1}^{k} \omega\left(r_{ij}\right)} \tag{2.14}$$

2.1.2.2 Stabilizing the Evolutionary Procedure

The issue of convergence instability is that large oscillations often occur in the objective function of the evolutionary history. The reason is that the discrete nature of the design variables make the evolution difficult to converge. An example of this can be seen in figure 2.7 in the compliance history of the evolution. In the left figure, a sample of topology optimization without any fix for this instability can be seen which has many spikes in its compliance history. In the right figure, the compliance history is almost completely free from any unwanted instabilities in its convergence and a more efficient structure is also produced.

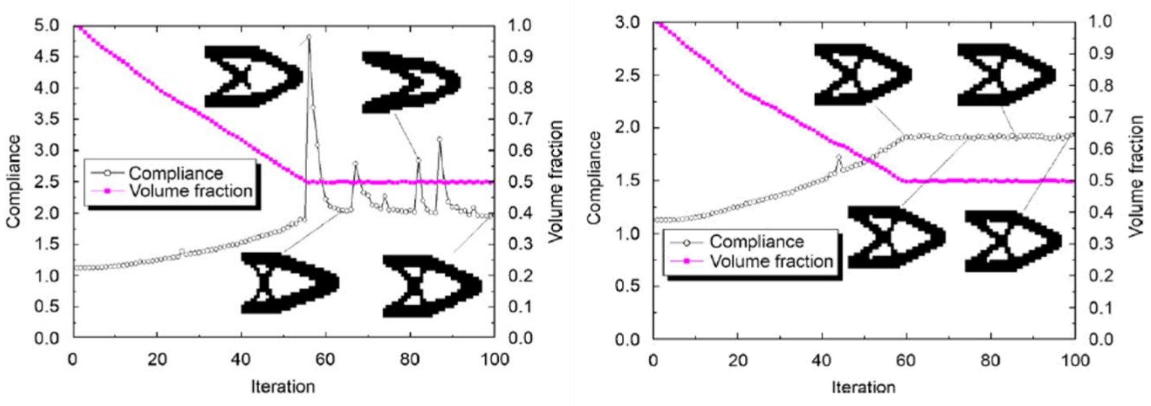


Figure 2.7: Example of unstable evolutionary convergence (left) and a stable convergence (right) [8]

To overcome the evolutionary instability, an averaging scheme is implemented in which the elemental sensitivity numbers are averaged with its historical information [8]. The updated sensitivity number includes the entire history of the sensitivity from the previous iterations, and the implemented equation only slightly affects the search path of the optimization algorithm which results in very little change in the final solution. The averaging scheme is outlined in equation (2.15).

$$\alpha_i = \frac{\alpha_i^k + \alpha_i^{k-1}}{2} \tag{2.15}$$

2.1.2.3 The Continuation Method to Avoid a Local Optimum

Although theoretically it is not guaranteed that a global optimum can be reached even though a unique optimal solution exists [57], the continuation method of topology optimization has been broadly researched for this very purpose [21, 56]. This uncertainty was introduced when the penalization factor p was implemented into the convex topology optimization problem rendering it a non-convex problem [58]. To mitigate this, a heuristic approach is taken where the topology optimization is run in cycles starting at the material penalty exponent equal to 1 ($p = 1$) and then running the optimization until the convergence criterion have been met. This renders it back to a convex problem with a solvable global optimum. The resulting design of this cycle is used as the initial guess for the subsequent cycle of topology optimization where the penalty exponent is slightly increased. This way, grey regions (elements of intermediate densities) change locally into black-and-white regions (elements that a purely solid of void) which is of concern for the not yet described SIMP method.

An alternate approach has been proposed and explored in which the filter radius of the mesh independency filter described in section 2.1.2.1 is cleverly repurposed to ensure a convex solution at the beginning and then changed to allow for convergence to a black-and-white solution [48, 59]. This is done in a similar cyclical manner as the penalty exponent method except that the filter radius (r_{min}) is initialized with a very larger value and then it is gradually decreased to allow for convergence to a black and white solution. This continuation method is greatly increased the computation time required for topology optimization but helps mitigates some of the drawbacks that the material interpolation scheme introduces into the more advantageous SIMP method.

2.1.3 Solid Isotropic Material with Penalization (SIMP)

One of the most popular methods of Topology Optimization in both academia and commercial software’s is the Solid Isotropic Material with Penalization (SIMP) [35, 60]. This method is highly comparable to the BESO method and only requires a few changes for its implementation. Equations (2.2) to (2.5) are still used to define the optimization problem with only a minor change to (2.5). This change to equation (2.5) of the BESO method is the primary differentiating factor between the two methods which now defines each element as an isotropic material with variable density. This is to say that the discrete nature of the design variable is not optimized over a continuous domain with a box constraint as follows:

$$0 < x_{min} \leq x_i \leq 1 \tag{2.16}$$

Optimizing over the continuous domain has the added advantage of improved convergence characteristics that were an issue of a discrete design variable-based optimization. A drawback of SIMP is the compliance of the final structure is often approximated to be higher than that of other methods which may be attributed to the over estimation of the strain energy of elements with intermediate densities [1]. The implementation of the SIMP algorithm is similar to the BESO flow chart of figure 2.5 except that the processes of calculating the nodal sensitivity numbers and calculating the target volume of the next iteration are made redundant and therefore removed from the process. When re-evaluating the dual formulation of the optimization problem, the sensitivity of the objective function now becomes twice that of the BESO method (equation 2.7 and 2.8) and is as follows:

$$\alpha_i = \frac{\partial C}{\partial x_i} = -p x_i^{p-1} u_i^T K_i^0 u_i \tag{2.17}$$

To solve the sensitivity analysis of the optimization problem, a bisection-based optimization technique is often employed such as the Optimality Criteria (OC) [31, 61] or another method known as the Method of Moving Asymptotes (MMA) [62] or the Globally Convergent Method of Moving Asymptotes (GCMMA) [63] is often used as well. This paper will only deal with the OC method. The new OC-based updating scheme which replaces equations (2.10) and (2.11) of the BESO method is formulated as:

$$x_i^{k+1} = \begin{cases} \max\left(x_{min}, \, x_i^k - m\right) & if \; x_i^k \beta_i^\eta \leq \max\left(x_{min}, \, x_i^k - m\right) \\ \min\left(1, \, x_i^k + m\right) & if \; x_i^k \beta_i^\eta \geq \min\left(1, \, x_i^k + m\right) \\ x_i^k \beta_i^\eta & otherwise \end{cases} \tag{2.18}$$

Where m, is the move limit which takes on a positive value, η is the numerical damping coefficient (typically 0.5) and β_i is found from the optimality condition according to equation (2.19) which uses the Lagrangian multiplier determined by virtue of a bisection method or a Newton method [22].

$$\beta_i = \lambda^{-1} p x_i^{p-1} u_i^T K_i^0 U_i \tag{2.19}$$

The filtering scheme of the BESO method (equations 2.12 – 2.14) that was used to solve the issues of mesh dependency and checker boarding is replaced with equation 2.20 and equation 2.21. This filtering method operates on the continuous nature of the design domain.

$$\frac{\partial C}{\partial x_i} = \frac{1}{x_i \sum_{j=1}^{N} H_{ij}} \sum_{j=1}^{N} H_{ij} x_j \frac{\partial C}{\partial x_i} \tag{2.20}$$

Where N, is the number of elements in the finite element mesh, r_{ij} is the distance between element i, and element j, and where H_{ij} is the mesh independent weight factor is calculated as:

$$H_{ij} = r_{min} - r_{ij} \tag{2.21}$$

A sample evolution of a SIMP-based topology optimization is shown in figure 2.8. The code provided in the appendix of [64] is utilized to optimize the topology of a cantilevered beam with the settings of nelx = 120, nely = 60, volfrac = 0.5, penal – 3, rmin = 3.5, and er = 0.01. It should be noted that the SIMP based topology optimizations require an additional step of interpreting the design produced by SIMP algorithms. The nearly black and white design field is thresholder at a density of 0.5 to provide a design that is completely solid or void.

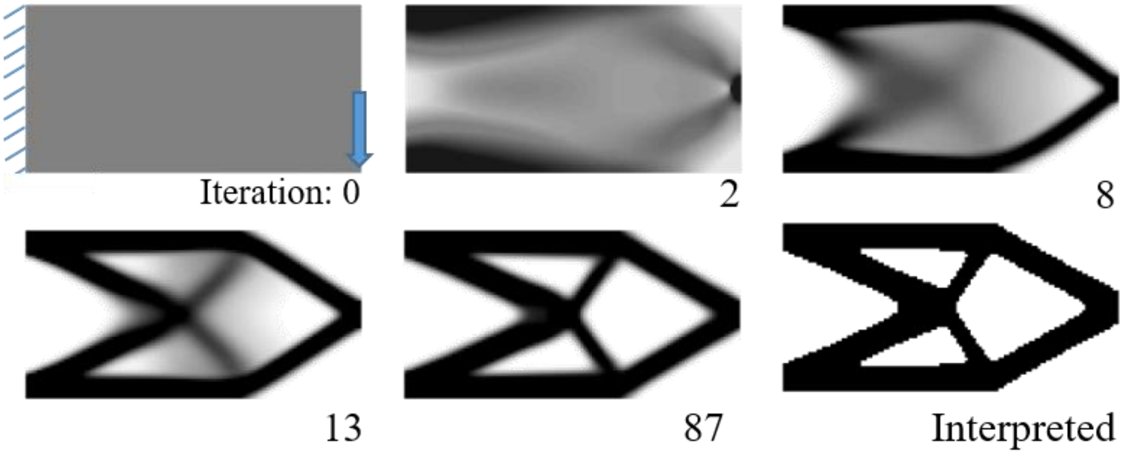


Figure 2.8: Sample evolutionary history of a SIMP algorithm.

2.2 On the Manufacturing of Topology Optimized Structures

The highly efficient and organic-like structures resulting from topology optimization are often difficult or sometimes even impossible to manufacture using traditional manufacturing methods [20]. Manufacturing these structures often requires special tailoring of the topology optimization algorithm for the specific employed manufacturing technique [65, 66]. Some examples of manufacturing constrained topology optimized structures are by means of stamping die components [67], machining [68], molding/casting [69], tolerance-based design [70], and additive manufacturing [71].

Additive manufacturing is by far the most highly investigated technique to produce the topology optimized structures. This is because it the layer-by-layer manufacturing technique allows for fabrication that is almost completely independent of the part's geometric complexity. It has been shown that AM is a cost-effective way to produce plastics, metals, ceramics and even composites of high complexity and small batch sizes [19] and combined with topology optimization, it has the potential to replace conventional manufacturing and design processes [18]. Although AM allows for greater freedom in design for manufacturing, it is not without its own limitations that should be considered [72]. These limitations will be explored in section 2.4.

2.3 Additive Manufacturing

The technology of Additive Manufacturing (AM), also referred to as 3D printing, relies on layer-by-layer material deposition and solidification which provides the potential benefits (over traditional manufacturing) of part consolidation, weight reduction, functional customization, personalization and even aesthetics [73, 74]. Layer-by-layer manufacturing is the process of producing a 3-dimensional object by stacking a number of 2½ dimensional layers on top of each other [75] as can be visualized in the 2½ D decomposition of a hemisphere in figure 2.9. The process of converting a 3D CAD (Computer Aided Design) model into a sequence of finite layers is referred to as slicing. The contours of the slices (layers) are then determined in order to generate a set of machine instructions, and these derived contours are useful for all layer-based additive manufacturing machines.

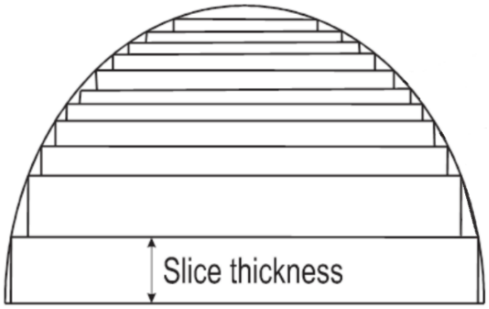


Figure 2.9: A cross-sectional view of the reconstruction of a hemisphere by using 2 1/2D layers of various slice thicknesses [75]

There exist many types of additive manufacturing machines, almost all of which are layer based [76], however only four are commonly used in academia and industry for the manufacturing of- or near net shaping of- end use parts. These common types are Stereolithography (SLA), Fused Deposition Modelling (FDM), Selective Laser Sintering (SLS), and Laser Engineered Net Shaping (LENS). This thesis focuses primarily on a type of SLA printer, but the other common type are briefly described below as the methods presented later on are still relevant to their implementations.

Stereolithography is the oldest form of additive manufacturing [77] which has widespread applications in the modern world [78]. SLA is a process which consists of curing/solidifying a liquid polymer at locations in which a UV LASER or a Digitally Light Processed (DLP) UV-based lamp shine upon the liquid. For a LASER based system, a build platform is able to move up and down as a piston inside of a vat of liquid resin and it will start one layer thickness from the top of the resin line. The LASER then traces out the 2D cross section giving a thickness (2 ½D) based on how far the build platform is lowered into the vat and the characteristics of the LASER beam. A DLP style SLA system functions in much the same way with only two differences, first the LASER UV source is replaced with a household-like theatre projector capable of solidifying the entire cross-section at once rather than tracing it out with the LASER, and second, the build platform starts at the bottom of the vat and the light is shown into the bottom of the VAT through a UV permeable membrane. These two stereolithographic processes can be seen below in figure 2.10.

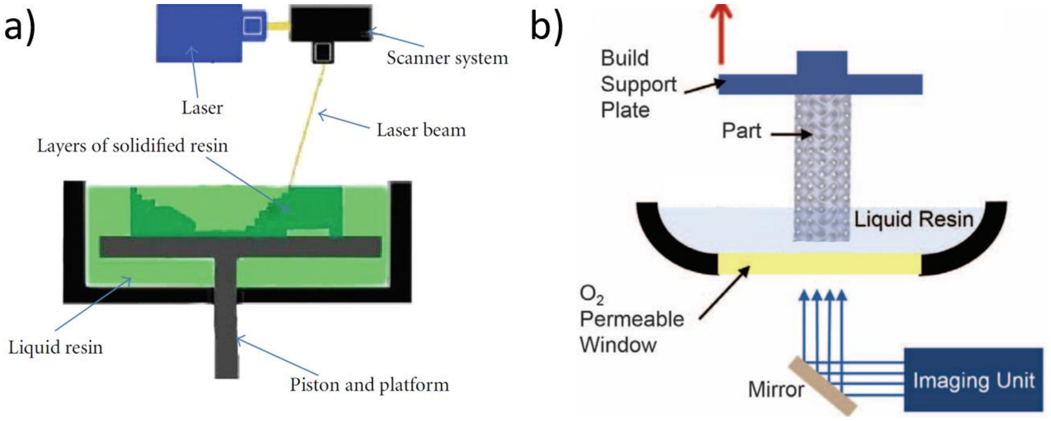


Figure 2.10: A visualization of a) a LASER type SLA [76] and b) a DLP type [79] stereolithographic-based additive manufacturing process.

Fused Deposition Modelling (FDM) -based additive manufacturing processes rely on feeding a thin thermoplastic-matrix filament through a print head that melts the plastic and deposits the liquefied material into a specific location. The print head deposits materials in beads which are traced to form the cross-sectional layer of the 3D object to be manufactured. This is the most cost-effective processes as it requires little post-processing, an inexpensive machine and cheap materials but is slow and has a poor z-resolution compared to the others as well as exhibiting the highest effects of material anisotropy. The print head is usually attached to a Cartesian 3-DOF gantry system but has been successfully implemented on 6-DOF arm as well [80] which allows for easier manufacturing of complex shapes. The process of FDM is visualized in figure 2.11 below.

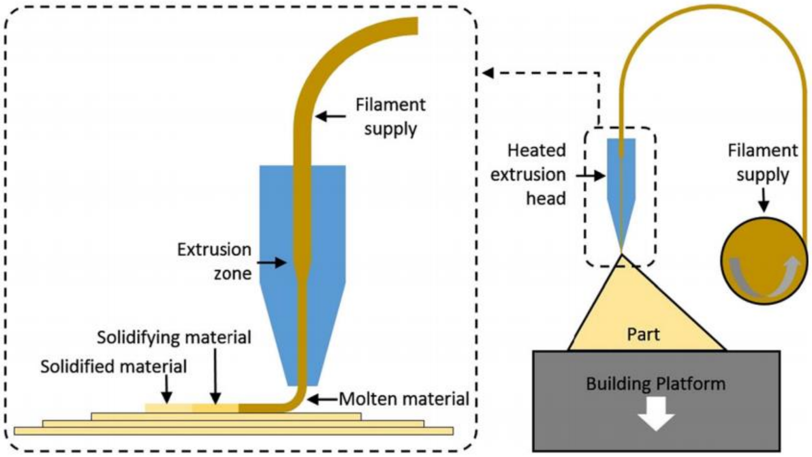


Figure 2.11: A visualization of the Fused Deposition Modelling Process [81]

Selective LASER Sintering (SLS), Selective LASER Melting (SLM), and Direct Metal LASER Sintering (DMLS) are all instantiations of the same concept that functions in much the same way as a LASER-based SLA machine [81]. The main difference between an SLA and an SLS/SLM manufacturing machine is that instead of curing a material from a liquid state into a solid state, the stock material is in the form of a fine powder and is then sintered or melted layer-by-layer by means of a LASER of much higher power than those found on an SLA machine. LASERs found on an SLS machine can range from 7 W for plastics and upwards of 200 W for metals while the LASERS on an SLM machine can be upwards of 400 W since they need to actually melt the powdered material rather than sinter it. DMLS is similar to SLS and SLM processes but differ in that they are limited to only allowing for the utilization of metallic materials. These SLS/SLM/DMLS machines have a large selection of materials available comparable to the FDM process and produce parts with less material anisotropy but this comes at the cost of expensive and hazardous, multi-stage machines as well as expensive material and post-processing costs and it also requires trained personal for its safe and effective operation. A visualization of these types of machines is found below in figure 2.12.

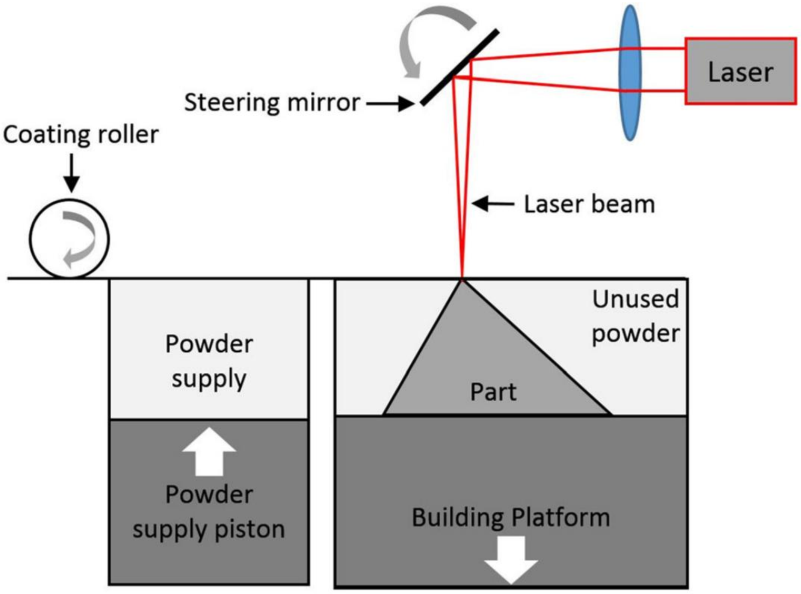


Figure 2.12: A visualization of the Selective LASER Sintering (SLS), Selective LASER Melting (SLM) and Direct Metal LASER Sintering (DMLS) process [81]

The last investigated form of additive manufacturing is known as LASER Engineered Net Shaping (LENS) [81-83]. Similar to an SLM and FDM combined process, the LENS process uses powdered material and a LASER to melt the material but it differs from SLM in that the delivery nozzle (that surrounds the LASER beam) injects the powder stream directly into the focused LASER beam. The printing head (combined LASER and material deposition nozzle) moves in a similar fashion to the FDM process which allows for a LENS-based manufacturing system to make parts that are much larger than the SLS/SLM –based processes but comes with the drawback of lower accuracy and poorer surface finishes. The primary application of this process is for near net shaping of parts on a hybrid manufacturing center capable of both traditional subtractive manufacturing (i.e. CNC mill/ lathe) and additive manufacturing in the same machine. A visualization of the LENS process is seen in figure 2.13.

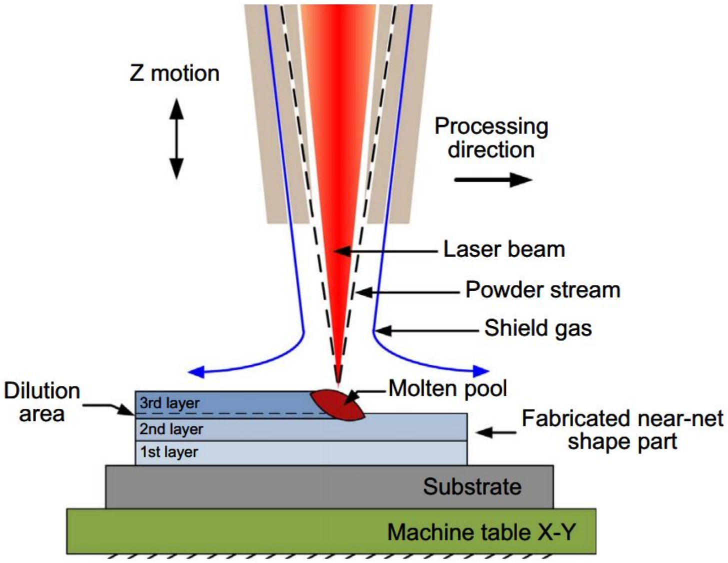


Figure 2.13: A visualization of the LENS process [83]

2.4 On the Constraints of Additive Manufacturing

The list of the four aforementioned additive manufacturing processes are the most common forms but this list is by no means inclusive; the reader may refer to the relevant cited literature for exhaustive lists on all of the modern processes. The DLP style of a SLA – based printer is the main focus of this paper because I was responsible for the design and manufacturing of one of these printers but this focus is rather irrelevant to the applications of the presented techniques since all AM machines have nearly identical limitations. The main limitations are material anisotropy [84, 85], the requirement of supporting structures for overhanging surfaces [4, 5, 86, 87], relatively poor surface finish [75, 88-92], minimum member thickness [56, 93, 94], the necessity of post-processing [95, 96], and AM specific residual stresses [97, 98]. All of the six additive manufacturing drawbacks outlined have been extensively studied within the context of topology optimization.

2.5 Topology Optimization for Additive Manufacturing

In 2011, a serious lack of AM- friendly TO solutions was identified as a serious bottleneck in the industrial implementation of topology optimization [74]. In recent years, tremendous progress has been made to address the six primary limitations of additive manufacturing outlined in section 2.4 [72] as well as to incorporate some unique advantages that AM brings such as porous infill design [99, 100] or functionally graded materials [99, 101]. The main issues that should always be addressed when additively manufacturing a topology optimized structure is the anisotropic nature of layer based manufacturing and the support structure requirements for manufacturing.

