ULSAN NATIONAL INSTITUTE OF
SCIENCE AND TECHNOLOGY

Master's Thesis

NON-PERIODIC LATTICE STRUCTURE DESIGN FOR ADDITIVE MANUFACTURING

Adrian Matias Chung Baek

Department of Mechanical Engineering

Graduate School of UNIST

2020

ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY

NON-PERIODIC LATTICE STRUCTURE DESIGN FOR ADDITIVE MANUFACTURING

Adrian Matias Chung Baek

Department of Mechanical Engineering

Graduate School of UNIST

ULSAN NATIONAL INSTITUTE OF
SCIENCE AND TECHNOLOGY

Non-Periodic Lattice Structure Design for Additive Manufacturing

A thesis
submitted to the Graduate School of UNIST
in partial fulfillment of the
requirements for the degree of
Master of Science

Adrian Matias Chung Baek

12.24.2019

Approved by

____

Advisor
Namhun Kim

Non-Periodic Lattice Structure Design for Additive Manufacturing

Adrian Matias Chung Baek

This certifies that the thesis of Adrian Matias Chung Baek is approved.

12/24/2019

Advisor: Namhun Kim

Duck Young Kim

Young-Bin Park

Abstract

As manufacturability of lattice structures has been relaxed with the availability of additive manufacturing (AM) technology, the study of cellular structure optimization has seen a rapid development during the past decade. Numerous design approaches for lattice structures have been proposed to help designers fabricate efficient lattice model. Generally, these approaches demand for unbearable computational cost and prior knowledge. To overcome the drawbacks of existing methods, Choi et al. proposes a simple framework of generating non-periodic lattice structures using topologically pre-optimized building blocks. However, this method does not properly consider the manufacturability of the lattice structure by neglecting additive manufacturing constraints in the design process. This thesis suggests a strategy to consider manufacturing constraints for the AM process in a contemporary lattice structure generation framework, in this case, Choi et al. work. The proposed method is devised to take full advantage of the already existing components, i.e. building block library, in order not to add complexity in the overall process. Considering the manufacturability of the lattice designs, an algorithm derived from the STL slicing method is introduced in the selection process to replace unprintable building blocks for optimal microstructure. Finally, numerical examples are presented, and reasonable solutions have been obtained to show the feasibility of the proposed method.

Contents

Abstract v

Contents i

List of Figures iii

List of Tables v

List of Equations v

1. Introduction 1

1.1 Background 1

1.2 Research Objectives 2

1.3 Outline 3

2. Literature Review 4

2.1 Lattices 4

2.2 Design for Additive Manufacturing 6

2.3 Lattice Structure Design 7

2.4 Non-periodic Lattice Structure Design 9

2.4.1 Brief Overview of Design Method 9

2.4.2 Generation of Building Block Library 10

2.4.3 Allocation of Building Blocks 10

2.4.4 Performance and Limitations 11

2.5 Additive Manufacturing Constraints 13

2.6 STL slicing algorithm 13

3. Methodology: Optimization of the Process for Additive Manufacturing 16

3.1 Model Description 16

3.2 Generation of microstructure unit cell 18

3.3 Optimization of the Process considering AM constraints 24

4. Design and FE simulation 28

4.1 Design Strategy 28

4.1.1 Design of the Library 28

4.1.2 Case Study – Cantilever beam 29

4.1.3 Case Study – Simply Supported Beam 31

4.2 Results of the Generated Structures 32

4.2.1 Cantilever beam 32

4.2.2 Simply Supported Beam 36

4.2.3 Manufacturing performances 36

5. Conclusion 40

5.1 Limitations and Future works 40

Reference 42

Appendix A: Constraints number of building blocks 46

Appendix B: MATLAB Code for counting constraints number 50

List of Figures

Figure 2.1 Strut-based and Skeletal-TPMS based cellular topologies 5

Figure 2.2 Examples of structural design optimization. (a) size optimization, (b) shape optimization, (c) topology optimization........ 7

Figure 2.3 The flowchart of the non-periodic lattice generation method........ 9

Figure 2.4 Visual demonstration of creating and filling hypothetical space. 10

Figure 2.5 Lattice structure of simply supported beam example. 11

Figure 2.6 Sliced planes of $\boldsymbol{B}_{8,-1,0}$. 12

Figure 2.7 Part with (a) overhang and (b) internal void........ 13

Figure 2.8 STL in *ASCII* file format 14

Figure 2.9 Possible cases of the facets with the slicing plane........ 14

Figure 3.1 Flow chart for concurrent method with added steps........ 17

Figure 3.2 Fishbone diagram of variables affecting lattice structure 18

Figure 3.3 Unit cubic cell optimized varying the normal stress........ 20

Figure 3.4 Unit cubic cell optimized varying the target length........ 21

Figure 3.5 The front (top) and orthographic (bottom) view of the optimal microstructure. 23

Figure 3.6 Triangulated STL representation of the surfaces of a simple cube shape 24

Figure 3.7 A sliced STL of a simple cube shape 24

Figure 3.8 Slicing algorithm. 25

Figure 3.9 Cross sections of 2_1_-1.5 building block 25

Figure 3.10 The overhang angle and the thickness of the strut. 26

Figure 3.11 Right skewed distribution graph 27

Figure 4.1 Process of preparing building blocks. 29

Figure 4.2 Cantilever beam example with loading conditions 30

Figure 4.3 Gridded hypothetical geometry of cantilever beam 30

Figure 4.4 Simply supported beam with distributed load. 31

Figure 4.5 Gridded hypothetical geometry of simply supported beam 31

Figure 4.6 Abaqus-MATLAB interface for generating the optimized lattice structure 32

Figure 4.7 FEA result of the cantilever beam 32

Figure 4.8 Front view of the lattice structure construted using Choi et al. library. 33

Figure 4.9 Lattice structure constructed using enhaced library. 34

Figure 4.10 Lattice structure generated using proposed algorithm 35

Figure 4.11 Comparison of the lattice structures with and without AM constraints 36

Figure 4.12 Stress distribution contour of the lattice structures with and without constraints 37

Figure 4.13 Regions of a building block needing support materials 38

Figure 4.14 Decrease of inner support materials in a replaced building blocks 39

List of Tables

Table 2.1 Definition of each case 15
Table 3.1 ANOVA results for minimum safety factor 19
Table 3.2 Details of the parameters used for optimizing optimal microstructure 22
Table 4.1 Stress conditions of Choi et al. library 28
Table 4.2 Added stress conditions 29
Table 4.3 Representative stresses of each grid and selected building blocks 33
Table 4.4 Comparison of the results of optimization methods 35
Table 4.5 Comparison of the mechanical performances of the generated lattice structures 37
Table 4.6 Printing performances 38
Table 4.7 Manufacturing cost 39

List of Equations

Equation 2.1 11
Equation 3.1 19

1. Introduction

1.1 Background

Throughout history, diverse manufacturing techniques and processes have been introduced, each having significant impacts on the global economy. For instance, the introduction of moving Model T assembly line by Henry Ford in the 20th century resulted not only in time and cost-savings, but also in a change of supply chains. Today, a relatively new technology called additive manufacturing (AM) has begun to upend the established economics of production. AM, broadly known as 3D printing, follows a fundamentally different process than conventional subtractive manufacturing methods. Instead of creating products by removing parts of initial material, AM operates in an additive manner where layers of material are stacked up in guidance of a digital file, making the fabrications of new shapes and geometrical features (shape complexity, material complexity, hierarchical complexity, functional complexity) possible [1]. Early AM applications were restricted to models and prototypes because of the low quality of the printed products, but with recent technological and material developments, the use of AM has been expanded into various industry sectors such as motor vehicles, aerospace, machinery, electronics, and medical products. According to the Wohlers Report 2018, the use of AM is still less than a tenth of 1% of total manufacturing output in 2017. Nevertheless, as AM technologies have immense potential to become a mainstream manufacturing process [2], the demands and investments for 3D printing technologies are expected to grow more in the coming years [3].

With the substantial growth of AM, designs with high geometric complexity have also caught the attention of the people. Cellular structures, including foams, lattice structures and honeycombs, are, specially, of interest. These structures are constructed by containing material only where it is needed for particular application [4], resulting into lightweight, strong and unique characteristics that bring a variety of benefits and open up new opportunities [5]. Unfortunately, these structures are not fit to be designed with the traditional Design for Manufacturing (DFM) which focus on relatively simple design geometries to alleviates manufacturing difficulties with the goal of keeping costs down. With the AM technologies, the geometric complexity has negligible influence on the total cost [6]. To take full advantages of this great opportunity and assist designers in exploring the unknown design space, new approaches known as design for additive manufacturing (DFAM) have been developed. DFAM is defined by Rosen [7] as the synthesis of shape, size, structures and material compositions to maximize the product capabilities and achieve desired performances and objectives.

Due to cellular structure becoming a research hotspot, the study of lattice structure optimization has seen a rapid development both in research and industrial applications. Typical DFAM methods to generate these complex concepts include topology optimization, design for multiscale structures, and multi material design. In particular, topology optimization has shown to be an efficient design approach in generating lightweight designs. Its use can be even seen for microstructures with prescribed or extreme properties such as bulk modulus maximization, negative Poisson's ration or zero thermal expansion coefficients [8]. Considerable number of theoretical and computational works on state-of-the-art lattice generation methodologies basing on the theories of topology optimization have been done [9]. These design methods can produce efficient optimized lattice geometries; however, most of them demanded for considerable computation cost.

To decrease the computational cost, methods such as size matching and scaling (SMS), relative density mapping (RDM), and branches of these methods, have been developed [3,7,10]. In the same sense, Choi et al. proposed a non-periodic lattice generation design that uses pre-optimized blocks. Even more, the process was kept as simple as possible for the purpose of generalization.

Nevertheless, most of these methods neglect the one most fundamental issue: the manufacturability constraints of AM [11].

1.2 Research Objectives

It is true that the AM has relaxed the fabricating limitation of lattice structures. However, manufacturing constraints, which have a significant influence on the printing quality and performances, and the mechanical properties of lattice struts, still remain. There are couple of existing design methods dealing with the problem, but these simply consider the constraints on the thickness of struts and end up with a substantial increase of computational cost [11].

To solve the mentioned issue, this study sets the following objectives:

- To suggest a strategy to consider manufacturing constraints for the AM process in a contemporary lattice structure generation framework.
- To consider other geometrical parameters beside the thickness of struts.
- To ensure desired printing quality with the consideration of manufacturability.
- To avoid drastic increasement of complexity or cost.

For these purposes, numerous simulations and analysis are done using commercial tools. Also, existing concept is adopted to write custom code. Lastly, several assumptions in steps are made to simplify the process.

1.3 Outline

The remainder of this thesis is organized as follows. In Chapter 2, related researches to the topic of the thesis were reviewed. In Chapter 3, the additional phases needed, the generation of building block with optimal lattice structure and the optimization of the process, to consider the manufacturability to ensure desired printing quality are described. Chapter 4 shows the effectiveness of proposed framework by using two examples presented in existing methods. Lastly, the thesis will be concluded with the conclusions and future research directions.

2. Literature Review

2.1 Lattices

Natural cellular structures which include bones have been an attractive field to people for centuries due to their high specific strength and stiffness provided by their porous structure [12]. Numerous attempts to mimic these structures in modern technical materials have been made, leading to many different types of manufactured cellular structure as well as a variety of means of fabricating them. Most common forms of cellular structure are the foams, honeycombs and lattices. Among these, lattices are flexible to achieve a wide range of different desired physical properties. Hence, lattice structures have been studied in large spectrums and diverse aspects. For instance, the mechanical properties of lattices have been often a function of the relative density, the solid constituent, and the unit cell architecture. If the material and the relative density are fixed, the mechanical properties of the lattice structures would sorely depend on the architecture of the unit cell. Several research studies focused on optimizing the cell topology in a manner so that it would have enhanced mechanical properties with the least amount of material invested [13].

Lattice structures are generally categorized based on their dimension as 2.5D or 3D or based on their mechanical response as being either bending-dominated or stretch-dominated. According to their unit cell topology, they can be further categorized into strut-based or triply periodic minimal surfaces (TPMS) [12]. Representative strut-based and triply periodic minimal surfaces topologies are illustrated in Figure 2.1.

Strut-based topologies are often preferred for their simplicity of design as well as efficient material distribution. Furthermore, they fully embrace the opportunities presented by AM [14,15]. Common strut-based cell topologies vastly studied are body-centered cubic (BCC) and face-centered-cubic (FCC), or other topologies such as cubic, octet-truss and diamond. By calculating the Maxwell number, it is also possible to know whether these structures have bending-dominated or stretch-dominated behaviors.

Lattices with cubic symmetry with octet-truss and the Kelvin topologies have been vastly investigated [16]. However, recent focus has shifted towards mathematically defined lattice architectures, TPMS based topologies [17]. TPMS are complex 3D topologies that minimize surface area for a given boundary and construct the lattices by periodically repeating in perpendicular directions [18]. Consequently, these surfaces split the lattice space into two or more interlocked domains, each being single connected component with no enfolded voids. Also, the curvature of these surfaces leads to a smoother transition at the connection point of the structure's components, potentially offering improved fixation over strut-based structures. Studies comparing the TPSM-based over strut-based lattices through compression tests suggest that the curvature structure is able to absorb 43.5% more energy than a rectangular structure [11]. In other studies, gyroid structures even showed greater specific energy absorption than BCC structures [19]. Overall, lattice structures with TPMS components show advantages over strut-based structures in terms of performances. Still, further research on comparative performances of strut-based and TPMS lattice are needed.

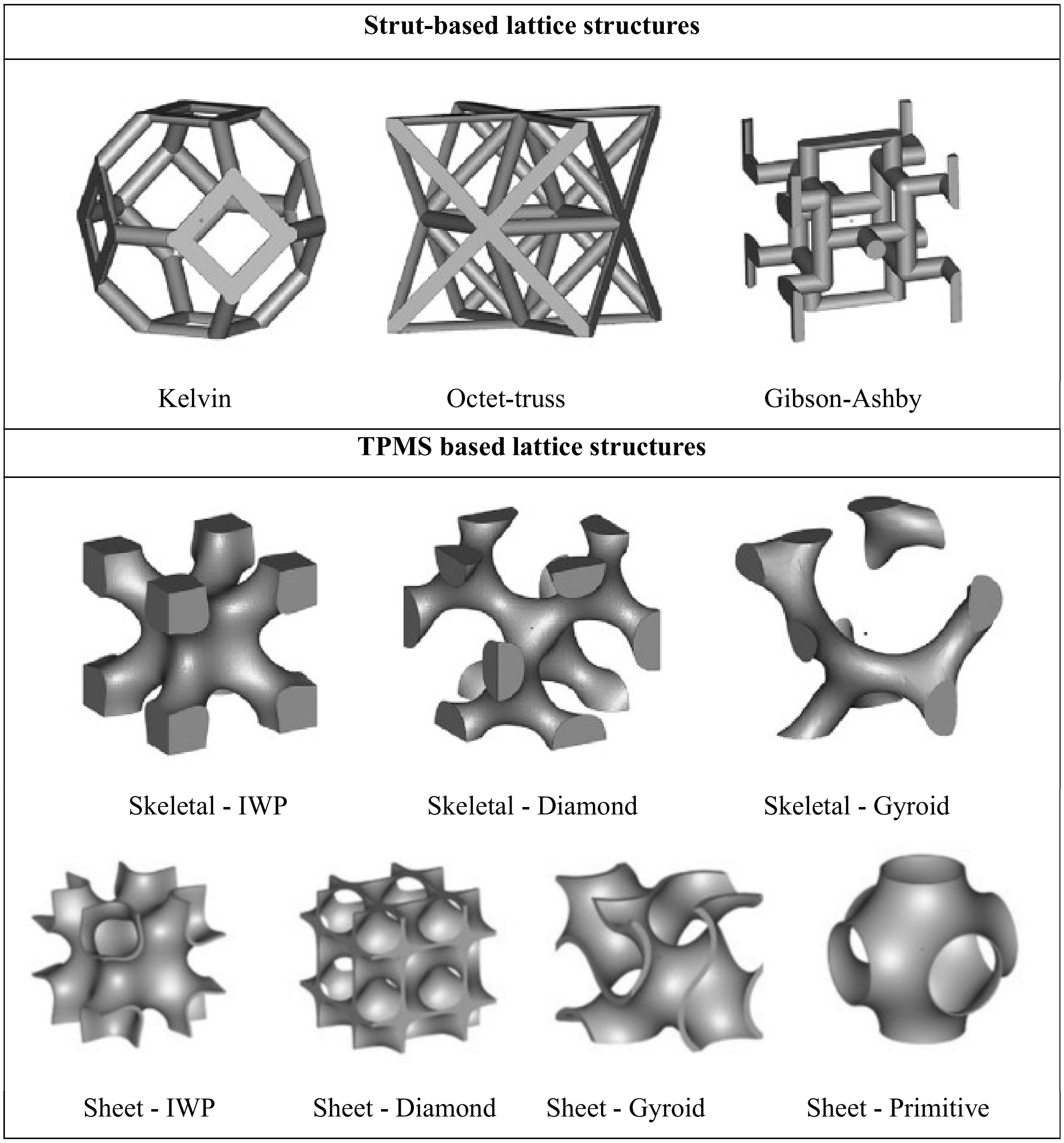


Figure 2.1 Strut-based and Skeletal-TPMS based cellular topologies [20]

2.2 Design for Additive Manufacturing

Traditionally, DFM has been used in manufacturing as a mean of the eliminating the manufacturing difficulties and minimizing the costs. However, the emergence of AM technologies promoted changes in DFM, as AM has different limitations from those of subtractive manufacturing method, leading to DFAM, which can take the advantage of the unique capabilities of AM. These unique capabilities include:

- Shape complexity: make possible the building of virtually any shape including customized geometries and enable shape optimization.
- Material complexity: enable the manufacture of parts with complex material compositions and designed property gradients.
- Hierarchical complexity: possible to design and fabricate hierarchical multi-scale structures from the microstructure through geometric mesostructured (sizes in the millimeter range) to the part-scale macrostructure.
- Functional complexity: can fabricate functional devices directly using AM machines by embedding components and kinematic joints in the building process [21,22].

