Clemson University
TigerPrints

All Theses

Theses

12-2015

Design of a Meta-Material with Targeted Nonlinear Deformation Response

Zachary Satterfield
Clemson University, zsatter@clemson.edu

Follow this and additional works at: https://tigerprints.clemson.edu/all_theses

Part of the Mechanical Engineering Commons

Recommended Citation

Satterfield, Zachary, "Design of a Meta-Material with Targeted Nonlinear Deformation Response" (2015). *All Theses*. 2245.
https://tigerprints.clemson.edu/all_theses/2245

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.

DESIGN OF A META-MATERIAL WITH TARGETED NONLINEAR DEFORMATION RESPONSE

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mechanical Engineering

---

by
Zachary Tyler Satterfield
December 2015

Accepted by:
Dr. Georges Fadel, Committee Chair
Dr. Nicole Coutris, Committee Member
Dr. Gang Li, Committee Member

ABSTRACT

The M1 Abrams tank contains track pads consist of a high density rubber. This rubber fails prematurely due to heat buildup caused by the hysteretic nature of elastomers. It is therefore desired to replace this elastomer by a meta-material that has equivalent nonlinear deformation characteristics without this primary failure mode. A meta-material is an artificial material in the form of a periodic structure that exhibits behavior that differs from its constitutive material. After a thorough literature review, topology optimization was found as the only method used to design meta-materials. Further investigation determined topology optimization as an infeasible method to design meta-materials with the targeted nonlinear deformation characteristics. Therefore, a method was developed in this thesis to logically and systematically design meta-material unit cells using engineering principles to achieve the desired nonlinear response. This method, called the Unit Cell Synthesis Method, requires the designer to have a fundamental understanding of the geometric nonlinearity of an elemental geometry. One or more of these elemental geometries are then systematically combined into a unit cell. A size optimization is performed on promising unit cell concepts to tune the geometry and converge its response towards that of the target. Application of this method was successful in generating a meta-material to meet the response of the rubber pad. The method represented in this thesis is meant to serve as a framework for future designers to develop meta-materials for nonlinear targeted responses.

DEDICATION

To my family and friends.

ACKNOWLEDGEMENTS

This research would not have been possible without the support and collaboration from many individuals. I would like to thank my parents, Nathaniel and June, for their unwavering support. I would not have made it this far without them. Thanks to my sister, Heather, for her unrelenting competition and love that has motivated my success. Thanks to Kara, who has unconditionally supported me throughout my research and the writing of this thesis.

I would like to thank all of my committee members for contributing their knowledge and support throughout this research project. I would like to thank Dr. Fadel for providing me with this research opportunity and guiding me through the many stages of the research process. Thanks to Dr. Li for his expertise in finite elements and the enthusiasm he brought to this research. Thanks to Dr. Coutris for her expertise in elasticity and attention to detail throughout this research.

I would like to thank Matthew Castanier, Bill Bradford, and the folks at US Army TARDEC for their financial support of this research. I would also like to thank Christopher Cardine of General Dynamics and Joshua Goossens of Tenneco for their collaboration and providing an industrial perspective.

I would like to thank my lab-mates in EIB 136 and friends who have contributed to this research. Thanks to Rahul Renu, Vijay Sarthy, and Anthony Garland for their insightful discussions. Thanks to Neehar Kulkarni for his close collaboration and discussion with this research.

TABLE OF CONTENTS

Abstract ii

Dedication iii

Acknowledgements iv

Table of Tables viii

Table of Figures ix

Chapter 1. Introduction 1

1.1. Overview of Abrams Military Track Pad System 1

1.2. Motivation for Replacing Elastomer Track Pad 3

1.3. Motivation for Designing a Meta-Material 6

1.4. Research Questions 8

1.5. Thesis Outline 9

Chapter 2. Literature Review 10

2.1. Methods to Design and Optimize Meta-Materials 10

2.2. Applications in Designing Meta- Materials 22

2.3. Conclusions 26

Chapter 3. Designing via Topology Optimization 28

3.1. Objective 28

3.2. TO in Optistruct 28

TABLE OF CONTENTS (CONTINUED)

3.3. Discussion 33

3.4. Conclusions 40

Chapter 4. Designing via Engineering Principles 41

4.1. Objective 41

4.2. Method of Evaluation 41

4.3. Design of the Brick Unit Cell 46

4.4. Design of the BrickOval Unit Cell...52

4.5. Design Considerations 62

4.6. Conclusions 70

Chapter 5. A Unit Cell Synthesis Method for Meta-Material Design 73

5.1. Method Introduction 73

5.2. Method Description 75

5.3. Discussion 83

5.4. Conclusion 85

Chapter 6. Conclusions and Future Work 87

6.1. Conclusions 87

6.2. Broader Impact..88

6.3. Future Work 89

TABLE OF CONTENTS (CONTINUED)

Chapter 7. References 91

Appendices 96

Appendix A. Python Script for Brick Design 97

Appendix B. Python Script for BrickOval Design 106

TABLE OF TABLES

Table 4.1 Target % Vertical Deformation Values 45

Table 4.2 Full Factorial Parameters and Values 51

Table 4.3 Dimensions of Optimized UC...61

Table 4.4 % Difference in Bulk Deformation Comparison Across Load Cases ...64

Table 4.5 Comparison of Bulk Deformation between Scaled Unit Cells 66

Table 5.1 Possible Connection Configurations...76

TABLE OF FIGURES

Figure 1.1. Components of the T186LL Track Link [2]..1

Figure 1.2. 2D Representation of a Road Wheel and Track Link [3].....................2

Figure 1.3. Track Backer Pad Failures Due to Chunking [2] 3

Figure 1.4. Decaying of Elastomer Properties w.r.t. Cycling Stress [6].................4

Figure 1.5. Thermal Map of M1 Abrams Track Pad and Road Wheel [2].............5

Figure 1.6. Loading and Unloading Stress-Strain Curves for (Left) Elastomer, and (Right) Linear Elastic Material [8] 6

Figure 1.7. Methodology to Optimize Meta-Material to Achieve target Properties [10]..7

Figure 1.8. Ashby Chart Comparing Materials According to Loss Coefficient and Young's Modulus [3,9] 8

Figure 2.1. Microscopic Unit Cell Variables in Square Void [19].......................11

Figure 2.2. Example (left) Boundary Conditions and (right) Solution using the HM [22]..13

Figure 2.3. Example Solutions of SIMP Method with (left) p=1.5 and (right) p=3 [22]..14

Figure 2.4. The Effect of Varying “p” in the SIMP Method [23].........................15

Figure 2.5. Example of (left) a Polyhex Prototile and (right) a Parallelogram Fundamental Domain of this Pattern [26]..18

Figure 2.6. Fundamental Domains for an L-Shaped Prototile [27] 18

TABLE OF FIGURES (CONTINUED)

Figure 2.7. Procedure Overview for Applying IHP to the Design of Meta-Materials [28] 19

Figure 2.8 Level Set TO of Cantilever Beam: (a) Boundary Conditons, (b) Initialization with Many Small Holes and (c) its Solution, (d) Initialization with Few Large Holes and (e) its Solution [32] 20

Figure 2.9. A Three-Phase Meta-Material (a) Unit Cell, and (b) in a 3x3 Array via TO [40] 24

Figure 2.10. Unit Cell of a Honeycomb Unit Cell before (left) and after (right) SO for Maximum Energy Absorption [42] 26

Figure 3.1. Initial UC with boundary conditions and uniaxial loading 29

Figure 3.2. Representative UC in Optistruct 29

Figure 3.3. UC topology solution to single load case, (right) top, and (left) bottom 31

Figure 3.4. Uniaxial tension curve for UC with single targeted compliance 31

Figure 3.5. Hypothetical uniaxial tension curve with two targeted compliances ..32

Figure 3.6 Illustrative Comparison of Target Responses in a Linear and Nonlinear IHP 37

Figure 3.7. Linear elastic TO a) boundary conditions with aspect ratio “x” by “y” and b-e) results with different aspect ratios 38

Figure 4.1 Current Elastomer Stress-Strain Response [3] 42

TABLE OF FIGURES (CONTINUED)

Figure 4.2 Example Methods of Determining Material Properties of (a) Rubber and (b) Meta-Material 44

Figure 4.3 Load and Boundary Conditions for Rubber Pad 45

Figure 4.4 Uniaxial Compression Target Response 46

Figure 4.5 FFB Free Body Diagram and Geometric Nonlinearity based on Aspect Ratio 47

Figure 4.6 FFB with Boundary Conditions Preserved..48

Figure 4.7 Modified Strut to Allow UC Periodicity and Connectivity.................48