Material anisotropy can be classified into two categories, 1) Process induced- and 2) intrinsic- anisotropy [102]. The process induced anisotropy is caused by the layered nature of the manufacturing process [103] resulting in mechanical properties varying in relation to the build orientation and the geometry of the part as well as heavily effecting the fatigue life. Intrinsic anisotropy of the material is a result of anisotropic lattice materials which is actually desirable in many cases for forming functionally graded materials [104].

Some methods to overcome the process induced anisotropy have been proposed such as the stress-based method [25, 105] or the sensitivity-based approach [106]. The stress-based approach is concerned with aligning the material orientation along the principal stress direction which proves useful for shear “weak” orthotropic materials but is however limited in its applicability to complex loading conditions. The sensitivity-based approach is more broadly applicable and is applied by treating the material orientation as a design variable referred to as Continuous Orientation Optimization (COO) but has the drawback of being easily caught in a local optimum. An approach that is completely independent from the topology optimization process itself is by increasing the degree of freedom of the printing nozzle to allow for printing of features in the same part in different orientations [107] or to simply determine the optimal orientation to additively manufacture the part that minimizes the effects of its anisotropic nature [108]. This concept of orientation optimization plays a vital role in the process introduced later in this thesis for the reason for the role of support slimming and minimization of manufacturing costs.

One of the main, and certainly the most studied topics in the discussion of topology optimization for additive manufacturing is of overhanging surfaces where the effect of the inclined angle of the overhanging surface directly correlates to the need of supporting structure [72]. This has been widely experimentally studied [86, 87], showing that that overhanging surfaces induce higher residual stresses and warp easier as the inclined angle becomes smaller relative to the build platform. In order to prevent small inclined build angles from warping, additional support structure should be included alongside the main part being printed. The support structure is often scaffolding-, lattice- or tree-like lattice structures erected from the build platform to the overhanging surface, and often printed using the base material. This supporting structure adds to the required time to print the material, increases wasted material, increases post processing times and negatively effects the surface finish of the part [109]. It as also been showing that 40% to 70% of the costs associated with additive manufacturing are associated with the support structures, their removal and correcting the surface that made contact with on the work piece [72].

Similar to practically every approach to mitigating the effects of the AM limitations on the manufacturing of topology optimization structures, these constraints can either be accounted for in the TO algorithm itself or have its effects minimized in a post processing stage following the TO step. For the instance of reducing the supporting structures, many approaches have been implemented for either cases such as self-supporting topology optimization algorithms at the cost of structural efficiency [74, 110-114], post-modification of the topology to eliminate unsupported regions at the cost of breaking the strict volumetric constraint of TO [115], part orientation optimization “support slimming” which does not completely eliminate the need for supporting structures [4, 5. 116], or just simply design the support structures in a clever way [86, 117-119]. Further constraining the topology optimization algorithm for print ready designs is often the most desirable scenario because of the costs associated with supporting structures but support slimming techniques are used when high structural performance is more desirable than minimizing production costs; a comparison between a self-supporting (print ready) design compared to one that is not is seen in figure 2.14.

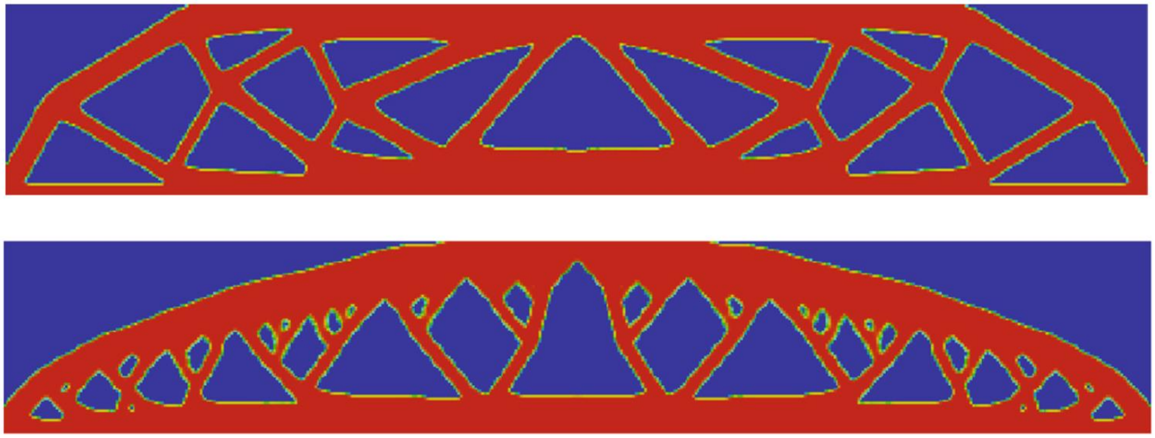


Figure 2.14: An optimal topology for the MBB benchmark problem with no overhang constraint (top) and a print-ready solution with a minimum self-supporting angle of 45° (Bottom) [111]

2.6 Process Planning

A recommended process to achieve a topology optimized structure and its subsequent additive manufacturing has been recommended in [2]. In this publication, it is recommended that this process be fast and flexible to the dynamic customer. It is also outlined that employing a common digital platform and connectivity between the various platforms is a key requirement in order to achieve an integrated level of intelligent design and manufacturing systems. There are nine outlined characteristics that a process should embody like i) **quick** formulation of required product functions, ii) **adaptive** modification of design to the dynamic nature of the manufacturing process, iii) **accurate** transfer of digital design specification to the manufacturing pre-processing stage, iv) **customized** and tailored design for the manufacturing specific process, v) **compatible** generation of machine and inspection specific instructions, vi) **dynamic** collection of digital data from the entire process for informed on the spot decision making , vii) **intelligent** processing of the process specific data and analytics for proper fault diagnosis and defect detection, viii) **compensative** adjustment of the machine instructions to compensate for systematic errors found in the process and, ix) **corrective** operations for defects detected on the final product.

The design and manufacturing process outlined in that paper is a five-step process which was described in chapter 1 of this thesis. This process was outlined as i) problem definition, ii) topology optimization, iii) build orientation optimization, iv) slicing, and v) manufacturing. Chapters 3-6 offer various improvements to the process. Frist, chapter 3 offers the ability for the user to gain control over convergence characteristics of the topology optimization algorithm implemented in step 2 of the outlined process. Chapter 4 and 5 present novel methods for evaluating specifically topology optimized structures for their support structure requirements based on the build angles of overhanging surfaces. This methodology can be used in step 2 of the process to tailor make the TO algorithm to accommodate the AM constraints or this methodology can be used in step 3 to minimize the manufacturing errors and costs. Chapter 6 presents a novel slicing approach that can also be used in step 2 to tailor design the TO algorithm to accommodate for the AM related constraints. The methodology proposed in chapter 6 combined with the methodologies of chapters 4 and 5 eliminate the need to convert the topology optimized structure to an STL representation. This allows for a reduction in computational time, manufacturing errors and enables a more intelligent process via new feedback mechanisms not before realized. These proposed improvements to the process of structural topology optimization and its subsequent additive manufacturing are in line with the characteristics of intelligent design and manufacturing systems.

Chapter 3. Convergence Control for Topology Optimization

3.1 Introduction

The Bidirectional Evolutionary Structural Optimization (BESO) TO algorithm contains a number of control parameters including target volume fraction and/or maximum local stress, material penalty exponent, minimum elemental density, filter radius, mesh resolution, and the focus of this chapter, the Evolutionary Rate (ER). Typically, the ER is set to a static value in the range of 0.5 % to 5 % where a smaller value of ER will result in small design changes between iterations and requiring more computational time in order to reach the volumetric convergence criteria. The reverse is typically true as well (although a very high value of ER can cause the optimization to become unstable and reach a highly non-optimal solution) however, using a smaller value of ER has the advantage of a more stable evolution and typically will result in a more efficient structural design. This chapter aims to convert the static BESO optimization to one that is adaptive in response to feedback of its evolutionary history and progress towards target goals ideally allowing for smaller computational time requirements for structures of even higher efficiency.

Feedback is a highly desirable trait in a process and has often had revolutionary results allowing for drastic performance improvements [120]. Proportional-Integral-Derivative-(PID) controllers are very common in engineering applications to create adaptive processes which relies on a feedback mechanism because of their general applicability to most control systems [121]. In particular, if the differential mathematical model of the plant is not known, an analytical control method can therefore not be used and this is when a PID controller is most useful. This is the case for topology optimization and is the rationale behind choosing a PID controller to design an adaptive BESO algorithm. The volumetric error signal is proposed to be used in the feedback mechanism to dynamically change the algorithms' evolutionary rate. It will later be validated through example that the BESO algorithm behaves as a second-order dynamic linear system and thus validating the implementation of a controller designed for such a use.

For reference, a sample plot of the convergence for a 2D cantilevered beam under bending condition can be seen in figure 3.1 which will serve as a benchmark for comparing the developed convergence control algorithm. This is the result of using a 120 x 60 element finite element mesh of 0.001mm thick plates, a target volume fraction of 0.5, an evolutionary rate of 0.02, material penalty exponent of 3 and a filter radius also equal to 3. The curve containing the white squares is the volumetric history of the convergence starting at an initial guess of every element a solid eventually reaching its target volume in 34 iterations and meeting the convergence criteria after46 iterations. It is worth noting that this has a piece-wise linear style convergence which does not resemble the unit-step response of a second –order dynamic system and has much room for improvement. The curve with the black circle data points is the objective function at each iteration of the optimization. The motivation behind the applied control theory is to remove large amounts of material at the beginning of the optimization when there exists a lot of inefficient material and for only small amounts of material to be removed towards the end as the volumetric error approaches zero.

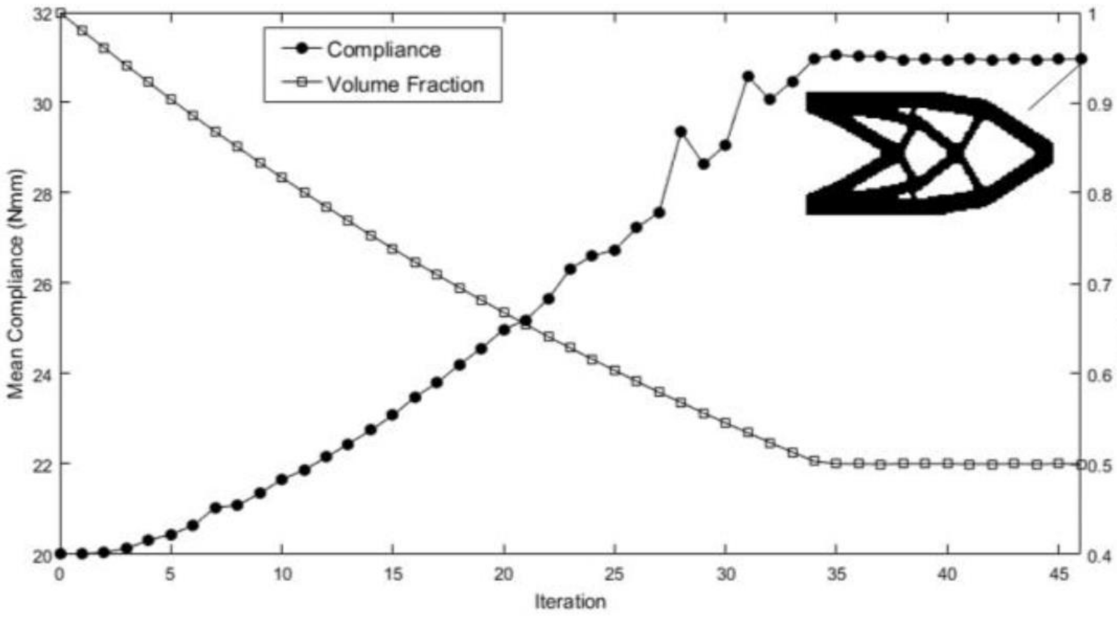


Figure 3.1: Sample convergence history of a short cantilever beam resulting from the standard BESO algorithm.

3.2 PID Control of a Second Order Linear System

In control theory, a standard PID controller is a three-term function each of which operate independently of one another and offer unique trade-offs in performance. Namely the functionalities are the proportional term, the integral term and the derivative term. This transfer function is expressed mathematically as:

$$G(s) = K_p + K_I \frac{1}{s} + K_D s \tag{3.1}$$

The proportional term provides an overall control action which drives the process towards its target proportionally relative to the error signal. The integral term reduces steady-state errors through low-frequency compensation via an integration of the error signal and the derivative term improves the transient response through high-frequency compensation via differentiation of the error signal. The differentiation term is also used to help reject disturbances to the system which in the case of topology optimization, can be seen as spikes in the volumetric convergence history due to members being broken from one iteration to the next. The degree of which each the three individual terms affects the output control is determined by manually adjusting the respective gain values (K_P, K_I, K_D). The effects that increasing these gain values has on the response of the system is seen in table 3.1.

Table 3.1: Effects of independent P, I, and D gain adjustments

	Rise Time	**Overshoot**	**Settling Time**	**Steady State Error**	**Stability**
Increasing $\boldsymbol{K_P}$	Decrease	Increase	Small Increase	Decrease	Degrade
Increasing $\boldsymbol{K_I}$	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing $\boldsymbol{K_D}$	Small Increase	Decrease	Decrease	Minor Change	Improve

In the context of convergence for topology optimization, rise time refers to the number of iterations it takes to reach the target volume for the first time, overshoot is the maximum peak value of the response curve (volumetric error history) measured relative to the target volume. Settling time refers to the number of iterations for the response curve to reach and stay within a predefined range ($\cong 2$ %) of the final target volume. Steady state refers to the difference between the final achieved structural volume and the predefined target volume.

A convenient classification of different types of process convergences is by means of a unit-step response and its associated damping coefficient (ζ). A family of these response curves can be seen in figure 3.2 where the x-axis can be thought of as the number of iterations and the y-axis can be thought of as the current iterations structural volume. In control theory, a damping coefficient of $\zeta < 0$ is considered unstable and will diverge from the targeted value. A system with a damping coefficient $\zeta = 0$ is considered to be marginally stable and corresponds to an indefinite sinusoidal oscillation around the targeted value. If the system is classified as having a damping coefficient of $0 < \zeta < 1$, then this is known as an underdamped scenario which means that the error signal exhibits a damped sinusoidal oscillation, eventually settling down to a value which may or may not be the target value (depending on whether a steady state error exists or not). A system with a damping coefficient $\zeta = 1$ is considered to be critically damped (often times this is the ideal convergence) which experiences no oscillations nor any overshoot before reaching a steady state, and this type of convergence also exhibits the fastest possible convergence in response to a unit step input. Finally, a process with a convergence exhibiting a damping coefficient $\zeta > 1$ is considered to be overdamped and similar to its critically damped brother, this response will not oscillate nor overshoot but will not reach a steady state as quickly. For topology optimization, the response is in terms of the volumetric history of the evolutionary procedure and the error signal is the difference in the target volume vs the volume at a given iteration.

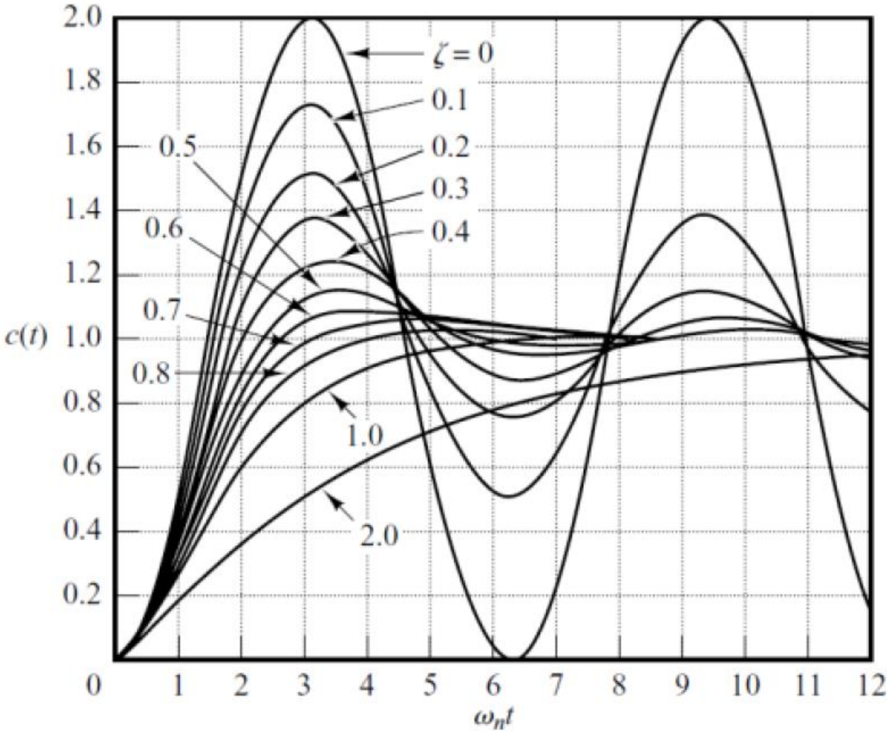


Figure 3.2: Unit-step response curves for a second order system with varying damping ratios [121]

3.3 Methodology

The plot of volume versus iteration number is conceptualized to be any second-order function which starts at an arbitrary volume and then gradually approached and eventually reaches the prescribed target structure volume. For simplicity in explaining the outlined approach, a convergence resembling a decaying exponential, somewhat similar to the form of a critically damped system is used for deriving the three terms of the controller. Figure 3.3 demonstrates this sample convergence as well as the process to approximate the three volumetric error terms as iteration i. The proportional term is found to simply be the height in this volumetric error curve, the additional integral error introduced in this iteration is represented by the shaded box region using a simple rectangle centered at iteration i, and derivative of the error curve is determined as the instantaneous slope of the curve.

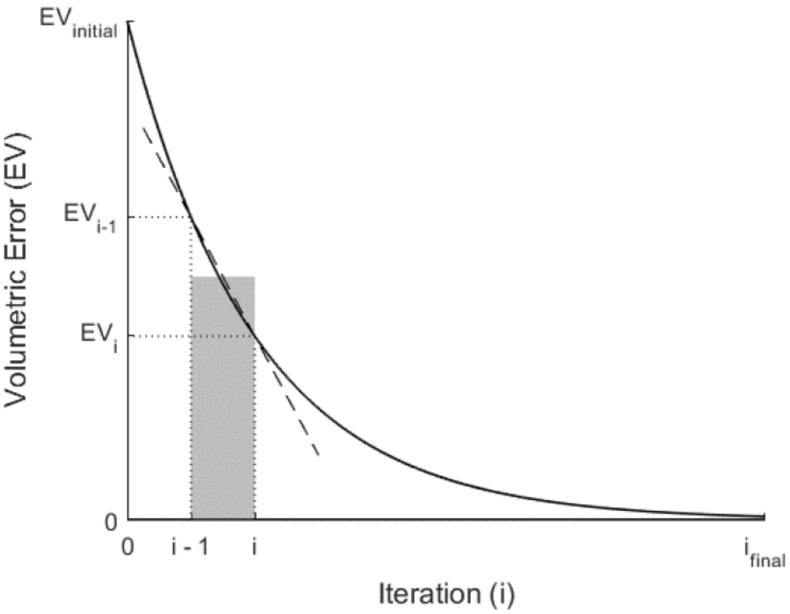


Figure 3.3: Sample volumetric error plot with a visualization of the proportional- (height), additional integral- (shaded) and the derivative- (dashed line) error.

These outlined volumetric error values, and the previously discussed associated user defined gain values are used to calculate the evolution rate for the subsequent iteration. This formulation of the instantaneous evolutionary rate is calculated according to equation (3.2) where E_{V_i}, $\int_0^i E_V$, and E'_{V_i} are the respective instantaneous proportional-, integral-, and derivative volumetric errors and K_P, K_I, and K_D are the gain values associated with their respective volumetric errors.

$$ER_i = K_P E_{V_i} + K_I \int_0^i E_V + K_D E'_{V_i} \tag{3.2}$$

The volumetric error at a given iteration E_{V_i} is calculated simply as the difference between the current iterations volume V_i and the prescribed target volume fraction V^*. This is expressed mathematically in equation (3.3) as:

$$E_{V_i} = V^* - V_i \tag{3.3}$$

The integral volumetric error $\int_0^i E_V$ is calculated to be the total area under the volumetric error curve and approximated using rectangles centered between the previous iteration $i - 1$ and the current iteration i. At iteration i, this new addition of the integral error (shaded region of figure 3.4) is summed with the integral error of the previous iteration ($\int_0^{i-1} E_V$). This integral of the volumetric error is approximated as:

$$\int_0^i E_V \approx \int_0^{i-1} E_V + \frac{E_{V_{i-1}} + E_V}{2}(i - (i-1)) \tag{3.4}$$

Note that if the integral volumetric error is not being calculated from one iteration to the next, the term $(i - (i-1))$ should reflect this as it is simply the width of the approximated rectangle. If the error term is being calculated at each iteration, then the term $(i - (i-1))$ evaluates to simply 1 and is therefore redundant and should be removed when implemented.

The derivative of the volumetric error (E_V') is simply the instantaneous rate of change in the volumetric error curve at iteration i. Since the actual function of the error curve is unknown, this term can be approximated as the slope of the line segment passing through the volumetric error at the current iteration and the volumetric error of the previous iteration. Like the integral volumetric error calculation, the $(i - (i-1))$ can be omitted if the term is evaluated at every iteration; if not, the 1 should be changed to reflect the number of iterations between each calculation of the volumetric error terms. The term E_V' is approximated as:

$$E_V' = \frac{dE_{V_i}}{di} \approx \frac{E_{V_i} - E_{V_{i-1}}}{(i - (i-1))} \tag{3.5}$$

In order to incorporate the new Evolutionary Rate procedure into the TO BESO algorithm, a minor modification needs to be made to the next iterations target volume described by equation (2.9). The old equation does not allow for the next iterations volume to cross the target volume line at all which would eliminate the possibility for underdamped types of convergences. To mitigate this problem, equation (3.6), a continuous form of this volumetric updating equation should be used in lieu of equation (2.9). This new equation works in this case but not in the case of the standard static ER type of BESO because ER is allowed to take on a negative value in the case of PID control. The proposed volumetric updating scheme is:

$$V_{i+1} = \min\left(\max(V_i - ER, 0), 1\right) \tag{3.6}$$

Tuning a PID controller refers to choosing a set of gain values (K_P, K_I, K_D) to meet the desired convergence characteristics for a given process. In the case of topology optimization, it is desirable to minimize the number of iterations because in some cases, iterations take hours each to complete. It is also preferable that the convergence of the optimization allows for the most efficient structure possible. With these preferences in mind, a critically damped type of convergence is best suited for topology optimization because 1) it allows for the quickest convergence of the optimization algorithm and 2) the nature of the convergence that is rapid when it is far from the target and slow, minor changes when the target volume has almost been met. The small changes at the end allows for fine adjustments to be made to the structure that would not be possible with the traditionally used static evolutionary rates.