Although DFAM is a derivation from DFM, in practice design knowledge, tools, rules, processes, and methodologies will be substantially different [3]. Hence, new approaches to the design process and design practice are required, including approaches to explore, complex design spaces [23-25]; to integrate material, mesostructures, and multi-scale design considerations [24-26]; and to overcome the “cognitive barriers” imposed by conventional techniques [27]. According to Rosen, following requirements are necessary: represent and design with large number of shape elements; efficiently search design spaces; represent complex material compositions, ensure their physical meaning, and determine their mechanical properties; ensure the manufacturability of specified shapes, material structures, and properties [23].

With such guidelines, diverse studies to develop new design approaches have been done. Existing literatures regarding DFAM can be categorized into three groups. The first group proposed design methods for specific AM processes or design process. For example, Ponche et al. described a new numerical chain-based design method which can improve the output while considering manufacturing process parameters [28]. In case of the second group, researches to push the boundaries of AM were conducted. Lastly, the third group focused on making guidelines for general DFAM [2].

Due to the endless design space opened by the availability of AM, the development of DFAM still remains the principal challenge of AM.

2.3 Lattice Structure Design

As this study focuses primarily on the development of design method for lattice structure, further contents will be regarding DFAM for lattice structures.

Along with the aforementioned mechanical advantages of lattice structures, the emerging availability of AM techniques has motivated the development of lattice-based structure design, both in research and industrial applications. Researches on lattice structure design can be sorted into two categories: investigation on structures to find the optimal lattice structure or proposition of new approaches to generate these lattice structures. The former investigates the advantages and disadvantages of diverse lattice structures. For example, Deshpande et al. investigated the properties of octet-truss lattice structures and found that the stretching-dominated properties of these structures offered significant potential in lightweight design [16]. Harryson et al. pointed out that the imperfections on the struts should be modeled in order for the FE model results to be in agreement with those of the experiments [29].

The latter is the studies on the methodology to enhance the process of producing lattice structures more efficiently. Typical structural design optimization for cellular structures are size, shape, and topology optimization. Size optimization finds the optimum cross-sectional areas of elements (Figure 2.2 (a)), shape optimization shifts the nodal positions within boundary of the structure (Figure 2.2 (b)) , whereas topology optimization optimizes material layout (Figure 2.2 (c)) [30]. Topology optimization has proven to be superiority over shape and size optimizations as it can find unintuitive and unanticipated designs.

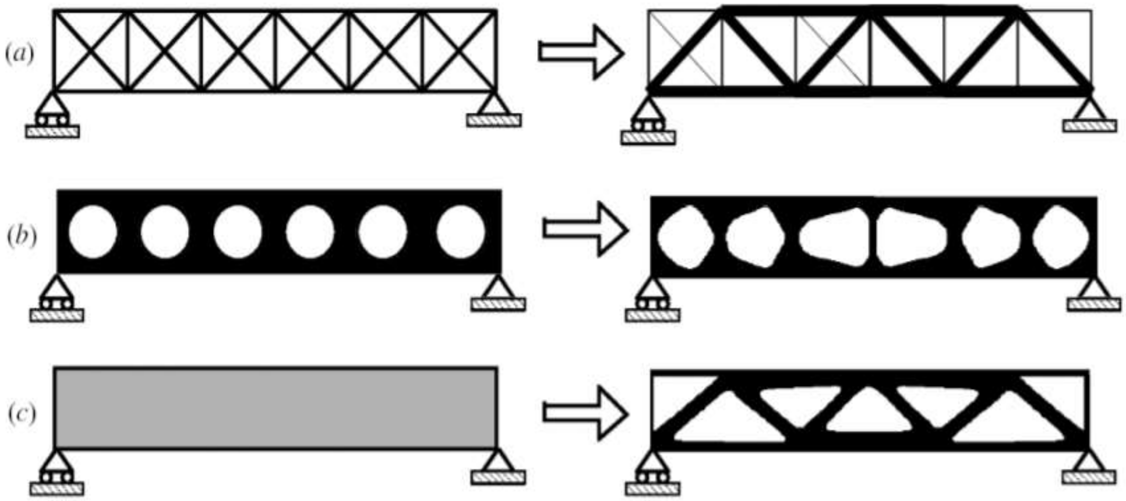


Figure 2.2 Examples of structural design optimizations. (a) size optimization, (b) shape optimization, (c) topology optimization [30]

Several works with implementations or integrations of topology optimization on other concepts are observed. Zhang et al. optimized unit cells using topology and strain energy-based method [8]. In case of Huang et al., he proposed a bidirectional evolutionary method to design cellular materials for maximum bulk or shear modulus [8] whereas Chen et al. proposed a moving iso-surface threshold method [31].

Researches on multiscale topology optimization have also been progressed as ideal multi scale design probably achieve structure with optimal topologies at macro- and micro-level simultaneously [31]. Rodrigues et al. introduced a two-scale optimization model to optimize material distribution by allowed the variance of material variables from point to point [32]. Xia and Breitkopf proposed not only a nonlinear multi-scale framework for concurrent structure design, but also a computationally efficient model using separated representations [33].

The design of lattice structures is not limited to topology optimization. Other optimization algorithms also exist such as the ground structure optimization methods, a truss-based size optimization which retains only elements with non-zero cross-sectional values [34]. Chu et al. made comparisons of two optimization algorithms, particle swarm optimization and Levenberg-Marquardt method by optimizing 2D lattice structures [35]. Most of these studies are focused on two-dimensional example structures. In recent years, optimization on three-dimensional examples has received great attentions. Wieding et al. optimized a 3D bone scaffold case using beam element FE model [36]. However, with the increasing number of design variables, computational cost has intensively increased.

To circumvent this limitation, Chang and Rosen to develop a method called the size matching and scaling (SMS), which works by reducing the multivariable optimization problem into a two-variable problem. Further improvements based on SMS method were done by Nguyen et al. [7] and Alzahrani et al. [4] to generate lattice structures for complex shapes. It is a matter of fact that SMS method has provided great computational cost reduction. However, the process taken to optimize is not user-friendly for designers to adapt. It requires users some knowledge of the processes.

To improve the performance of structures with lower computational costs and aid designers in fully benefitting from AM, Choi et al. developed an effective strategy to design lattice structures.

2.4 Non-periodic Lattice Structure Design

2.4.1 Brief Overview of Design Method

The non-periodic lattice structure design generation framework was proposed by Choi et al. with the goal of providing great computational cost reduction while maintaining the performance of the target structure as well as simplifying the optimization process of obtaining optimal design so that it is more accessible for people with limited knowledge of DFAM [37].

The design methodology consists of two components, a library of optimized cells and an algorithm to allocate the cells at proper location and generate the lattice structure, as shown in Figure 2.3. As a prerequisite, the library must be built and ready beforehand. The library stores unit cubic cells topology optimized under different stress conditions, namely building blocks. If this condition is met, the process starts with the FEA of the given design space using the loading and boundary conditions to collect the elements' stresses, nodal coordinates and connectivity data. With the data collected, building blocks are called and placed on their respective location, creating a lattice structure. Details of each component are discussed in the following sections.

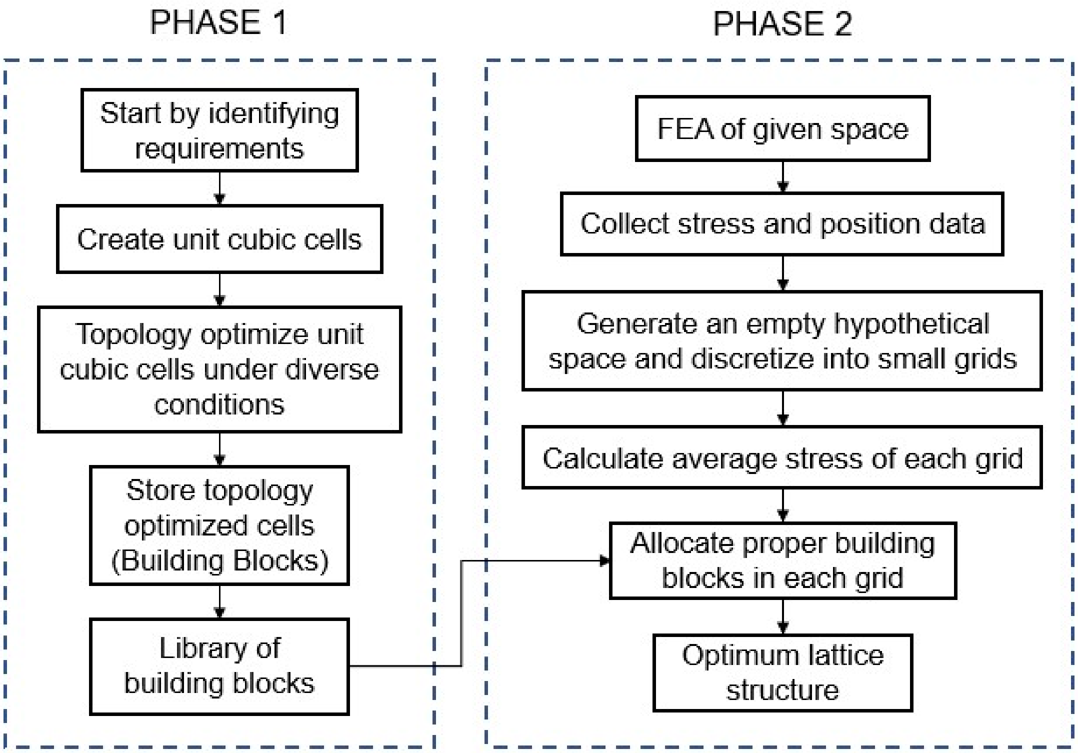

PHASE 1
PHASE 2
Start by identifying requirements
Create unit cubic cells
Topology optimize unit cubic cells under diverse conditions
Store topology optimized cells (Building Blocks)
Library of building blocks
FEA of given space
Collect stress and position data
Generate an empty hypothetical space and discretize into small grids
Calculate average stress of each grid
Allocate proper building blocks in each grid
Optimum lattice structure

Figure 2.3 The flowchart of the non-periodic lattice generation method

2.4.2 Generation of Building Block Library

The creation of building blocks is the one of the crucial steps in this methodology. A unit cubic cell must be prepared two requirements: fixed frame to guarantees the connectivity between the building blocks when assembled and fillets to avoid sharp corners which are sources of stress concentration singularity problem. The cubic cell is then topology optimized with safety factor of 2 under different loading and bounding conditions. The resulting structures are saved in the library as B_{ijk} where the subscripts *i,j,k* indicates the normal principal and shear stresses.

2.4.3 Allocation of Building Blocks

The structure geometry is meshed using solid brick elements. Then, a FE analysis is invoked on the meshed structure to collect the stress information. The mesh node information (elements' nodal coordinates and connectivity data) is also required. As the following step, an empty hypothetical geometry containing the bounding dimensions of the original space is created and discretized into hexahedral shape grids which should not be larger in size than the mesh elements. These grids are then filled with the collected data. A visual representation of the gridded structure is depicted in Figure 2.4.

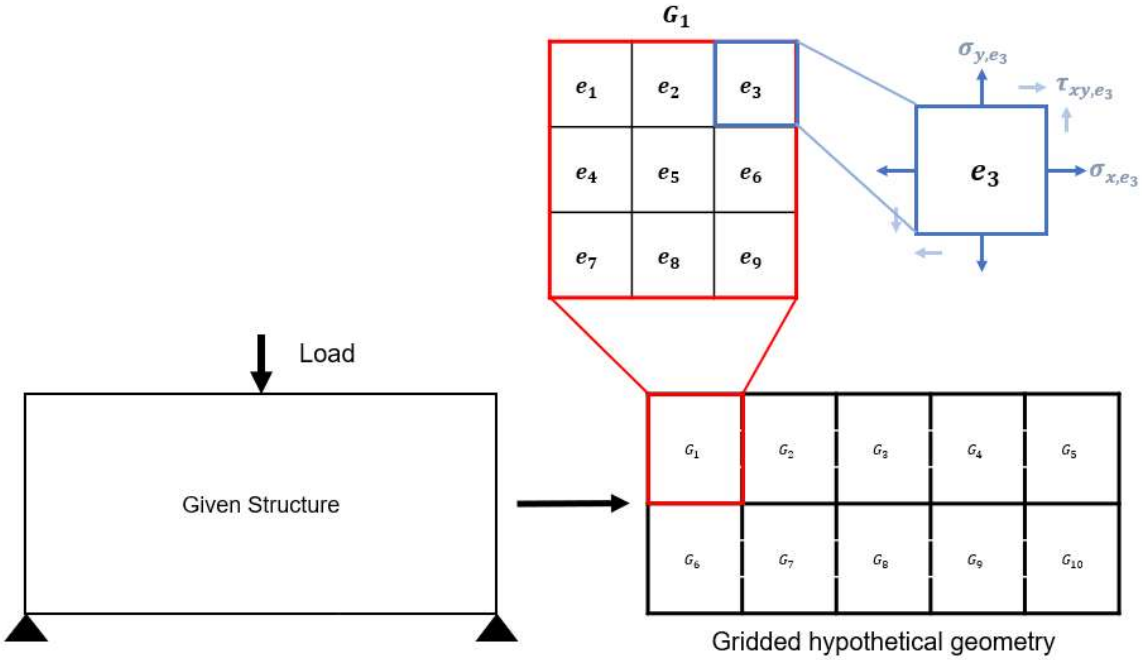


Figure 2.4 Visual demonstration of creating and filling hypothetical space

Once the stress values are all mapped in the grid, the average stress of each grid is determined. The average stress $\sigma_{G_n,(x,y,xy)}$ is equivalent to the summation of all stresses in a grid divided by the number of elements in the same grid.

$$\sigma_{G_n,(x,y,xy)} = \frac{\sum \sigma_{e,(x,y,xy)}}{t} \qquad \textbf{Equation 2.1}$$

where n denotes the grid number, e is the element number, and t is the number of elements in a grid.

In selecting the building block for each grid, there are two rules. First, the stresses used to optimize the cubic cell $\sigma_{B_{ijk}}$ must be equal or greater than the average stress of the grid regardless of the sign. Second, the product of the mentioned stresses must have a positive sign. Following the rules, each grid is matched with the appropriate building block. If all grid is filled, the lattice structure is obtained.

2.4.4 Performance and Limitations

Cho et al. used a simply supported beam example under plane stress for simplicity of the problem. Also, a total of 245 combinations of stresses were used to build the building block library. Going through the mentioned steps, a lattice structure as in Figure 2.5 was generated. Structural analysis was conducted to compare the result with that of a conventional topology optimized structure and validate the proposed framework. According to the comparison results, a significant computational cost reduction was achieved at the minimal cost of mechanical performances.

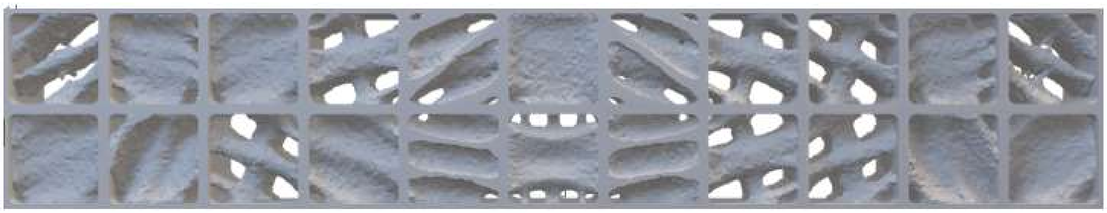


Figure 2.5 Lattice structure of simply supported beam example

Although, this approach can decrease the computational cost, it seems to have limitations. For instance, the design suggested is infeasible as some of the assembled building blocks, i.e. $B_{8,-1,0}$ shown in Figure 2.6, cannot be fabricated by AM processes without the use of support materials, which increases not only the build time, but also the overall cost. Simply speaking, AM constraints haven't been considered in the optimization process, limiting the fabrication of the suggested part using AM. In addition, not sufficient comparison with other existing lattice generation methods are made to truly show the efficiency of the proposed method.