Figure 4.8 "Brick" UC with Parameters 49

Figure 4.9 "Brick" UC in a 4x4 Tessellation 50

Figure 4.10 Full Factorial Analysis of 4x4 "Brick" Tessellation...........................51

Figure 4.11 Oval Geometric Free Body Diagram Nonlinearity based on Parameter Sensitivities 53

Figure 4.12 Predicted Effect Of Combining Oval with FFB in a UC 54

Figure 4.13 Integration of Oval with FFB in UC..55

Figure 4.14 Load Path of (top) Springs and (bottom) Geometries in Parallel.......55

Figure 4.15 Example 4x4 Tessellation of BrickOval UC......................................56

Figure 4.16 Optimization Setup in ModeFrontier...59

Figure 4.17 Optimization Design History and Objective Convergence 60

Figure 4.18 Optimized Meta-Material Deformation Response 61

Figure 4.19 Optimized BrickOval Design in a 4x4 Array 61

TABLE OF FIGURES (CONTINUED)

Figure 4.20 Bulk Property Homogenization Limit Results for BrickOval UC 64

Figure 4.21 Bulk Deformation Convergence towards Homogeneous Properties..65

Figure 4.22 Max Stress and Stress Distribution Comparison of a) Datum, b) 0.1 Scale, and c) 10.0 Scale UC's 67

Figure 4.23 Ashby Chart Comparing Material Strength and Elastic Modulus [9] 69

Figure 4.24 Initial Titanium Results via Manual Tuning 70

Figure 5.1 Synthesis UC Design Method Flowchart 74

Figure 5.2 Example EFGs and their General Nonlinear Behavior 75

Figure 5.3 1st Order Connection Configuration 77

Figure 5.4 2nd Order Connection Configuration 77

Figure 5.5 Decomposition of a Meta-Material into RVE and Tessellation of UC 79

Figure 5.6 Example Meta-Material with Uniaxial Loading (left) and after Deformation (right) 80

CHAPTER 1. INTRODUCTION

1.1. Overview of Abrams Military Track Pad System

The M1 Abrams tank weighs upwards of 63 tons and can travel at speeds up to 45 mph [1]. The track system provides a robust means for the tank to traverse a diverse range of environments. The current track system used in the M1 tank is the T186LL and a dissected track link can be seen in Figure 1.1 [2].

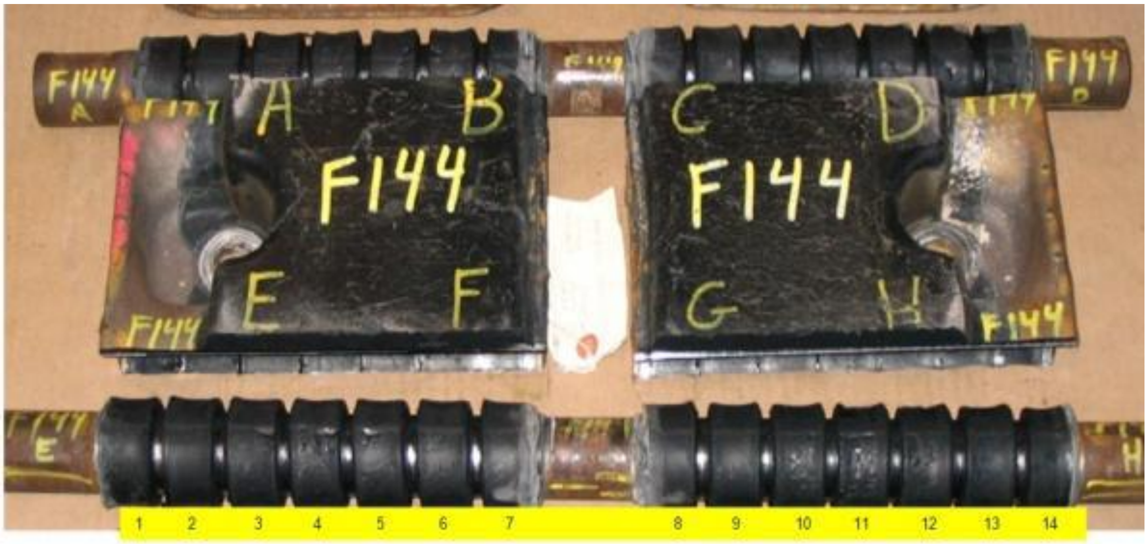


Figure 1.1. Components of the T186LL Track Link [2]

The track system is comprised of individual track links that are connected via steel linkages and bushings. Under standard operation, the track links come into contact with road wheels which support the weight of the tank. A diagram showing this interaction can be seen in Figure 1.2 [3]. In this figure, it can be seen that the track link consists of three primary components: the ground pad, the steel plate, and the backer pad. The primary focus of this research will be on the backer pad.

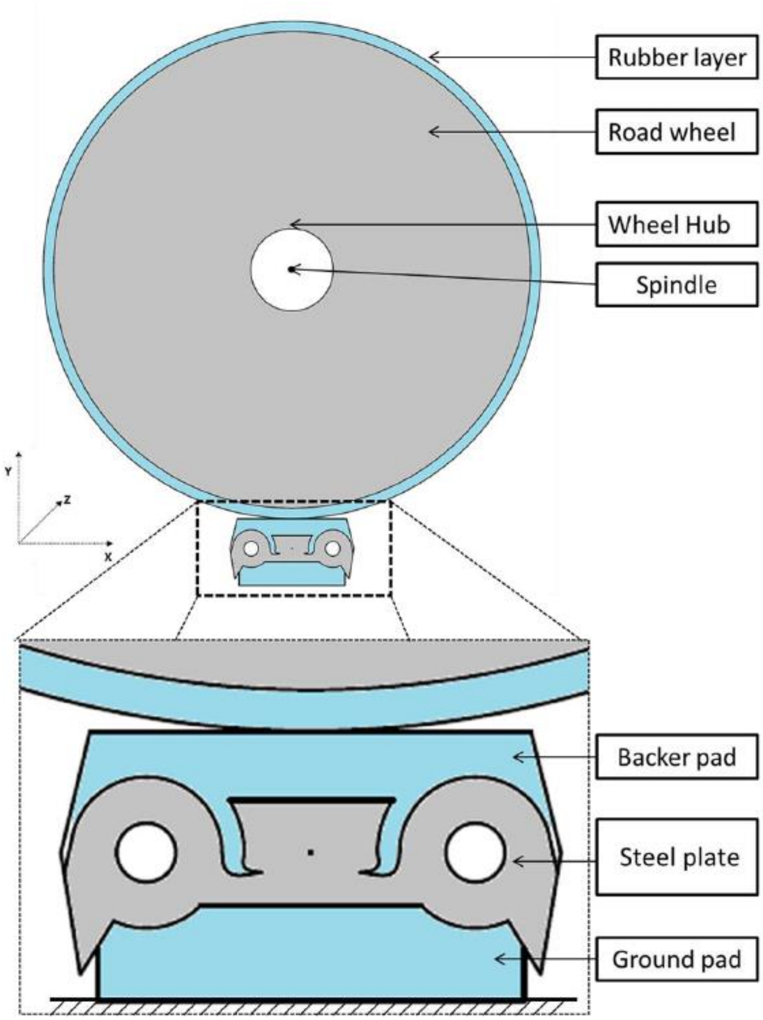


Figure 1.2. 2D Representation of a Road Wheel and Track Link [3]

The track pad has several functions including road surface protection, sound-deadening, traction on hard surfaces, and most importantly vibration reduction. The track pad has similar functions to those of automotive pneumatic tires. The pad must support the vehicle weight as it experiences both compressive and shear loadings, operate at high speeds for a long duration, and must have low wear for durability and replacement considerations [4].

The current track backer pad is composed of Styrene Butadiene Rubber (SBR) combined with a filler material. The filler contains short fiber reinforcements that act to resist tearing, chunking, and abrasion [1]. Chunking is defined as a loss of material which typically results from crack propagation and thermal degradation of the rubber [2]. Examples of chunking on the backer pad are shown in Figure 1.3.