Tuning a PID controller appears to be conceptually easy as it is only tasked with choosing three terms but it may be difficult in practice such as if there are multiple conflicting convergence objectives like quick and stable [122]. There exist many methods to experimentally determine the gain values for the desired case of a critically damped process but the majority of them rely on first starting with small proportional gain value then gradually increasing it until the convergence becomes marginally stable. This is not however possible with TO because as the proportional gain is increased (or the static ER for that matter), the optimization algorithm breaks down far ahead of the convergence becoming marginally stable. Instead, an alternate tuning process is proposed.

For tuning the PID controller, it is suggested to start with a derivative gain of zero and a very small value of the integral gain (0.0001 < kI < 0.001) then slowly increasing the proportional gain until just after the algorithm has reached similar performance (e.g. similar compliance and settling time) as the standard algorithm. The user should then start increasing the derivative gain to make the volume convergence plot resemble that of a critically damped system. If the system seems to take too long to converge, the user should increase the integral gain slightly. The authors have find that the controller exhibits similar performance characteristics for varying mesh resolutions so it is suggested that the tuning be carried out on a low-resolution mesh then applied to the applicable high-resolution mesh.

3.4 Results and Discussion

The goal of first example of a topology optimization with the implemented control theory is shown in figure 3.4 that achieves the same structural performance to that of the benchmarked example of figure 3.1 but with a more desirable convergence. For this particular example, tuning values of $K_P = 0.125$, $K_D = 0.35$, and $K_I = 0.002$ with all of the other settings remaining the same as the benchmark example. Although the resulting topology is slightly different, it can be seen that the resulting compliance of the structure is identical (30.94 Nmm) but the optimization met the convergence criteria in six fewer iterations. This is likely as result of the far steeper initial evolution and then very small evolutionary rates used towards the end of the convergence allowing for fine design adjustments.

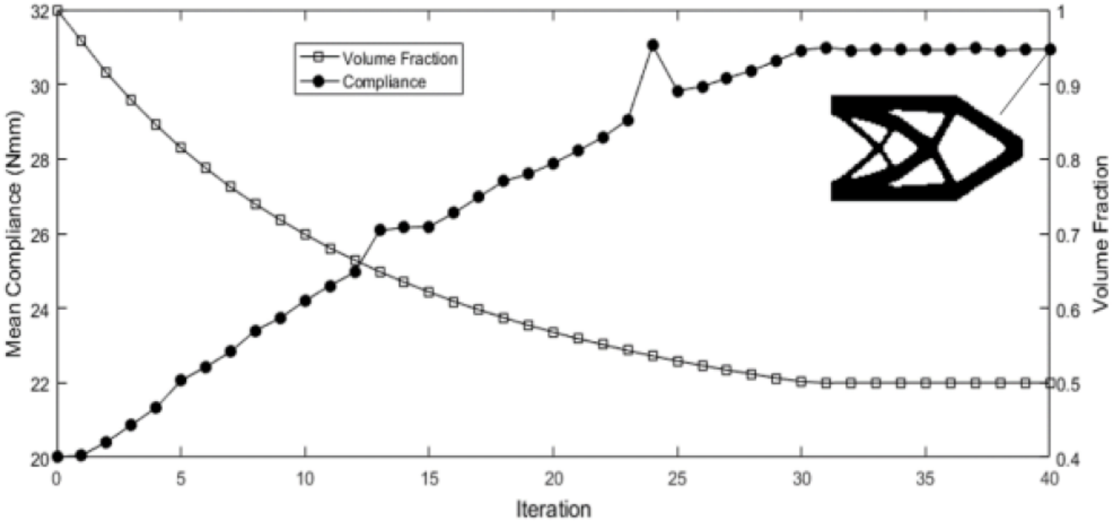


Figure 3.4: Convergence history for the benchmarked cantilever with PID controlled evolutionary rate tuned at $K_P = 0.125$, $K_D = 0.35$, and $K_I = 0.002$.

A second example on the implementation of the PID controlled evolutionary rate is presented in figure 3.5 with the goal of validating the use of control theory design for second-order dynamic systems. In literature, the volumetric convergence always takes on a sort of piece-wise linear convergence with no signs of actually behaving like a second order system so this example aims to demonstrate that the behavior of the optimization is at least that of a second order system. In order to achieve this, a large value for the integral and proportional gains should be used to induce overshoot but not too much that the optimization becomes unstable. A tuning of the three gain values of $K_P = 0.2$, $K_D = 0.3$, and $K_I = 0.04$ was found to achieve this desired goal. The results of this optimization and its convergence history can be seen in figure 3.6. It is clear that the system behaves like that of an underdamped system with a damping coefficient of approximately $\zeta = 0.3$. The volume of the structure overshoots its target volume of 50 % and reaches a volume fraction of 29 % and then overshoots the target again before eventually settling down at the target volume fraction. The structure at the time of these maximum overshoots can be seen in the figure as well as the final resulting structure.

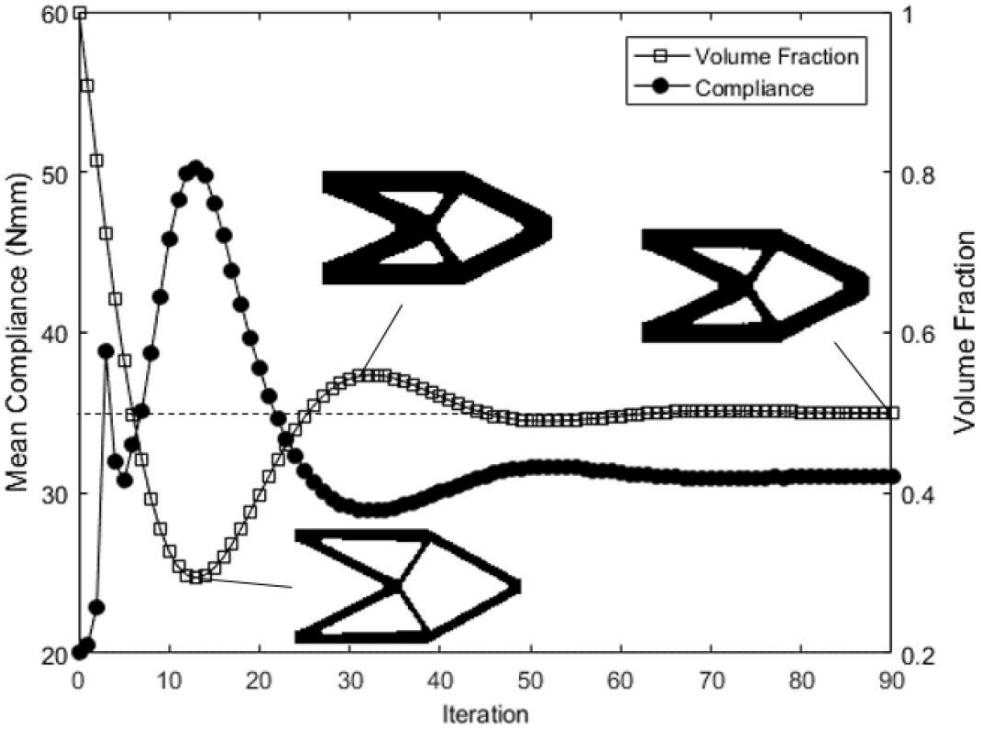


Figure 3.5: Convergence history for the benchmarked cantilever with PID controlled evolutionary rate tuned at $K_P = 0.2$, $K_D = 0.3$, and $K_I = 0.04$.

The resulting structure has a resemblance to the results of the cantilevered beam at low resolutions of mesh which suggests that this form of overshooting the target volume could be used for mesh independent topology optimization in lieu of the filtering scheme described in section 2.1.2.1. It is also worth noting that the evolutionary rate of the BESO algorithm plays the similar role as the move limit in the mathematical programming of the step size of the SIMP algorithm [123]. This resemblance in the algorithm structure suggests that PID control of the SIMP algorithm's move limit is a possible avenue for future research.

3.5 Conclusion

This chapter presented an improvement to the BESO based topology optimization algorithm which allows the user control over various aspects of the convergence of the optimization which has not before been realized. It is shown that the PID control of the evolutionary rate is an effective control parameter to turn the static optimization algorithm into one that is adaptive. It is also suggested that the highly underdamped case of this adaptive BESO algorithm has potential to serve as a computationally efficient way to ensure mesh independence. Another potential avenue of research based on this work is to extend this control theory to the SIMP algorithm for control of its bi-section method's move limit.

Chapter 4. On Density Gradients for Overhanging Feature Analysis

4.1 Introduction

Overhanging surfaces is the main geometric constraint for additive manufacturing as outlined in the recent paper on the topic [72]. It is shown that as the inclined angle of an overhanging surface relative to the build platform directly correlates to the need of supporting structure because these surfaces induce higher residual stresses and therefore warp easier as experimentally validated in [87]. In order to prevent these inclined build surfaces (overhanging surfaces), supporting structure should be printed simultaneously alongside the part itself to help support these surfaces and prevent warping. Each particular additive manufacturing process has its own self-supporting surface inclination angle at which an overhanging surface can be printed without supports such as the generally accepted 45° for FDM and SLM type printers of 25° for DLP style SLA printers a visualization of these minimum self-supporting build angles can be seen in figure 4.1. These minimum self-supporting build angles are both machine and process dependent and are usually experimentally determined but they could be reasonably approximated with a good multiphysics simulation tool which takes into account the transient and highly dynamic properties of a material undergoing solidification at the edge of a layer.

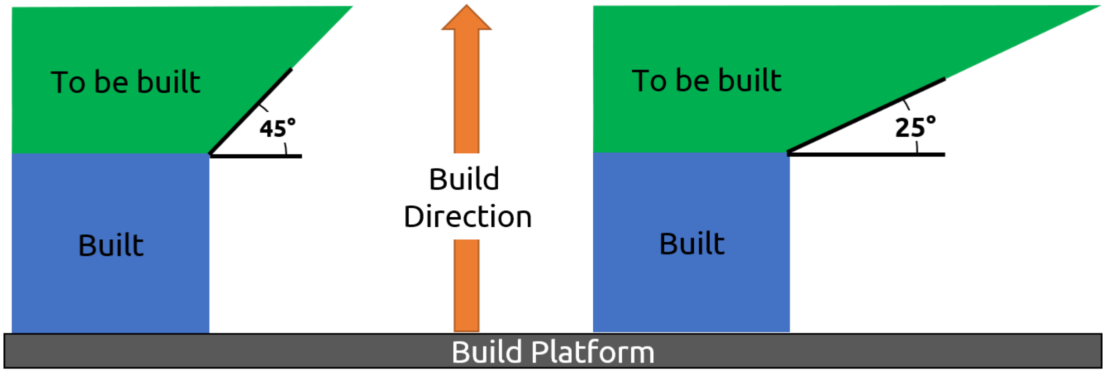


Figure 4.1 Visualization of a 45° (left) and a 25° (right) overhanging surface.

It is crucial to identify which overhanging surface require the need for supporting structure in order to successfully additively manufacture an object and therefore it is important to have a useful algorithm to do so. Typically, the results of the topology optimization are converted to an STL (Standardized Tessellation Language) formatted representation by means of a marching cubes algorithm [124] which allows for easy identification of the overhanging surface angle by means of the normal vector of the triangular facets. One goal of this thesis is to eliminate the need for this unnecessary STL conversion and another is to allow for layer-based manufacturing planning to be performed in the topology optimization loop which is highly inefficient if the structure needs to be converted to an STL format at every iteration. The proposed methodology involves approximating the density gradient of the design matrix in the SIMP based algorithm or the sensitivity gradient when concerned with the BESO algorithm to be used for determining the build angles.

4.2 Elemental Density Gradients in 2D

The definition of a gradient is the vector formed by the operator ∇ acting on a scalar function at given point in a scalar field. In the context of this thesis, the given point refers to a given element and the scalar field is the matrix of elemental density values. The density gradient vector is composed of both a scalar magnitude and a direction which points in the direction of maximum increasing magnitude. For visual reference, figure 4.2 depicts 9 sample elements with their associated density values and the grey/white regions represents an overhanging surface of a SIMP based design. The blue arrow is a visual representation of the density gradient vector ∇x_e point in the direction θ_e of maximum increasing density.

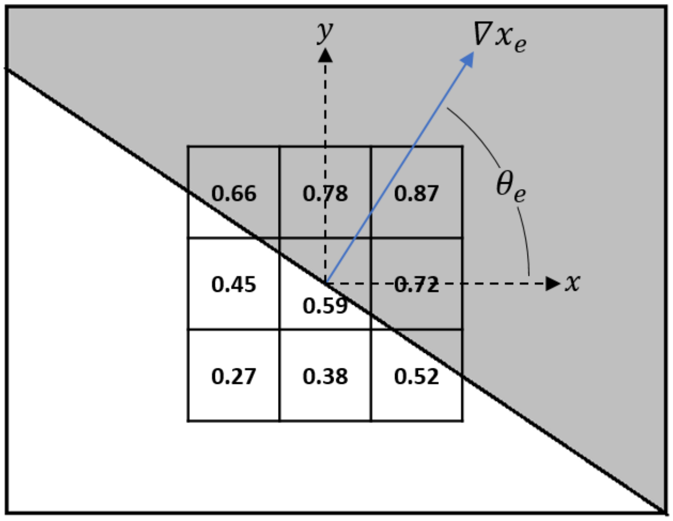


Figure 4.2: A visual of the elemental density gradient vector (blue) measured at an element lying on the edge/ surface of the SIMP part

Mathematically, the density gradient vector (∇x_e) is conveniently represented in its polar form by both its magnitude ($|\nabla x_e|$) as well as the vector's associated direction (θ_e) measured from the positive x-axis. This is expressed as:

$$\nabla x_e = |\nabla x_e| \angle \theta_e \tag{4.1}$$

To calculate the magnitude and the direction of the gradient vector, the Cartesian components need to first be calculated. The magnitude and direction of the gradient vector expressed in terms of its Cartesian components is as follows:

$$|\nabla x_e| = \sqrt{\nabla x_{e_x}^2 + \nabla x_{e_y}^2} \tag{4.2}$$

$$\theta_e = \tan^{-1} \frac{\nabla x_{e_y}}{\nabla x_{e_x}} \tag{4.3}$$

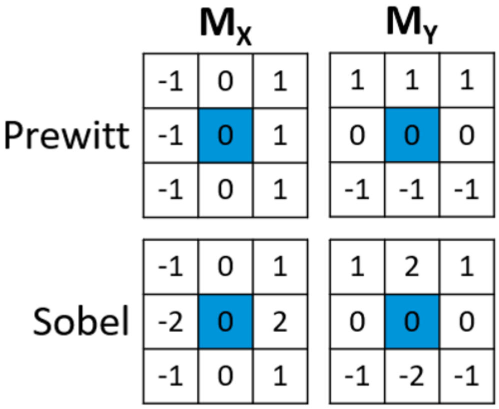
Since the SIMP representation of the structural design is represented by finite elements and not a continuous function, the Cartesian components of the elemental density gradient vector cannot be exactly calculated and therefore must be approximated. For a convenient approximation method, the elements of the finite element mesh will initially be assumed to be all perfect squares / cubes. This representation allows for a similar analog comparison between the SIMP structural representation and a greyscale image. This is convenient because there exists well established techniques in the field of computer vision to approximate the Cartesian components of the gradient vector based on the light intensity (greyness) of the individual pixels. This methodology is used in computer vision to determine the magnitude of the gradient vector for edge detection algorithms [125] but is repurposed in this application to determine the overhanging surface angle of a work piece relative to the build platform. This implementation has been demonstrated in terms of topology optimization in works [113, 114] for 2D examples already but has not been demonstrated in 3D likely because the 3D computer vision algorithm has not yet been presented in extension to 3D. Note that it will be shown later how this 2D computer vision algorithm will be extended into 3D and then generalized as a filter (like the SIMP filter of equation 2.20) for a finite mesh composing of any type elements, not just squares. It is worth mentioning for validation purposes that another matrix convolution kernel from computer vision known as the Gaussian filter has been demonstrated as an effective mesh dependency/ scale invariance filter in lieu of equation (2.20) [55].

The Cartesian components of the elemental density gradient vector are approximated by means of matrix convolution with either the Prewitt (standard) or the Sobel (filtered) kernel matrices. An x-axis kernel matrix is used for approximating the density gradient across the y-axis (in the x-direction) and a separate y-axis kernel matrix is used for approximating the density gradient across the x-axis (in the y-direction). For a detailed explanation on matrix convolution, the reader may refer to chapter 3 the open sourced textbook [125] and may refer to chapter 5 for its use specifically in approximating gradient vectors of discrete scalar fields. The Cartesian components of the density gradient vector are approximated by using the following two matrix convolution operations:

$$\nabla x_{e_x} = \boldsymbol{x} * M_x \tag{4.4}$$

$$\nabla x_{e_y} = \boldsymbol{x} * M_y \tag{4.5}$$

Where $\boldsymbol{x}$ is the entire density field and M_x and M_y are the convolution kernels which are chosen to be either the Prewitt the Sobel matrices. The Sobel mask is essentially just the Prewitt mask that has been combined with the Gaussian distance-based weighting for noise suppression purposes and is therefore the choice for implementation, however the Prewitt masks shows well the operating principles of these kernels and are thus displayed as well. It is worth noting that exists other kernels that serve similar purposed such as the Roberts mask or the Sobel-Feldman [126] but they are not explored in this thesis. The pair of Prewitt and the pair of Sobel masks (kernels) are shown in figure 2.3.



	M_X			M_Y		
Prewitt	-1	0	1	1	1	1
	-1	0	1	0	0	0
	-1	0	1	-1	-1	-1
Sobel	-1	0	1	1	2	1
	-2	0	2	0	0	0
	-1	0	1	-1	-2	-1

Figure 4.3: Two Examples of pairs of masks used to approximate the Cartesian components of gradient vector in a discrete scalar field.

For the sake of computational efficiency, the gradient vectors need to only be analyzed at perimeter/surface elements of the structure. These specific elements are determined by checking their immediate neighborhood; if the element is a solid element and a neighboring element of it is a void, then this element lies on an edge/surface and should be included in the gradient analysis. A sample of this analysis can be seen in figure 4.4 for the angles are visualized as a specific color and are measured relative to the positive x-axis.

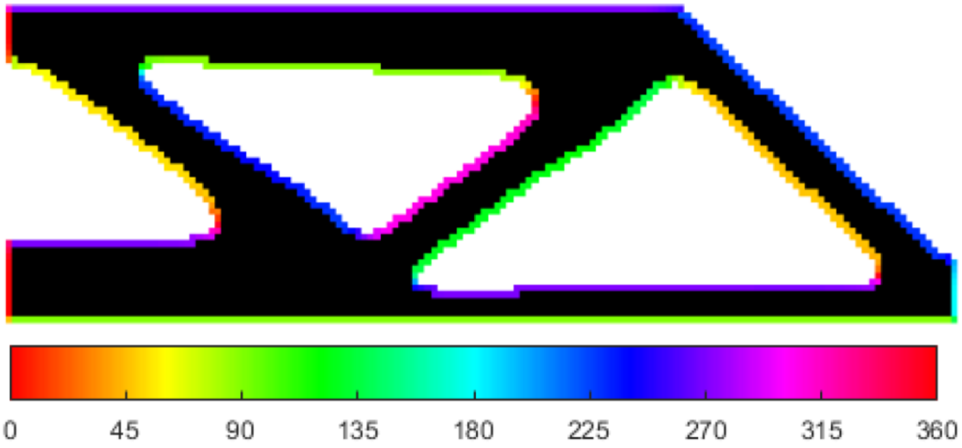


Figure 4.4: A Visualization of the density gradient vector directions at the perimeter elements measured CCW from the element's local positive x-axis

Once the direction of the gradient vectors have been determined at the perimeter/surface elements, the overhanging surfaces and their corresponding build angles can then be determined. This is done by simply determining which of the perimeter elements have a gradient direction of $0 < \theta_e < 180°$. The build angle of the surface is only measured from 0 to 90° as it is measured relative to the build platform and not the positive x-axis. The visualization of the structures overhanging surfaces and its corresponding build angles can be seen below in figure 4.5.

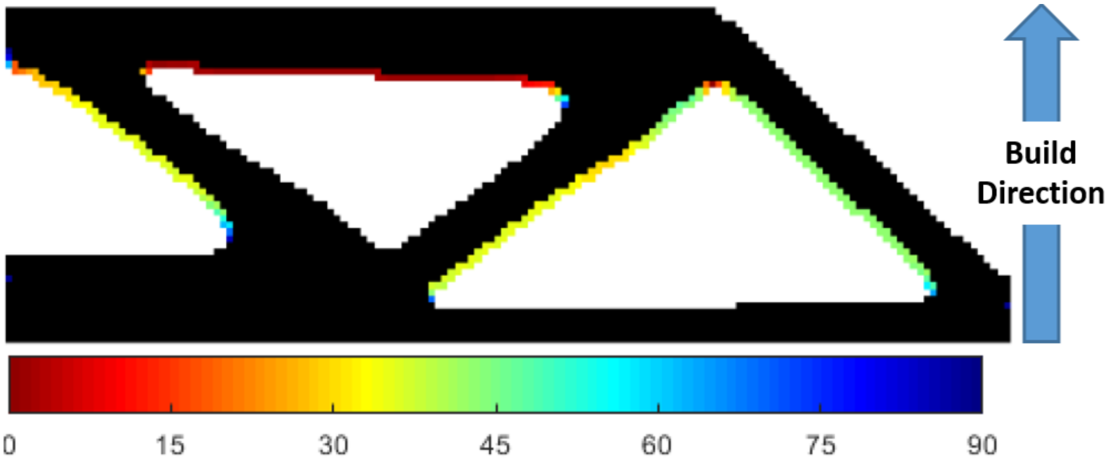


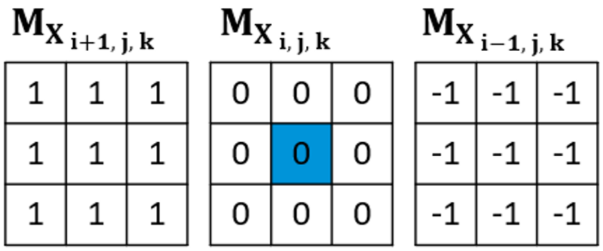
Figure 4.5: A Visualization of the overhanging surface build angles

4.3 Elemental Density Gradients in 3D

As far as I know, it has not been shown in literature how to calculate the density gradient vectors for a 3D finite element representation of a topology optimized structure. This section will overview first how to approximate the density gradient vectors for a commonly used cube-only mesh and then extend this derivation into a filter-based approach applicable to a finite element mesh consisting of any type and combination of elements. In order to achieve this, the derivation of the 2D computer vision gradient masks will be briefly explored then a set of masks to perform the same task in 3D is presented.