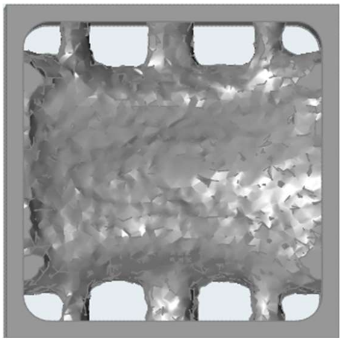
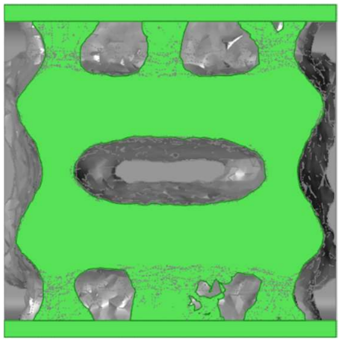
XZ plane	XY plane
	

Figure 2.6 Sliced view of $\boldsymbol{B_{8,-1,0}}$

2.5 Additive Manufacturing Constraints

TO and AM are well suited for each other. However, designs from topology optimization are often not AM friendly. Indeed, it has been observed that optimized structures with surfaces close to being horizontal tend to be distorted when printed without supports [38]. Such horizontal regions are called overhangs. Other inherent constraints in AM processes include enclosed voids, system capability (minimum fabricable size), material compatibility, surface finish. Most of these constraints demand for temporary support material to prevent the printed part from deforming. These support structures are made of either dissolvable materials or same materials as the main body and leads to additional removal procedure which can be automatic or manual. The removal process in a post-process is automatic if dissolvable material is employed, but if single-material manufacturing is employed, it is performed manually [39]. Hence, such sacrificial material increases not only total material usage and build time, but also, time required in post-fabrication treatments. Some recent works make efforts to reduce the involved removal process by adjusting the printing direction or the shape of the design [40,41]. In some cases, however, these advancements in post-processing are meaningless. For instance, when desired design contains interior voids, the existence of support structures becomes more troublesome because they can be hard to removed or sometimes even inaccessible as illustrated in Figure 2.7. Unremoved support structures counteract the goal of the optimization as shapes with a particular volume or property cannot be achieved. At last but not least, wherever support structures are in contact with the part, they result in a poor surface finish [42].

Therefore, the effect of the support structures is another factor to be considered in the AM process. If possible, the use for such structures should be avoided. This, however, is not feasible. Instead, support structure minimization is of significant interest within the AM community. Even the slightest decrease in use of support material can be significant economically as material costs make up for 18%, the largest percentage cost for metal AM, and post-fabrication costs 8% of AM product cost.

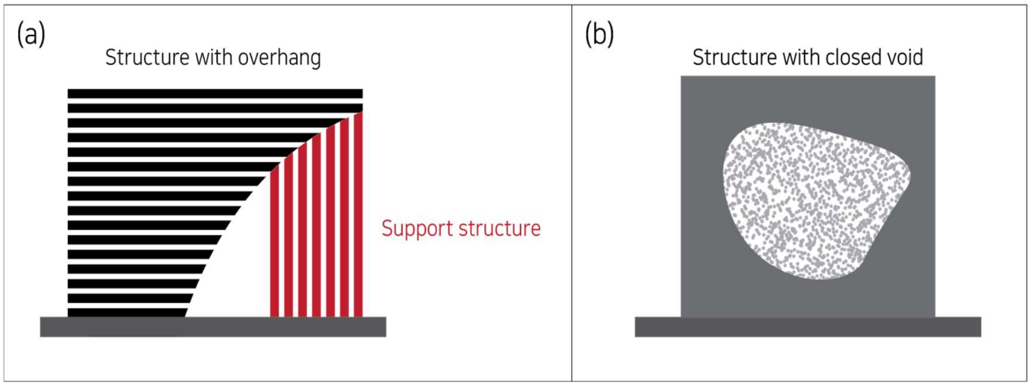


Figure 2.7 Part with (a) overhang and (b) internal void

2.6 STL slicing algorithm

In AM processes, STL file format, binary or ASCII, is the de-facto data exchange standard. STL file defines the surface of the target object through a list of triangle facet data. Each facet consists of a unit normal, a line perpendicular with length of 1, and three vertices or corners for each coordinate as illustrated in Figure 2.8 [43].

```
solid name
facet normal n_i n_j n_k
          outer loop
                    vertex v1_x v1_y v1_z
                    vertex v2_x v2_y v2_z
                    vertex v3_x v3_y v3_z
          end loop
facet normal (second)
          outer loop
                    .....
                    .....
          end loop
end facet
endsolid name
```

Figure 2.8 STL in *ASCII* file format

Through process planning, the STL file is converted into a G-code, a printing instruction needed to produce the target object in a 3D printer. The slicing of STL file using STL slicing algorithm is a key step in process planning, in which the contour data of the desired slicing level is obtained by assigning the triangular facets into each respective line segment. Assuming that the design is oriented in the z-direction, six different positional relationships between the slicing plane and the corresponding facet are observed as in Figure 2.9 and Table 2.1.

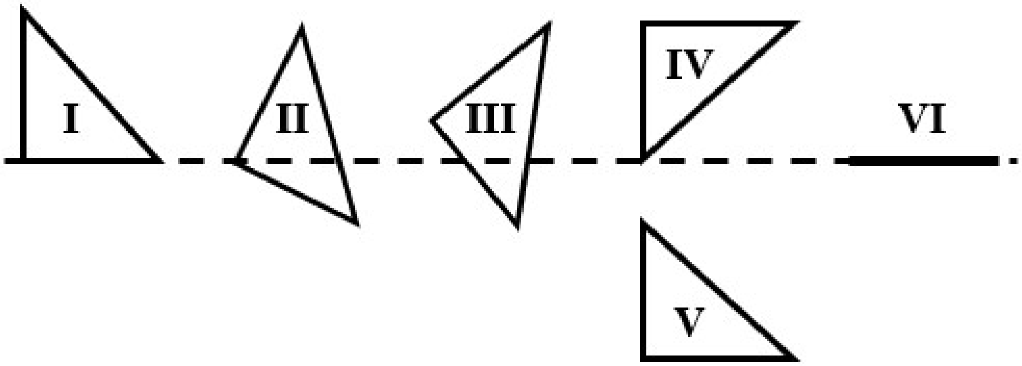


Figure 2.9 Possible cases of the facets with the slicing plane

The points generated for each case of interaction are 4, 3, 2, 0, 2, 6, respectively. Only case I, II, II are considered and the rest can be ignored. This is because case V will not produce a line segment while case VI is a redundancy, so it must be ignored to avoid overlapping segments. Lastly, it is obvious to ignore case IV as it does not intersect with any plane [44,45].

Table 2.1 Definition of each case

Case	Description
I	Two vertices of facet on the sliding plane
II	Plane bisecting through one vertex
III	Plane bisecting through two sides
IV	One vertex of facet on the sliding plane
V	No intersection
VI	Triangle facet on the plane

Once all points on each slicing plane are detected, lines of the slice data must be created. Two associated points are grouped together to form a line. One thing to keep in mind is that there can be overlapping lines due to facets sharing edges at the slicing plane. To define the contour, these repeating lines must be either removed or fused into a single line. If all representative lines that characterize the boundary of the design are obtained, line grouping is conducted to rearrange the vertices. This is to form a closed loop that define the hatch patch boundary needed for more efficient subtractive or additive processes [46].

3. Methodology: Optimization of the Process for Additive Manufacturing

In spite of its great potential, AM has many limiting factors to consider, even more when topology optimization is involved in the process. These factors, in most cases, do not prevent the manufacturing of the design since AM processes work in a layer-by-layer manner, but they do result in increased fabrication and clean-up costs. Hence, designers must always keep in mind the manufacturability while conceptualizing a design.

Choi et al. proposed a modern lattice generation methodology with descent efficiency, but they have left out the crux of the matter, the manufacturing performance. As multiple topology optimized building blocks are used to generate the lattice structure, the suggested design often ends up being AM unfriendly and requires significant amount of sacrificial structures to be manufactured.

This section discusses a method to impose AM constraints to Choi et al. lattice generation method to get solutions that are designed for AM.

3.1 Model Description

Including an external program or considering constraints after the optimization would not only spoil the optimum solution, but also diminish the effectiveness of the method. Thus, the simplest and most fitting way of considering additive manufacturing constraints in the work of Choi et al. is proposed. Instead of incorporating complicated and computationally expensive optimization algorithms, this approach replaces the building blocks with cellular structures and minimizes the need for supports structures.

The proposed methodology consists two parts: the generation of the lattice microstructure that will replace the infeasible building blocks and the selection and replacement of the current building blocks. In generating the optimum lattice microstructure, an investigation of the effects of the parameters is included. And for the selection of the building blocks that need to be replaced, the constraint elements in each building block are counted by inspecting its skeleton using an algorithm written in MATLAB software. Figure 3.1 shows the flow chart of the concurrent method with the key steps added.

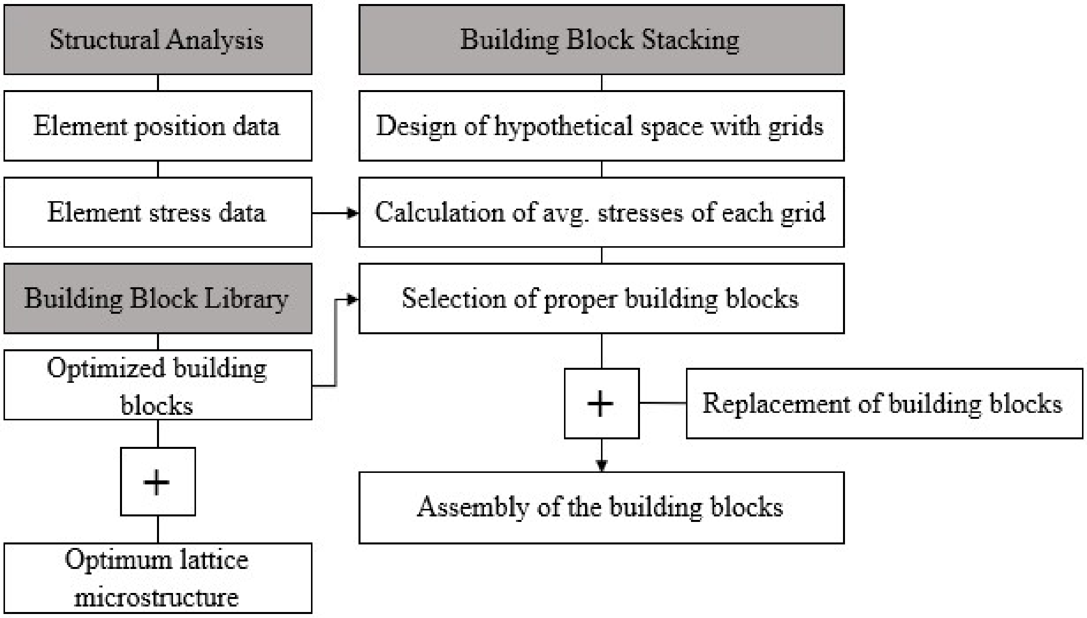

Structural Analysis
Building Block Stacking
Element position data
Design of hypothetical space with grids
Element stress data
Calculation of avg. stresses of each grid
Building Block Library
Selection of proper building blocks
Optimized building blocks
+
Replacement of building blocks
+
Assembly of the building blocks
Optimum lattice microstructure

Figure 3.1 Flow chart for concurrent method with added steps

3.2 Generation of microstructure unit cell

Topological shape optimization leads to the best shape possible, but, at the same time, it frequently outputs an unpredictable design with rough surfaces and multiple overhang angles due to not having any explicit or implicit restriction. On the other hand, lattice optimization, although having some restrictions, gives out a design with steady and AM friendly structure. In these aspects, lattices can be a better option to assure and improve the printing performances.

As many of the building blocks in the library seem infeasible without the use of support materials, a lattice microstructure that can satisfy diverse loading conditions is desired. Figure 3.2 is a fishbone diagram of adjustable variables in Inspire Altair commercial optimization software used for the topology optimization of the unit cells, that affect the structure of lattice. The highlighted variables are the variables considered in this study. Inspire provides lattice optimization function with various objectives: maximize stiffness, maximize frequency, and minimize mass. To be consistent with the work of Choi et al., optimization of lattice with minimize mass objective is preferred in this study. Other design variables such as target length, minimum and maximum diameters, and percentage of fill, must be also set in order to run the optimization. Depending on how these variables, including the boundary conditions and load, lattices with greatly varying constituents and safety factor can be obtained. For the optimal building block, the critical values of these parameters must be defined.

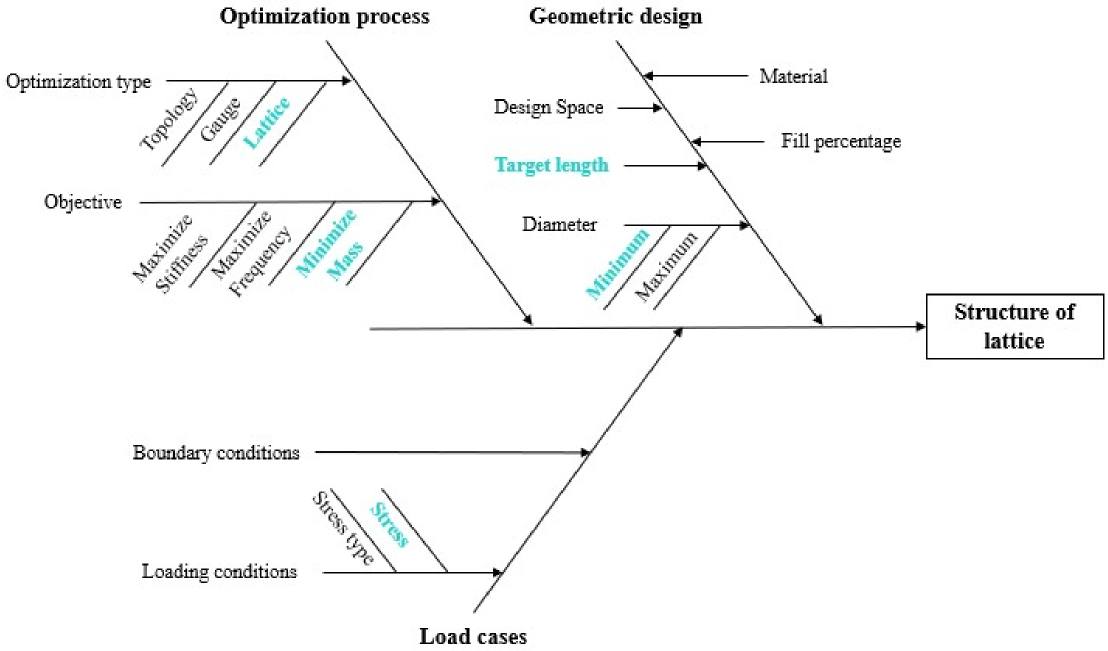


Figure 3.2 Flow chart for concurrent method with added steps

As a mean to determinate the influential parameters, a statistical analysis was conducted using the Minitab commercial tool. But first, data needed for the analysis were collected by calling forth the unit cells used for creating building blocks and running lattice optimization on them. A total of 245 models were optimized to collect the normal and shear stresses, the design variables, and the resulting minimum safety factors. In the process, some exceptions were made. First, the maximum and minimum diameters were not noted because these were coupled with the target length and automatically changed with it. Secondly, each model was optimized increasing the target length by 10 mm in a range of 10 mm to 90 mm. Increasing length was fixed at the tens place because lower values could lead to minimum diameters with 0.1 millimeter. At last, the design space was completely filled with lattice.

From the analysis, the influences of the parameters on the minimum safety factor were evaluated. Table 3.1 summarizes the results from ANOVA. It is observed that the p-values of the normal stresses in the x direction and the target length are less than 0.05 meaning that they have significant influences on the minimum safety factor. This is also inferred from the regression model

$$\begin{aligned} minSF = & -3.311 + 0.1063x + 0.14168\, L\; - 0.005816\, x \cdot L \\ & -0.004170\, y \cdot L + 0.001056\, x \cdot y \cdot L \end{aligned} \qquad \textbf{Equation 3.1}$$

where *x* and *y* are the normal stresses in x and y directions, respectively, and *TL* is the target length. The coefficients of the normal stress in x direction and the target length are relatively larger than that of the other parameters.

Table 3.1 ANOVA results for minimum safety factor

	DF	Adj SS	Adj MS	F-value	P-value
$\boldsymbol{\sigma_x}$	1	110.1	110.1	6.33	0.012
$\boldsymbol{\sigma_y}$	1	42.2	42.2	2.43	0.120
$\boldsymbol{\tau_{xy}}$	1	0.1	0.1	0.00	0.948
Target length	1	18136.6	18136.6	1043.08	0.000
Error	1730	30131.7	17.3		
Total	1745	51263.9			

S	R-sq	Adj R-sq	Predicted R-sq
4.16138	41.22%	41.05%	40.84%

However, this outcome does not match with simulation results of the lattice optimization. As shown in Figure 3.3, even when the normal stress differed with target length set at 70 mm, no or minimal changes in the structure occurred. Thus, it was concluded that the lattice depended only on the target length.