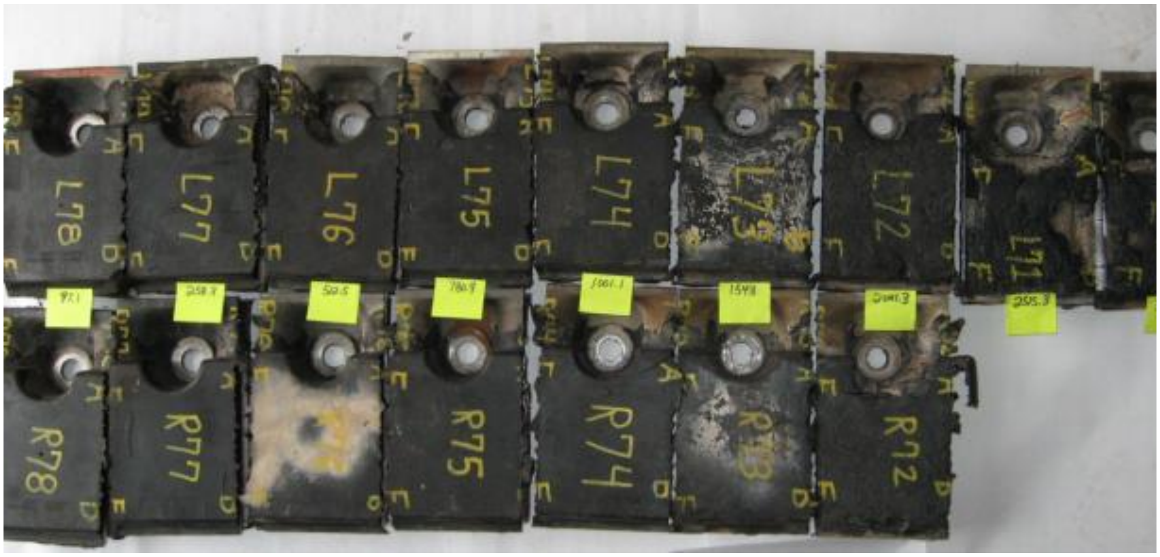


Figure 1.3. Track Backer Pad Failures Due to Chunking [2]

1.2. Motivation for Replacing Elastomer Track Pad

The rubber backer pad on a M1 Abrams tank operates as a dampening interface between the metal track link and the road wheel. Under normal operation, the backer pad experiences high strain rate and cyclic loading conditions. These severe conditions result in limited fatigue life of the pad and lead to increased tank downtime and high maintenance costs. Previous research has been completed in order to understand the primary mode of failure in an effort to prolong the track pad’s serviceable life.

Lesuer et al. in [5] were some of the first to begin experimental investigations in failure of tank track pads in 1983. Lesuer et al. in [6] developed some of the first computer models to understand track pad failure in 1985. Their work showed that track pad life is a function of temperature, environment, and number of loading cycles. One of their results, shown in Figure 1.4, relates increased temperatures and cycling stresses to decaying residual strength [6].

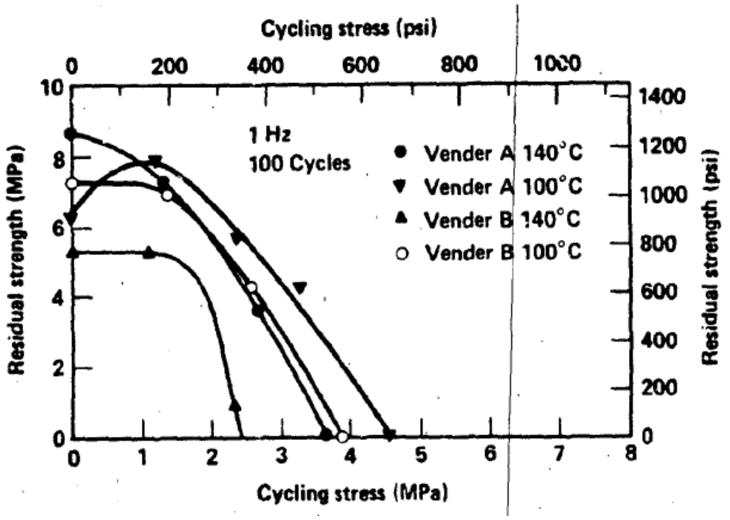


Figure 1.4. Decaying of Elastomer Properties w.r.t. Cycling Stress [6]

In [7], Mars and Ostberg defined a 2-term Ogden hyperelastic model of the current elastomer, performed dynamic simulations to capture the effects of a rollover event, and used these to estimate damage accumulation via a fatigue life solver. A rollover event can be defined as the entire process for a track pad to complete a single revolution in the track system. In this single revolution, seven successive road wheels initialize contact with the pad and traverse across the top of the pad via rotation. The rollover event completes when the track pad traverses the top of the track system which is not in contact with the ground. Within this rollover event, the road wheel causes high strain rates to occur and the elastomer experiences mostly compressive and some shear deformations (especially when turning). The goal of the authors was aimed at understanding how damage developed in the elastomer part.

Ostberg and Bradford explored the loading distribution of the road wheel on the backer pads in [2]. The infrared image in Figure 1.5 shows the temperature of the backer pad after 12 miles at a constant 40 mph. Consistent operation at elevated temperatures degrades elastomer material properties. Thus, it is necessary to minimize energy losses, specifically those due to thermal energy caused by the deformation of the backer pad [2].

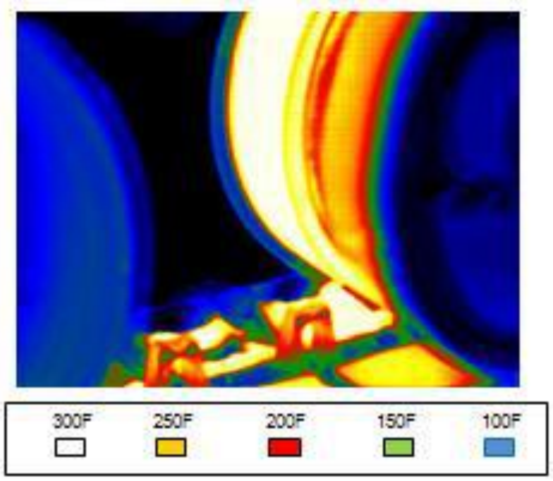


Figure 1.5. Thermal Map of M1 Abrams Track Pad and Road Wheel [2]

The primary source of elevated temperatures within the backer pad elastomer is due to the material's inherent loss coefficient. Elastomers are nonlinear materials with respect to their stress-strain response. However, hysteresis is exhibited in the loading cycle (or loading and unloading) as depicted on the left side of Figure 1.6. The area between these two curves is energy loss, most of which converts into thermal energy. The heat generated due to one cycle of loading cannot dissipate away from the elastomer at a fast enough rate due to the elastomers' low thermal conductivity [9]. This leads to the high buildup of thermal energy shown in Figure 1.5.

1.3. Motivation for Designing a Meta-Material

Rodger Walser coined the term "meta-material" in 1999 and discussed a strategy in which to design meta-materials for a desired purpose in [11]. Meta-materials are a class of artificial materials that are so named due to their designed purpose of achieving specific global properties. Specific definitions vary in literature but the objective remains the same throughout [12]. These artificial materials are designed to achieve behavior that cannot be found in nature in response to some need or application. In the context of this thesis, a meta-material is an artificial material that contains unit cells at the local level which, combined through tessellation, creates a global medium that can be considered homogeneous but with properties different from the base material. The uniqueness of global physical behavior in meta-materials emerges from the combination of the constitutive material and geometry of the unit cells. A graphical representation of how

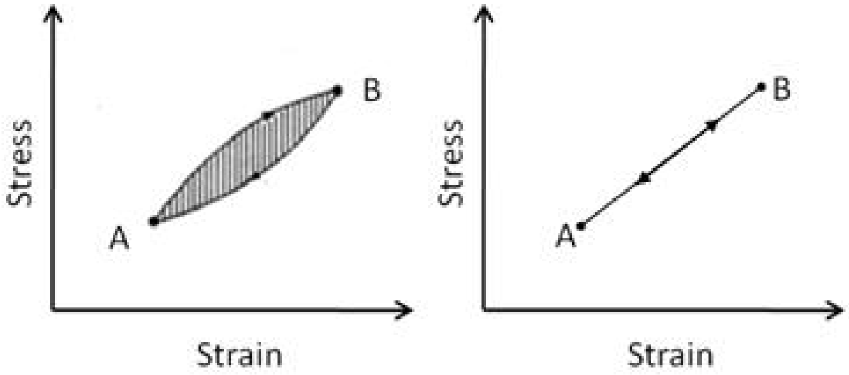


Figure 1.6. Loading and Unloading Stress-Strain Curves for (Left) Elastomer, and (Right) Linear Elastic Material [8]

meta-materials can be tuned to achieve targeted properties via optimization is shown in Figure 1.7.