In the case of the x-axis Prewitt mask, it operates by simply finding the difference in magnitude across the y-axis which itself when calculated has its own magnitude. That is to say the greater the difference between the magnitudes across the local y-axis of the element, then the greater the magnitude that the x-axis gradient component will take on. In the case of the y-axis Prewitt mask, the difference in magnitude across the x-axis is analyzed. This suggests that for analysis of the 3D gradient in the x-axis, the difference in magnitude should be analyzed across the local y-z-plane of a particular element. Likewise, for the y-component of the 3D gradient vector, the difference in magnitude should be determined across the x-z plane, and for the z-component, the y-z-plane should be used.

In 2D, the gradient masks consist of only a 3x3 matrix, but when extended into 3D it should be considered as a 3x3x3 kernel for the matrix convolution. This 3-dimensional kernel is best thought of as set of three, 3x3 masks for visual interpretation. A sample set of these Prewitt-equivalent masks for use in approximating the gradient in the x-axis is shown in figure 4.6.



$\mathbf{M}_{\mathbf{X}\,i+1,j,k}$

1	1	1
1	1	1
1	1	1

$\mathbf{M}_{\mathbf{X}\,i,j,k}$

0	0	0
0	0	0
0	0	0

$\mathbf{M}_{\mathbf{X}\,i-1,j,k}$

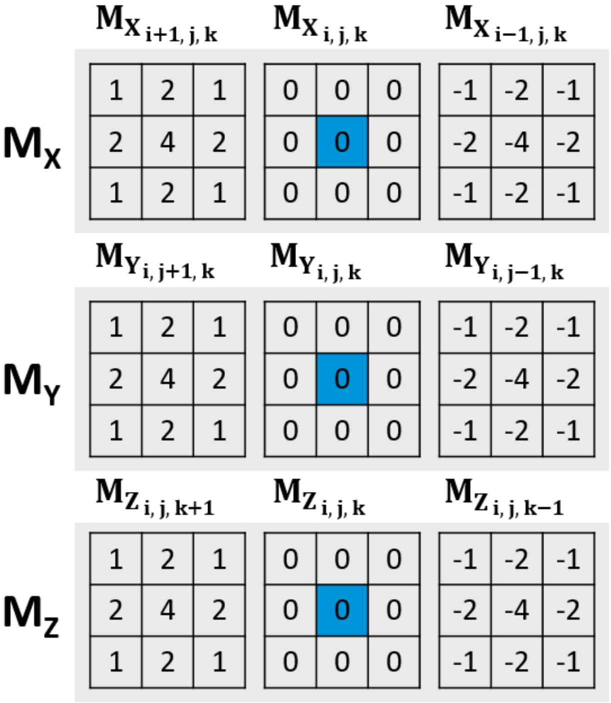
-1	-1	-1
-1	-1	-1
-1	-1	-1

Figure 4.6: 3x3x3 Prewitt-Equivalent kernel for approximating the x-component of the 3D gradient vector

Although this form of 3D Prewitt equivalent kernel would function well for its intended use case, it is shown that additionally weighting the elements of the masks based on their distance from the central element helps reduce noise which in turn provides an approximation that is truer to the actual value [125]. In 2D, the Sobel mask is the result of this additional distance-based weight factor within the kernel which uses an additional Gaussian distribution on top of the Prewitt form of gradient approximation. Essentially, any element of the Prewitt is multiplied by the Gaussian distance-based weight. The equation of a Gaussian distribution is as follows:

$$g(d) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{d^2}{2\sigma^2}} \tag{4.6}$$

Where d the distance between two elements is, $g(d)$ is the distance based Gaussian weight used for the Sobel-equivalent kernel and σ is the spread of the distribution which controls how heavily the elements further from the central element are weighted. Note that the term $\frac{1}{\sigma\sqrt{2\pi}}$ is a constant value for all distances which when implemented in matrix convolution is made redundant because it will be divided out anyways. It is suggested that this term can simply be omitted during implementation [125]. The variable σ is a user-defined constant which should be chosen such that all of the weights turn out as integer values which makes display in the ask form simpler. A more profound choice of this variable will be recommended later when this is implemented into its filter-based form. Figure 4.7 presents a set of three masks that are suggestion for use in approximating the density gradient vector in 3D. The matrix convolution is very similar to that of 2D in that the blue element of the mask is centered at the element of interest, then the other elements of the mask correspond to position of elements in relation to the central element. The weights of the mask are multiplied by their corresponding element’s density, summed up and divided by the total sum of the weight values.



$$\mathbf{M_X} \quad \begin{array}{ccc} \mathbf{M_X}_{\mathbf{i+1,\,j,\,k}} & \mathbf{M_X}_{\mathbf{i,\,j,\,k}} & \mathbf{M_X}_{\mathbf{i-1,\,j,\,k}} \\ \begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} & \begin{bmatrix}-1&-2&-1\\-2&-4&-2\\-1&-2&-1\end{bmatrix} \end{array}$$

$$\mathbf{M_Y} \quad \begin{array}{ccc} \mathbf{M_Y}_{\mathbf{i,\,j+1,\,k}} & \mathbf{M_Y}_{\mathbf{i,\,j,\,k}} & \mathbf{M_Y}_{\mathbf{i,\,j-1,\,k}} \\ \begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} & \begin{bmatrix}-1&-2&-1\\-2&-4&-2\\-1&-2&-1\end{bmatrix} \end{array}$$

$$\mathbf{M_Z} \quad \begin{array}{ccc} \mathbf{M_Z}_{\mathbf{i,\,j,\,k+1}} & \mathbf{M_Z}_{\mathbf{i,\,j,\,k}} & \mathbf{M_Z}_{\mathbf{i,\,j,\,k-1}} \\ \begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} & \begin{bmatrix}-1&-2&-1\\-2&-4&-2\\-1&-2&-1\end{bmatrix} \end{array}$$

Figure 4.7: Proposed set of Sobel-equivalent masks to approximate the 3D density gradient vector.

The explanation of calculating the magnitude of the gradient vector and its angles relative to the x-y and the y-z plane of the polar form of the vector are omitted because they are not useful in the implementation of the analysis. What is important, is the build angle of the surface elements relative to the build platform of the additive manufacturing machine. This is achieved by using the dot product between the density gradient vector and the build platform. The user defined normal vector of the build platform/ direction vector of the build direction is of the form:

$$\vec{n} = (n_x, n_y, n_z) \tag{4.7}$$

And the gradient vector is in the form:

$$\overrightarrow{\nabla x_e} = (\nabla x_{e_x}, \nabla x_{e_y}, \nabla x_{e_z}) \tag{4.8}$$

Analogous to equation (4.4) and (4.5), the Cartesian components of the gradient vectors for a given element are calculated by means of matrix convolution between the density field and the gradient masks of figure 4.7. The operation will return a matrix that is the same size as the original density matrix for each Cartesian components, which are approximated as:

$$\boldsymbol{\nabla x_x} = \mathbf{x} * M_{X_{i+1,j,k}} + \mathbf{x} * M_{X_{i,j,k}} + \mathbf{x} * M_{X_{i-1,j,k}} \tag{4.9}$$

$$\boldsymbol{\nabla x_y} = \mathbf{x} * M_{Y_{i,j+1,k}} + \mathbf{x} * M_{Y_{i,j,k}} + \mathbf{x} * M_{Y_{i,j-1,k}} \tag{4.10}$$

$$\boldsymbol{\nabla x_z} = \mathbf{x} * M_{Z_{i,j,k+1}} + \mathbf{x} * M_{Z_{i,j,k}} + \mathbf{x} * M_{Z_{i,j,k-1}} \tag{4.11}$$

As mentioned, the build angle of a particular surface element (γ_e) is calculated using the inverse cosine of the dot product between the gradient vector and the build platform/build direction. This is shown mathematically as:

$$\gamma_e = \cos^{-1}(\overrightarrow{\nabla x_e} \cdot \vec{n}) \tag{4.12}$$

A visualization of these build angles for the overhanging surfaces of a 3D topology optimized structure is presented in figure 4.8. The left image of the figure is a visualization of the build angles directly overlain on top of the finite element mesh itself where the color of the facet corresponds to the build angle of that particular element in relation to the build platform. The right picture in the figure is a smoothed version of the structure which is more visually appealing and could even be considered as slightly easier to manufacture.

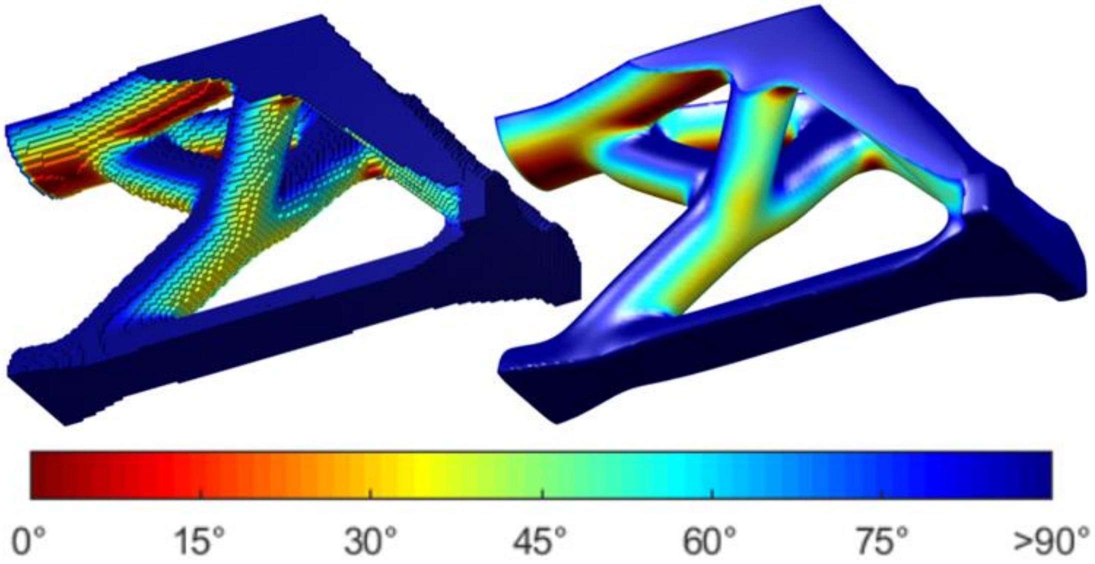


Figure 4.8: Sample analysis of the build angles of overhanging surfaces on a topology optimized structure in its finite element form (left) and a smoothed isosurface form (right)

A customized MATLAB script to calculate the 3D density gradient vectors has been included in Appendix A which may simply be adapted into the benchmark 3D topology optimization code provided in Appendix C of [35]. To approximate the density gradient vector, simply add this line of code after line 90 of the aforementioned topology optimization code.

```
91    [magnitude, direction] = DensityGradient(x);
```

The code provided in Appendix A of this paper should be included as either an auxiliary function to your script or as a standalone script contained within the working directory of the program. The function will return two matrices of the same size as the density matrix, one containing the magnitude of the elemental density gradient vector, and the other is the angle of the vector measured from the XZ-plane (assumed build platform).

To visualize the results, the `display_3D` function can be simply modified to display the direction of the density gradient vector using a color map such as the jet color scheme to plot the elements instead of plotting them as greyscale visualization of the densities similar to figure 4.8.

4.4 Filter Approach for a Mesh Consisting of any combination of Finite Element Types

The methods presented in this chapter thus far are concerned with topology optimizations whose design domains consisted of only squares (in 2D) and cubes (in 3D) but most practical applications decompose the design domain into multiple types of finite elements. A reconstruction of the computer vision derivation to form the masks of the previous sections is performed to achieve a continuous form of density gradient analysis based on the SIMP filtering presented in equations (2.20) and (2.21) which is applicable to any finite element mesh.

An approximation of the Cartesian components of the density gradient analysis is presented in equations (4.13 – 4.16). The familiar SIMP filtering scheme is used with some minor changes. First, the filter is split into three components to analyze the gradient in the x-, the y- and the z- directions individually with an additional directionality term added ($\frac{d_{ij}}{|d_{ij}|}$) where d_{ij_x} is the x-component of the distance between elements i and j. These functions essentially perform the exact same operations as the matrix convolution.

$$\nabla x_{e_x} = \frac{1}{\sum_{j=1}^{N} G_{ij}} \sum_{j=1}^{N} G_{ij} x_j \frac{d_{ij_x}}{|d_{ij_x}|} \tag{4.13}$$

$$\nabla x_{e_y} = \frac{1}{\sum_{j=1}^{N} G_{ij}} \sum_{j=1}^{N} G_{ij} x_j \frac{d_{ij_y}}{|d_{ij_y}|} \tag{4.14}$$

$$\nabla x_{e_z} = \frac{1}{\sum_{j=1}^{N} G_{ij}} \sum_{j=1}^{N} G_{ij} x_j \frac{d_{ij_z}}{|d_{ij_z}|} \tag{4.15}$$

The filter weight (H_{ij}) of equation (2.20) can be used in lieu of the new proposed Gaussian weight factor (G_{ij}) but it is recommended that the Gaussian weight factor is used for the purpose of noise reduction in order to achieve accurate approximations. The proposed weight factor for this filter as follows:

$$G_{ij} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{d_{ij}^2}{2\sigma^2}} \tag{4.16}$$

This Gaussian weight factor has a component known as the spread factor (σ). It is recommended that this factor reflect the length scale used in the topology optimization process so that the local gradient analysis does not encompass multiple members and to ensure that the surface in question is adequately analyzed. It is recommended that the spread factor is taken as a third of the filter radius used in the SIMP optimization such as presented in equation (4.17).

$$\sigma = \frac{r_{min}}{3} \tag{3.1}$$

4.5 Conclusion

In this chapter, an improvement to the traditional topology optimization and additive manufacturing process is proposed which allows for the evaluation of the build angles associated with the overhanging surfaces. A Computer vision method is implemented for the 2D cases and then this method is extended into 3D to allow for the analysis of more realistic structures. Identifying and classifying the surfaces that require supporting structure is critical to the end-use success of the additively manufactured part. The work from this chapter will combine with the next two chapters to make a further improvement to the process by eliminating the need to convert the structure into an STL file ultimately saving computational time as well as a reduction in geometrical errors. Chapter 5 directly uses these methods to perform a novel build orientation optimization.

Chapter 5. Build Orientation to Minimize Manufacturing Costs

5.1 Introduction

The functionality of AM produced parts is highly dependent on its geometrical and its dimensional accuracy as well as its surface integrity [88]. As previously outlined, supporting structure is required to be printed alongside the work piece to support overhanging surfaces with small build angles to prevent surface warping and to ensure dimensional accuracy. These support structures ("supports") are instrumental in providing a successfully produced AM part but are not without their own limitations. First, the supports are often times made using the same material as the primary work piece which is sometimes expensive, particularly in the case of DMLS types of prints. Second, every support requires manual labor to be removed and again in the case of a DMLS type of print, this removal can be expensive as it can require the use of expensive machine tools which require a skilled worker to operate. Third, these supports are attached directly to the surface of the part which directly affects the surface integrity that they are attached to, even if post machining/ post surface treatment is implemented. Lastly, a method of support slimming is often implemented to minimize the volume of Required Support Volume (RSV) but this does not necessarily minimize manufacturing costs because although the amount of supporting structure is reduced via part orientation optimization, the manufacturing time and associated costs may be greatly increased. This concept of support slimming by means of part orientation optimization is first explored then a more general and novel approach to minimize the total manufacturing costs is proposed.

5.2 Support Slimming by Means of Part Orientation Optimization

Support slimming has been extensively studied in recent years as the popularity of AM in both academia and industry has increased. This thesis aims to achieve support slimming by means of part orientation optimization although other methods exist like clever support structure design, self-supporting work piece designs or modification of the work piece to become self-supporting but these methods come with their own associated drawbacks. Build orientation optimization refers to the reorienting the work piece relative to the build direction of an AM process. When the part is re-oriented, the build angles of the overhanging surfaces relative to the build platform vary and therefore an orientation that minimizes the volume of supporting structure may be found.

Part orientation optimization has been studied on the STL representation of the structure in [127, 128] but has not been extensively studied in the finite element representation of the structure [5]. It is worth noting here that Chapters 4-6 combine in such a way that makes the conversion from the finite element mesh to an STL file unnecessary which ultimately saves computational effort. Chapter 4 presented a method for analyzing the build angles of overhanging surfaces which will be used in this chapter for the part orientation optimization analysis. Building off of figure 4.5 of the previous chapter, an analysis of RSV can be performed. This is done by looking at each element on the overhanging surfaces and determining whether or not it is below the minimum self-supporting build angle; if the build angle is lower than the self-supporting build angle then a support is required and first starts at the overhanging surface then is extended downwards until it reaches another surface or it reaches the build platform. This process is continued for all of the elements that lie on the overhanging surface and the total volume of these supports is summed up to total as the RSV. A sample analysis of the RSV is performed and visualized in figure 5.1 on the results of figure 4.5 where the additional grey areas of the figure represent the locations of supporting structure for any surface under the common self-supporting build angle of 40°.

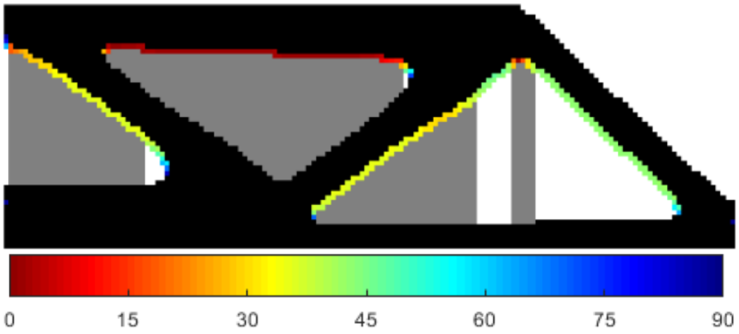


Figure 5.1: The grey-shaded areas represent location of required supporting structures at a prescribed minimum self-supporting angle of 40°

The minimum self-supporting build angle is AM-process and even machine specific therefore the RSV algorithm should be able to account for these variations. Based on the build angles of the overhanging surfaces, the support requirements can change according to the self-supporting build angle and this variation is pictured in figure 5.2. The same topology optimized structure of figure 4.5 is used and analyzed at minimum self-supporting build angles of 15° (top left), 30° (top right), 45° (bottom left), and 60° (bottom right). It is shown that as the minimum self supporting build angle increases, so to does the RSV.

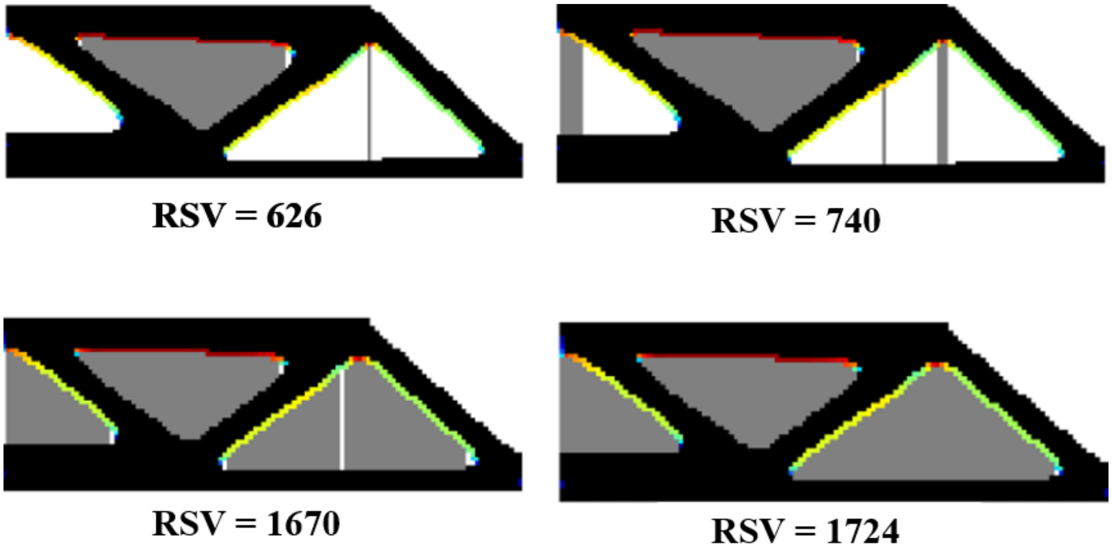


Figure 5.2: RSV calculations of minimum self-supporting build angles of 15° (top left), 30° (top right), 45° (bottom left), and 60° (bottom right).

The goal of this section is to minimize the amount of Required Support Volume (RSV) by means of part orientation optimization. In other words, in which orientation should the work piece oriented within the additive manufacturing machine to minimize the total amount of required support structure for a prescribed minimum self-supporting build angle. As an example, this analysis was carried out on the structure of figure 4.5 at 1° increments of rotation about the z-axis. A plot of this RSV vs. build orientation is shown in figure 5.3.

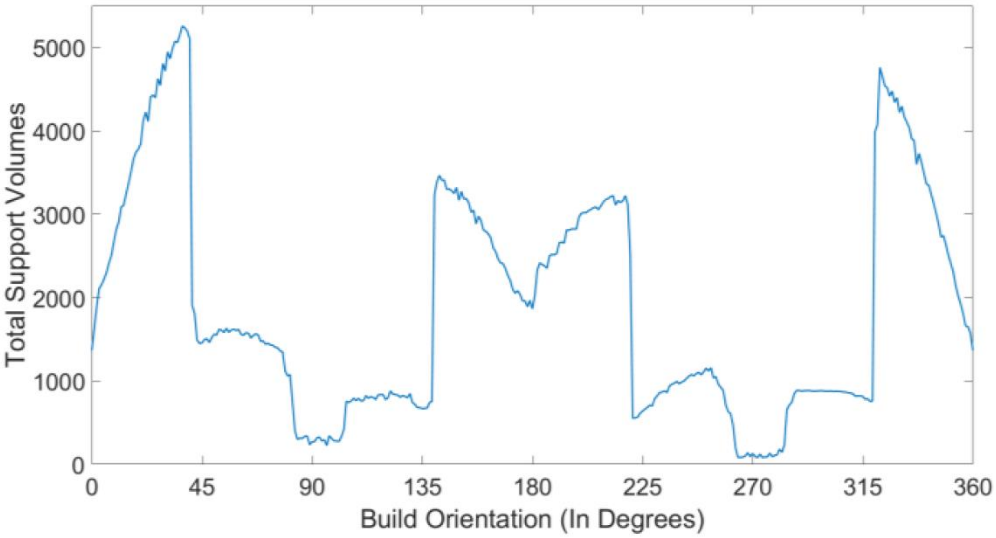


Figure 5.3: Analysis of the build orientation vs the RSV at a minimum self-supporting angle of 40°.

The brute-force style of optimization analysis of figure 5.3 indicates that the orientation of 38° produced the largest RSV value and the build orientation of 273° resulted in the minimized RSV value. A visualization of the support requirements at these orientations is shown in figure 5.4. A brute-force minimization style is not ideal for practical application but is utilized in this paper to demonstrate the landscape of the optimization problem. In practical implementation, an efficient optimization algorithm should be implemented that can deal with the many local optimum and the sharp slopes that are present in these generated landscapes. This sort of brute-force generated build orientation optimization landscape will be utilized for the remainder of the example problems to get an idea of the problems landscape complexity.