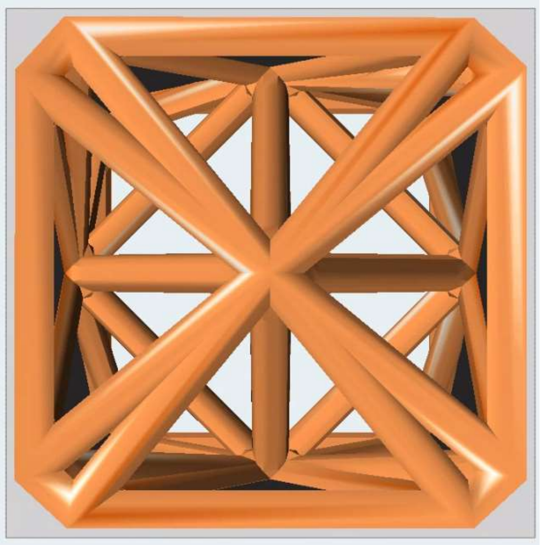
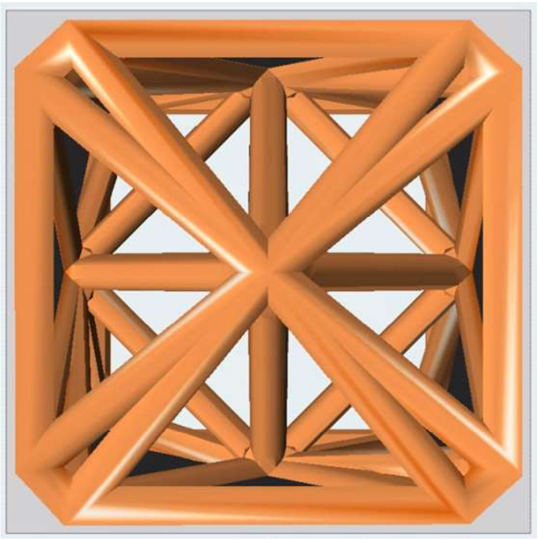
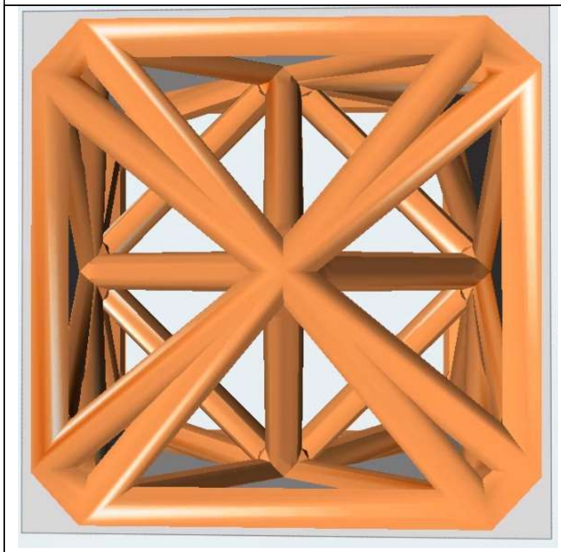
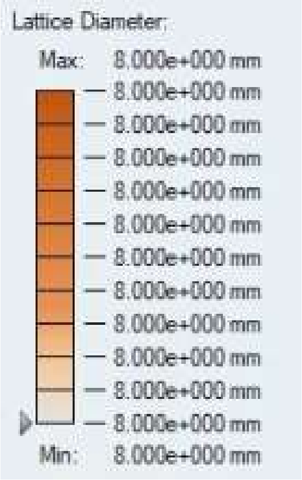
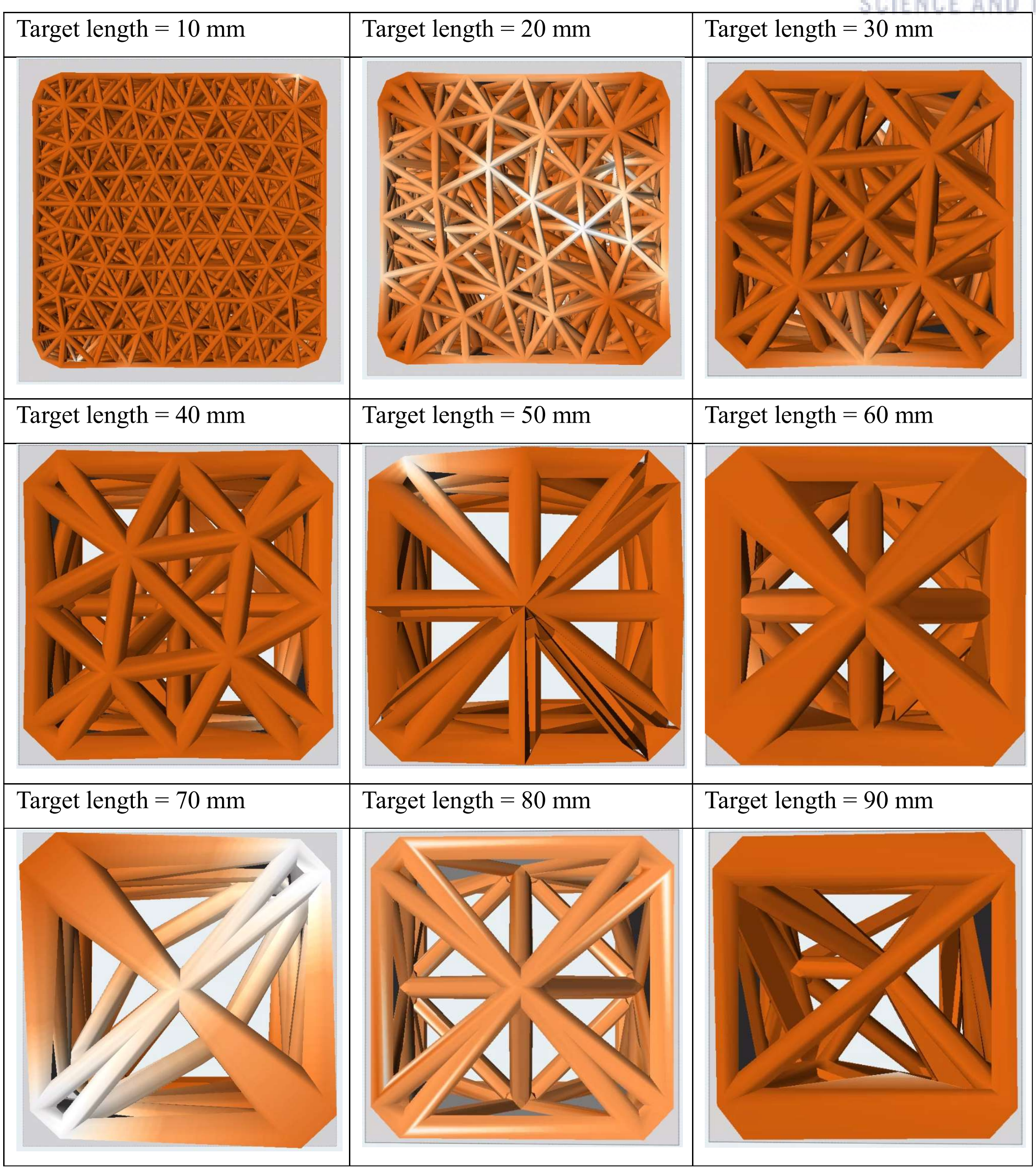
σ_x_σ_y_τ_{xy} = -2_0_0 (MPa)	σ_x_σ_y_τ_{xy} = 4_0_0 (MPa)
	
σ_x_σ_y_τ_{xy} = 8_0_0 (MPa)	Range of lattice diameter
	 Lattice Diameter: Max: 8.000e+000 mm 8.000e+000 mm 8.000e+000 mm 8.000e+000 mm 8.000e+000 mm 8.000e+000 mm 8.000e+000 mm 8.000e+000 mm 8.000e+000 mm 8.000e+000 mm 8.000e+000 mm 8.000e+000 mm Min: 8.000e+000 mm

Figure 3.3 Unit cubic cell optimized varying the normal stress

Based on this conclusion, the unit cubic cell was optimized once again varying the target length as before. Although the critical value of the target length came out as 90 mm from the statistical analysis, verification was needed. The resulting structures are shown in Figure 3.4



Target length = 10 mm

Target length = 20 mm

Target length = 30 mm

Target length = 40 mm

Target length = 50 mm

Target length = 60 mm

Target length = 70 mm

Target length = 80 mm

Target length = 90 mm

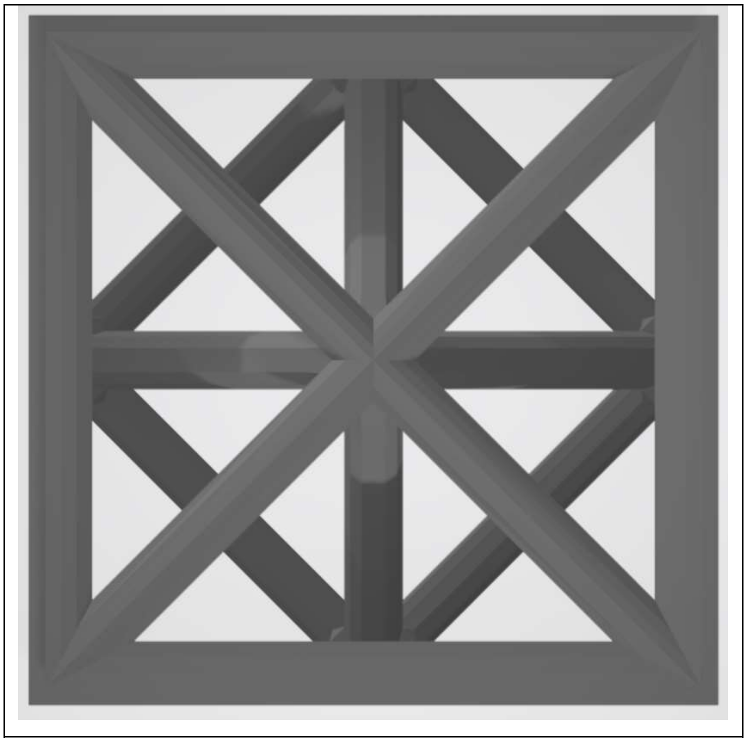
Figure 3.4 Unit cubic cell optimized varying the target length

The lattices with target length 80 mm and 90 mm had the most stable structures. Also, these were the only ones with minimum safety factors greater than 2. Between the two design, the one with 80mm length was selected over the one with 90mm because it contained more uniform elements and less components requiring support materials. All design variables for generating the optimal microstructure are given in Table 3.2.

Table 3.2 Details of the parameters used for optimizing optimal microstructure

Objective	Minimize Mass	
Parameter	Value	Unit
$\boldsymbol{\sigma_x}$	8	MPa
$\boldsymbol{\sigma_y}$	4	MPa
$\boldsymbol{\tau_{xy}}$	3	MPa
Target length	80	mm
Minimum diameter	8	mm
Maximum diameter	16	mm
Fill %	100	%
Cell size	100	mm

Finally, the obtained microstructure showed in Figure 3.5 is stored in the library in stereo-lithography (STL) format.



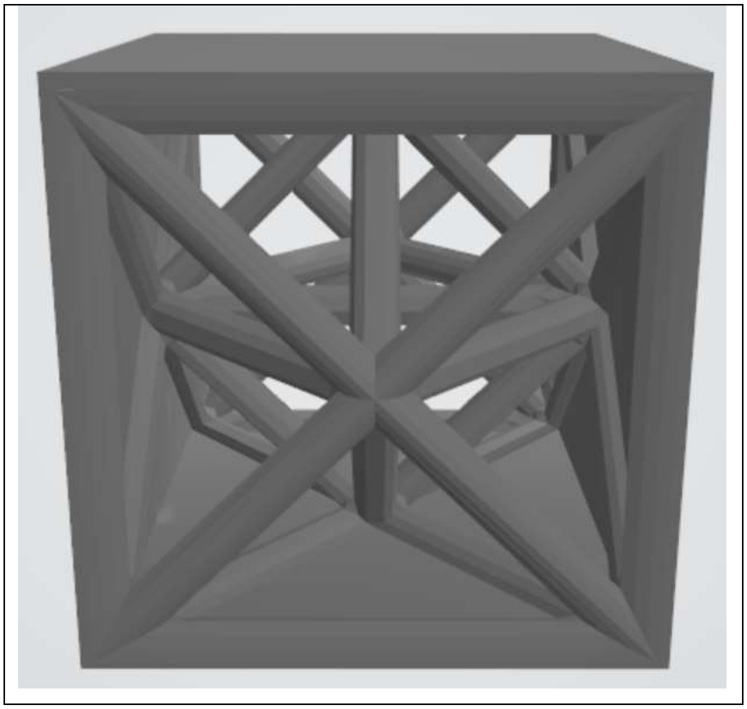


Figure 3.5 The front (top) and orthographic (bottom) view of the optimal microstructure

3.3 Optimization of the Process considering AM constraints

To effectively replace the building blocks subjected to additive manufacturing constraints, their layout must be evaluated. Unfortunately, these blocks are stored in the library as STL, the standard format for AM processes. STL is the precise approximated boundary representation of an object. However, this file only holds 3D surface mesh information, an extensive list of triangle facets with coordinates of three vertices and the normal oriented to the exterior of the solid, and not geometrical description of the domain. A representation of STL and its components is presented in Figure 3.6.

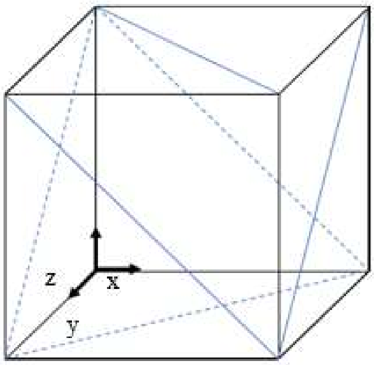


Figure 3.6 Triangulated STL representation of the surfaces of a simple cube shape

Direct use of STL file for analytical purpose is infeasible. Of course, methods of converting STL to other existing CAD format for finite element mesh (FEM) exist. Yet, this could result into a considerable increase of computational load and complexity, which is against the requirement of this study: to keep the model as simple as possible and computationally cheap. So instead, one of the crucial steps in process planning, STL slicing, is adopted. STL slice literally means the slicing of a STL file as shown in Figure 3.7. The information gathered from slicing the part's triangulated surfaces is generally used for tool path generation for each layer and conversion of the tool path to suitable data.

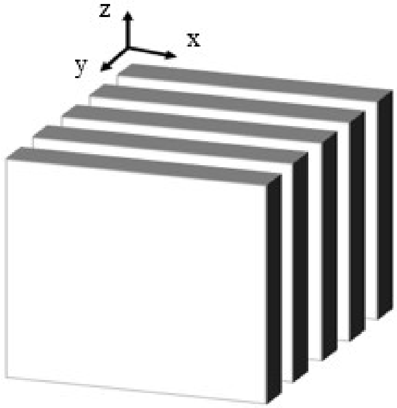
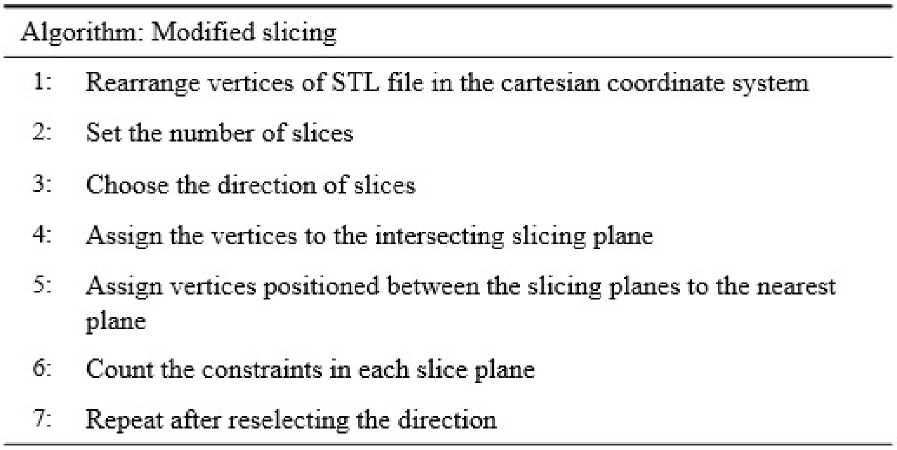


Fig 3.7 A sliced STL of a simple cube shape

In this case, the information will be used to generate the boundary of each building blocks sliced layer and consider the constraints. Because existing STL slicing algorithms are not available, a custom MATLAB code was developed. A summary of the steps taken in the algorithm is given in Fig 3.8.