Based on the source of thermal energy in the current track pad, the undesired heat generation could be eliminated by replacing the high loss elastomer with a material that exhibits little to no loss. As shown in the right side of Figure 1.6, linear elastic materials do not exhibit the same hysteresis as elastomers. Therefore, such materials are desired as replacement to the current rubber track pad. Ashby presents a useful material comparison based on the loss coefficient and Young's Modulus material properties in Figure 1.8. As shown, most metals have a desired low loss coefficient. However, the Young's Modulus of metals is several orders of magnitude too high for the intended application – replacing the elastomer track pad. Since all properties listed in this Ashby chart are those of solid, homogeneous materials, it is reasonable to expect that designing a meta-material with a metallic constitutive material can yield the desired Young's Modulus without significantly compromising the loss coefficient.

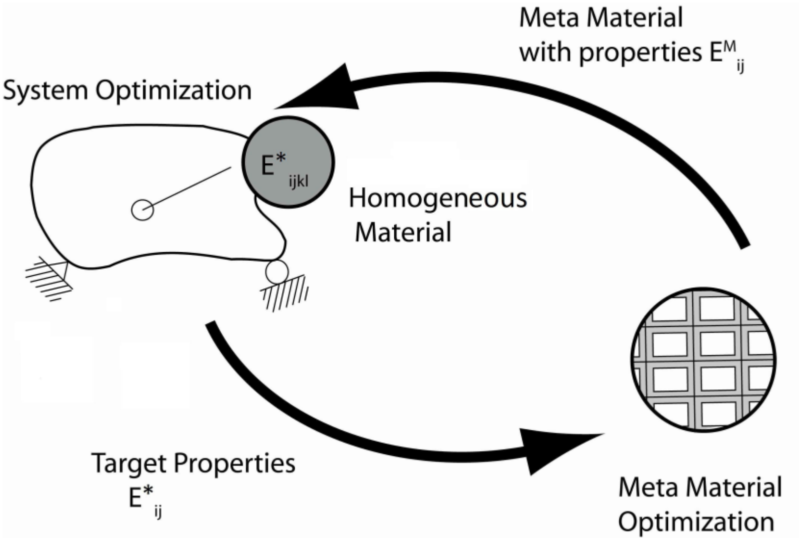


Figure 1.7. Methodology to Optimize Meta-Material to Achieve target Properties [10]

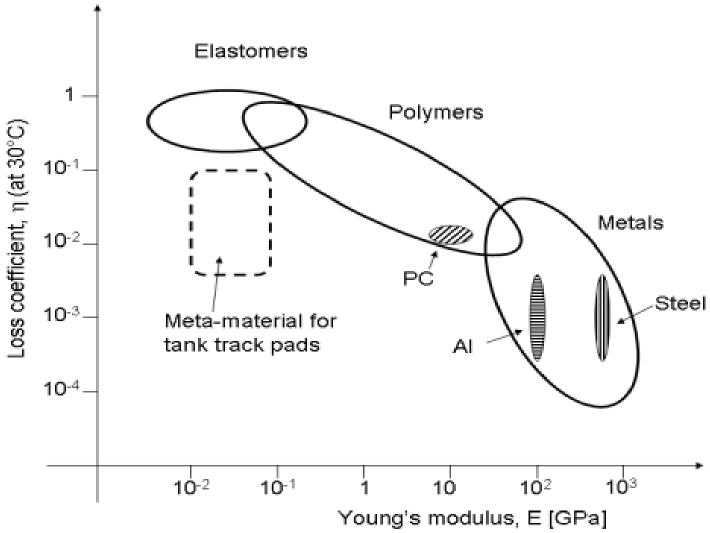


Figure 1.8. Ashby Chart Comparing Materials According to Loss Coefficient and Young's Modulus [3,9]

1.4. Research Questions

This thesis focuses on answering the following questions by testing their associated hypotheses:

R1. Can a meta-material be developed in which the global behavior of the resultant medium exhibits nonlinear compressive behavior similar to that of the current elastomer?

H1. A meta-material can be designed whose global behavior in compression is of similar nonlinearity to that of the current elastomer.

R2. What method can be used to successfully develop a meta-material to meet the nonlinear deformation response of the current application?

a. Is topology optimization a feasible method to accomplish this?

b. If not, can a method be developed to accomplish this?

H2. A meta-material can be developed using topology optimization to meet the material behavior requirements.

1.5. Thesis Outline

This thesis is organized into six chapters. The current chapter has introduced the motivation and research objectives of this work.

Chapter 2 will review the current literature about the two primary methods that can be used to design meta-materials, including the merits and applications of each.

Chapter 3 describes the methods used and the results of designing a unit cell with the desired properties via topology optimization and the lessons learned from that work.

Chapter 4 describes the method to design a unit cell with the desired properties via a fundamental understanding of beam shapes, tuning the geometry with parametric optimization, and presents the resulting geometry.

Chapter 5 is a discussion of the results obtained in Chapters 3 and 4, lessons learned, and presents a possible generalization of the method used in Chapter 4 to create a design framework for meta-material design.

Chapter 6 concludes this work by summarizing the methods used, obtained results, and significance and laying out areas of future work.

CHAPTER 2. LITERATURE REVIEW

In an effort to understand the best approach to successfully solve the design problem, a literature review of relevant historic and contemporary research in the field was conducted. This literature review is organized into existing approaches to design meta-materials by topology optimization and tune existing topologies via size optimization. Example applications in these areas for designing meta-materials are also presented.

2.1. Methods to Design and Optimize Meta-Materials

2.1.1. Topology Optimization

Topology optimization (TO) is a numerical optimization approach requiring little input from the designer that often yields novel unit cell geometries. There are two primary methods that have been developed to implement TO: the Homogenization Method (HM), and Level Sets.

2.1.1.1. Fundamentals of the Homogenization Method

The HM was born from the mathematical theory of homogenization, or relation between macro- and micro-level properties [13]. This theory was adapted for use in TO by creating a method to determine effective properties of heterogeneous media to enable implementation in the finite element method [14]. The HM, therefore, combines homogenization theory with a finite element solver to solve an optimization problem that aims to determine the optimal topology of some structure. Bendsøe and Kikuchi developed and first applied the HM in TO in [15]. Hassani and Hinton developed an extensive mathematical formulation of the HM, and several variants, in [16–18].

The HM designs a topology by optimizing the material distribution within the discretized design domain via the addition or removal of material in microstructures within this domain. These microstructures can have voids with various shapes. One such example is shown in Figure 2.1 where the microscopic unit cell and void are in the shape of a square and the void is defined by its height, width, and rotation. The microstructure of each unit cell in the domain can vary from completely solid, completely void, or any range in between.

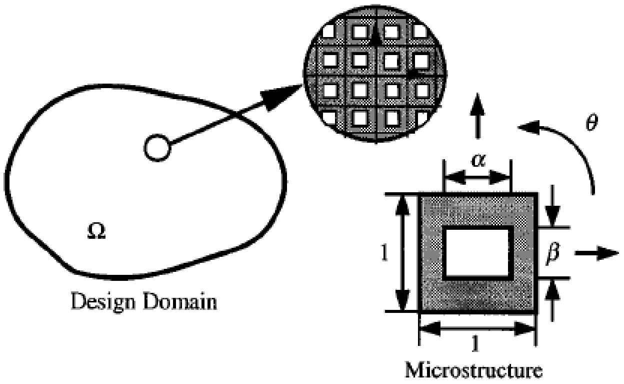


Figure 2.1. Microscopic Unit Cell Variables in Square Void [19]

The optimization algorithm modifies these unit cell microstructures, or design variables, based on data of the finite element results and the update scheme chosen to improve the objective function value. These new cell microstructures are then related to effective properties for finite element purposes using homogenization theory and submitted for the next iteration. This iterative process continues until some convergence criteria is met [19]. While both gradient and non-gradient based optimization algorithms applied to

TO problems can be found in literature, Sigmund convincingly proves in [20] that gradient based algorithms are superior in computation time, computation cost, and yield more optimal topologies.

In order to perform TO using the HM, the optimization procedure must be formalized. While the objective function is problem dependent, an example of the classic TO problem for minimizing compliance given a volume fraction constraint is given [21] below

$$
\begin{aligned}
\min_{x}:&\ \ c(x) = U^T K U = \textstyle\sum_{e=1}^{N} x_e u_e^T K_0 u_e \\
subject\ to:&\ \ \frac{V(x)}{V_0} = f \\
:&\ \ KU = F \\
:&\ \ 0 < x_{min} \leq x \leq 1
\end{aligned}
\tag{2.1}
$$

where $c(x)$ is the objective function to minimize strain energy, f is the volume fraction constraint, $KU = F$ is the finite element solution, and the remaining constraint is the range of allowable element densities [21]. The minimum of the element density range is constrained to be above zero, denoted by x_{min}, to prevent singularity issues in the stiffness matrix in performing the finite element analysis.