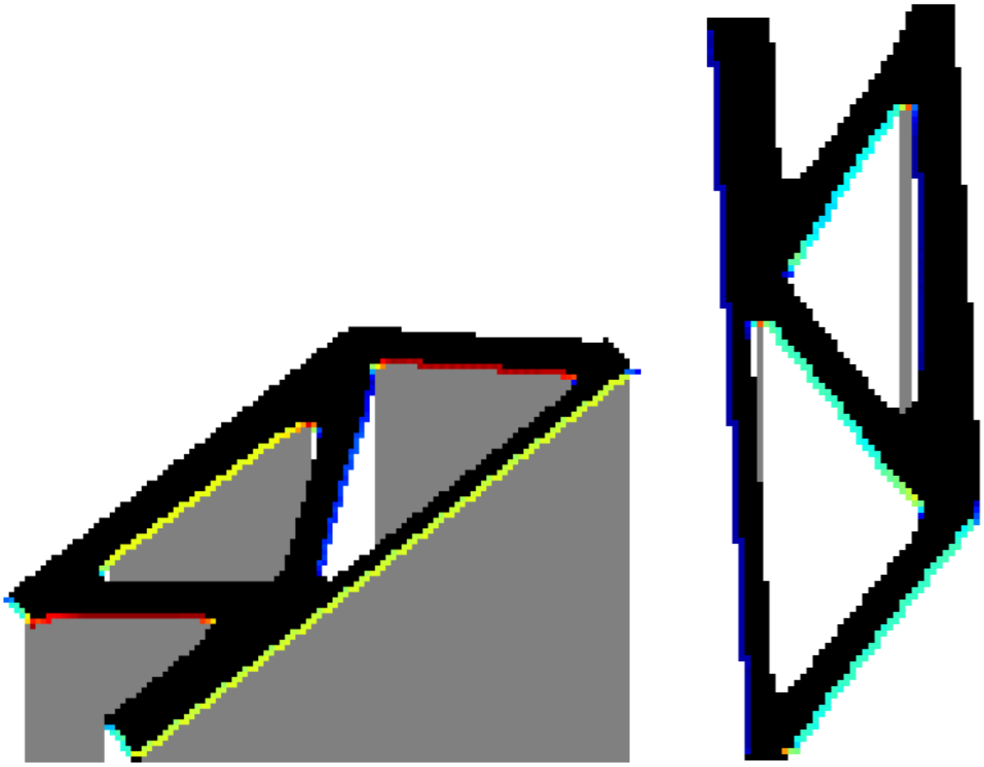


Figure 5.4: Build orientation that maximizes (left) and minimizes (right) the required support volume at a minimum self-supporting build angle of 40°.

It is clear from figures 5.2 that the build orientation plays a significant role in the amount of RSV to successfully print the topology optimized structure and figure 5.2 demonstrates that even small changes in the build orientation can lead to drastic changes in the RSV value. This signifies the utility of optimizing the build orientation based on support structure requirements but figure 5.3 also shows that it may come at the cost of increase manufacturing costs of other forms. This additional cost is present because layer-based manufacturing times are primarily concerned with the number of layers that need to be printed (i.e. the height of the part) and in figure 5.3, even though the RSV is minimized, the structure stands as tall as possible. This additional manufacturing cost will be accounted for in the next section in a novel cost formulation of any given AM process but before that the RSV based support slimming orientation optimization will be demonstrated in 3D.

5.3 Examples of RSV-Based Build Orientation Optimization

The applicability of the minimum required support volume analysis is limited if it is not demonstrated to be viable in 3D. For this purpose, table 5.1 and table 5.2 demonstrate the results of this analysis on three different 3D topology optimized structures. The first problem is a benchmark problem often referred to as a "Mitchell structure" or "3D Wheel". This structure is particularly chosen because of its two planes of symmetry. Its structure was topologically optimized using the code provided in the appendix of [35] with a total of 171 500 discrete elements comprising its finite mesh. The second structure is the standard cantilever benchmark problem consisting of 211 050 finite elements and one plane of symmetry. The last structure analyzed is a non-benchmark problem consisting of 171 500 finite elements with no spatial symmetry. The second and third structures were topologically optimized using the built-in feature of ANSYS™ 18.2. The density matrix of the final result was exported from ANSYS post processing and imputed into the implemented program in MATLAB™ for custom overhanging feature analysis and support slimming analysis.

Table 5.1 column 1 shows the setup of three different topology optimization problems with their problem definitions and resulting optimal structures visualized in column 2. Colum 3 of table 5.1 shows the RSV based build orientation optimization landscape where the structure is first rotated about the x-axis then about the y-axis by the amount specified in the graph. It should be noted here that if there is a geometric plane of symmetry in the design of the structure, there will be a corresponding plane of symmetry in the optimization landscape as well. Accounting for this during implementation could save significant computational time by only searching the non-redundant portions of the optimization domain.

Column 1 of table 5.2 demonstrates the required supporting structure locations of the three different work pieces in their minimum RSV orientation. The grey lines indicated supports that would start at a surface and then extend down to the build platform and the green lines indicate support structures that would start at an overhanging surface and extend down to another surface. These double surface supports should be more heavily penalized because

Table 5.1: Three structural design problems, results of topology optimization, and corresponding landscapes of RSV-based build orientation optimization

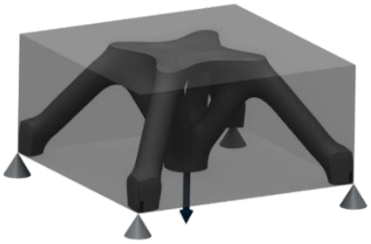
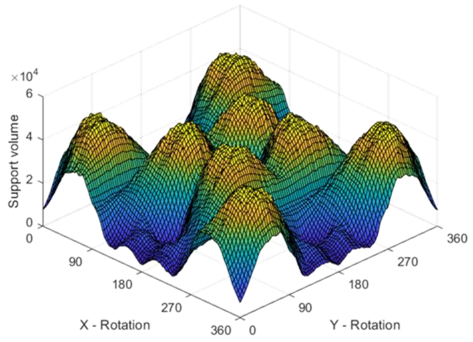
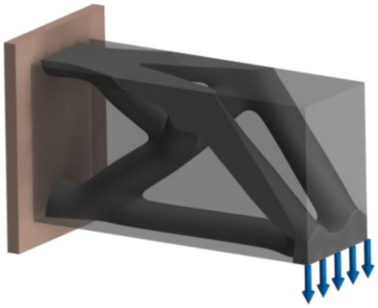
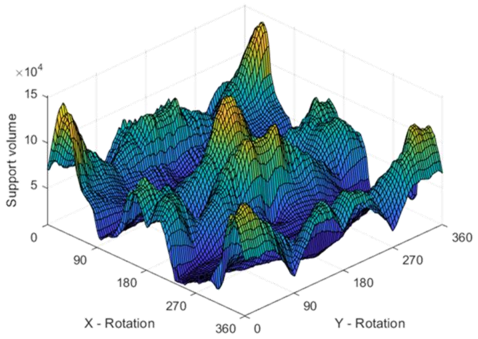
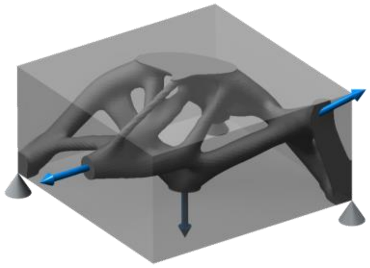
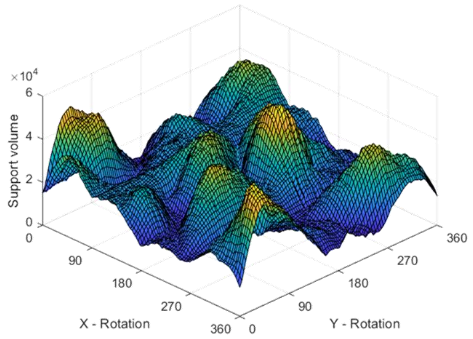
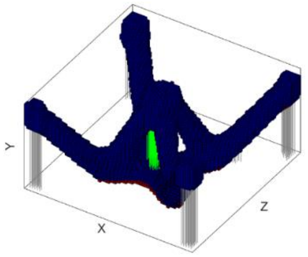
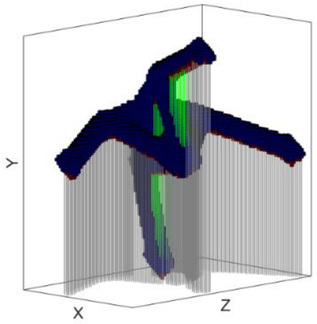
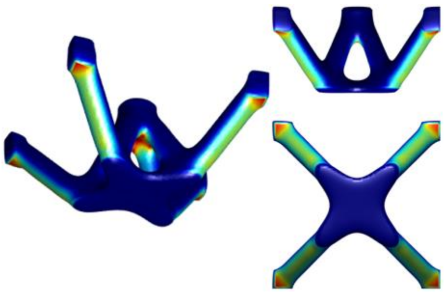
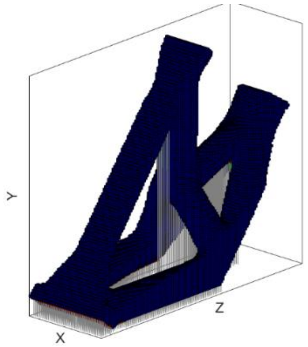
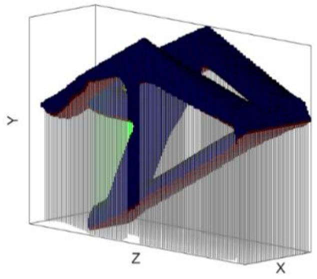
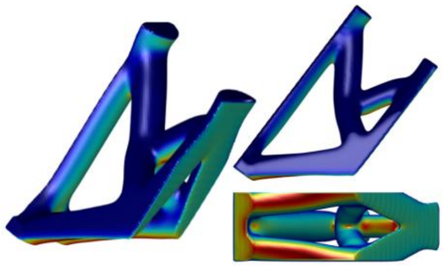
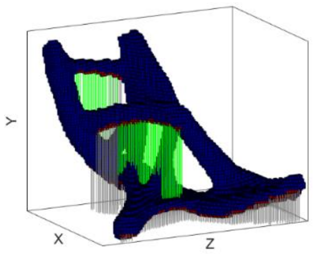
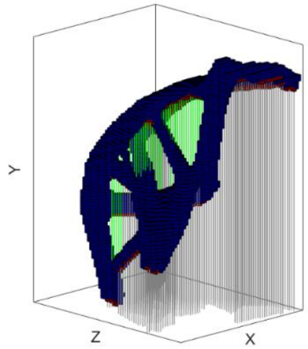
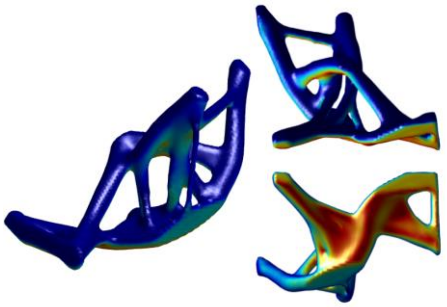
	Structural design problem and resulting topology	**RSV-based build orientation optimization landscape**
Top. Opt. Software: [35] Nelx: 70 Nely: 35 Nelz: 70 Two planes of symmetry		
Top. Opt. Software: ANSYS 18.2 Nelx: 45 Nely: 67 Nelz: 70 One plane of symmetry		
Top. Opt. Software: ANSYS 18.2 Nelx: 70 Nely: 35 Nelz: 70 No symmetry		

Table 5.2: Part orientations that minimizes (column 1), and maximizes (column 2) RSV with surface angle visuals of the minimum orientation (column 3).

Minimum support orientation	**Maximum support orientation**	**Surface overhang angle w.r.t the build platform visualization @ min support orientation**
		
		
		

they are detrimental to the surface quality of twice the surface area as that of the grey supports. Column 2 of table 5.2 demonstrates the built orientation that maximizes the RSV for visual comparison with the minimum RSV build orientation. Colum 3 of this table shows a smoothed visual overlay of the build angles over the structure in its minimal RSV orientation. The colour scheme of this colour overlay can be referred to the scale of figure 4.8.

For reference, the cantilevered beam was printed in its minimum and maximum RSV orientation to validate the reduction of supporting structure. The structure was converted to an STL format and loaded into the PreForm software associated with the Formlabs 2 printer where the supporting structure was automatically generated as shown in the left half of the Figure 5.5. The structures were then printed (as seen in the right of figure 5.5) and weighed to analyze the difference in supporting structure weight. In the minimum RSV orientation, the structure + support structure weighed 19 g and in the max RSV orientation, the combined structures weighed 24 g.

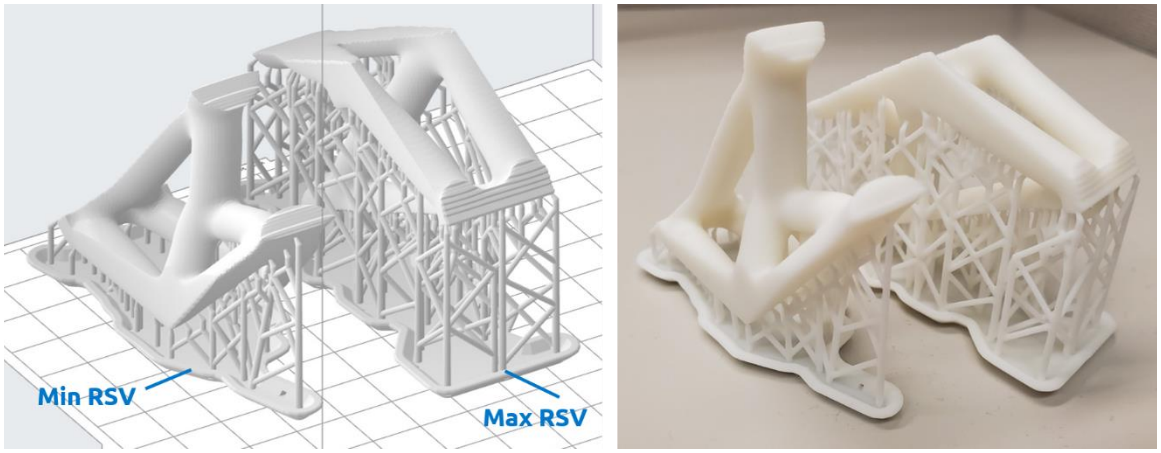


Figure 5.5: Topology optimization in its minimum and maximum RSV orientation. The left figure shows the structure and its supports in the “PreForm” software and the resulting structure after printing on the Formlabs 2 printer in the right half.

5.4 Part Orientation Optimization with General AM Considerations

Although it is shown in the previous section how the build orientation can be selected based on RSV, it is useful to have a more general discussion here about the optimum build orientation. In particular, each additive manufacturing process has its own specifications and requirements. In some of them, minimizing RSV is the main objective while in the others, minimizing the total printing area, minimizing the total number of layers, or minimizing the total surface area needing post-processing and finishing operations, or a combination of these can be the main objective. For this reason, this section proposes a more general approach to find the optimum build orientation for a given additive manufacturing process based on the density gradient based build angle analysis and the associated support structure requirement study. The total manufacturing cost can be represented by the following formula:

$$C_T = \alpha_m * RSV + \alpha_n * N + \alpha_a * PA + \alpha_f * FA \qquad (5.1)$$

Where C_T is the total cost for fabrication, RSV is the total required support volume, N is total number of required layers, PA is the summation of total printing area for all of the layers and FA is the total area requiring finishing due to artifacts of the support structures. Parameters, α_m, α_n, α_a , α_f are associated costs indices for support material, layer changing setup, area of printing, and finishing operations, respectively. It should be mentioned that equation 5.1 appears linear in nature but as can been seen in figure 5.6, the main parameters are highly nonlinear which renders the equation itself nonlinear.

As a case study, we can look at the cantilevered beam as an example provided in table 5.1. Figure 5.6 a), b), c), d) respectively presents the build orientation optimization landscape for each one of the individual objective functions including RSV, number of layers at a constant 0.1mm layer thickness, total printing area, and the part surface area affected by supporting structure.

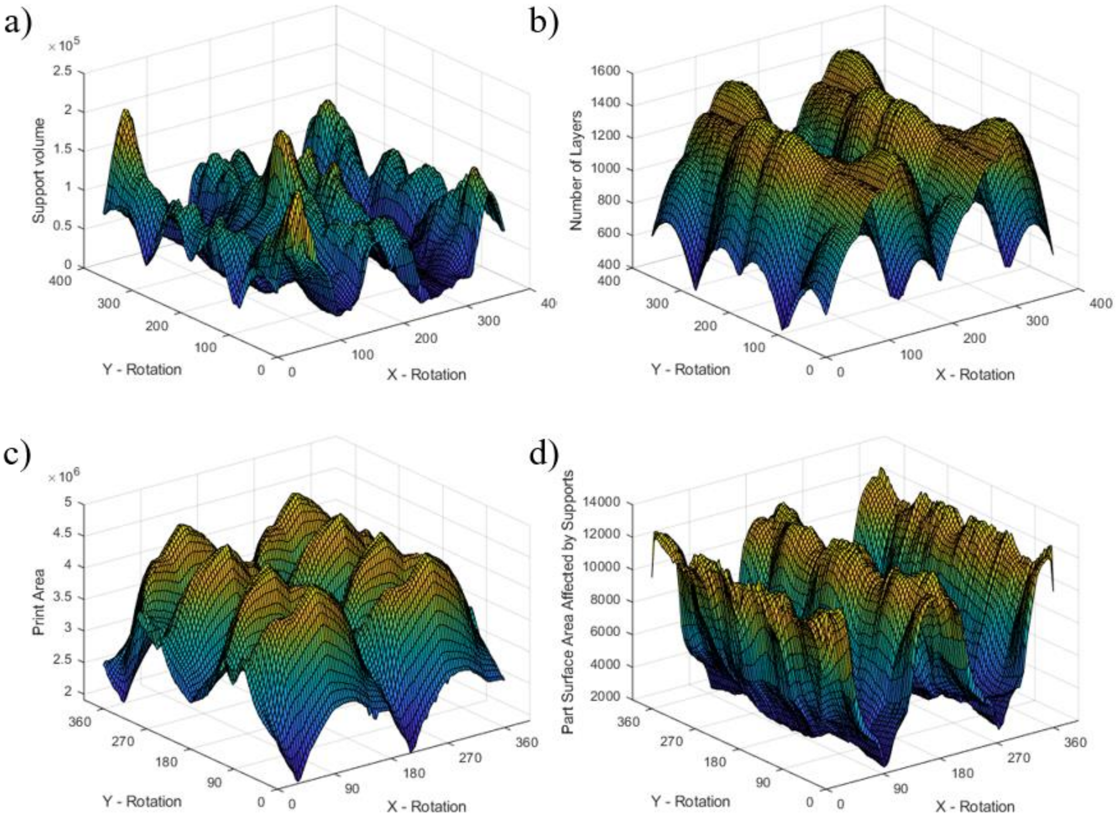


Figure 5.6: Build orientation optimization landscape for various objective functions; using only a) RSV, b) number of layers, c) printing area, and d) finishing area, as the only objective function.

For example, for $\alpha_m = 0.003\ \frac{cents}{mm^3}$, $\alpha_n = 0.4\ \frac{cents}{N}$, $\alpha_a = 0.0001\ \frac{cents}{mm^2}$, and $\alpha_f = 0.05\ \frac{cents}{mm^2}$, the multi-objective build orientation optimization landscape is pictured in figure 5.7. It is found that the minimum cost to manufacture the cantilevered beam on an FDM based machine is $10.03 and the maximum cost to fabricate the part would be $18.50. This analysis theoretically leads to a potential cost savings of 84% to manufacture the part based on its orientation in the additive manufacturing machine. It be seen that the additional parameters of optimization add significant complexity to the optimization landscape opposed to the original RSV based optimization which indicates an effective global optimization technique should be implemented. The optimum for this example was found by using a direct search method of the orientation optimization landscape.

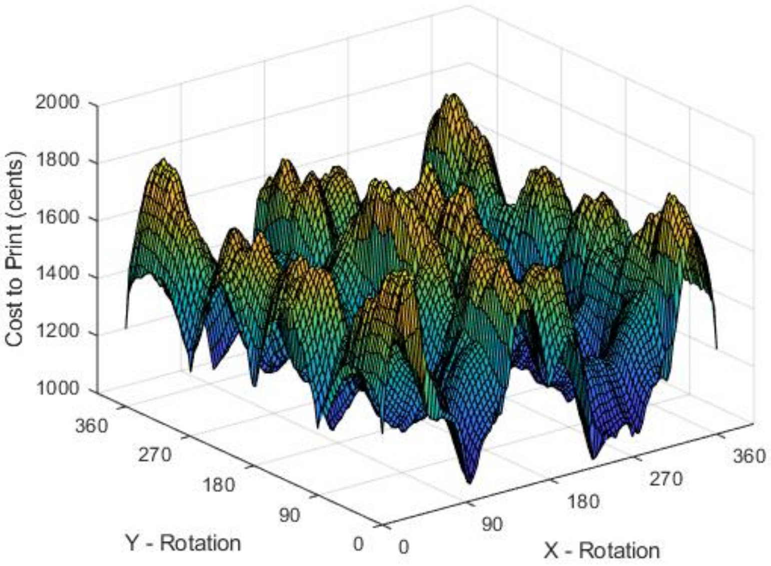


Figure 5.7: Multi-dimensional build orientation optimization landscape for a specific additive manufacturing process with given cost indices.

5.5 Conclusion

It is shown in various examples that the density gradient based build angle analysis of the overhanging surfaces presented in chapter 4 serves as a useful tool to determine support structure requirements. This analysis is first presented to perform support slimming by minimized the Required Support Volume (RSV) my means of re-orienting the work piece within an additive manufacturing machine. The analysis is further extended to a more general case considering additional AM considerations. The build orientation optimization landscapes of these minimum RSV-based analysis is shown for consideration when choosing an optimization algorithm. This analysis plus the FEA based slicing presented in the next chapter combine to eliminate the need for converting the FEA mesh representation of the structure to that of an STL definition.

Chapter 6. Slicing of Finite Element Structural Representations

6.1 Introduction

Slicing is the process which produces a set of 2D contours that form a layer-based representation of a 3D model. These layers are typically of uniform thickness but recent advancements allow users to adaptively slice objects allowing for varying layer thickness to minimize manufacturing errors and time [89]. In topology optimization applications, the slicing is almost exclusively performed on a triangular facet surface representation of the 3D object (STL) because this is readily available commercially software to do so. Although slicing the STL structural representation is a well-developed and –established method, it has the drawback of requiring that the original finite element mesh representation of the structure be converted to the STL format. This intermediate step of re-representing the structure introduces deviations due to simplifications of the true geometric shape as well as adding unnecessary additional computational time to the entire process. Eliminating the STL conversion also opens up the possibility of direct communication between the topology optimization algorithms and the manufacturing pre-processing stage which has been successfully implemented by a colleague to reduce based AM costs [129]. It is also worth stating that this sort of simultaneous iterative communication between the design and manufacturing stages of the process promotes flexible and agile deigns [130] for future more advanced applications of topology optimization and additive manufacturing processes. This chapter outlines the methodology necessary to perform finite element-based slicing for the generation of additive manufacturing machine instructions.