The process starts by bringing the vertices from the STL file and rearranging them in the cartesian coordinate system. The normal vectors are omitted because graphical visualization of the solid body is not required. The space generated is then sliced using slicing planes. The vertices that touches these planes would be used to create the contour of the sliced building blocks. The gaps between adjacent slices are kept small to precise image of the cross sections. Some vertices which are positioned between the slicing planes are rounded, so that they are assigned to the nearest plane. Fig 3.9 shows the resulting cross sections of a building block.



```
Algorithm: Modified slicing
1:  Rearrange vertices of STL file in the cartesian coordinate system
2:  Set the number of slices
3:  Choose the direction of slices
4:  Assign the vertices to the intersecting slicing plane
5:  Assign vertices positioned between the slicing planes to the nearest
    plane
6:  Count the constraints in each slice plane
7:  Repeat after reselecting the direction
```

Fig 3.8 Slicing algorithm

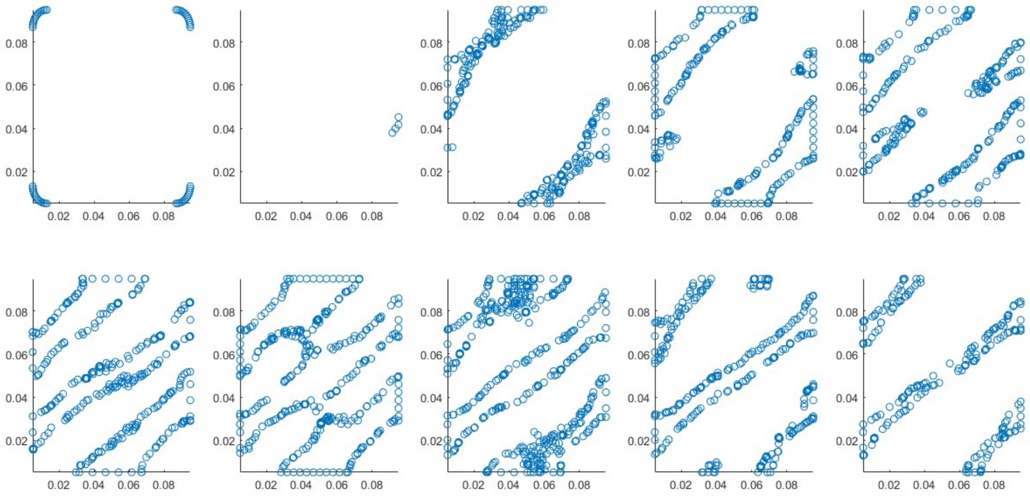


Figure 3.9 Cross sections of 2_1_-1.5 building block.

Now that the cross sections of each building block can be examined, the additive manufacturing constraints in each slice plane are located and counted. The most important constraints to be taken into account in this study are the overhang angles and the minimum cross-section parameter value that can be fabricated using AM machines. The angles of the struts and the lengths of the strut diameter at the boundaries of each cross section were calculated as shown in Figure 3.10. If the two values were lower than 35° and 1 mm, respectively, they would be counted as constraints. The mentioned steps are repeated by slicing the building block in different direction. Once found the constraints in each sliding plane, they are summed to obtain the representative constraints number of the building block.

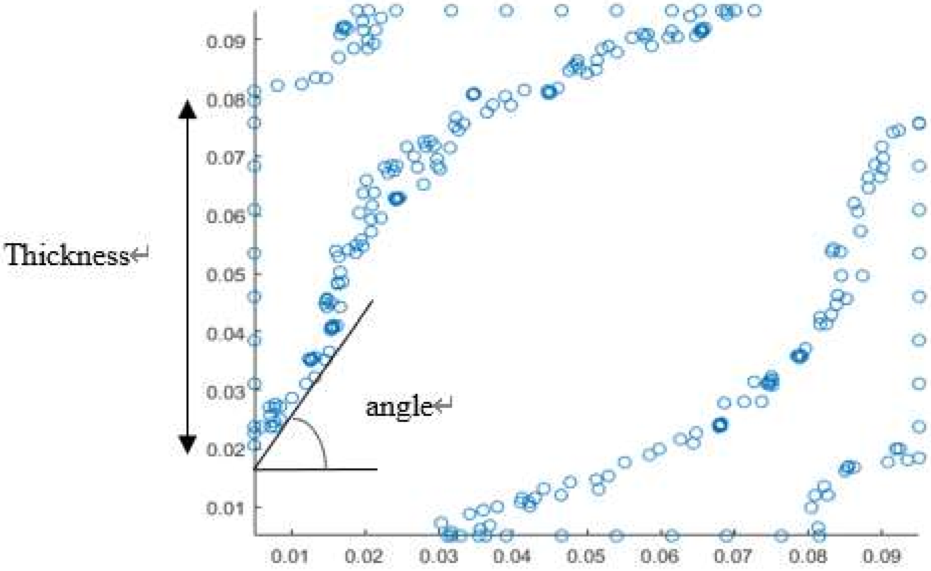


Figure 3.10 The overhang angle and the thickness of the strut

In order to determine which building block to replace, a critical value for the constraints number must be defined. Hence, the building blocks requiring support structures in Choi et al. library were manually identified and analyzed using the slicing algorithm to find their respective number of constraints. By plotting the calculated number of constraints, a right skewed distribution graph can be obtained as shown in Figure 3.11. Through statistical hypothesis testing, a critical value of 20 was set. If the number of constraints of a building block comes out to be greater than 20, then the corresponding building block would be regarded as an AM unfriendly structure and replaced with the microstructure designed in the previous section.

Specified procedures are added in between the selection and the assembly process of existing framework.

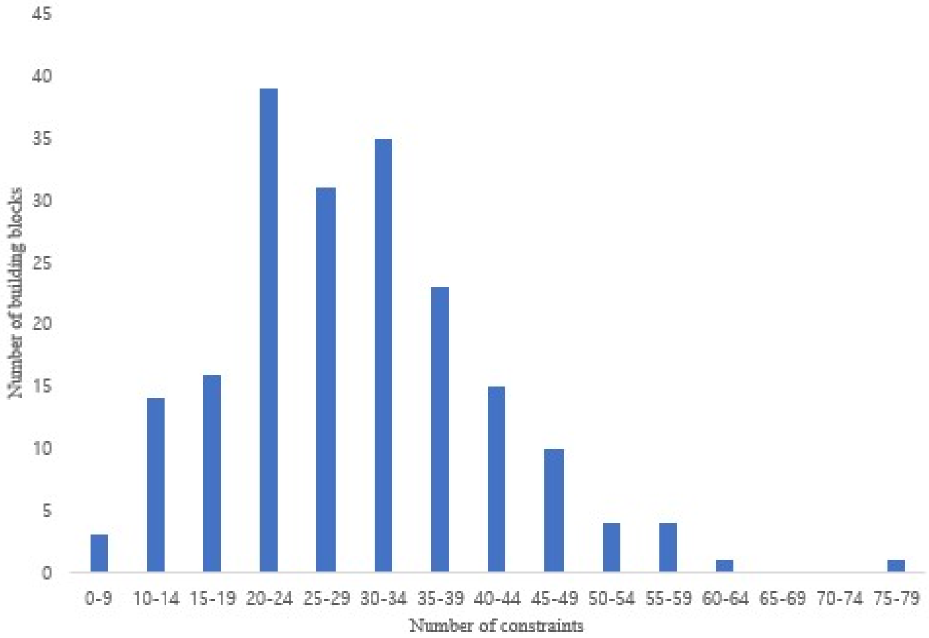


Figure 3.11 Right skewed distribution graph

4. Design and FE simulation

The details of methodology to consider the AM constraints have been described. In this section, numerical investigations are conducted to present the feasibility and the efficiency of the proposed method. Two examples are especially chosen for these purposes. The first example is a cantilever beam which has been used in studies by Chang and Rosen [10], Alzahrani et al. [4], and Gorguluarslan et al. [5]. By comparing the results from these studies with that obtained from the proposed method, computational effectiveness is shown. The second example is a simply supported beam with a distributed load previously adopted by Choi et al. [37]. This example was used to compare the mechanical performances of the generated structures with that of the conventional topology optimized structure. The resulting lattice structure of simply supported beam with and without AM constraints were additively printed on a fused deposition machine (FDM-type) to verify the manufacturability performances.

4.1 Design Strategy

4.1.1 Design of the Library

The building block library, one of the most essential components of this methodology, must be prepared. However, the library accessible in Choi et al. have been sorely developed to design their examples. Hence, supplementary stress conditions, in addition to the ones listed in Table 4.1, would be required.

Table 4.1 Stress conditions of Choi et al. library

$\boldsymbol{\sigma_x}$	8	4	2	0	-2	-4	-8
$\boldsymbol{\sigma_y}$	4	2	1	0	-1	-2	-4
$\boldsymbol{\tau_{xy}}$		3	1.5	0	-1.5	-3	

Unit: MPa

For convenience, the stress conditions of the building blocks needed in the examples were calculated by analyzing the FEA results in advance. The newly attained stress conditions are listed in Table 4.2.

The building blocks are built first by creating a cubic cell with an outer frame and fillets with a radius of 0.008 m in Altair INSPIRE, developed to perform structural optimization. Supports are applied as bounding conditions at each corner of the cubic cell. The model is then optimized using the Altair INSPIRE solver with the given loading conditions. The last step is to generate an STL file, which is the default format for AM processes. The process of building is presented in Figure 4.1.

Table 4.2 Added stress conditions

σ_x	-1	-0.5		0.5	1
σ_y		-0.5	0	0.5	
τ_{xy}				0.5	

Unit: MPa

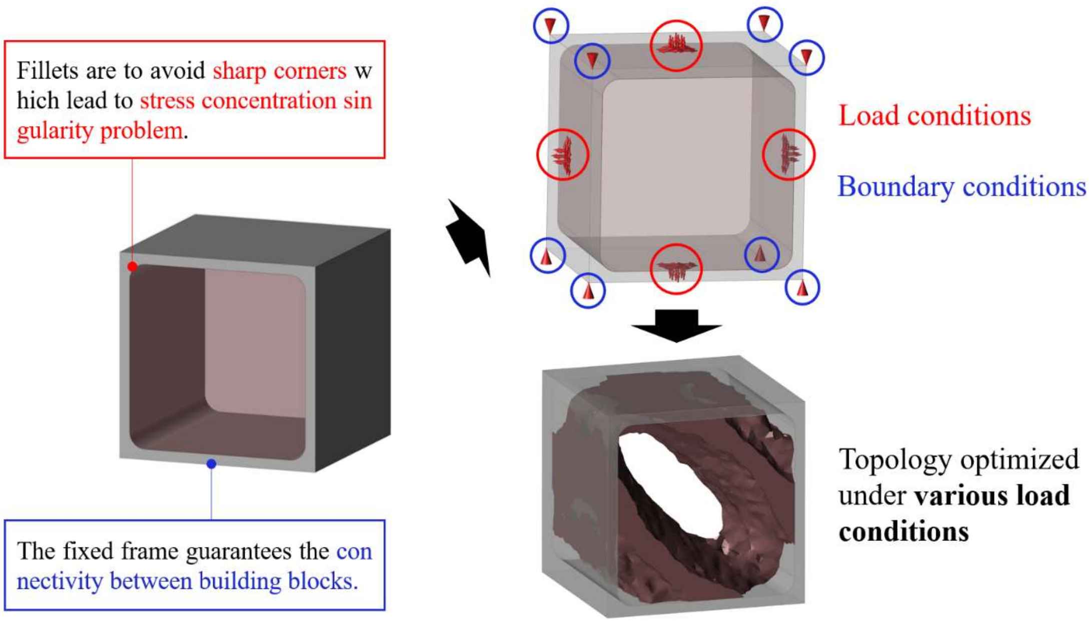

Fillets are to avoid sharp corners which lead to stress concentration singularity problem.
Load conditions
Boundary conditions
Topology optimized under various load conditions
The fixed frame guarantees the connectivity between building blocks.

Figure 4.1 Process of preparing building blocks

4.1.2 Case Study – Cantilever beam

A cantilever beam with geometric parameters of 50 mm in length, 20 mm in height and 10 mm in width is designed with lattice cells. This problem has importance as the optimization results with existing methods, i.e. SMS, RDM, and an optimization method with MFD algorithm can be obtained from works of Chang and Rosen [10], Alzahrani et al. [4], and Gorguluarslan et al. [5]. And compared with the results obtained. The conditions needed for the investigation are constants with those of existing studies. The cantilever beam is fixed at the back end and two downward load of 10 N is applied at the front end, as shown in Figure 4.2. It is assumed that the elastic modulus (E) of material used for the beam is 1960 MPa. A hypothetical space with the same dimensions with the cantilever beam is produced. This space is then discretized into $10 \times 10 \times 10\ \text{mm}^3$ brick elements. The resulting space has 10 grids as seen in Figure 4.3.

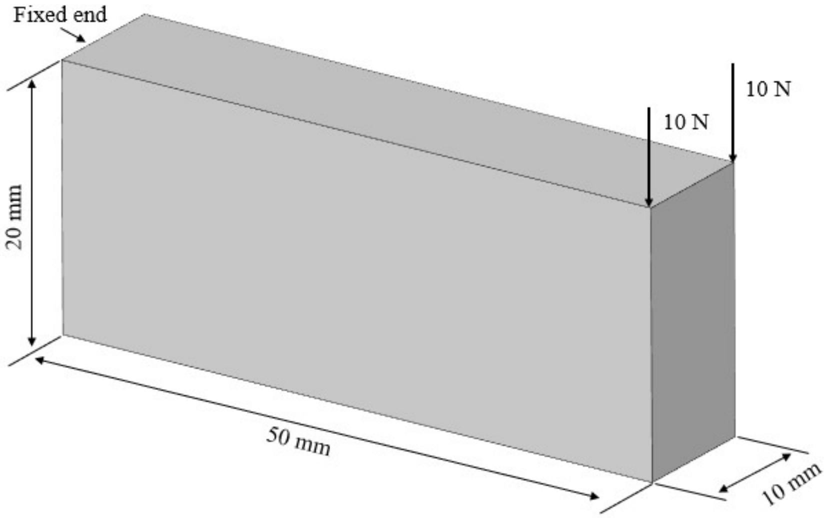


Figure 4.2 Cantilever beam example with loading conditions.

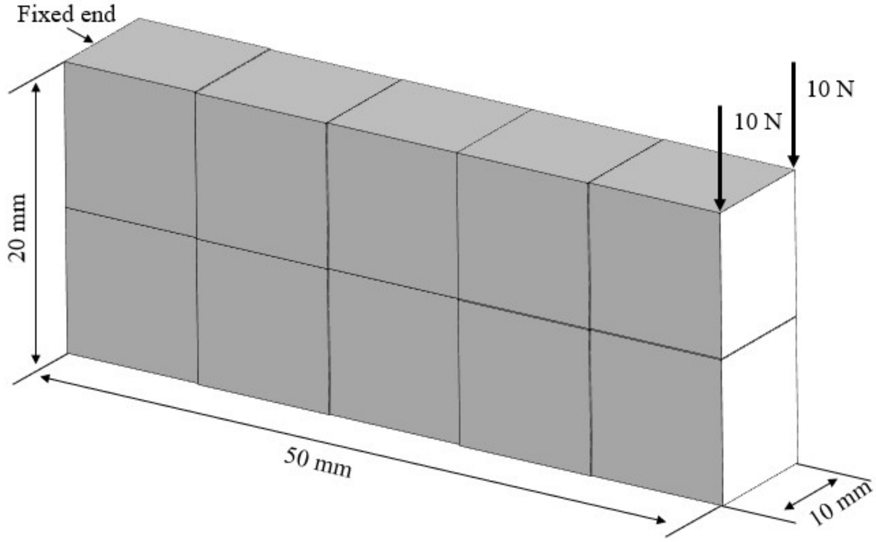


Figure 4.3 Gridded hypothetical geometry of cantilever beam

4.1.3 Case Study – Simply Supported Beam

A simply supported beam with distributed load is considered. The beam, having 220 mm length, 40 mm height and the width of 20 mm, is fixed at both ends by a 20 mm long plate and an uniform pressure with a loading domain of 20 mm is applied at the center top edge as seen in Figure 4.4. Design space is constructed with acrylonitrile butadiene styrene (ABS) material having a Young's Modulus of 2 GPa, a yield stress of 45 MPa and a density of 1.06 $\mathrm{kg/m^3}$.

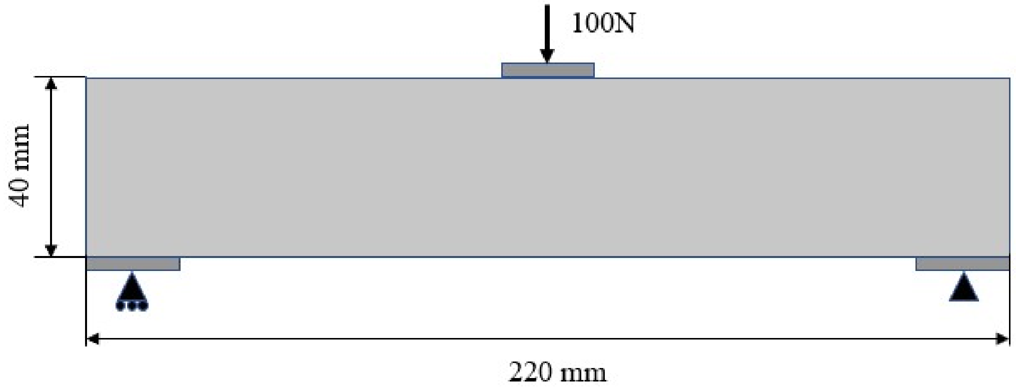


Figure 4.4 Simply supported beam with distributed load

Like the previous case, an empty geometry sharing the boundary dimensions of the original design is generated and meshed into 22 grids resulting in Figure 4.5.