An example implementing the TO problem described in equation (2.1) for minimizing compliance is shown in Figure 2.2. This left figure depicts the boundary conditions of a cantilevered beam with a single point load on the free end. The optimized topology for a structure with maximum stiffness is shown in the right figure. Here, the black portions indicate solid material, white indicates void of material, and shades of grey represent some gradient of material and voids. Due to manufacturing constraints limiting the production of microstructures represented by shades of grey, modifications of the original HM have been developed to improve manufacturability.

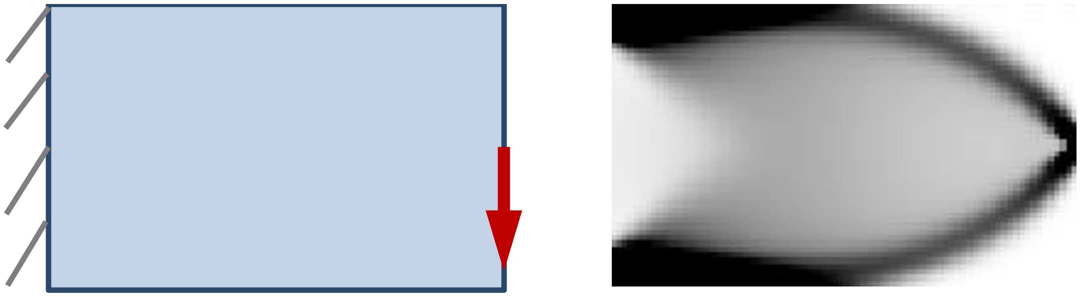


Figure 2.2. Example (left) Boundary Conditions and (right) Solution using the HM [22]

2.1.1.2. The SIMP Method

One of the first adaptations of the HM in TO is the Solid Isotropic Material with Penalization method (SIMP). The purpose of this variant is to eliminate topologies that are not manufacturable. The SIMP method accomplishes this by penalizing design variables if their density lies between 0 (completely void) or 1 (completely solid). The penalization is implemented by raising the element density to an exponential factor of "p" in the objective function. Extending the optimization formulation represented in equation (2.1) to enforce penalization under the SIMP method is shown below.

$$\begin{aligned} \min_{x}:&\ c(x) = U^T K U = \textstyle\sum_{e=1}^{N} x_e{}^p u_e^T K_0 u_e \\ subject\ to:&\ \frac{V(x)}{V_0} = f \\ :&\ KU = F \\ :&\ 0 < x_{min} \leq x \leq 1 \end{aligned} \tag{2.2}$$

It is useful to note that the SIMP method yields the original HM by setting the penalization factor to one. As this penalization factor increases, the intermediate densities are shown to be removed from the solution. Using the same boundary conditions as in Figure 2.2 (left), results of using the SIMP method are shown in Figure 2.3 by setting the penalization to 1.5 (left) and 3 (right) [22]. Note that the solution with penalization of 1.5 yields a more defined solid topology than the solution in Figure 2.2, however there still exists intermediate densities in the solution. The solution with penalization of 3 yields a defined solid solution with little intermediate densities along the solid-void borders. Thus, the solution with penalization factor of 3 yields the best manufacturable solution and was found to be the best penalization factor for eliminating intermediate densities in the literature.

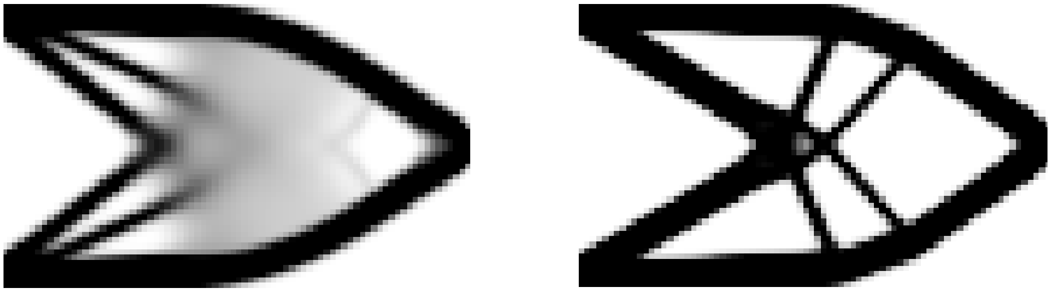


Figure 2.3. Example Solutions of SIMP Method with (left) p=1.5 and (right) p=3 [22]

It is also important to note that a specific drawback of the SIMP method is the effect that increasing the penalization factor has on the objective function. As the penalization factor increases, the converged solution will yield objective function values that are less optimal. Since the objective function is a measure of the desired value, this phenomenon can be graphically portrayed in Figure 2.4 [23]. In this figure, normalized stiffness is compared to the volume fraction across different penalization values. The curves shown are theoretical maximums at the specified penalization and volume fraction. This graph expresses how enforcing the manufacturing constraint via the SIMP method severely limits physical performance as there is a significant gap between the theoretically optimal solution (with penalization factor of 1) and the manufacturable optimal solution (typically with penalization factor of 3). For example, consider the minimum compliance example of a cantilever beam with design space and loading conditions shown in Figure 2.2 (left). The solution with penalization factor of 1 is shown in Figure 2.2 (right) as a black, white, and gradient solution. However, the solution with penalization factor of 3, shown in Figure 2.3 (right), is strictly a black and white solution. In the latter, it is clear where the boundaries of the material are and a part can then be manufactured. However, in the grayscale solution (with penalization factor of 1), it is unclear how to manufacture these intermediate densities.

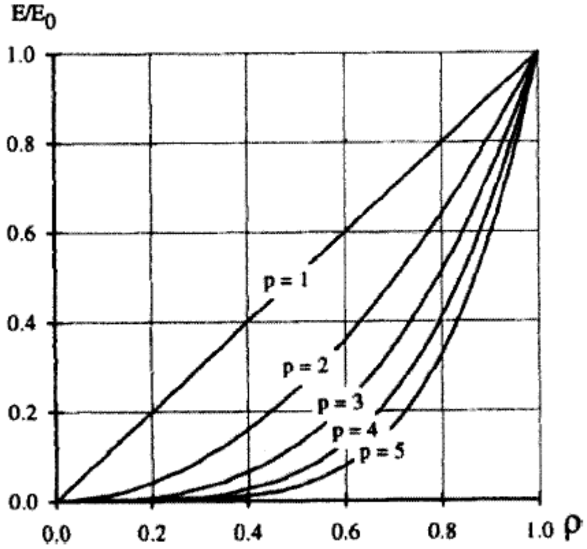


Figure 2.4. The Effect of Varying "p" in the SIMP Method [23]

2.1.1.3. The Inverse Homogenization Problem

The HM can be used to solve two general classes of problems: the standard homogenization problem, and the inverse homogenization problem (IHP). The former has been previously described as in the minimum compliance example (2.1.1.1). In this class of problems, some homogenized property is being optimized in the objective function (e.g. compliance) while some constraint(s) is applied (e.g. volume). However, the IHP reverses this optimization problem. Instead, the objective function may be to optimize volume with a constraint on the homogenized elastic properties, or elasticity tensor. This allows the designer to target specific material properties.

The IHP was first introduced by Sigmund in [24]. In this paper, Sigmund explained how the optimization problem could be formulated to target linear elastic material properties. The base cell, Y, is rectangular in $\mathbb{R}_2$ and is defined as

$$Y =]0, Y_1^0[x]0, Y_2^0[\tag{2.3}$$

where Y_1^0 and Y_2^0 are the horizontal and vertical lengths of the unit cell, respectively. The global properties of the material can then be defined as

$$E_{ijkl}^H = \frac{1}{Y}\int_Y E_{ijpq}(\varepsilon_{pq}^{0(kl)} - \varepsilon_{pq}^{*(kl)})dY \tag{2.4}$$

where $\varepsilon_{pq}^{0(kl)}$ is the desired pre-strain defined for uniaxial tension along the x- and y-direction, and pure shear. $\varepsilon_{pq}^{*(kl)}$ is the resulting solution of Y-periodicity. The optimization procedure is described using optimality criteria and Lagrangian multipliers. Sigmund then provided examples in 2D for various target properties while minimizing volume. Sigmund later expands on this work in [25] by applying the approach specifically to a strain energy formulation and applying it to both 2D and 3D examples. These two papers serve as the foundation for designing structures via TO with prescribed material properties.