6.2 Methodology

This section proposes a five-step procedure for finite element-based slicing. Two separate examples will be investigated at each stage of the methodology. This first example is a test piece designed specifically to test known failure modes of slicers such as slicing at the intersection of horizontal and vertical walls of convex corners and ray-tracing capabilities. The second example is the result of a topology optimized hook to demonstrate the slicer's applicability to topology optimization problems. The CAD model of the first example and the design domain / problem definition of the second example (topology optimized hook) are shown in figure 6.1.

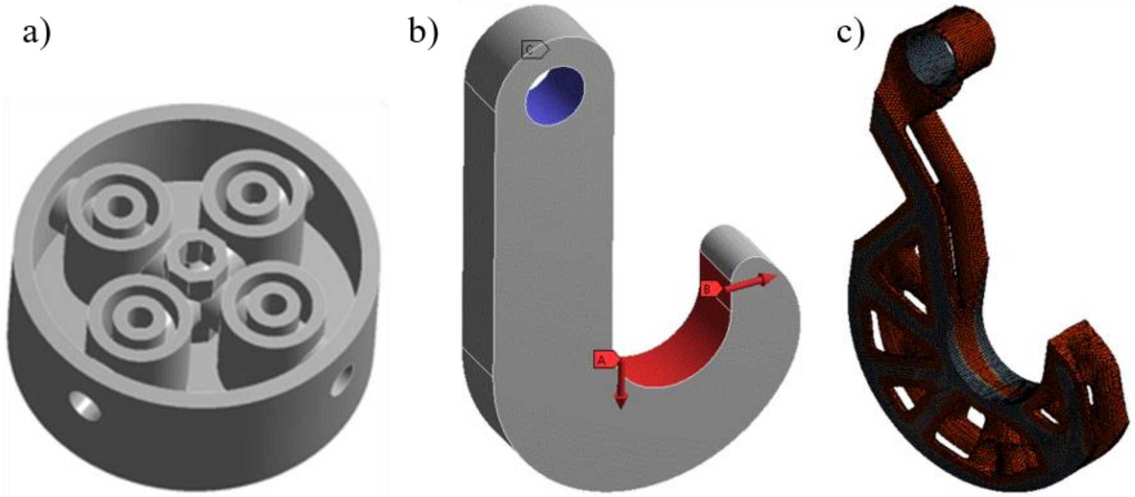


Figure 6.1: a) CAD model of the first test piece, b) design domain of hook design problem and c) optimal hook topology which will serve as test piece number 2

The five outlined steps of this methodology are i) Transferring the finite element mesh from the topology optimization post-processor to the additive manufacturing preprocessor, ii) Element filtration to remove any unnecessary computations iii) Finding the intersection of the elements with the slice plane iv) Forming the contours of the slice's cross-section, and v) Producing a set of AM-process and machine specific instructions to produce the structure.

Step i) Transfer of structure represented by solid finite-elements to the AM preprocessor

The design domain of a topology optimized structures is meshed into a group of solid finite elements for the sake of numerical structural analysis. The design of the structure is represented by whether or not a finite element from the design domain should be included in the optimal topology. Often times hundreds of thousands, to millions of finite elements that are used to describe the topology optimization's design domain in order to achieve a well-defined surface as well as an accurate approximation of the true structural performance. It is common for both academia and industry to implement commercial finite element meshers and solvers because they are readily available in these environments and simple to interface with.

The user typically has control over element sizes, types and mesh refinement strategies in order to adequately evaluate their structure. Finite elements are not necessarily constrained to be any particular size, shape or even complexity but these commercial FEA softwares commonly decompose 3D models into a subset of only four different finite elements types to represent structural problems. The four fundamental solid finite element types are known as a cuboid, a tetrahedral, a pyramid and a prism. These four element types are visualized in table 6.1 along with some relevant geometric properties. Note that the colour scheme used for the elements in this table will remain consistent with any figures which utilize visual finite elements.

The information pertaining to the locations and geometry of the finite element mesh is stored and transferred from the FEA post processor to the AM pre-processor in a series of two matrices. Table 6.2 and Table 6.3 are visual representations of the two matrices with the first table storing information related to the nodal locations of each node and the second table contains information of the elements which consist of at most 8 node indices in a specific order used to represent the individual elements. These matrices are stored in a text file by the topology optimization process and then read in by the AM pre-processor for slicing.

Table 6.1: The four fundamental finite element types with some relevant geometric properties

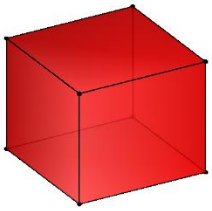
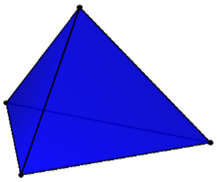
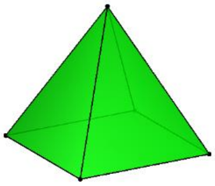
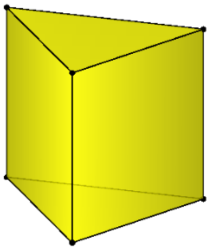
Element Name	Visual	# of Faces	# of Nodes	# of Edges
Cuboid		6	8	12
Tetrahedral		4	4	6
Pyramid		5	5	8
Wedge		5	6	9

Table 6.2: Matrix of nodal information of the finite element mesh

Nodes			
Node #	**Cartesian Coordinates**		
1:	X_1	Y_1	Z_1
2:	X_2	Y_2	Z_2
⋮	⋮	⋮	⋮

Table 6.3: Matrix of the element information of the finite element mesh.

Elements									
Element #	**Element Type**	**Node Number**							
1:	Hex	$N_{1,1}$	$N_{1,2}$	$N_{1,3}$	$N_{1,4}$	$N_{1,5}$	$N_{1,6}$	$N_{1,7}$	$N_{1,8}$
2:	Wed	$N_{2,1}$	$N_{2,2}$	$N_{2,3}$	$N_{2,4}$	$N_{2,5}$	$N_{2,6}$	-	-
3:	Pyr	$N_{3,1}$	$N_{3,2}$	$N_{3,3}$	$N_{3,4}$	$N_{3,5}$	-	-	-
4:	Tet	$N_{4,1}$	$N_{4,2}$	$N_{4,3}$	$N_{4,4}$	-	-	-	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Figure 6.2 presents the finite element decomposition of both the slicing test work piece and a low-resolution decomposition of the hook's design domain as well as the resulting optimal topology which also represented in finite elements. Example 1 (slicing test piece) consists of 5222 solid elements and 14 010 nodes which are divided by 2 323 pyramid-, 2 019 tetrahedral-, 651 cuboid-, and 229 wedge-type elements. Example 2 (topology optimized hook) is comprised of 24 651 elements, 102 445 nodes which are divided by 21 094 cuboid-, 2 298 pyramid-, 979 tetrahedral and 290 edge-type finite elements. These meshes of figure 6.2 a) and c) will be used as examples in the upcoming steps.

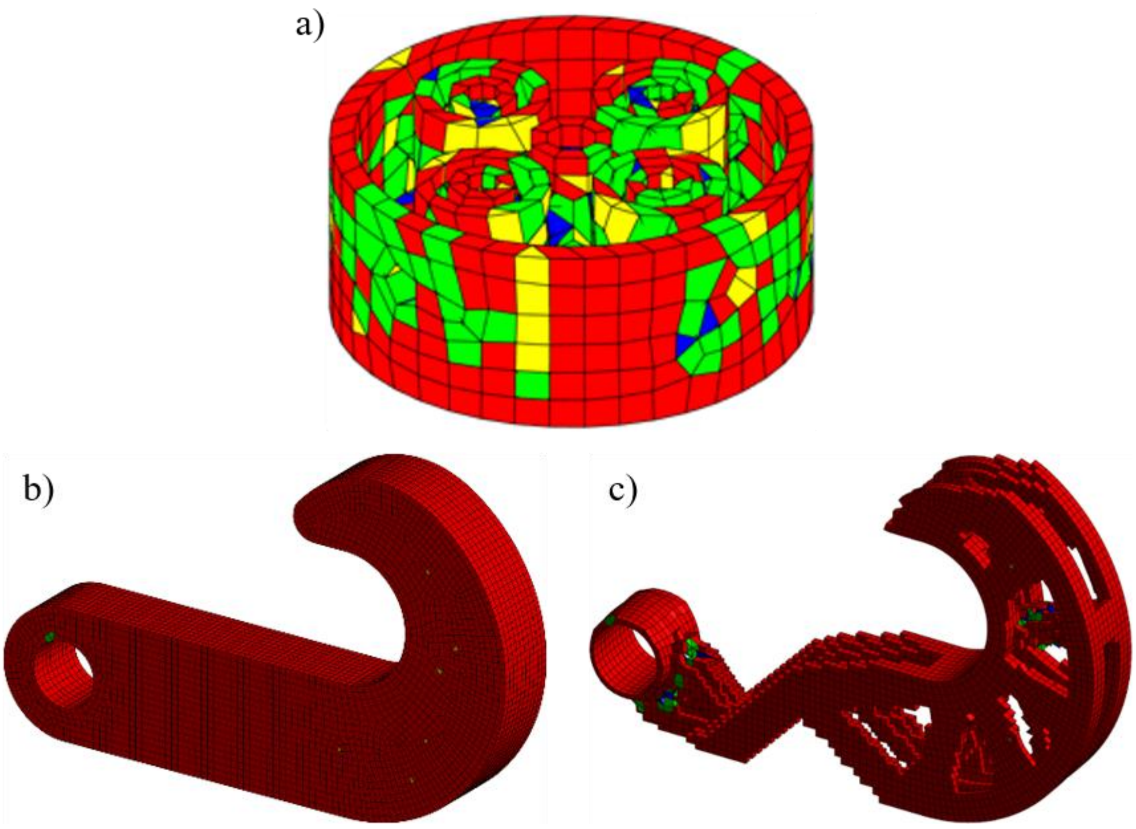


Figure 6.2: Finite element decomposition of a) the slicing test piece (test example 1), b) the hook's design domain, and c) the hook's optimal topology (test example 2)

Step ii) Element filtration

The second step is to determine which of the elements lie on the slice plane for a process called element filtration. The slice plane is a virtual plane that runs parallel (perpendicular to the build direction) and at an offset to the build platform which defines the bottom of a 2 ½D slice. The intersection of the finite element mesh and this slice plane will form the cross section of the specific layer ultimately sued to form the contours and then the machine instructions. It important to simply determine which of the elements contain the slice plane to prevent unnecessary intersection checks on all of the elements.

The element filtration is performed by first determining the minimum and maximum height of the nodes relative to the build platform. If the height of the slice plane lies between the min and max nodes of an element, then that particular element will not be filtered out and therefore be passed on to the next step of slicing. Figure 6.3 shows a visualization of the filtration process on the two test pieces where the pink plane is the slice plane, the grey elements are filtered out and the red elements are passed on to step 3 of the slicing process.

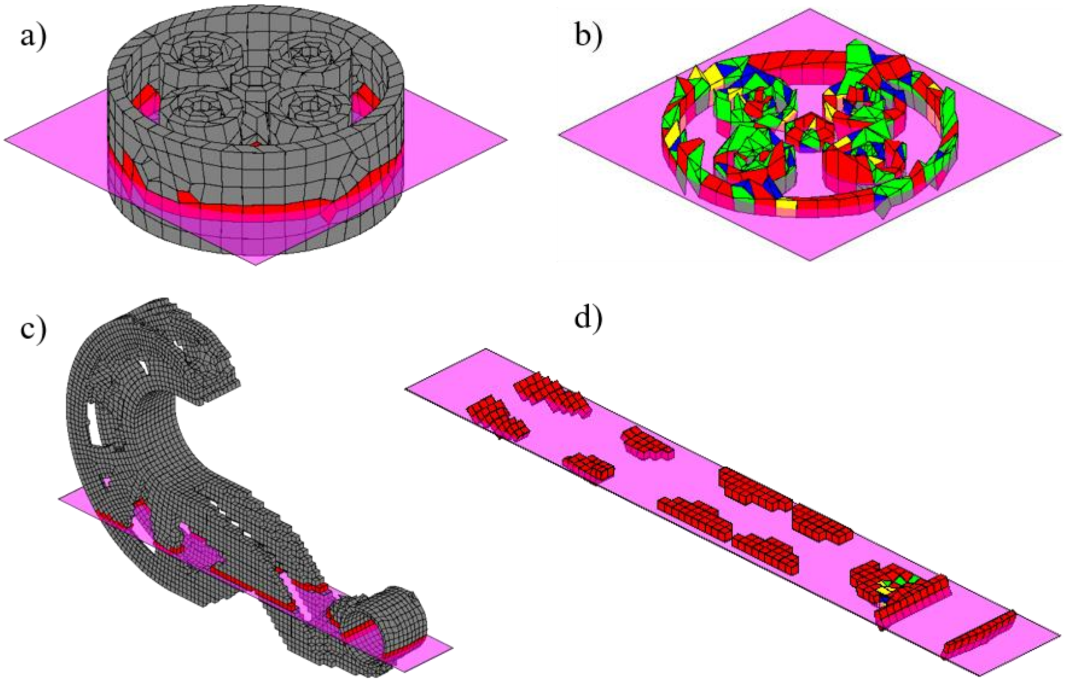


Figure 6.3: a) and c) show the intersection of the slice plane with the finite element mesh of examples 1 and 2 respectively. b) and d) are elements remaining post filtration.

From the element filtration examples presented in figure 6.3, the effectiveness of this filtration can be analyzed. For example 1, only 684 of the original 5 222 elements remain resulting in an 87% decrease in the number of elements that need to be analyzed in the next step. In the case of example 2, only 958 of the original 24 651 elements remain which results in a 96% reduction of elements. This shows that the element filtration is a worth while step in the process and that the greater the resolution of the mesh, so to is the benefits of the filtration step.

Step iii) Intersection of filtered elements with the slice plane

This task of the process determines precisely the location of where the edges of the elements intersect with the slice plane. The points of intersection of the element edges with the slice plane is used because of the simple process to find this point when the edge is represented as a parametric line segment and the slice plane is represented as a parametric plane. Particularly, for every facet of a finite element that intersects the slice plane, two of its edges will intersect at a point with the plane. These points are connecting to form an edge of the slice. The two intersecting points are stored in a list called contour nodes and the edges formed between 2 points for each facet are stored in a list called the contour edges. Table 6.4 depicts how the contour node information is stored and table 6.5 depicts how the contour edges are stored and used for communication in the remaining steps.

Table 6.4: Matrix representing how the contour node information is stored.

Contour Nodes		
Node #	**Cartesian Coordinates**	
1:	X_1	Y_1
2:	X_2	Y_2
⋮	⋮	⋮

Table 6.5: Matrix representing how the contour edge information is stored.

Contour Edges		
Edge #	**Contour Nodes**	
1:	$CN_{1,1}$	$CN_{1,2}$
2:	$CN_{2,1}$	$CN_{2,2}$
⋮	⋮	⋮

As mentioned, the contour nodes will be calculated as the intersection of the element edges and the slice plane. The contour edges are formed between the nodes (x_0, y_0, z_0) and (x_1, y_1, z_1) and the parametric term t is used. The parametric representation of the edge represented as a line segment is mathematically expressed with the three following equations as:

$$x = x_0 + t(x_1 - x_0) \tag{6.1}$$

$$y = y_0 + t(y_1 - y_0) \tag{6.2}$$

$$z = z_0 + t(z_1 - z_0) \tag{6.3}$$

The parametric representation of the slice plane is the following:

$$Ax + By + Cz + D = 0 \tag{6.4}$$

Where the slice plane is defined by a normal vector ($N = [A, B, C]$) formed by three arbitrary points (V_0, V_1, and V_2) that lie at the correct slice height. This normal vector is calculated as the cross product of the 2 vectors formed by the three points as shown below in equation (6.5).

$$N = [A, B, C] = (V_1 - V_0)\ X\ (V_2 - V_0) \tag{6.5}$$

The D term of parametric plane equation is the offset of the plane. It is simply calculated as the negative dot product of the normal vector and the point that is common between the two lines used to form the normal vector of the plane. This term is calculated as follows:

$$D = -(N \cdot V_0) \tag{6.6}$$

Performing the intersection of the element edges and the slice plane will result in contour node intersection points which are used to form the contour edges list. This list contains a number line segments that form the cross-section of the layer. Figure 6.4 shows the sample contour edges (black line segments) of the examples explored in this chapter and the figure is also colourized to demonstrate which elements the line segments were derived from.

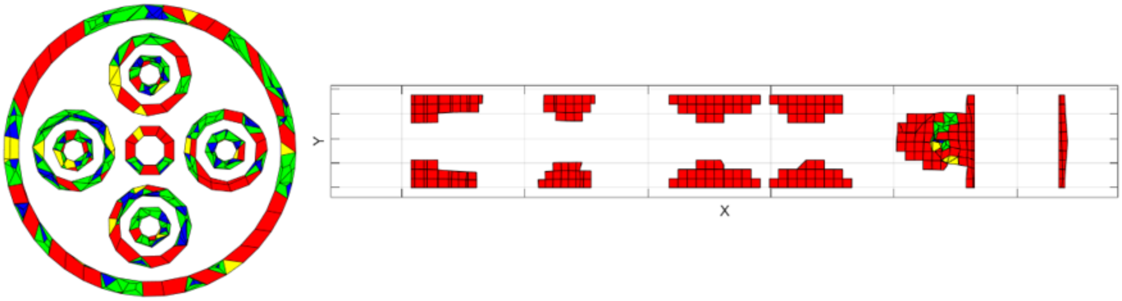


Figure 6.4: The contour edges (black line segments) that are formed when the elements are intersected with the slice plane.

Step iv) Forming the contours of the slice

The contours of the slice refer to a set of polygons formed by non redundant contour edges generated in the previous step. The individual polygons represent regions of either solid material or they signify a void of material. The contours are ultimately used to generate the machine instructions for the AM machine whether its used in a ray tracing algorithm to generate a black and white image for a DLP style SLA printer or whether the polygons are bridged at intervals to form a set of tool paths to be used on laser-/print head-based printers. The process of converting the contour edges identified in step three to the outlined contour polygons is explained in this step.

Step four of the slicing process is a three-stage process in itself. First the redundant edges of the identified contour edges are removed from the list, second the edges are formed into a set of polygons, and lastly the polygons are identified as either solid or void by means of ray tracing.

The first stage of this step is to remove edges that are doubled up in the contour edges list. These doubled up edges are referred to as redundant because they do not contribute significant information relative to the slice contours. The redundant edges are a result of two elements sharing a face in the finite element mesh which leads to two overlapping contour edges being formed in the intersection step (step 3). Removing the redundant edges from the contour edges list will leave only perimeter edges which are of sole concern when finding the contour polygons. The stage of removing these redundant edges is visualized in figure 6.5.

It was identified in the previous step that the finite element mesh of example 1 produced 2 448 contour edges when intersected with the slice plane but after removal of the redundant edges, only 316 edges remained. For example 2, There were 1153 contour edges before removal of the redundant edges and only 267 remained after this stage of step 4. This resulted in 88% less contour edges in the case of example 1 and 77% less edges for the case of example 2. This edge removal is visualized in figure 6.5.

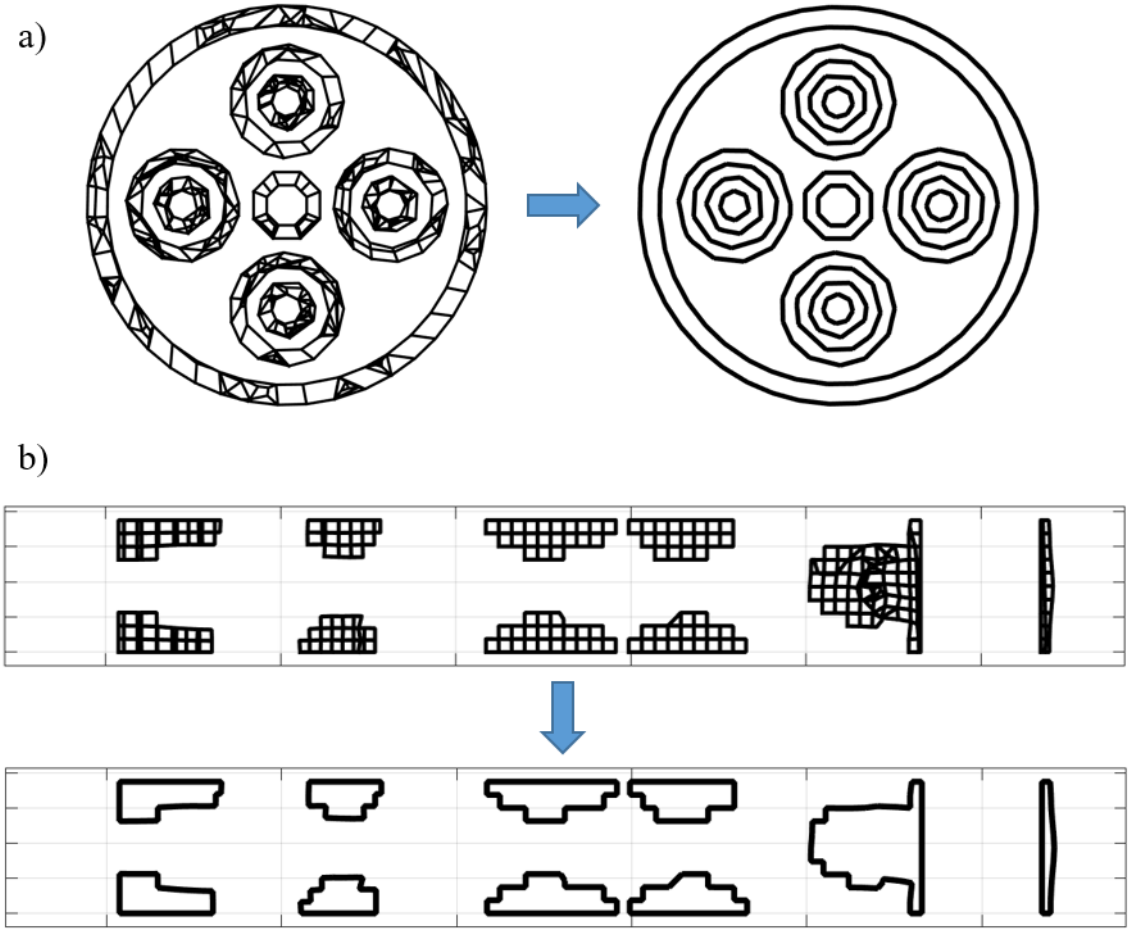


Figure 6.5: Results of removing the redundant contour edges for a) example 1 and b) example 2

Stage two out of three for this step is to form the remaining contour edges into a set of polygons. A polygon is simply a closed-set of edges where the start of the head of the first edge is coincident with tail of the second edge. This sequence continues until the head of a sequential line segment is coincident with the tail of the first edge belonging to the set. When this condition is met, a polygon is formed. It should be seen in the results of figure 6.5 that for example 1, there exists 20 individual polygons and for example 2, there exists 10 individual polygons.