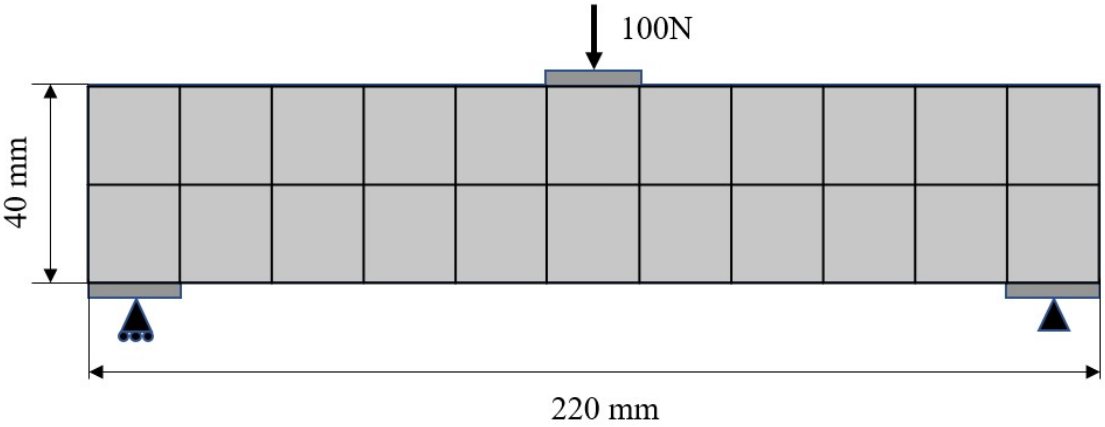
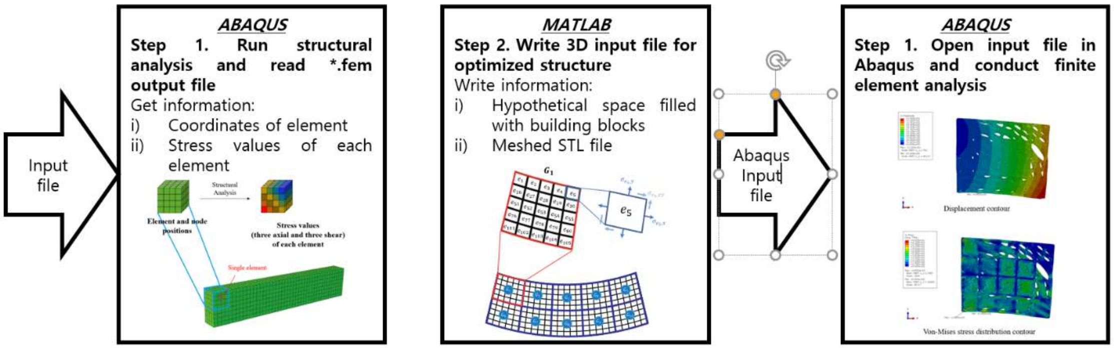


Figure 4.5 Gridded hypothetical geometry of simply supported beam

4.2 Results of the Generated Structures

4.2.1 Cantilever beam

To obtain the results presented here, a three-step Abaqus (a commercial software)-MATLAB interface has been used as illustrated in Figure 4.6. In step 1, a finite element (FE) model of the cantilever beam is modeled and meshed into 10000 elements. FE analysis is then invoked using this model to evaluate the stress information required as seen in Figure 4.7. In step 2, the MATLAB code read the structure information (i.e. stress values and node coordinates of each element) from the Abaqus input and output files. In each grid of the empty hypothetical space, 1000 elements with their corresponding stress information are placed and the average stresses were calculated. Based on the calculated values, the STL file of the lattice structure of the cantilever beam is generated. As STL file only contains 3D surface mesh information, it is converted into Abaqus input file using an open source MATLAB code. In step 3, the input file is opened in Abaqus for further FEA of the constructed lattice structure.


Input file

ABAQUS

Step 1. Run structural analysis and read *.fem output file

Get information:

i) Coordinates of element

ii) Stress values of each element

MATLAB

Step 2. Write 3D input file for optimized structure

Write information:

i) Hypothetical space filled with building blocks

ii) Meshed STL file

Abaqus Input file

ABAQUS

Step 1. Open input file in Abaqus and conduct finite element analysis

Structural Analysis
Element and node positions
Stress values (three axial and three shear) of each element
Single element
G_1
e_5
Displacement contour
Von-Mises stress distribution contour

Figure 4.6 Abaqus-MATLAB interface for generating the optimized lattice structure

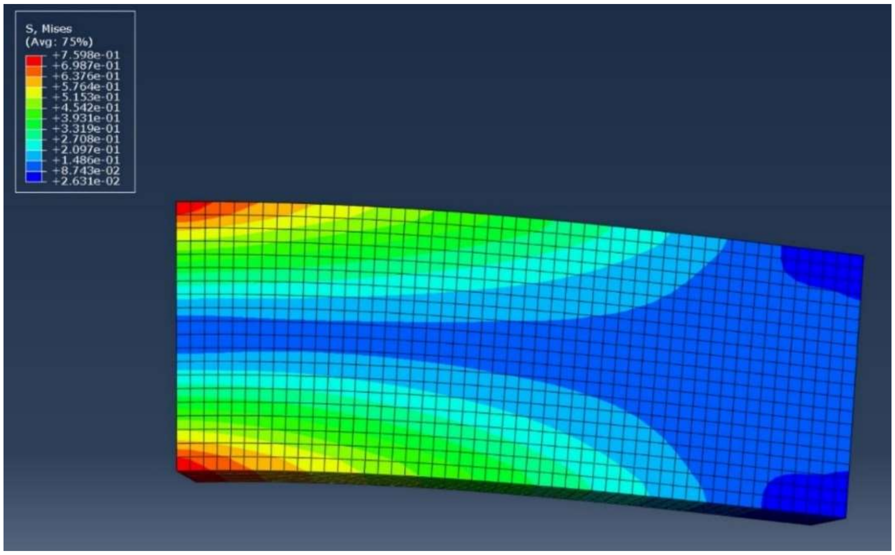


Figure 4.7 FEA result of the cantilever beam

One problem arises while generating the lattice structures. The calculated representative stresses of each grid are relatively small than the ones in the work of Choi et al., but since their library is specifically produced for their evaluation, appropriate building blocks are not available to generate the optimized structure of this study. Thus, a lattice structure with unnecessary elements and a worse performance is constructed as seen in Figure 4.8. To overcome this issue, extra set of stress conditions are used to optimize the new building blocks. The selected building blocks for each grid and the representative stresses are summarized in Table 4.3. The grid numbers represent the sequence of the grid, which is from the upper left corner to the lower right corner.

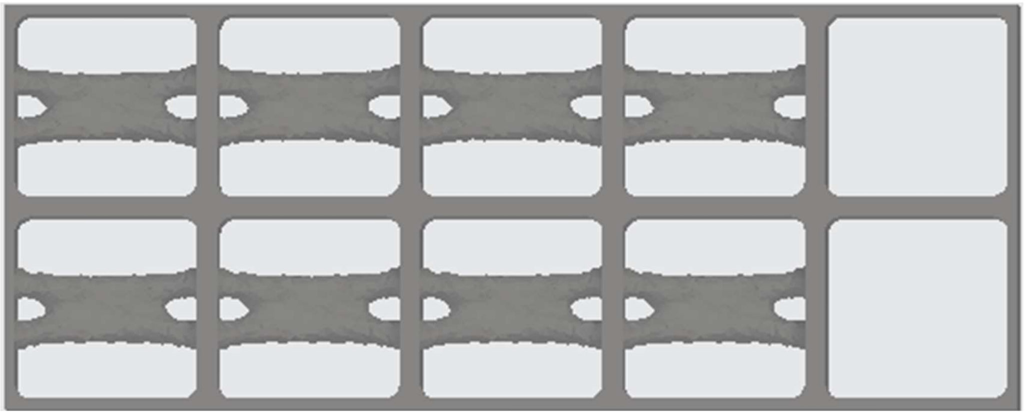


Figure 4.8 Front view of the lattice structure constructed using Choi et al. library

Table 4.3 Representative stresses of grid and selected building blocks

Grid	$\sigma_{G_n,x}$ (MPa)	$\sigma_{G_n,y}$ (MPa)	$\tau_{G_n,xy}$ (MPa)	Building Blocks (Cho et al.) $\sigma_{B,x}\_\sigma_{B,y}\_\tau_{B,xy}$	Building Blocks (Added) $\sigma_{B,x}\_\sigma_{B,y}\_\tau_{B,xy}$
1	0.336	0.0028	0.05	2_0_0	1_0.5_0.5
2	0.263	-0.00015	0.05	2_0_0	1_0_0.5
3	0.188	0	0.05	2_0_0	1_0_0.5
4	0.113	-0.00015	0.05	2_0_0	1_0_0.5
5	0.0404	-0.0028	0.05	0_0_0	0.5_-0.5_0.5
6	-0.336	-0.0028	0.05	-2_0_0	-1_-0.5_0.5
7	-0.263	0.00015	0.05	-2_0_0	-1_0_0.5
8	-0.188	0	0.05	-2_0_0	-1_0_0.5
9	-0.113	0.00015	0.05	-2_0_0	-1_0_0.5
10	-0.0404	0.00283	0.05	0	-0.5_0_0.5

Figure 4.9 shows the resulting lattice structure after the addition of the building block. The structure has a more comprehensive form with more elements.

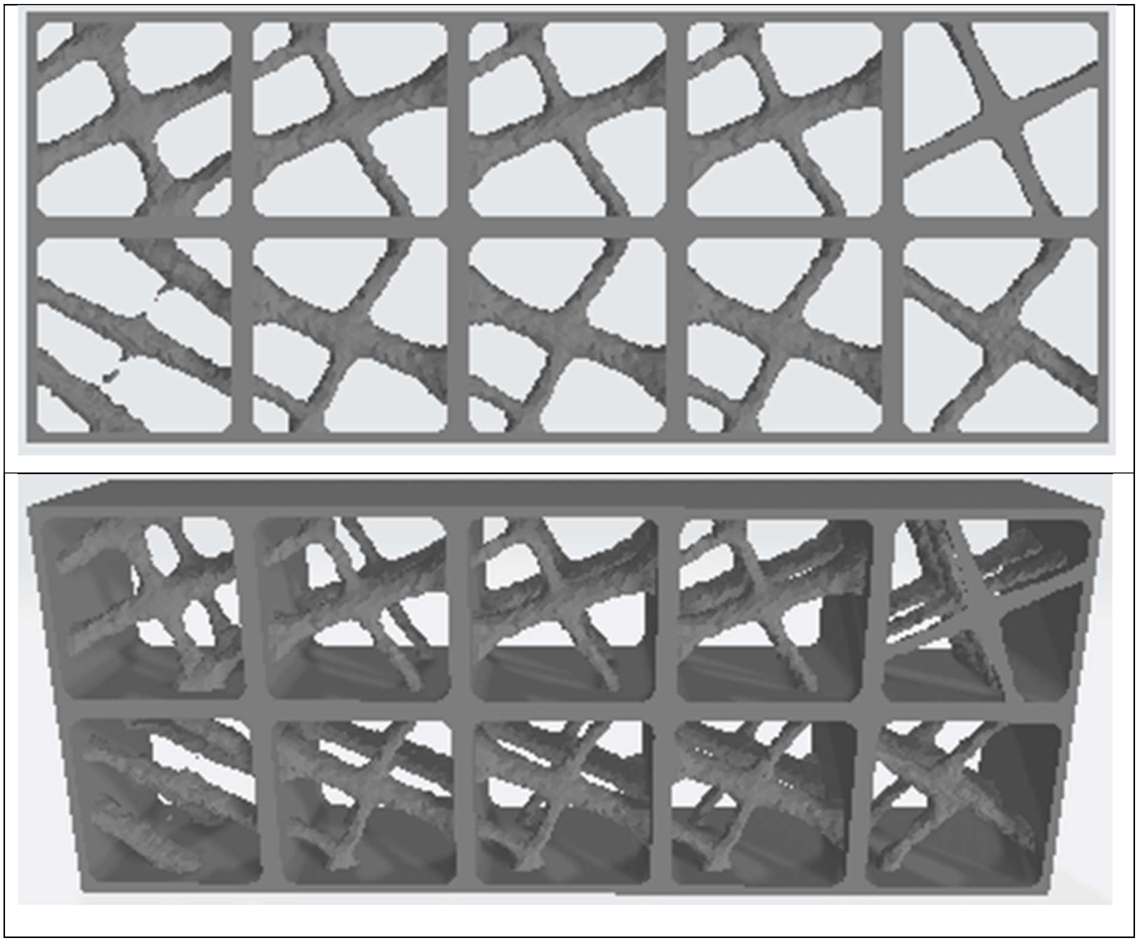


Figure 4.9 Lattice structure constructed using enhanced library

The selected building blocks are then analyzed for additive manufacturing constraints using the slicing algorithm. Building blocks in grid 1, 5 and 10 have number of constraints greater than the critical value. Therefore, they are substituted with the optimal microstructure resulting in an optimized structure in Figure 4.10. By comparing the Figure 4.9 and Figure 4.10, it is possible to tell that the building blocks that had more struts with overhang angles are replaced. The displacements and the optimization times are also obtained using a computer with 64Gb RAM. These results are compared with the results obtained from existing methods in Table 4.4. It is seen in Table 4.4 that a displacement result of 0.368 mm is obtained when proposed algorithm is used, which is lower than the result obtained using Choi et al. method. This result is also the lowest among the used methods. Also, the elapse times for Choi et al. and the proposed methods are 1.4 s to 1.5 s and 9.1 s, respectively. The optimization of the first optimized structure using the existing library takes shorter time due to the repeating building blocks. The computational time of Choi et al. is also lower than existing methods, as optimization takes only 1.5 s. However, computational time for the proposed method is a multiple of the Choi et al. This is obvious because the proposed method is an extension work of Choi et al, so it will require more computational effort for optimization. Still, it can be concluded that the proposed method does not drastically increase the computational time of the former method.

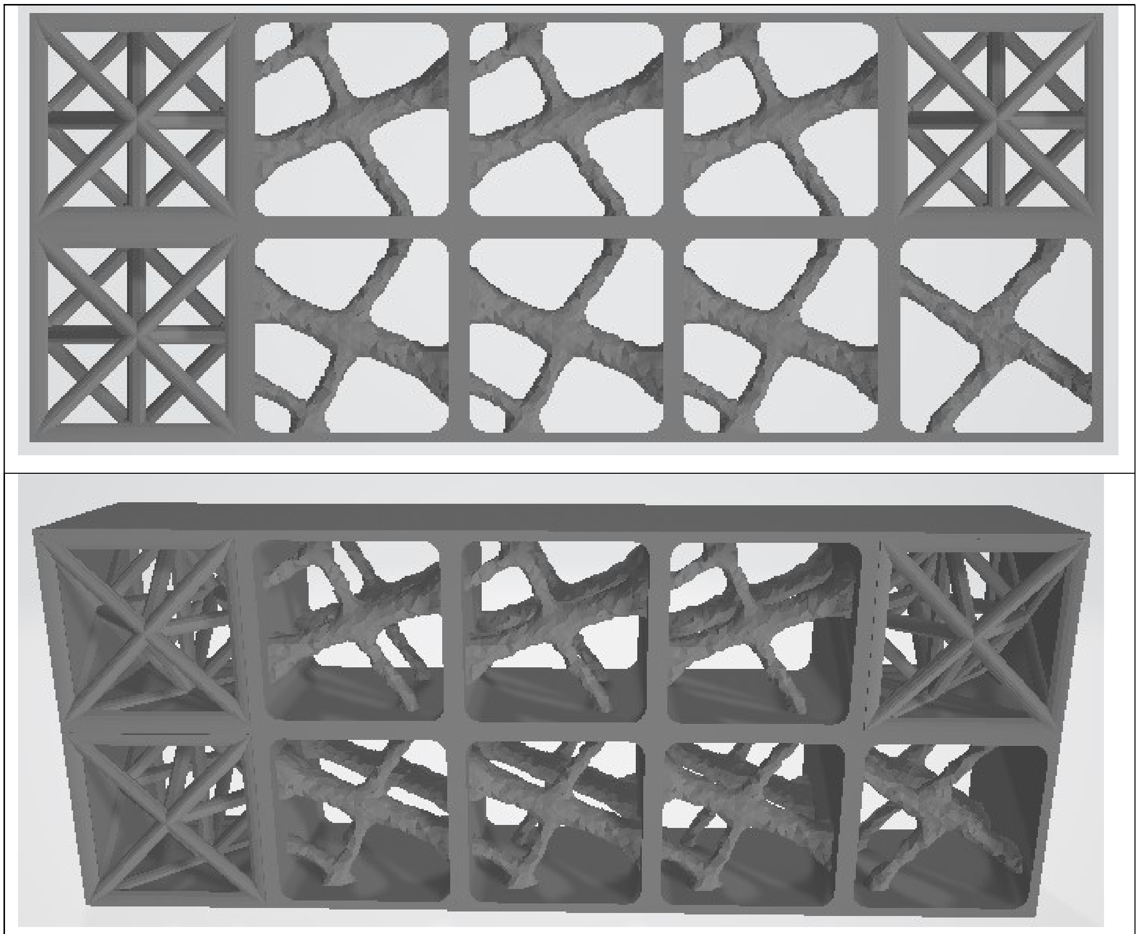


Figure 4.10 Lattice structure generated using proposed algorithm

Table 4.4 Comparison of the results of optimization methods

Optimization method	Displacement results (mm)	Time (s)
SQP	1.204	4.5
MFD	1.204	4.5
SQP – Phase 1	0.428	12
MFD – Phase 1	0.439	10
MFD – Phase 2	0.388	8

SMS [10]	0.573	3
RDM [4]	0.747	0
Choi et al.	0.407	1.4 – 1.5
Proposed method	0.368	9.1

Next, only the proposed approach is implemented for the following example, as its effectiveness compared to the existing methods has already been shown in this example.