2.1.1.4. Considerations for the Design of Meta-Materials

Designing meta-materials using TO requires properly defining an appropriate unit cell design space, enforcing periodic boundary conditions, and developing the optimization problem as an IHP. Each of these issues will be briefly discussed in the following section.

One of the fundamental considerations in the design of meta-materials via TO is defining the base unit cell, or design domain. Bénard and Diaz discuss in [26] how periodic tilings, or prototiles, of cells can be defined in different geometric patterns. They prove that all prototiles can be represented in fundamental domains, or parallelograms, defined by vectors v_1 and v_2 from a corner of the tile. Another constraint for fundamental domains is that their area must be equal to that of the prototile. In Figure 2.5, the left image shows a geometric pattern using Polyhex, or hexagonal, prototiles. The right image shows the same pattern represented by a parallelogram fundamental domain. A primary reason to identify such a prototile is for characterization into a domain that can be discretized into common finite elements.

In a later work ([27]), Bénard and Diaz also prove that fundamental domains may not be unique to a given pattern, as shown in Figure 2.6. However, in this figure, note that domain S is not fundamental as it fails the area constraint. In reference to symmetric domains used in TO, the authors note that representing the microstructure domain using

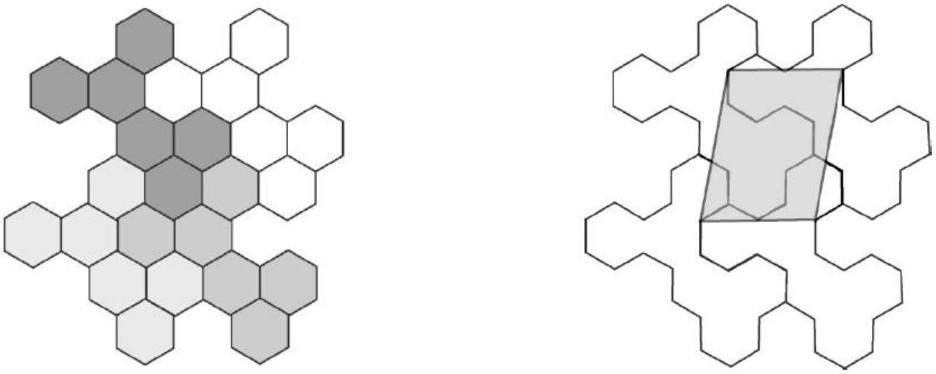


Figure 2.5. Example of (left) a Polyhex Prototile and (right) a Parallelogram Fundamental Domain of this Pattern [26]

other geometries than squares allows one to enforce other symmetry conditions that can increase the design space and the chance of developing new solutions.

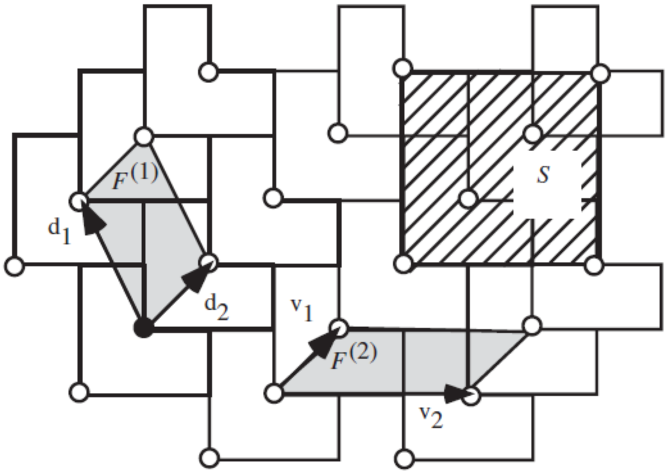


Figure 2.6. Fundamental Domains for an L-Shaped Prototile [27]

Meta-materials can be designed via TO to target specific material properties per an adapted IHP. Once the fundamental domain, Y, has been established, the effective properties of Y are assumed to be homogenous throughout the meta-material. The connectivity between unit cells is just as important as the internal topology of the unit cell.

Therefore, enforcing periodic boundary conditions are essential. Once these boundary conditions are in place, the designer must only consider a single unit cell design space, Y.

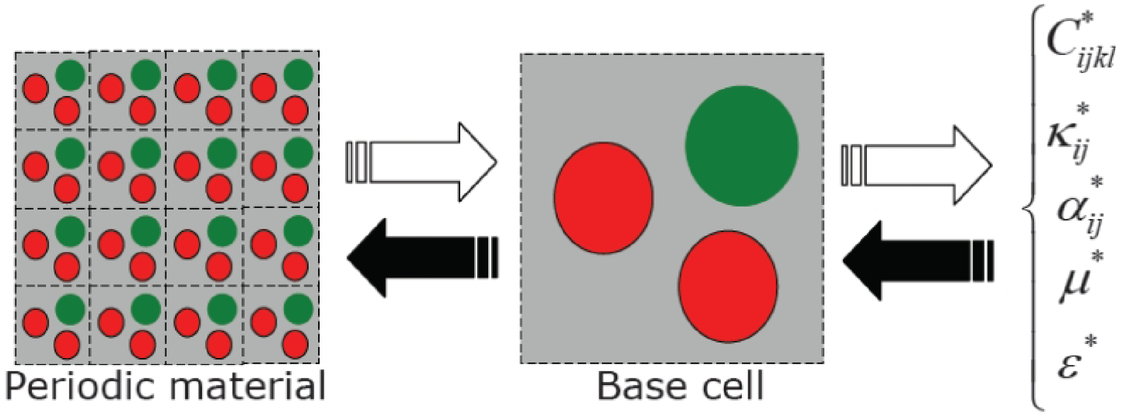


Figure 2.7. Procedure Overview for Applying IHP to the Design of Meta-Materials [28]

In [28], Sigmund explains how the IHP can be applied to the design of meta-materials. The IHP process as applied to meta-materials is shown in Figure 2.7 (depicted by black arrows) and compared to the traditional homogenization problem (depicted by white arrows). Sigmund develops the optimization setup for a meta-material IHP and presents three examples of its application in the design of meta-materials with prescribed elastic and electromagnetic properties.

2.1.1.5. Level Set Method

Level Set approaches in topology optimization employ a level set model embedded in a scalar function of a higher dimension to represent a structural boundary. This method operates by moving the structural boundary that is implicitly defined in the level set function that is driven by the objective function. A thorough numerical formulation of a level set in TO is given by Wang et al. in [29]. Additionally, a literature review of level set formulations and applications is given by van Dijk et al. in [30]. Level set methods typically involve tight coupling with a finite element solver which limits implementation with a commercial finite element solver [31]. Additionally, optimized designs are typically highly dependent on the initial guess of material distribution [32]. This phenomenon can be seen in Figure 2.8 where the number of holes in the initial material distribution is directly correlated to the number of holes in the optimized topology. Unfortunately, there were no instances in literature where level sets were used to design meta-materials.