A scenario can occur when forming the set of polygons in this stage where the polygons do not guarantee to form into proper polygons when the methodology just described is used. This situation arises when two separate polygons share a mutual point such as the scenario of topology optimization checker boarding which cases this stage to breakdown. To overcome this, when forming the contour edges into ordered lists of edges aligned sequentially head-to-tail, there needs to be an additional check that needs to be met. In order to add a line segment in sequence to the previous one when forming the polygon, the sequential edges must have matching direction vectors when the result of the cross product between the edge in question (point in the direction of tail to head) and the edge's normal vector is determined. This ensures that the edge in question is properly determined as part of a particular polygon or not. Note that the edge's normal vector is a vector perpendicular to the edge that points away from the centroid of the element it was derived from. This scenario is not present in the 2 outlined examples but is worth noting for implementation.

Stage three of this step is to determine each of the polygons is a void or whether it a solid contour. The difference between the two is that a solid contour should be filled with material and a void contour should not be filled with material. In the case of layer-based or print head-based AM machines, the tool paths generated by void contours are extended outwards until they meet a solid contour and the solid contours are extended inwards until they meet a void contour for when peripheral based path planning is implemented. To determine whether a polygon is solid or void, a simple ray-tracing algorithm is implemented. A visualization of the ray tracing algorithm can be seen below in figure 6.6.

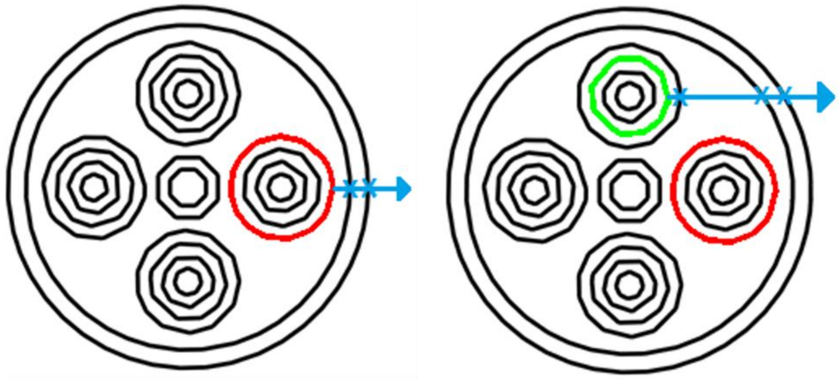


Figure 6.6: A sample ray-tracing visualization for a solid polygon with an even number of intersection (left) and a void polygon with odd number of intersections (right).

The ray tracing algorithm functions by first taking the right most point of a polygon and tracing a ray from this point onwards to positive infinity. A ray is represented by equations (6.1 – 6.3) where $t \geq 0$ and used to determine the intersection of all other edges in the contour edges list by forming the edges as line segment defined by the same equations where $0 \leq t \leq 1$. The number of intersections is summed up and if it is a positive or a zero sum, then the polygon is identified as a solid polygon. If the number of intersections between the ray and the list of edges is odd, then the polygon is a void polygon. The results of this stage are shown on the outlined examples of 1 and 2 in figure 6.7. In this figure, solid polygons are drawn in red and void polygons are drawn in green. In example 1, 10 of the polygons are solid and 10 of the polygons are void. In example 2, all of the polygons are considered to be solid.

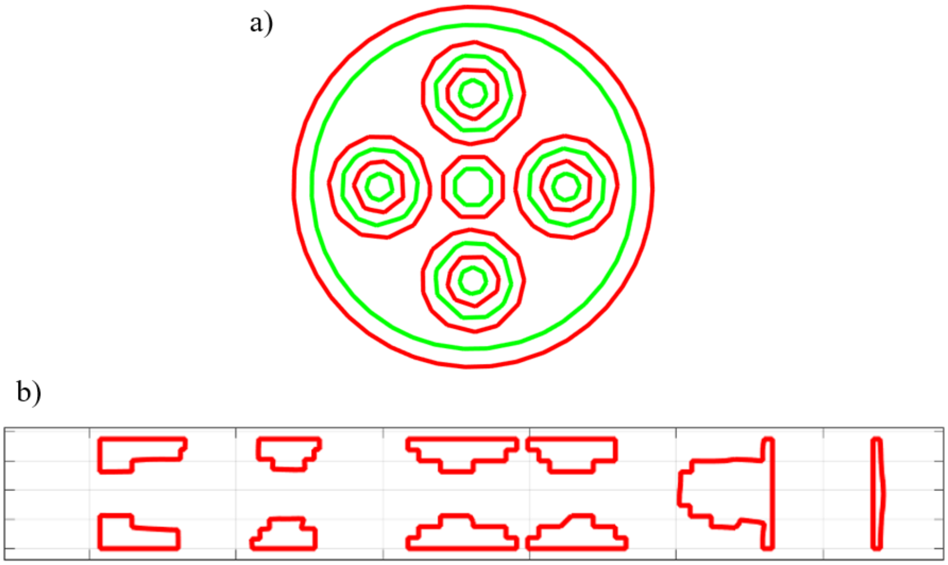


Figure 6.7: Contour polygons identified as either solid (red) of void (green) for a) example 1 and b) example 2

Step v) Producing AM machine instructions from the contour polygons

Step five is highly dependent on the implement AM process as well as the specific machine therefore it is not explained in depth. The contour polygons formed in the previous step are all that is necessary for forming either the path planning in the case of print head- / laser-based machines or for black and white picture generation such as the DLP style SLA process. This second option is explored for demonstrating the machine instruction generation but is should be noted that the outlined methodology is not limited to just this machine or even the AM specific process.

Figure 6.8 a) is a demonstration of the black and white image that is generated as machine instructions for a sample slice of example 1 in the case of a DLP style SLA printer. Figure 6.8 b) is also a sample image that would be generated necessary to print a layer of the outlined example 2. These images are white in regions that the projector should be shining bright in order to solidify the liquid resin. The regions that are black are associated with the absence of light which will be areas of the layer that does not solidify the liquid resin.

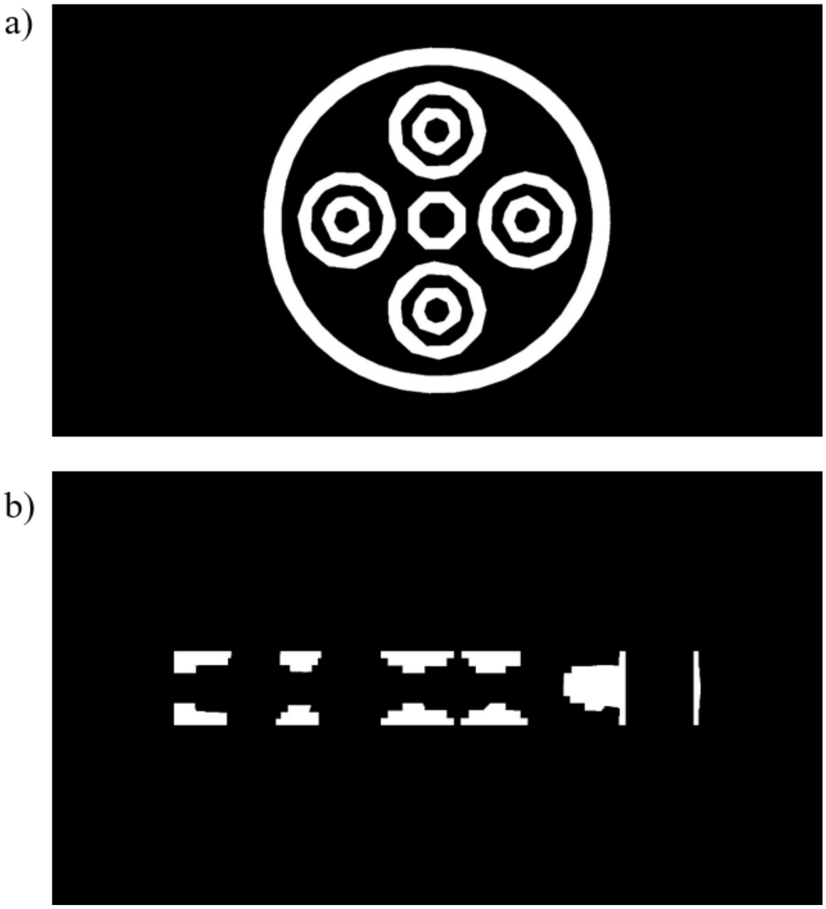
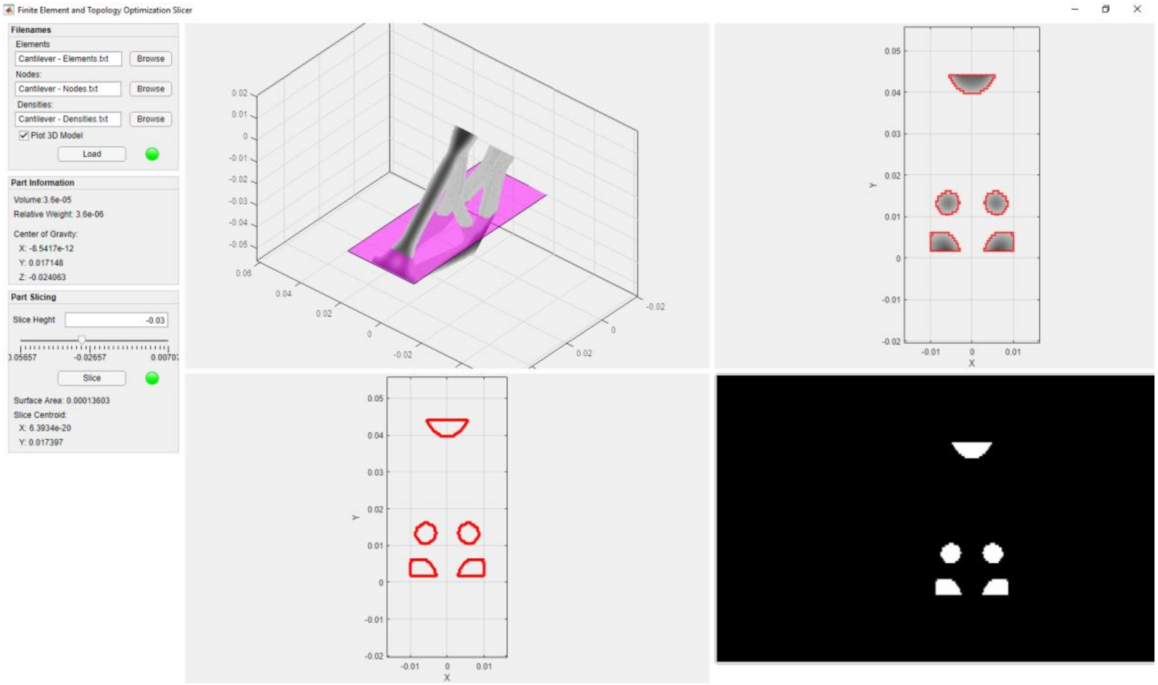


Figure 6.8: Sample machine instructions that would be sent to produce a layer on a DLP style SLA printer for a) example 1 and b) example 2.


Finite Element and Topology Optimization Slicer

Filenames

Elements: Cantilever - Elements.txt

Nodes: Cantilever - Nodes.txt

Densities: Cantilever - Densities.txt

☑ Plot 3D Model

Part Information

Volume:3.6e-05

Relative Weight: 3.6e-06

Center of Gravity:
X: -8.5417e-12
Y: 0.017148
Z: -0.024063

Part Slicing

Slice Heght: -0.03

Surface Area: 0.00013603

Slice Centroid:
X: 6.3934e-20
Y: 0.017397

Figure 6.9: Sample snap shot of the developed GUI slicer demonstrated on the cantilever beam of chapter 5 in its minimum RSV build orientation.

For the purposed of testing the slicing algorithm, a GUI based version of this slicing process was developed. This app loads in the node and elements defined by table 6.2 and table 6.3 and prompts the user for a slice height. When the object is sliced with this information, various steps of the slicing process are visualized including the final black and white image which would actually be used in an AM part. To demonstrate this program, the cantilever of chapter 5 is loaded and sliced in its build orientation that minimizes the RSV value. This sample slice can be seen in figure 6.9.

6.3 Conclusion

In conclusion, a methodology was proposed to slice structures represented as a mesh of finite elements such as the use case in topology optimization. The finite element slicer combined with the support structure requirement analysis of chapters 4 and 5 combine to eliminate the need for converting the structure to an STL file. The test finite element slicing software developed was also implemented in a novel fashion to constrain the topology optimization process to minimize the manufacturing time of a toolpath-based AM machine.

Chapter 7. Summarization, Future works, and Conclusion

This thesis was written with the goal of outlining and improving the existing state-of-the-art process of structural topology optimization design and employed additive manufacturing techniques. The developed improvements were investigated with the spirit of industry 4.0 and the hypothetical industry 5.0 in mind which were employed in various aspects of the design and manufacturing process. A five-step design process was explored that would start with the problem definition of a structural design problem on one side, and output an additively manufactured part on the other.

The proposed improvements to the process include not before realized control over various convergence characteristics of a popular topology optimization algorithm known as BESO (chapter 3). Second, an extension of a computer vision-based density gradient analysis of the topology optimization's design domain into a 3D form that is applicable to a finite element mesh consisting of any combination of element types (chapter 4). This analysis is used in the third proposed improvement to determine the build angles and support structure requirements of overhanging surfaces in additive manufacturing applications which is then further utilized to determine an optimal build orientation that minimizes manufacturing costs (chapter 5). A fourth improvement suggests that the computationally heavy, and geometric complexity reducing step of re-representing the finite element form of the topology optimized structure into the STL form can be eliminated. This is achieved by proposing a novel methodology for slicing a finite element mesh (chapter 6) in combination with the second and third proposed improvements which unite to eliminate the need for an STL representation altogether. Throughout the development of these improvements, some possible avenues of future work have been identified.

During the investigation of convergence control for topology optimization, a difficulty in tuning a PID controller is identified. A proposed solution of a multi-objective optimization algorithm is suggested to find the pareto-optimal set of solutions for minimizing the number of iterations to converge vs minimizing the stiffness compliance. I have begun investigation the NSGA-II multi-objective metaheuristic algorithm to investigate if there exists a robust convergence profile that is an optimal trade-off of computational time and structural efficiency across a wide range of initial conditions. This type of analysis could be extended to find a set of robust controller tuning gain values which give similar performance over a wide variety of initial boundary conditions and topology optimization settings.

A third suggested improvement may be to experimentally determine the set of cost indices to replace the sort of back of the envelope calculated ones that were examined. To achieve this, an experimental setup should be designed to gain insight into the true values of the four cost indices (α_m, α_n, α_a , α_f). This first index α_m is associated with the cost of material per unit volume, the second index α_n is associated with the cost per layer which can be determined by finding the associated machine costs per average layer manufacturing time. The third index α_a and the fourth index α_f are associated with the total area that needs to be printed and the area that needs to be post processed respectively. This work will be carried out as I continue into my doctoral studies.

A fourth avenue of future work, and in fact an ongoing effort by Mr. Jankovics would be to implement the developed finite element slicer into the topology optimization algorithm to constrain it directly in order to minimize manufacturing cost. This could be achieved by minimizing non-printing moves of the printing head of a LASER-/print head-based 3D printer. An experimental study should also be carried out to determine the actual, if any reduction in computation time that slicing the finite element model directly brings over converting to an STL then slicing it. An extension of the entire process could be explored which automates the problem definition prior to the proposed process or automating any necessary-post processing of the additively manufactured part.

In conclusion, Topology optimization promises highly efficient and on-demand designs while additive manufacturing brings manufacturing that is independent of geometric complexity. It is clear that the young field of topology optimization for additive manufacturing has much improvement to be made leading to many exciting possibilities for avenues of future research.

References

[1] Huang, X. and Xie, M., 2010. *Evolutionary topology optimization of continuum structures: methods and applications*. John Wiley & Sons.

[2] Bender D., and Barari A., 2019, Customized Process Planning for Additive Manufacturing– Proceedings of IFAC IMS 2019 – In Press

[3] Bender, D. and Barari, A., 2018. Convergence Control for Topology Optimization. Proceedings of CSME 2018

[4] Bender D., and Barari A., 2019, On Finding the Density-Gradient Vectors of a 3D Structure Resulting from Topology Optimization for Additive Manufacturing, Journal of Structural and Multidisciplinary Optimization – In Press

[5] Bender, D. and Barari, A., 2018, August. Overhanging Feature Analysis for the Additive Manufacturing of Topology Optimized Structures. In *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. V01AT02A044-V01AT02A044). American Society of Mechanical Engineers.

[6] Bender D., and Barari A., 2019, Multi-Element Slicing Approach for Intelligent Additive Manufacturing Preprocessing – Proceedings of CSME 2019 – In Press

[7] Bender D., and Barari A., 2019, Direct Solid Element Slicing for Additive Manufacturing – Proceedings of ASME IDETC 2019 – In Press

[8] Huang, X. and Xie, Y.M., 2007. Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method. *Finite Elements in Analysis and Design*, *43*(14), pp.1039-1049.

[9] Dede, E.M., Joshi, S.N. and Zhou, F., 2015. Topology optimization, additive layer manufacturing, and experimental testing of an air-cooled heat sink. *Journal of Mechanical Design*, *137*(11), p.111403.

[10] Luo, Q. and Tong, L., 2015. Design and testing for shape control of piezoelectric structures using topology optimization. *Engineering Structures*, *97*, pp.90-104.

[11] Torquato, S., Hyun, S. and Donev, A., 2002. Multifunctional composites: optimizing microstructures for simultaneous transport of heat and electricity. *Physical review letters*, *89*(26), p.266601.

[12] Zhou, S. and Li, Q., 2008. Design of graded two-phase microstructures for tailored elasticity gradients. *Journal of Materials Science*, *43*(15), pp.5157-5167.

[13] Beghini, L.L., Beghini, A., Katz, N., Baker, W.F. and Paulino, G.H., 2014. Connecting architecture and engineering through structural topology optimization. *Engineering Structures*, *59*, pp.716-726.

[14] Bobby, S., Spence, S.M., Bernardini, E. and Kareem, A., 2014. Performance-based topology optimization for wind-excited tall buildings: A framework. *Engineering Structures*, *74*, pp.242-255.

[15] Cui, C., Ohmori, H. and Sasaki, M., 2003. Computational morphogenesis of 3D structures by extended ESO method. *Journal of the International Association for Shell and Spatial Structures*, *44*(1), pp.51-61.

[16] Sigmund, O., 2000. Topology optimization: a tool for the tailoring of structures and materials. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, *358*(1765), pp.211-227.

[17] Polavarapu, S., Thompson, L.L. and Grujicic, M., 2009, January. Topology and free size optimization with manufacturing constraints for light weight die cast automotive backrest frame. In *ASME 2009*

International Mechanical Engineering Congress and Exposition (pp. 641-655). American Society of Mechanical Engineers.

[18] Marchesi, T.R., Lahuerta, R.D., Silva, E.C., Tsuzuki, M.S., Martins, T.C., Barari, A. and Wood, I., 2015. Topologically optimized diesel engine support manufactured with additive manufacturing. *IFAC-PapersOnLine*, *48*(3), pp.2333-2338.

[19] Atzeni, E., Iuliano, L., Marchiandi, G., Minetola, P., Salmi, A., Bassoli, E., Denti, L. and Gatto, A., 2014. Additive manufacturing as a cost-effective way to produce metal parts. In *High Value Manufacturing: Advanced Research in Virtual and Rapid Prototyping-Proceedings of the 6th International Conference on Advanced Research and Rapid Prototyping, VR@ P* (Vol. 2013, pp. 3-8).

[20] Chang, K.H. and Tang, P.S., 2001. Integration of design and manufacturing for structural shape optimization. *Advances in engineering software*, *32*(7), pp.555-567.

[21] Rozvany, G.I., 2009. A critical review of established methods of structural topology optimization. *Structural and multidisciplinary optimization*, *37*(3), pp.217-237.

[22] Bendøse, M.P. and Sigmund, O., 2003. Topology Optimization: Theory, Methods and Applications. ISBN: 3-540-42992-1.

[23] Sigmund, O. and Maute, K., 2013. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, *48*(6), pp.1031-1055.

[24] Bendsøe, M.P. and Kikuchi, N., 1988. Generating optimal topologies in structural design using a homogenization method. *Computer methods in applied mechanics and engineering*, *71*(2), pp.197-224.

[25] Suzuki, K. and Kikuchi, N., 1991. A homogenization method for shape and topology optimization. *Computer methods in applied mechanics and engineering*, *93*(3), pp.291-318.

[26] Xie, Y.M. and Steven, G.P., 1993. A simple evolutionary procedure for structural optimization. *Computers & structures*, *49*(5), pp.885-896.

[27] Mattheck, C. and Burkhardt, S., 1990. A new method of structural shape optimization based on biological growth. *International Journal of Fatigue*, *12*(3), pp.185-190.

[28] Querin, O.M., Steven, G.P. and Xie, Y.M., 1998. Evolutionary structural optimisation (ESO) using a bidirectional algorithm. *Engineering computations*, *15*(8), pp.1031-1048.

[29] Young, V., Querin, O.M., Steven, G.P. and Xie, Y.M., 1999. 3D and multiple load case bi-directional evolutionary structural optimization (BESO). *Structural optimization*, *18*(2-3), pp.183-192.

[30] Bendsøe, M.P., 1989. Optimal shape design as a material distribution problem. *Structural optimization*, *1*(4), pp.193-202.

[31] Zhou, M. and Rozvany, G.I.N., 1991. The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, *89*(1-3), pp.309-336.

[32] Sethian, J.A. and Wiegmann, A., 2000. Structural boundary design via level set and immersed interface methods. *Journal of computational physics*, *163*(2), pp.489-528.

[33] Wang, M.Y., Wang, X. and Guo, D., 2003. A level set method for structural topology optimization. *Computer methods in applied mechanics and engineering*, *192*(1-2), pp.227-246.

[34] Logan, D.L., 2011. *A first course in the finite element method*. Cengage Learning.

[35] Liu, K. and Tovar, A., 2014. An efficient 3D topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, *50*(6), pp.1175-1196.

[36] Hinton, E. and Sienz, J., 1995. Fully stressed topological design of structures using an evolutionary procedure. *Engineering computations*, *12*(3), pp.229-244.

[37] Rozvany, G.I. and Querin, O.M., 2002. Combining ESO with rigorous optimality criteria. International journal of vehicle design, 28(4), pp.294-299.

[38] Bendsøe, M.P. and Sigmund, O., 1999. Material interpolation schemes in topology optimization. *Archive of applied mechanics*, *69*(9-10), pp.635-654.