4.2.2 Simply Supported Beam

The goal of this example is to investigate if the suggested design can have a comparable performance with that of Choi et al and the conventional topology optimization. The same steps as the previous case were taken, except that, for the FE model of simply supported beam, 1408 elements were used. The lattice structure obtained is compared with that of the other method as shown in Figure 4.11. As it is difficult to compare the two structures just by looking, numerical analysis is conducted. It should be noted that the design must be voxelized since it is in STL file format. The voxelization is done using an open source code in MATLAB and the FEA is performed in Abaqus. The results of the analysis are listed in Table 4.5. The stress distributions of the two lattice structures obtained from the analysis are also illustrated in Figure 4.12. The suggested lattice structure design shows similar stress distribution contour with that of Choi et al. However, there are some significant difference in values between the two structures.

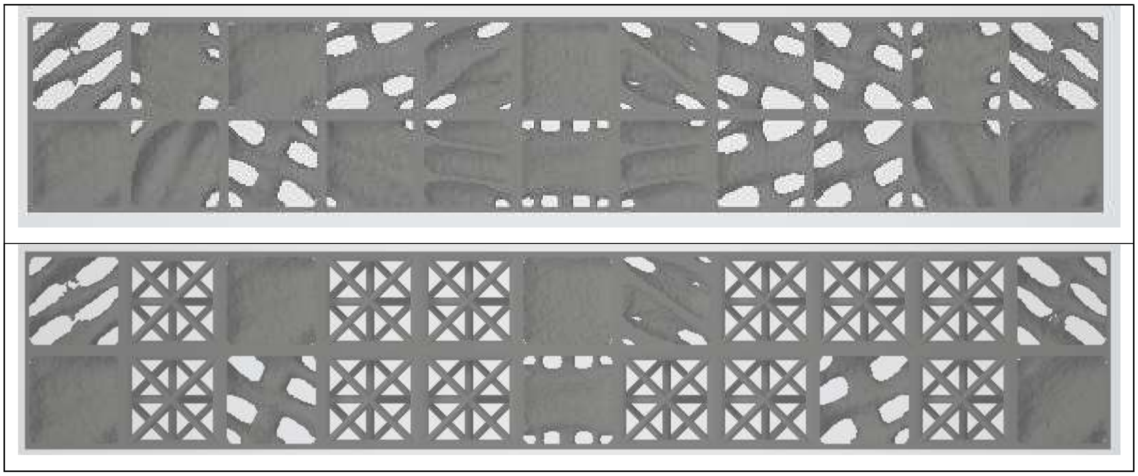


Figure 4.11 Comparison of the lattice structures with and without AM constraints

Table 4.5 Comparison of the mechanical performances of the generated lattice structures

	Weight [g]	Displacement results [mm]	Maximum stress [MPa]	CPU Time [seconds]
Original design	185.56	0.05147	5.687	-
Topology optimization	93.56	0.0786	2.627	4779
Choi et al.	94.51	0.1126	3.205	5.6
Proposed approach	73.9	0.2935	8.594	27.3

As seen in the Table 4.5, the optimized lattice-based structure has a displacement of 0.2935, which is larger than the displacement values of the structures from other methods. Furthermore, the maximum stress is almost three times of Choi et al. method. This drop of mechanical performance, which can be also inferred as a drop of stiffness, may be due to the reduction of weight, which was drastically decreased to 73.9 g. On the other hand, much like the results from previous example, a longer computational time is needed. This increase in time is again due to the incorporation of the slicing algorithm and the increased complexity of the problem.

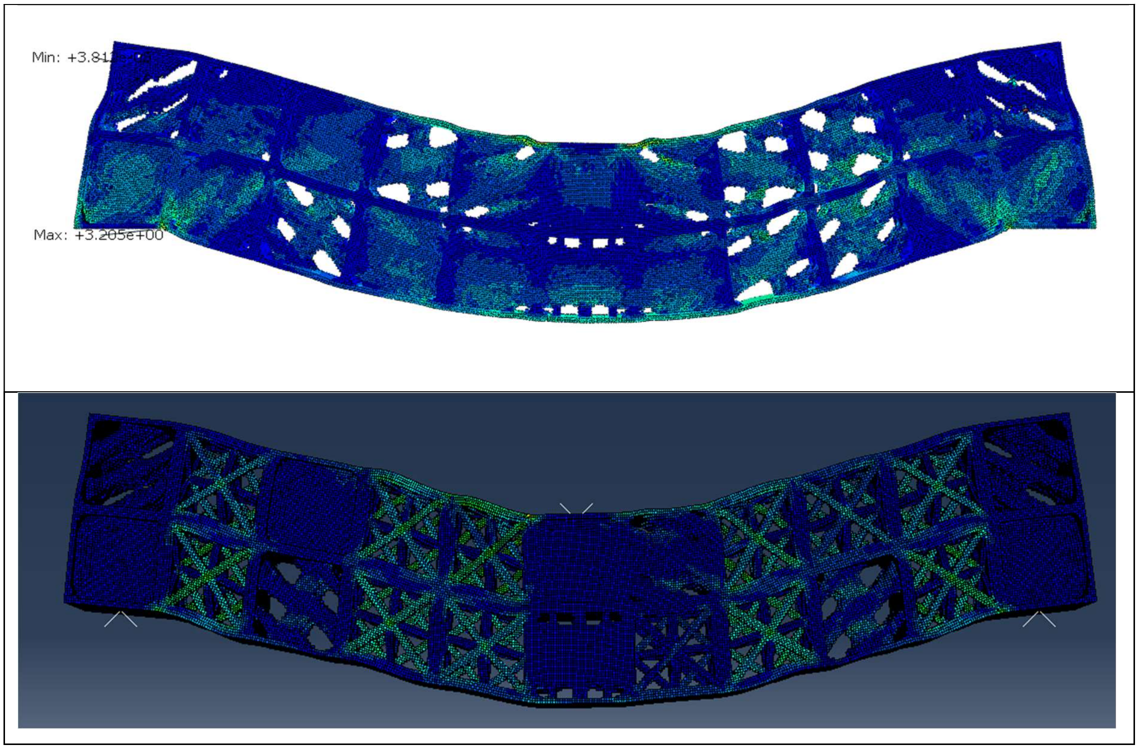


Figure 4.12 Stress distribution contour of the lattice structures with and without constraints

4.2.3 Manufacturing performances

The lattice structure of the cantilever beam example is printed with a limit orientation of 35 degree using a fused deposition machine (Stratasys uPrint). ABS (Acrylonitrile butadiene styrene) is set as the material. The same material, but in different form, is used for the support materials. Basic grid support is selected for the printing. Meshmixer, a support simulation tool, is also conducted to calculate the support material needed. For this simulation, a tree-like supports is preferred. The results are listed in Table 4.6.

Table 4.6 Results showing printing performances

Method	Model volume [cm^3]	Support volume [cm^3]	Simulated support volume [cm^3]	Total volume [cm^3]	Printing time [minute]
Choi et al.	2.912	5.981	2.457	8.893	114
Proposed method	3.158	5.484	2.163	8.642	122

According to the table, the design volume has increased leading to a longer printing time. At the same time, the support material in the suggested cantilever design is lower than that of Choi et al. However, the decrease of support volume is not significant. The reason behind this result is because of the fixed frame, which was set as a requirement when generating the unit cubic cell to guarantee the connectivity. The use of supports cannot be avoided due to this constraint. In the Figure 4.13, the regions that need support materials are highlighted in red.

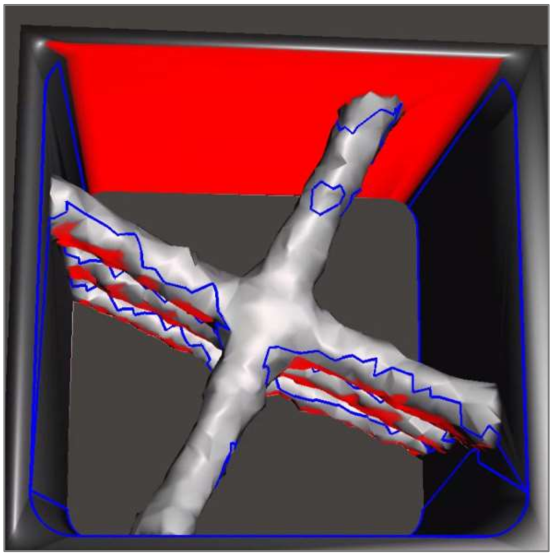


Figure 4.13 Regions needing support materials in a building block

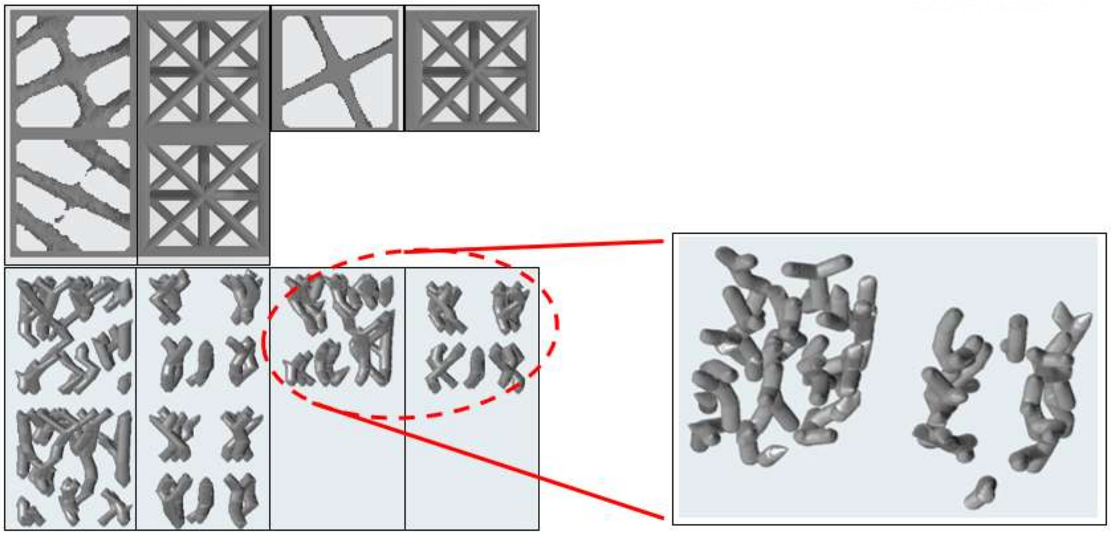


Figure 4.14 Decrease of inner support materials in replaced building blocks

Still, through suggested method, a decrease of inner support structures that lead to an increase of preprocessing is observed in the Figure 4.14.

This is clearer when calculating the manufacturing costs of printing both lattice structures. The results are summarized in Table 4.7. Even when printing a small scale design, the post processing cost has decreased significantly.

Table 4.7 Manufacturing cost

Machine		Stratasys uPrint (FDM)	Choi et al.		Suggested design	
		Unit Cost	Quantity	Sub Total	Quantity	Sub Total
Material	ABS	460.76 ₩/cc	2.912 cc	1,342 ₩	3.158 cc	1,455 ₩
	Support	449.13 ₩/cc	5.981 cc	2,687 ₩	5.484 cc	2,463 ₩
Processes	Printing	10,000 ₩/hour	114 min	19,000 ₩	122 min	20,334 ₩
	Post-processing	30,000 ₩/hour	117 min	58,500 ₩	107 min	53,500 ₩
Total			81,529 ₩		77,752 ₩	

5. Conclusion

In this thesis, an improved Choi et al lattice design framework that considers the geometric parameter values that deteriorate the manufacturing performances of AM is developed. To retain the simplicity of design framework, changes in the process are kept to a minimum. A custom-built STL slice algorithm is integrated into the selection process with the goal of directly inspecting the building blocks for AM manufacturing constraints. In the first phase, an optimal lattice microstructure is generated by lattice optimizing the unit cubit cell in the commercial tool INSPIRE. The parameters needed for the optimization are defined through ANOVA. In the second phase, analysis of the selected building blocks is conducted using the STL slice algorithm. Building blocks with constraints number smaller than the pre-determined critical value are replaced with the prepared structure.

Two numerical examples are used to show the robustness and efficiency of the proposed strategy. The first example, a cantilever beam example, is used to show that the proposed framework can achieve minimal computational cost even when considering the manufacturing constraints. It is shown that the proposed method has worse performance compared to alternative existing methods. Still the gap is not significant. In addition, taking into account the fact that the optimization is conducted with limited settings than the one used in the existing methods, the results can be seen as comparable. The second example is used to investigate the mechanical performances of the suggested lattice structure. Even though a lower stiffness with higher maximum Von-Mises stress are obtained, a significant weight reduction is achieved. Moreover, the computational efficiency is not hardly dropped even with more complex application. At last but not least, the resulting lattice structures of the cantilever beam example were additively printed on a FDM printer to verify the manufacturability performances. Even though the proposed method could not significantly reduce support volume, it still gives a design with less inner support structure which can positively affect the post processing.

5.1 Limitations and Future works

The proposed methodology has been developed with several assumptions, which leads to some limitations. Couple of the parameters couldn't be adjusted. For instance, final volume of the lattice structure couldn't be adjusted. Also, the substituting building block hasn't been designed to be fully supportless. Lastly, assumptions or fundamental rules for multi-scale optimization haven't been covered. Taking these into account, this work must be further developed as follows.

- The suggested design from proposed method must be further validated through mechanical testing.
- Real-world application of the lattice structure optimization should be done.
- Refinement of the building block library is required.

Reference

1. Vayre, B., Vignat, F., & Villeneuve, F. (2012). Designing for additive manufacturing. *Procedia ClrP, 3,* 632-637.
2. Ranjan, R., Samant, R., & Anand, S. (2017). Integration of design for manufacturing methods with topology optimization in additive manufacturing. *Journal of Manufacturing Science and Engineering, 139*(6), 061007.
3. Thompson, M. K., Moroni, G., Vaneker, T., Fadel, G., Campbell, R. I., Gibson, I., ... & Martina, F. (2016). Design for Additive Manufacturing: Trends, opportunities, considerations, and constraints. *CIRP annals*, *65*(2), 737-760.
4. Alzahrani, M., Choi, S. K., & Rosen, D. W. (2015). Design of truss-like cellular structures using relative density mapping method. *Materials & Design, 85,* 349-360.
5. Gorguluarslan, R. M., Gandhi, U. N., Song, Y., & Choi, S. K. (2017). An improved lattice structure design optimization framework considering additive manufacturing constraints. *Rapid Prototyping Journal*, *23*(2), 305-319.
6. Cheng, L., Zhang, P., Biyikli, E., Bai, J., Robbins, J., & To, A. (2017). Efficient design optimization of variable-density cellular structures for additive manufacturing: theory and experimental validation. *Rapid Prototyping Journal*, *23*(4), 660-677.
7. Nguyen, J., Park, S. I., & Rosen, D. (2013). Heuristic optimization method for cellular structure design of light weight components. *International Journal of Precision Engineering and Manufacturing*, *14*(6), 1071-1078.
8. Wang, C., Zhu, J., Zhang, W. H., Li, S. Y., & Kong, J. (2018). Concurrent topology optimization design of structures and non-uniform parameterized lattice microstructures. *Structural and Multidisciplinary Optimization, 58*(1), 35-50.
9. Brackett, D., Ashcroft, I., & Hague, R. (2011, August). Topology optimization for additive manufacturing. In *Proceedings of the solid freeform fabrication symposium, Austin, TX* (Vol. 1, pp. 348-362). S.
10. Chang, P. S., & Rosen, D. W. (2013). The size matching and scaling method: a synthesis method for the design of mesoscale cellular structures. *International Journal of Computer Integrated Manufacturing*, *26*(10), 907-927.
11. Tang, Y., Dong, G., Zhou, Q., & Zhao, Y. F. (2017). Lattice structure design and optimization with additive manufacturing constraints. *IEEE Transactions on Automation Science and Engineering*, *15*(4), 1546-1562.
12. Maconachie, T., Leary, M., Lozanovski, B., Zhang, X., Qian, M., Faruque, O., & Brandt, M. (2019). SLM lattice structures: Properties, performance, applications and challenges. *Materials & Design*, 108137.