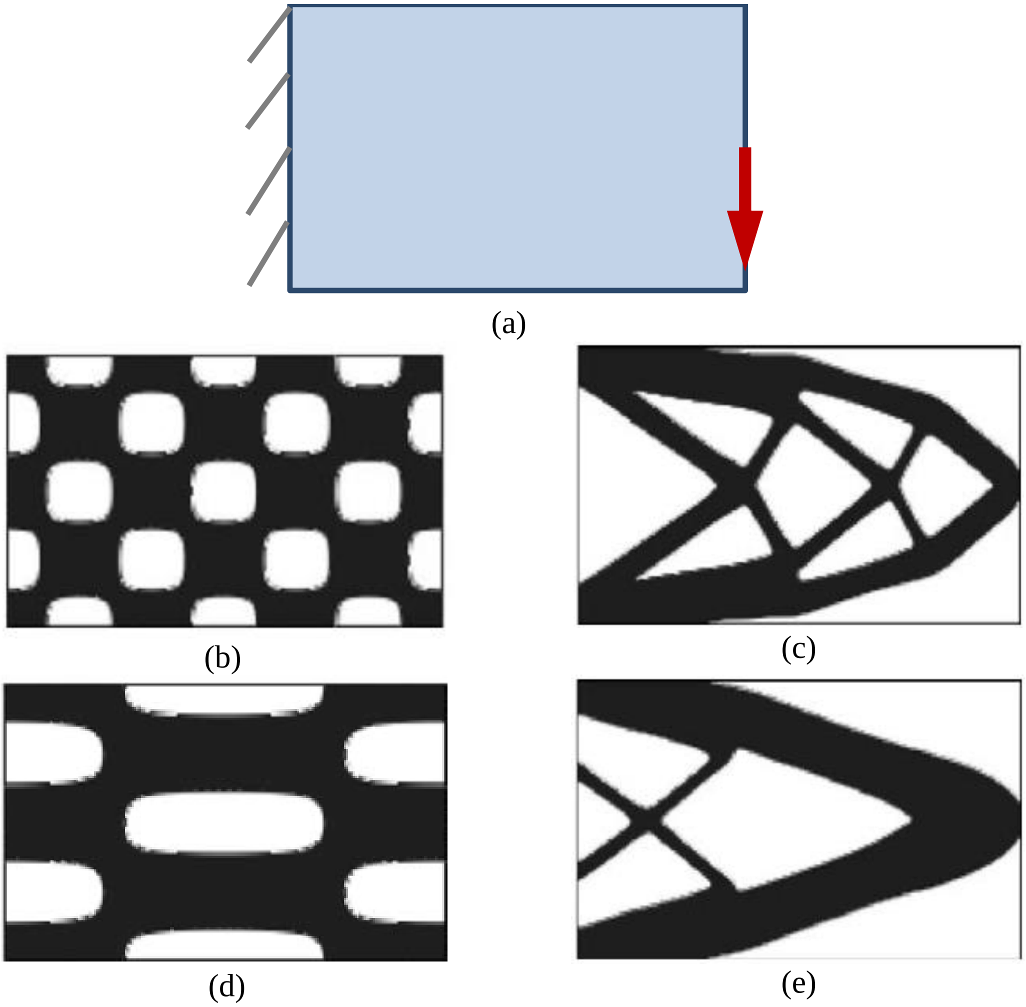


Figure 2.8 Level Set TO of Cantilever Beam: (a) Boundary Conditions, (b) Initialization with Many Small Holes and (c) its Solution, (d) Initialization with Few Large Holes and (e) its Solution [32]

2.1.2. Size Optimization

Size optimization (SO) is an approach that searches for optimal dimensions, orientation, and/or curvature of the unit cell given a topology. Therefore, the topology and variables, typically geometric dimensions, to be optimized must be specified by the designer. This fact distinguishes SO as a method to tune an existing topology instead of one to design a topology. The number of design variables is typically few, (less than 10). SO iterates over an optimization algorithm coupled with an analytical tool, such as finite element analysis, to calculate the objective function until it finds an optimal solution. The optimization problem SO solves is typically in regards to a whole system with known boundary conditions or the unit cell of a meta-material with periodic boundary conditions.

A variety of optimization algorithms can be employed in SO problems, each with their own merits. Considerations for choosing algorithms include required convergence time, available computational resources, computational cost per iteration, availability of a gradient evaluation, the number of design variables, and the size of the design domain. Gradient based approaches, such as sequential linear/quadratic programming, quasi-Newton, and feasible directions, yield fast convergence but require a gradient evaluation and are susceptible to local optima [33]. Non-gradient based approaches, such as genetic algorithms and particle swarms, are inspired by organic systems, can be considered robust by exploring much of the design domain, but typically converge slowly with high computational cost [34]. Unique heuristic methods are less common and problem specific, but may yield better results than traditional approaches [35,36]. Exhaustive search, though not a traditional optimization algorithm, is an effective and robust method for determining the global optima within a small design domain or if computational cost is not a concern. Additionally, Response Surfaces and Neural Networks are approximation tools that can be used in combination with several of the above algorithms to further decrease convergence time and computational cost [33,37,38].

2.2. Applications in Designing Meta- Materials

The following section describes applications of the previously defined methods in designing meta-materials. A large number of meta-material design problems were found in the literature. The literature sought to optimize properties of mechanical, piezo-electrical, thermal, fluid, or acoustic nature. In an effort to down-select, focus was placed on the type of application, complexity of the problem, and relevance to the current research. Since the properties being targeted in the current work are mechanical in nature, that will be the focus of applications presented herein.

2.2.1. Applications of Topology Optimization

Czech et al. in [39] designed a meta-material with targeted shear properties to serve as the shear band in a non-pneumatic wheel. This optimization process had two levels: a top-level wheel optimization targeting meta-material thickness and shear strength required to meet design specifications, and a bottom-level topology optimization targeting shear strength and meta-material thickness obtained in the top level, but using a different constitutive material. Traditional asymptotic homogenization (or HM) assumes the size of the unit cell is significantly smaller than the design domain in order to approximate homogenized material properties. The authors instead used a variant of the HM, called Volume Averaging Method, in combination with a SIMP approach to solve this IHP. The authors provide a detailed explanation of how the Volume Averaging Method relaxes the HM limitation between unit cell size and global design domain to ensure accuracy of the global optimization analysis in [10]. Three linearly elastic material models (polycarbonate, steel, and aluminum) were optimized and compared with an objective of minimizing volume and constraints of average and maximum contact pressure. A genetic algorithm coupled with a neural network was used for the top-level optimization procedure while the developed topology optimization scheme was used for the bottom-level optimization.

In [40], Gibiansky and Sigmund optimized a three-phase meta-material for extreme bulk modulus in order to determine minimum compliances across different volume fraction constraints. The novelty in this work is the emphasis on a three-phase meta-material. All the previous literature is in reference to a two-phase optimization (typically material and void). Thus, a three-phase meta-material is inherently a composite meta-material consisting of at least two different materials and void. The objective function maximized the bulk modulus of the periodic unit cell with constraints on the volume fraction of each phase, a symmetric effective elasticity tensor, and controls for numerical stability. The authors state this to be an IHP, however the presented optimization setup contradicts this statement as material properties are maximized instead of constrained. One example of a three-phase optimized meta-material resulting from this work is shown in Figure 2.9.

Carstensen et al. is the most recent attempt (as of the time of this review) to address both geometric and material nonlinearity in TO in [41] and apply it to a meta-material design problem aimed at maximizing energy absorption. In their approach, equilibrium is described in terms of the residual force vector to be solved by an iterative nonlinear finite element solver. The authors decouple the design domain from the finite element domain to eliminate numerical issues presented in large deformations of void elements when computing the objective function (strain energy). Geometric nonlinearity is appropriately accounted for by replacing the Cauchy stress tensor (used for small deformation) with the second Piola Kirchoff stress tensor (used for large deformation). To account for material nonlinearity, the authors relate the elastic constitutive matrix, plastic hardening modulus, and yield stress for each element based on the Von Mises yield function with isotropic hardening. The authors then apply their proposed setup to design a meta-material consisting of a bulk metallic glass material which typically has brittle failure in bulk form (feature size > 1 mm) but microscopically ductile. The objective function maximizes energy absorbed with constraints on volume fraction, minimum feature size, and periodic boundary conditions. Numerical and experimental comparisons of the TO results to those of a regular hexagonal honeycomb structure with equivalent volume fraction are presented.

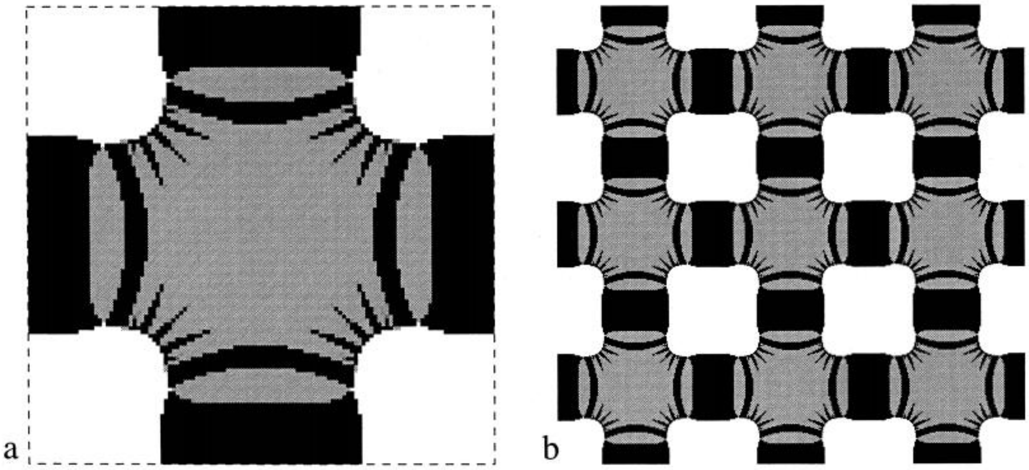


Figure 2.9. A Three-Phase Meta-Material (a) Unit Cell, and (b) in a 3x3 Array via TO [40]

The designed meta-material was shown to exhibit much better energy absorption behavior than the honeycomb, (though it would be interesting to compare the designed meta-material to the SO honeycomb designed in [42] with the same objective). While the authors present a high-level overview of their nonlinear approach in TO, a much more rigorous mathematical formulation is necessary to validate this method. As it stands, the authors present a useful extension in TO theory to account for nonlinear problems.