[39] Rietz, A., 2001. Sufficiency of a finite exponent in SIMP (power law) methods. *Structural and Multidisciplinary Optimization*, *21*(2), pp.159-163.

[40] Chiandussi, G., 2006. On the solution of a minimum compliance topology optimisation problem by optimality criteria without a priori volume constraint specification. *Computational Mechanics*, *38*(1), pp.77-99.

[41] Huang, X. and Xie, Y.M., 2010. A further review of ESO type methods for topology optimization. *Structural and Multidisciplinary Optimization*, *41*(5), pp.671-683.

[42] Huang, X. and Xie, Y.M., 2009. Bi-directional evolutionary topology optimization of continuum structures with one or multiple materials. *Computational Mechanics*, *43*(3), p.393.

[43] Zhang, W. and Sun, S., 2006. Scale-related topology optimization of cellular materials and structures. *International journal for numerical methods in engineering*, *68*(9), pp.993-1011.

[44] Huang, X. and Xie, Y.M., 2008. Optimal design of periodic structures using evolutionary topology optimization. *Structural and Multidisciplinary Optimization*, *36*(6), pp.597-606.

[45] Yang, X.Y., Xie, Y.M. and Steven, G.P., 2005. Evolutionary methods for topology optimisation of continuous structures with design dependent loads. *Computers & structures*, *83*(12-13), pp.956-963.

[46] Ansola, R., Canales, J. and Tárrago, J.A., 2006. An efficient sensitivity computation strategy for the evolutionary structural optimization (ESO) of continuum structures subjected to self-weight loads. *Finite elements in analysis and design*, *42*(14-15), pp.1220-1230.

[47] Huang, X. and Xie, Y.M., 2011. Evolutionary topology optimization of continuum structures including design-dependent self-weight loads. *Finite Elements in Analysis and Design*, *47*(8), pp.942-948.

[48] Sigmund, O., 1997. On the design of compliant mechanisms using topology optimization. *Journal of Structural Mechanics*, *25*(4), pp.493-524.

[49] Xie, Y.M. and Steven, G.P., 1996. Evolutionary structural optimization for dynamic problems. *Computers & Structures*, *58*(6), pp.1067-1073.

[50] Yang, X.Y., Xie, Y.M., Steven, G.P. and Querin, O.M., 1999. Topology optimization for frequencies using an evolutionary method. *Journal of Structural Engineering*, *125*(12), pp.1432-1438.

[51] Huang, X., Zuo, Z.H. and Xie, Y.M., 2010. Evolutionary topological optimization of vibrating continuum structures for natural frequencies. *Computers & structures*, *88*(5-6), pp.357-364.

[52] Chu, D.N., Xie, Y.M., Hira, A. and Steven, G.P., 1996. Evolutionary structural optimization for problems with stiffness constraints. *Finite Elements in Analysis and Design*, *21*(4), pp.239-251.

[53] Diaz, A. and Sigmund, O., 1995. Checkerboard patterns in layout optimization. *Structural optimization*, *10*(1), pp.40-45.

[54] Haber, R.B., Jog, C.S. and Bendsøe, M.P., 1996. A new approach to variable-topology shape design using a constraint on perimeter. *Structural optimization*, *11*(1-2), pp.1-12.

[55] Li, Q., Steven, G.P. and Xie, Y.M., 2001. A simple checkerboard suppression algorithm for evolutionary structural optimization. *Structural and Multidisciplinary Optimization*, *22*(3), pp.230-239.

[56] Sigmund, O. and Petersson, J., 1998. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural optimization*, *16*(1), pp.68-75.

[57] Stolpe, M. and Svanberg, K., 2001. On the trajectories of penalization methods for topology optimization. *Structural and Multidisciplinary Optimization*, *21*(2), pp.128-139.

[58] Rozvany, G.I.N., Zhou, M. and Sigmund, O. 1994. Optimization of topology. *Advances in Design Optimization*. H. Adeli (ed.), Chapman & Hall, London: 340–99.

[59] Sigmund, O. and Torquato, S., 1997. Design of materials with extreme thermal expansion using a three-phase topology optimization method. *Journal of the Mechanics and Physics of Solids*, *45*(6), pp.1037-1067.

[60] Sigmund, O., 2001. A 99 line topology optimization code written in Matlab. *Structural and multidisciplinary optimization*, *21*(2), pp.120-127.

[61] Rozvany, G.I. and Zhou, M., 1993. *Optimality criteria methods for large structural systems*. Univ.-Gesamthochschule.

[62] Svanberg, K., 1987. The method of moving asymptotes—a new method for structural optimization. *International journal for numerical methods in engineering*, *24*(2), pp.359-373.

[63] Zillober, C., 1993. A globally convergent version of the method of moving asymptotes. *Structural optimization*, *6*(3), pp.166-174.

[64] Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B.S. and Sigmund, O., 2011. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, *43*(1), pp.1-16.

[65] Liu, J. and Ma, Y., 2016. A survey of manufacturing oriented topology optimization methods. Advances in Engineering Software, 100, pp.161-175.

[66] Lazarov, B.S., Wang, F. and Sigmund, O., 2016. Length scale and manufacturability in density-based topology optimization. *Archive of Applied Mechanics*, *86*(1-2), pp.189-218.

[67] Azamirad, G. and Arezoo, B., 2016. Structural design of stamping die components using bi-directional evolutionary structural optimization method. *The International Journal of Advanced Manufacturing Technology*, *87*(1-4), pp.969-979.

[68] Zuo, K.T., Chen, L.P., Zhang, Y.Q. and Yang, J., 2006. Manufacturing-and machining-based topology optimization. *The international journal of advanced manufacturing technology*, *27*(5-6), pp.531-536.

[69] Harzheim, L. and Graf, G., 2006. A review of optimization of cast parts using topology optimization. Structural and multidisciplinary optimization, 31(5), pp.388-399.

[70] Schevenels, M., Lazarov, B.S. and Sigmund, O., 2011. Robust topology optimization accounting for spatially varying manufacturing errors. *Computer Methods in Applied Mechanics and Engineering*, *200*(49-52), pp.3613-3627.

[71] Zegard, T. and Paulino, G.H., 2016. Bridging topology optimization and additive manufacturing. *Structural and Multidisciplinary Optimization*, *53*(1), pp.175-192.

[72] Liu, J., Gaynor, A.T., Chen, S., Kang, Z., Suresh, K., Takezawa, A., Li, L., Kato, J., Tang, J., Wang, C.C. and Cheng, L., 2018. Current and future trends in topology optimization for additive manufacturing. *Structural and Multidisciplinary Optimization*, *57*(6), pp.2457-2483.

[73] Doubrovski, Z., Verlinden, J.C. and Geraedts, J.M., 2011, January. Optimal design for additive manufacturing: opportunities and challenges. In ASME 2011 international design engineering technical conferences and computers and information in engineering conference (pp. 635-646). American Society of Mechanical Engineers.

[74] Brackett, D., Ashcroft, I. and Hague, R., 2011, August. Topology optimization for additive manufacturing. In Proceedings of the solid freeform fabrication symposium, Austin, TX (Vol. 1, pp. 348-362).

[75] Sikder, S., Barari, A. and Kishawy, H.A., 2014, August. Effect of adaptive slicing on surface integrity in additive manufacturing. In *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. V01AT02A052-V01AT02A052). American Society of Mechanical Engineers.

[76] Wong, K.V. and Hernandez, A., 2012. A review of additive manufacturing. ISRN Mechanical Engineering, 2012.

[77] Hull, C.W., UVP Inc, 1986. Apparatus for production of three-dimensional objects by stereolithography. U.S. Patent 4,575,330.

[78] Bártolo, P.J. ed., 2011. *Stereolithography: materials, processes and applications*. Springer Science & Business Media.

[79] Tumbleston, J.R., Shirvanyants, D., Ermoshkin, N., Janusziewicz, R., Johnson, A.R., Kelly, D., Chen, K., Pinschmidt, R., Rolland, J.P., Ermoshkin, A. and Samulski, E.T., 2015. Continuous liquid interface production of 3D objects. Science, 347(6228), pp.1349-1352.

[80] Wu, C., Dai, C., Fang, G., Liu, Y.J. and Wang, C.C., 2017, May. RoboFDM: A robotic system for support-free fabrication using FDM. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1175-1180). IEEE.

[81] Bikas, H., Stavropoulos, P. and Chryssolouris, G., 2016. Additive manufacturing methods and modelling approaches: a critical review. *The International Journal of Advanced Manufacturing Technology*, *83*(1-4), pp.389-405.

[82] Atwood, C., Ensz, M., Greene, D., Griffith, M., Harwell, L., Reckaway, D., Romero, T., Schlienger, E. and Smugeresky, J., 1998. *Laser engineered net shaping (LENS (TM)): A tool for direct fabrication of metal parts* (No. SAND98-2473C). Sandia National Laboratories, Albuquerque, NM, and Livermore, CA.

[83] Cong, W. and Ning, F., 2017. A fundamental investigation on ultrasonic vibration-assisted laser engineered net shaping of stainless steel. *International Journal of Machine Tools and Manufacture*, *121*, pp.61-69.

[84] Yadollahi, A. and Shamsaei, N., 2017. Additive manufacturing of fatigue resistant materials: Challenges and opportunities. *International Journal of Fatigue*, *98*, pp.14-31.

[85] Ahn, S.H., Montero, M., Odell, D., Roundy, S. and Wright, P.K., 2002. Anisotropic material properties of fused deposition modeling ABS. *Rapid prototyping journal*, *8*(4), pp.248-257.

[86] Vanek, J., Galicia, J.A.G. and Benes, B., 2014, August. Clever support: Efficient support structure generation for digital fabrication. In *Computer graphics forum* (Vol. 33, No. 5, pp. 117-125).

[87] Wang, D., Yang, Y., Yi, Z. and Su, X., 2013. Research on the fabricating quality optimization of the overhanging surface in SLM process. *The International Journal of Advanced Manufacturing Technology*, *65*(9-12), pp.1471-1484.

[88] Barari, A., Kishawy, H.A., Kaji, F. and Elbestawi, M.A., 2017. On the surface quality of additive manufactured parts. *The International Journal of Advanced Manufacturing Technology*, *89*(5-8), pp.1969-1974.

[89] Sikder, S., Barari, A. and Kishawy, H.A., 2015. Global adaptive slicing of NURBS based sculptured surface for minimum texture error in rapid prototyping. *Rapid prototyping journal*, *21*(6), pp.649-661.

[90] Ippolito, R., Iuliano, L. and Gatto, A., 1995. Benchmarking of rapid prototyping techniques in terms of dimensional accuracy and surface finish. *CIRP annals*, *44*(1), pp.157-160.

[91] Lalehpour, A. and Barari, A., 2018. A more accurate analytical formulation of surface roughness in layer-based additive manufacturing to enhance the product's precision. *The International Journal of Advanced Manufacturing Technology*, *96*(9-12), pp.3793-3804.

[92] Kaji, F. and Barari, A., 2015. Evaluation of the surface roughness of additive manufacturing parts based on the modelling of cusp geometry. *IFAC-PapersOnLine*, *48*(3), pp.658-663.

[93] Guest, J.K., Prévost, J.H. and Belytschko, T., 2004. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International journal for numerical methods in engineering*, *61*(2), pp.238-254.

[94] Poulsen, T.A., 2003. A new scheme for imposing a minimum length scale in topology optimization. *International Journal for Numerical Methods in Engineering*, *57*(6), pp.741-760.

[95] Lalehpour, A. and Barari, A., 2016. Post processing for Fused Deposition Modeling Parts with Acetone Vapour Bath. *IFAC-PapersOnLine*, *49*(31), pp.42-48.

[96] Lalehpour, A., Janeteas, C. and Barari, A., 2018. Surface roughness of FDM parts after post-processing with acetone vapor bath smoothing process. *The International Journal of Advanced Manufacturing Technology*, *95*(1-4), pp.1505-1520.

[97] Wu, A.S., Brown, D.W., Kumar, M., Gallegos, G.F. and King, W.E., 2014. An experimental investigation into additive manufacturing-induced residual stresses in 316L stainless steel. *Metallurgical and Materials Transactions A*, *45*(13), pp.6260-6270.

[98] Salvati, E., Lunt, A.J.G., Ying, S., Sui, T., Zhang, H.J., Heason, C., Baxter, G. and Korsunsky, A.M., 2017. Eigenstrain reconstruction of residual strains in an additively manufactured and shot peened nickel superalloy compressor blade. *Computer Methods in Applied Mechanics and Engineering*, *320*, pp.335-351.

[99] Wu, J., Wang, C.C., Zhang, X. and Westermann, R., 2016. Self-supporting rhombic infill structures for additive manufacturing. *Computer-Aided Design*, *80*, pp.32-42.

[100] Radman, A., Huang, X. and Xie, Y.M., 2013. Topology optimization of functionally graded cellular materials. *Journal of Materials Science*, *48*(4), pp.1503-1510.

[101] Xia, Q. and Wang, M.Y., 2008. Simultaneous optimization of the material properties and the topology of functionally graded structures. *Computer-Aided Design*, *40*(6), pp.660-675.

[102] Zhang, P., Liu, J. and To, A.C., 2017. Role of anisotropic properties on topology optimization of additive manufactured load bearing structures. *Scripta Materialia*, *135*, pp.148-152.

[103] Guo, N. and Leu, M.C., 2013. Additive manufacturing: technology, applications and research needs. *Frontiers of Mechanical Engineering*, *8*(3), pp.215-243.

[104] Compton, B.G. and Lewis, J.A., 2014. 3D-printing of lightweight cellular composites. *Advanced materials*, *26*(34), pp.5930-5935.

[105] Diaz, A.R. and Bendsøe, M., 1992. Shape optimization of structures for multiple loading conditions using a homogenization method. *Structural optimization*, *4*(1), pp.17-22.

[106] 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. *International Journal for Numerical Methods in Engineering*, *101*(8), pp.571-605.

[107] Song, X., Pan, Y. and Chen, Y., 2015. Development of a low-cost parallel kinematic machine for multidirectional additive manufacturing. *Journal of Manufacturing Science and Engineering*, *137*(2), p.021005.

[108] Ulu, E., Korkmaz, E., Yay, K., Ozdoganlar, O.B. and Kara, L.B., 2015. Enhancing the structural performance of additively manufactured objects through build orientation optimization. *Journal of Mechanical Design*, *137*(11), p.111410.

[109] Calignano, F., 2014. Design optimization of supports for overhanging structures in aluminum and titanium alloys by selective laser melting. *Materials & Design*, *64,* pp.203-213.

[110] Jankovics, D., Gohari, H., Tayefeh, M. and Barari, A., 2018. Developing Topology Optimization with Additive Manufacturing Constraints in ANSYS®. *IFAC-PapersOnLine*, *51*(11), pp.1359-1364.

[111] Gaynor, A.T. and Guest, J.K., 2016. Topology optimization considering overhang constraints: Eliminating sacrificial support material in additive manufacturing through design. *Structural and Multidisciplinary Optimization*, *54*(5), pp.1157-1172.

[112] Langelaar, M., 2017. An additive manufacturing filter for topology optimization of print-ready designs. *Structural and multidisciplinary optimization*, *55*(3), pp.871-883.

[113] Driessen, A.M., 2016. Overhang constraint in topology optimisation for additive manufacturing: a density gradient based approach.

[114] Mass, Y. and Amir, O., 2017. Topology optimization for additive manufacturing: Accounting for overhang limitations using a virtual skeleton. *Additive Manufacturing*, *18*, pp.58-73.

[115] Leary, M., Merli, L., Torti, F., Mazur, M. and Brandt, M., 2014. Optimal topology for additive manufacture: A method for enabling additive manufacture of support-free optimal structures. *Materials & Design*, *63*, pp.678-690.

[116] Pham, D.T., Dimov, S.S. and Gault, R.S., 1999. Part orientation in stereolithography. *The International Journal of Advanced Manufacturing Technology*, *15*(9), pp.674-682.

[117] Huang, X., Ye, C., Wu, S., Guo, K. and Mo, J., 2009. Sloping wall structure support generation for fused deposition modeling. *The International Journal of Advanced Manufacturing Technology*, *42*(11-12), p.1074.

[118] Dumas, J., Hergel, J. and Lefebvre, S., 2014. Bridging the gap: automated steady scaffoldings for 3D printing. *ACM Transactions on Graphics (TOG)*, *33*(4), p.98.

[119] Strano, G., Hao, L., Everson, R.M. and Evans, K.E., 2013. A new approach to the design and optimisation of support structures in additive manufacturing. *The International Journal of Advanced Manufacturing Technology*, *66*(9-12), pp.1247-1254.

[120] Åström, K.J. and Hägglund, T., 2001. The future of PID control. *Control engineering practice*, *9*(11), pp.1163-1175.

[121] K. Ogata, *Modern control engineering*, 5th ed. [Delhi]: Pearson, 2016.

[122] Ang, K.H., Chong, G. and Li, Y., 2005. PID control system analysis, design, and technology. *IEEE transactions on control systems technology*, *13*(4), pp.559-576.

[123] Chu, D.N., Xie, Y.M., Hira, A. and Steven, G.P., 1997. On various aspects of evolutionary structural optimization for problems with stiffness constraints. *Finite Elements in Analysis and Design*, *24*(4), pp.197-212.

[124] Lorensen, W.E. and Cline, H.E., 1987, August. Marching cubes: A high resolution 3D surface construction algorithm. In *ACM siggraph computer graphics* (Vol. 21, No. 4, pp. 163-169). ACM.

[125] Shapiro, L.G. and Stockman, G.C., Computer Vision, March 2000.

[126] Ziou, D. and Tabbone, S., 1998. Edge detection techniques-an overview. *Pattern Recognition and Image Analysis C/C of Raspoznavaniye Obrazov I Analiz Izobrazhenii*, *8*, pp.537-559.

[127] Muir, M.J., Querin, O.M. and Toropov, V., 2014. Rules, Precursors and Parameterisation Methodologies for Topology Optimised Structural Designs Realised Through Additive Manufacturing. In *10th AIAA Multidisciplinary Design Optimization Conference* (p. 0635).

[128] Morgan, H.D., Cherry, J.A., Jonnalagadda, S., Ewing, D. and Sienz, J., 2016. Part orientation optimisation for the additive layer manufacture of metal components. *The International Journal of Advanced Manufacturing Technology*, *86*(5-8), pp.1679-1687.

[129] D. Jankovics and A. Barari, “Customization of Automotive Structural Components using Additive Manufacturing and Topology Optimization,” in 13th IFAC Workshop on Intelligent Manufacturing Systems, (Pre-print), 2019.

[130] Barari, A. and Pop-Iliev, R., 2009. Reducing rigidity by implementing closed-loop engineering in adaptable design and manufacturing systems. *Journal of Manufacturing Systems*, *28*(2-3), pp.47-54.

APPENDICES

A1. MATLAB Program: DensityGradient

```matlab
function [mag, dir] = DensityGradient(x)
[nely,nelx,nelz] = size(x);
mag = zeros(nely,nelx,nelz);
dir = zeros(nely,nelx,nelz);

xTemp = zeros(nely+2,nelx+2,nelz+2);
xTemp(2:nely+1, 2:nelx+1, 2:nelz+1) = x;

for ely = 2:nely+1
    for elx = 2:nelx+1
        for elz = 2:nelz+1
            magX = ((xTemp(ely,elx+1,elz) - xTemp(ely,elx-1,elz))*4 ...
               + (xTemp(ely,elx+1,elz-1) - xTemp(ely,elx-1,elz-1))*2 ...
               + (xTemp(ely,elx+1,elz+1) - xTemp(ely,elx-1,elz+1))*2 ...
               + (xTemp(ely+1,elx+1,elz) - xTemp(ely+1,elx-1,elz))*2 ...
               + (xTemp(ely-1,elx+1,elz) - xTemp(ely-1,elx-1,elz))*2 ...
               + (xTemp(ely+1,elx+1,elz+1) - xTemp(ely+1,elx-1,elz+1)) ...
               + (xTemp(ely+1,elx+1,elz-1) - xTemp(ely+1,elx-1,elz-1)) ...
               + (xTemp(ely-1,elx+1,elz-1) - xTemp(ely-1,elx-1,elz-1)) ...
               + (xTemp(ely-1,elx+1,elz+1) - xTemp(ely-1,elx-1,elz+1)))/16;

            magY = ((xTemp(ely+1,elx,elz) - xTemp(ely-1,elx,elz))*4 ...
               + (xTemp(ely+1,elx-1,elz) - xTemp(ely-1,elx-1,elz))*2 ...
               + (xTemp(ely+1,elx+1,elz) - xTemp(ely-1,elx+1,elz))*2 ...
               + (xTemp(ely+1,elx,elz+1) - xTemp(ely-1,elx,elz+1))*2 ...
               + (xTemp(ely+1,elx,elz-1) - xTemp(ely-1,elx,elz-1))*2 ...
               + (xTemp(ely+1,elx+1,elz+1) - xTemp(ely-1,elx+1,elz+1)) ...
               + (xTemp(ely+1,elx+1,elz-1) - xTemp(ely-1,elx+1,elz-1)) ...
               + (xTemp(ely+1,elx-1,elz-1) - xTemp(ely-1,elx-1,elz-1)) ...
               + (xTemp(ely+1,elx-1,elz+1) - xTemp(ely-1,elx-1,elz+1)))/16;

            magZ = ((xTemp(ely,elx,elz+1) - xTemp(ely,elx,elz-1))*4 ...
               + (xTemp(ely,elx-1,elz+1) - xTemp(ely,elx-1,elz-1))*2 ...
               + (xTemp(ely,elx+1,elz+1) - xTemp(ely,elx+1,elz-1))*2 ...
               + (xTemp(ely+1,elx,elz+1) - xTemp(ely+1,elx,elz-1))*2 ...
               + (xTemp(ely-1,elx,elz+1) - xTemp(ely-1,elx,elz-1))*2 ...
               + (xTemp(ely+1,elx+1,elz+1) - xTemp(ely+1,elx+1,elz-1)) ...
               + (xTemp(ely+1,elx-1,elz+1) - xTemp(ely+1,elx-1,elz-1)) ...
               + (xTemp(ely-1,elx-1,elz+1) - xTemp(ely-1,elx-1,elz-1)) ...
               + (xTemp(ely-1,elx+1,elz+1) - xTemp(ely-1,elx+1,elz-1)))/16;

            mag(ely-1,elx-1,elz-1) = sqrt(magX^2 + magY^2 + magZ^2);

            if magY < 0
                mult = -1;
            else
                mult = 1;
            end

            dt = mult*acosd((magX^2 + magZ^2) ...
            / ((sqrt(magX^2 + magZ^2))*(sqrt(magX^2 + magY^2 + magZ^2))));

            if isreal(dt)
                dir(ely-1,elx-1,elz-1) = dt;
            else
                dir(ely-1,elx-1,elz-1) = 0;
            end

            if isnan(dir(ely-1,elx-1,elz-1))
                dir(ely-1,elx-1,elz-1) = 0;
            end
        end
    end
end
```