13. Brandt, M. (Ed.). (2016). *Laser Additive Manufacturing: Materials, Design, Technologies, and Applications*. Woodhead Publishing.

14. Xiao, Z., Yang, Y., Xiao, R., Bai, Y., Song, C., & Wang, D. (2018). Evaluation of topology-optimized lattice structures manufactured via selective laser meling. *Materials & Design, 143,* 27-37.

15. Xu, S., Shen, J., zhou, S., Huang, X., & Xie, Y. M. (2016). Design of lattice structures with controlled anisotropy. *Materials & Design, 93,* 443-447.

16. Deshpande, V. S., Fleck, N. A., & Ashby, M. F. (2001). Effective properties of the octet-truss lattice material. *Journal of the Mechanics and Physics of Solids*, *49*(8), 1747-1769.

17. Abueidda, D. W., Bakir, M., Al-Rub, R. K. A., Bergström, J. S., Sobh, N. A., & Jasiuk, I. (2017). Mechanical properties of 3D printed polymeric cellular materials with triply periodic minimal surface architectures. *Materials & Design*, *122*, 255-267.

18. Yoo, D. J. (2014). Recent trends and challenges in computer-aided design of additive manufacturing-based biomimetic scaffolds and bioartificial organs. *International journal of precision engineering and manufacturing*, *15*(10), 2205-2217.

19. Maskery, I., Aboulkhair, N. T., Aremu, A. O., Tuck, C. J., & Ashcroft, I.A. (2017). Compressive failure modes and energy absorption in additively manufactured double gyroid lattices. *Additive Manufacturing, 16*, 24-29.

20. Al-Ketan, O., Rowshan, R., & Al-Rub, R. K. A. (2018). Topology-mechanical property relationship of 3D printed strut, skeletal, and sheet based periodic metallic cellular materials. *Additive Manufacturing, 19,* 167-183.

21. Rosen, D. W. (2014). Research supporting principles for design for additive manufacturing: This paper provides a comprehensive review on current design principles and strategies for AM. *Virtual and Physical Prototyping*, *9*(4), 225-232.

22. Booth, J. W., Alperovich, J., Chawla, P., Ma, J., Reid, T. N., & Ramani, K. (2017). The design for additive manufacturing worksheet. *Journal of Mechanical Design*, *139*(10), 100904.

23. Chu, C., Graf, G., & Rosen, D. W. (2008). Design for additive manufacturing of cellular structures. *Computer-Aided Design and Applications*, *5*(5), 686-696.

24. Rosen, D. W. (2007). Computer-aided design for additive manufacturing of cellular structures. *Computer-Aided Design and Applications*, *4*(5), 585-594.

25. Rosen, D. W. (2007, August). Design for additive manufacturing: a method to explore unexplored regions of the design space. In *Eighteenth Annual Solid Freeform Fabrication Symposium* (pp. 402-415). Austin, TX: University of Texas at Austin (freeform).

26. Mueller, B. (2012). Additive manufacturing technologies–Rapid prototyping to direct digital manufacturing. *Assembly Automation*, *32*(2).

27. Seepersad, C. C. (2014). Challenges and opportunities in design for additive manufacturing. *3D*

printing and Additive Manufacturing, *1*(1), 10-13.

28. Ponche, R., Kerbrat, O., Mognol, P., & Hascoet, J. Y. (2014). A novel methodology of design for Additive Manufacturing applied to Additive Laser Manufacturing process. *Robotics and Computer-Integrated Manufacturing*, *30*(4), 389-398.

29. Harrysson, O. L., Cansizoglu, O., Marcellin-Little, D. J., Cormier, D. R., & West II, H. A. (2008). Direct metal fabrication of titanium implants with tailored materials and mechanical properties using electron beam melting technology. *Materials Science and Engineering: C*, *28*(3), 366-373.

30. Wang, S. (2007). *Krylov subspace methods for toplogy optimization on adaptive meshes.*

31. Chen, W., Tong, L., & Liu, S. (2017). Concurrent topology design of structure and material using a two-scale topology optimization. *Computers & Structures*, *178*, 119-128.

32. Xia, L., & Breitkopf, P. (2014). Concurrent topology optimization design of material and structure within FE2 nonlinear multiscale analysis framework. *Computer Methods in Applied Mechanics and Engineering*, *278*, 524-542.

33. Xia, L., & Breitkopf, P. (2015). Multiscale structural topology optimization with an approximate constitutive model for local material microstructure. *Computer Methods in Applied Mechanics and Engineering*, *286*, 147-167.

34. Dorn, W. (1964). Automatic design of optimal structures. *J. de Mecanique*, *3*, 25-52.

35. Chu, J., Engelbrecht, S., Graf, G., & Rosen, D. W. (2010). A comparison of synthesis methods for cellular structures with application to additive manufacturing. *Rapid Prototyping Journal*, *16*(4), 275-283.

36. Wieding, J., Wolf, A., & Bader, R. (2014). Numerical optimization of open-porous bone scaffold structures to match the elastic properties of human cortical bone. *Journal of the mechanical behavior of biomedical materials*, *37*, 56-68.

37. Choi, H. J., Chung Baek, A. C., & Kim, N. H. (2019, August). Design of non-periodic lattice structures by allocating pre-optimized building blocks. In *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection.

38. Mirzendehdel, A. M., & Suresh, K. (2016). Support structure constrained topology optimization for additive manufacturing. *Computer-Aided Design*, *81*, 1-13.

39. Wu, J., Wang, C. C., Zhang, X., & Westermann, R. (2016). Self-supporting rhombic infill structures for additive manufacturing. *Computer-Aided Design*, 80, 32-42.

40. Hu. K., Jin, S., & Wang, C. C. (2015). Support slimming for single material based additive manufacturing. *Computer-Aided Design, 65*, 1-10.

41. Hu. K., Zhang, X., & Wang, C. C. (2015, August). Direct computation of minimal rotation for support slimming. In *2015 IEEE International Coference on Automation Science and*

Engineering (CASE) (pp. 936-941). IEEE.

42. Thomas, D. (2009). *The development of design rules for selective laser meling* (Doctoral dissertation, University of Wales).
43. Hu, J. (2017). Study on STL-based slicing process for 3D printing.*Solid Freeform.*
44. Adnan, F.A., Romlay, F. R. M., & Shafiq, M. (2018, April). Real-time slicing algorithm for Stereolithography (STL) CAD model applied in additive manufacturing industry. In *IOP Conference Series: Materials Science and Engineering* (Vol. 342, No. 1, p.012016). IOP Publishing.
45. Brown, A. C., & De Beer, D. (2013, September). Development of a stereolithography (STL) slicing and G-code generation algorithm for an entry level 3-D printer. In *2013 Africon* (pp. 1-5). IEEE.
46. Topçu, O., Taşcıoğlu, Y., & Ünver, H. Ö. (2011, May). A method for slicing CAD models in binary STL format. In *6th International Advanced Technologies Symposium (IATS'11)* (Vol. 163, pp. 141-145).

Appendix A: Constraints number of building blocks

Table A.1 Constraints number of building blocks requiring support structures

Model name	Number of constraints	Model name	Number of constraints
0_1_1.5.stl	21	2_1_-1.5.stl	13
0_1_-1.5.stl	21	2_-1_1.5.stl	27
0_-1_1.5.stl	14	2_-1_-1.5.stl	33
0_-1_-1.5.stl	12	-2_1_1.5.stl	29
0_2_0.stl	21	-2_1_-1.5.stl	21
0_-2_0.stl	12	-2_-1_1.5.stl	36
0_2_1.5.stl	21	-2_-1_-1.5.stl	16
0_2_-1.5.stl	25	2_1_3.stl	11
0_-2_1.5.stl	12	2_1_-3.stl	23
0_-2_-1.5.stl	10	-2_-1_3.stl	32
0_4_0.stl	25	-2_-1_-3.stl	11
0_-4_0.stl	26	2_2_0.stl	20
0_4_1.5.stl	21	2_-2_0.stl	24
0_4_-1.5.stl	39	-2_2_0.stl	11
0_-4_1.5.stl	39	-2_-2_0.stl	15
0_-4_-1.5.stl	21	2_2_1.5.stl	20
0_4_3.stl	24	2_2_-1.5.stl	19
0_4_-3.stl	40	2_-2_1.5.stl	30
0_-4_3.stl	40	2_-2_-1.5.stl	26
0_-4_-3.stl	24	-2_2_1.5.stl	19
2_0_0.stl	13	-2_2_-1.5.stl	29
-2_0_0.stl	13	-2_-2_1.5.stl	31
2_0_1.5.stl	21	-2_-2_-1.5.stl	8
2_0_-1.5.stl	19	2_2_3.stl	14
-2_0_1.5.stl	36	2_2_-3.stl	40
-2_0_-1.5.stl	21	2_-2_3.stl	20
2_1_0.stl	23	2_-2_-3.stl	20
2_-1_0.stl	21	2_4_0.stl	32
-2_1_0.stl	7	2_-4_0.stl	16

-2_-1_0.stl	19	-2_4_0.stl	28
2_1_1.5.stl	19	-2_-4_0.stl	15
2_4_1.5.stl	31	-4_1_1.5.stl	43
2_4_-1.5.stl	46	-4_1_-1.5.stl	54
2_-4_1.5.stl	23	-4_-1_1.5.stl	39
2_-4_-1.5.stl	19	4_1_3.stl	37
-2_4_1.5.stl	25	4_1_-3.stl	27
-2_4_-1.5.stl	30	-4_-1_3.stl	56
-2_-4_1.5.stl	35	-4_-1_-3.stl	19
-2_-4_-1.5.stl	14	4_2_0.stl	32
2_4_3.stl	51	4_-2_0.stl	19
2_4_-3.stl	60	-4_2_0.stl	11
-2_-4_3.stl	57	-4_-2_0.stl	22
-2_-4_-3.stl	29	4_2_1.5.stl	21
4_0_0.stl	26	4_2_-1.5.stl	31
-4_0_0.stl	25	4_-2_1.5.stl	30
4_0_1.5.stl	32	4_-2_-1.5.stl	39
4_0_-1.5.stl	38	-4_2_1.5.stl	41
-4_0_1.5.stl	57	-4_2_-1.5.stl	34
-4_0_-1.5.stl	41	-4_-2_1.5.stl	41
4_0_3.stl	33	-4_-2_-1.5.stl	30
4_0_-3.stl	34	4_2_3.stl	42
-4_0_3.stl	45	4_2_-3.stl	41
-4_0_-3.stl	31	-4_-2_3.stl	49
4_1_0.stl	27	-4_-2_-3.stl	28
4_-1_0.stl	17	4_4_0.stl	45
-4_1_0.stl	7	4_-4_0.stl	34
-4_-1_0.stl	25	-4_4_0.stl	20
4_1_1.5.stl	35	-4_-4_0.stl	47
4_1_-1.5.stl	23	4_4_1.5.stl	31
4_-1_1.5.stl	30	4_4_-1.5.stl	27
4_-1_-1.5.stl	36	4_-4_1.5.stl	49
-4_1_1.5.stl	45	4_-4_-1.5.stl	32

-4_4_1.5.stl	38	8_-1_-3.stl	28
-4_4_-1.5.stl	35	-8_1_3.stl	21
-4_-4_1.5.stl	39	-8_1_-3.stl	40
-4_-4_-1.5.stl	18	-8_-1_3.stl	17
4_4_3.stl	51	-8_-1_-3.stl	36
4_4_-3.stl	58	8_2_0.stl	31
-4_-4_3.stl	76	8_-2_0.stl	28
-4_-4_-3.stl	36	-8_2_0.stl	39
8_0_0.stl	22	-8_-2_0.stl	24
-8_0_0.stl	44	8_2_1.5.stl	24
8_0_1.5.stl	33	8_2_-1.5.stl	25
8_0_-1.5.stl	36	8_-2_1.5.stl	36
-8_0_1.5.stl	39	8_-2_-1.5.stl	25
-8_0_-1.5.stl	27	-8_2_1.5.stl	31
8_0_3.stl	34	-8_2_-1.5.stl	24
8_0_-3.stl	32	-8_-2_1.5.stl	48
-8_0_3.stl	24	-8_-2_-1.5.stl	40
-8_0_-3.stl	32	8_2_3.stl	28
8_1_0.stl	32	8_2_-3.stl	23
8_-1_0.stl	22	8_-2_3.stl	28
-8_1_0.stl	41	8_-2_-3.stl	31
-8_-1_0.stl	50	-8_2_3.stl	29
8_1_1.5.stl	37	-8_2_-3.stl	27
8_1_-1.5.stl	30	-8_-2_3.stl	35
8_-1_1.5.stl	33	-8_-2_-3.stl	29
8_-1_-1.5.stl	27	8_-4_0.stl	36
-8_1_1.5.stl	47	-8_4_0.stl	23
-8_1_-1.5.stl	44	8_4_1.5.stl	15
-8_-1_1.5.stl	25	8_4_-1.5.stl	23
-8_-1_-1.5.stl	45	8_-4_1.5.stl	32
8_1_3.stl	34	8_-4_-1.5.stl	30
8_1_-3.stl	31	-8_4_1.5.stl	22
8_-1_3.stl	28	-8_4_-1.5.stl	34

-8_-4_1.5.stl	29	8_4_-3.stl	22
-8_-4_-1.5.stl	21	-8_-4_3.stl	40
8_4_3.stl	24	-8_-4_-3.stl	23

Appendix B: MATLAB Code for counting constraints number

```
tic;
clear all
clc;


%% 파일 이름 갖고오기
stlpath = [pwd '\cad_files_변환1\'];
filepattern = fullfile(stlpath);
stlfile = dir(filepattern);
allfilenames = {stlfile.name};
allfilenames(4) = [];
allfilenames(4) = [];

stlnum = size(allfilenames,2);
%%

for m = 1:stlnum
    result(m,1) = string(allfilenames(1,m));
    filename = [stlpath char(allfilenames(1,m))];
    [x,y,z,c] = stlread(filename);

    temp = min(x);
    xmin = min(temp);
    xnew = -xmin+x;

    temp = min(y);
    ymin = min(temp);
    ynew = -ymin+y;

    temp = min(z);
    zmin = min(temp);
    znew = -zmin+z;

    %printing not possible = notp

    %%
    sze1 = size(x,1);
    sze2 = size(x,2);
    totalsze = sze1*sze2;

    for i = 1:totalsze
        newset(i,:) = [xnew(i),ynew(i),znew(i)];
    end

    %%
    sorted = unique(newset, 'rows');
    newsortedx = sortrows(sorted, 1);
    newsortedy = sortrows(sorted, 2);
    %sorted = sortrows(newset,1);
    %newsorted = unique(sorted, 'rows');
```

```
   sortedsze = size(newsortedx,1);

   %%
   i = 0;
   for i = 1:sortedsze
      newsortedx(i,1) = round(newsortedx(i,1)/0.0025)*0.0025;
      newsortedy(i,2) = round(newsortedy(i,2)/0.0025)*0.0025;
   end

   %%
   i = 0;
   k = 0;
   n = 0;
   notp = 0;
   steps = round(linspace(0, 0.1, 21),3);
   for j = 1:21
      yzlast = [];
      xzlast = [];
      for i = 1:sortedsze
         if newsortedx(i,1) == steps(j)
            k = k+1;
            yzlast(k,1) = newsortedx(i,2);
            yzlast(k,2) = newsortedx(i,3);
         end

         if newsortedy(i,2) == steps(j)
            n = n+1;
            xzlast(n,1) = newsortedy(i,1);
            xzlast(n,2) = newsortedy(i,3);
         end
      end


      size1 = size(yzlast,1);
      if size1 ~= 0;
         notp = notp + counting(yzlast);

      end

      size2 = size(xzlast,1);
      if size2 ~= 0;
         notp = notp + counting(xzlast);
      end
   end
   result(m,2) = notp;
end

toc;
```

Acknowledgement

I want to express my sincere gratitude to my advisor, Prof. Namhun Kim, for all his guidance and encouragement. Without his patience and support, it would have been difficult to complete my master course. His guidance not only expanded my knowledge in additive manufacturing, but also enlightened me so that I could think in the right direction. I would also like to thank my committees, Prof. Duck Young Kim and Prof. Young-Bin Park, for their insightful comments and advices.

My thank also goes to Dr. Eunju Park, Dr. Jooyeon Kwon, and Dr. Moise Busogi for the advices they gave me whenever I needed. Their suggestions on my research have been invaluable and helped me grow as a researcher.

I thank my colleagues in UCIM lab, specially my senior colleagues, Sangho Ha, Jungsik Kim, Donghwan Song, Haejoon Choi, Eunseo, Moon-yeong for sharing their priceless experiences. I am also grateful to my other colleagues such as Taeyang, Ga-hyung, Minso, Jageon for their assistances, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the two years period.

Also I thank my good friends in UNIST: Yunshin and Kyungtak. Finally, I express my profound gratitude to my family: my parents and sister, for their limitless love and support.