2.2.2. Applications of Size Optimization

Mehta in [43] performs SO on a proposed contact-aided regular and auxetic honeycomb topology to determine if such contact can cause stress relief as compared to the traditional alternatives. The optimization setup consists of a multi-objective function maximizing stress relief and gap height with three geometric manufacturing constraints. The Non-dominating Sorting Genetic Algorithm (NSGA-II) was chosen due to its ability to handle two objectives efficiently and discover the Pareto curve. Mehta applies these proposed compliant mechanism topologies in a morphing airfoil example.

Shultz et al. performs SO in [42] to design a honeycomb meta-material for maximized specific energy absorption experiencing in-plane crushing. The optimization procedure consists of maximizing specific energy absorption with a geometric manufacturing constraint and a free geometric variable while maintaining constant material volume. The Multi-Island Genetic Algorithm was chosen in combination with a Response Surface to explore the design domain. An in-depth explanation for choosing the specific genetic algorithm was not provided and it was noted that other algorithms may have been chosen. The unit cell can be seen before (left) and after (right) undergoing SO in Figure 2.10.

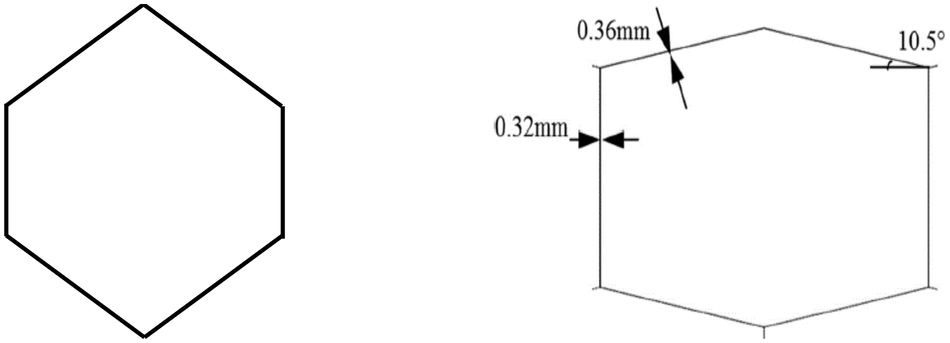


Figure 2.10. Unit Cell of a Honeycomb Unit Cell before (left) and after (right) SO for Maximum Energy Absorption [42]

Choi and Patel, in [44], propose a method to reliably handle random or uncertain constraints in the design of meta-materials. In one of their examples, the authors apply SO to a maximum stiffness truss structure problem with a pre-defined topology. The objective function is to minimize strain energy with member thickness and overall volume constraints. A gradient based Sequential Quadratic Programming algorithm was used due to the low number of design variables and availability of an analytical model of the system. The authors discuss probabilistic failure based on loading uncertainty in their results.

2.3. Conclusions

TO is a numerical method to distribute material within the design domain in an optimal layout to meet an objective. TO using the HM is the only method discovered in the literature to design meta-materials. Size optimization is a method that has been used in the literature to tune meta-materials with a pre-defined unit cell topology to meet an objective. Therefore, this method optimizes the dimensions of an existing unit cell topology instead of designing a novel one. Based on this literature review, TO with the HM will be used to design a meta-material to replace the tank track pad.

CHAPTER 3. DESIGNING VIA TOPOLOGY OPTIMIZATION

3.1. Objective

An existing commercially available TO tool was chosen to solve the meta-material design problem. The chosen TO tool was Optistruct 12.0 by Altair Engineering, Inc. This tool was chosen due to its use of a vetted HM formulation with SIMP, widespread use in industry, and availability. The software enables several manufacturing constraints including minimum feature size, maximum feature size, and constant cross-section. Optistruct is a solver that works in tandem with HyperMesh, an advanced meshing software from the same company.

3.2. TO in Optistruct

In order to determine the feasibility of using Optistruct to solve the design problem, two hypothetical tests were run. These test setups and their results are described in the following sub-sections.

3.2.1. Unit Cell Setup

The premise of the design problem is to match a nonlinear deformation curve under uniaxial loading with a meta-material. An example Unit Cell (UC) geometry with boundary conditions and a uniaxial load is shown in Figure 3.1. A uniform pressure, P, was applied on one end for the uniaxial load. Boundary conditions were enforced as shown to restrict movement in the y-direction and for symmetry in the x-y plane. Thus, a resulting topology would need to be mirrored in the x-y plane to fully represent the UC. The geometry was initially chosen with dimensions 10mm x 10mm x 1mm and the material was chosen to be common steel (E = 210 GPa, ρ = 7850 kg/m^3, μ = 0.30).

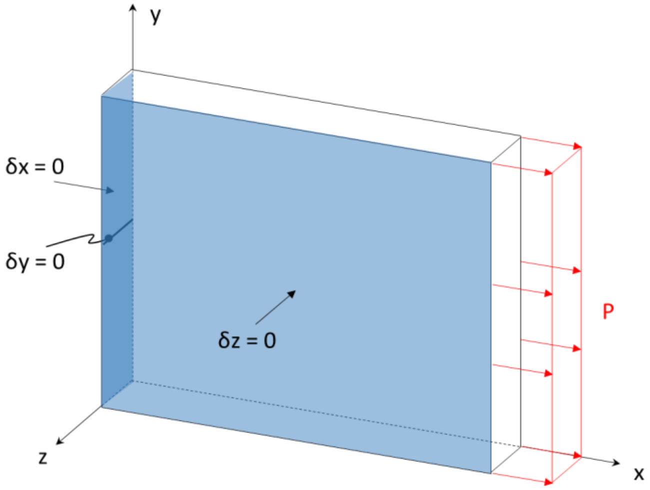


Figure 3.1. Initial UC with boundary conditions and uniaxial loading

This UC, along with the loading and boundary conditions, was created within HyperMesh as shown in Figure 3.1. In this figure, the green volume represents the design space, in which material can be distributed per the TO algorithm. The yellow volumes

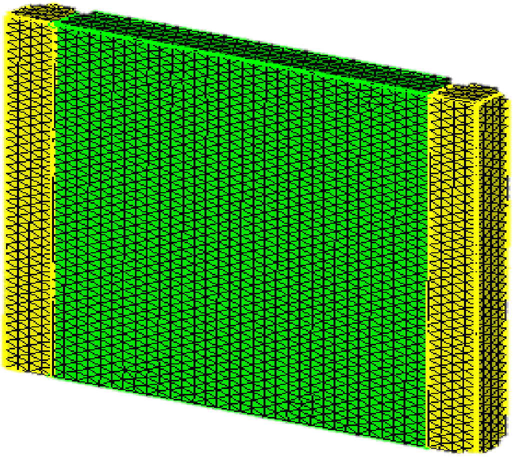


Figure 3.2. Representative UC in Optistruct

represent the non-design space where solid material must remain to ensure the boundary conditions and pressure loading locations remain intact throughout the optimization.

3.2.2. Single Load Case

The first test in Optistruct was constrained to target a single compliance given a single load case. Compliance, the inverse of stiffness, is a measure of strain energy as defined below as

$$C = \frac{1}{2} u^T K u = \frac{1}{2} \int_V \varepsilon^T \sigma dV \tag{3.1}$$

Compliance was used due to its availability in Optistruct as a response variable. The applied pressure was P_1 = 2,000 MPa and the targeted compliance of the UC was 3,685 MJ, which represents 5x the displacement of the same geometry with solid steel. A geometric nonlinear analysis was used to solve the equilibrium equations. The corresponding optimization setup can be seen in equation (3.2) below.

$$\begin{aligned} \min_{x}: &\ f(x) = \frac{V(x)}{V_0} \\ subject\ to: &\ C(P_1) = 3685 \text{ MJ} \\ : &\ KU = F \\ : &\ 0 < x_{min} \leq x \leq 1 \end{aligned} \tag{3.2}$$

The software was able to converge to a solution to this optimization problem and the resulting UC topology is shown in Figure 3.3 with color indicating element densities. The topology was not consistent through the thickness due to the boundary condition along the x-y plane preventing displacement in the z-direction. The UC geometry was then tested across several pressure loads to determine its stress-strain response. This response is shown in Figure 3.4. Note that a pseudo-strain in terms of overall UC deformation was calculated as

$$\%\ Deformation = \frac{\delta_{avg}}{L}(100) \tag{3.3}$$

where δ_{avg} is the average displacement at the end of the UC where the pressure is applied and L is the length of the UC (10 mm).

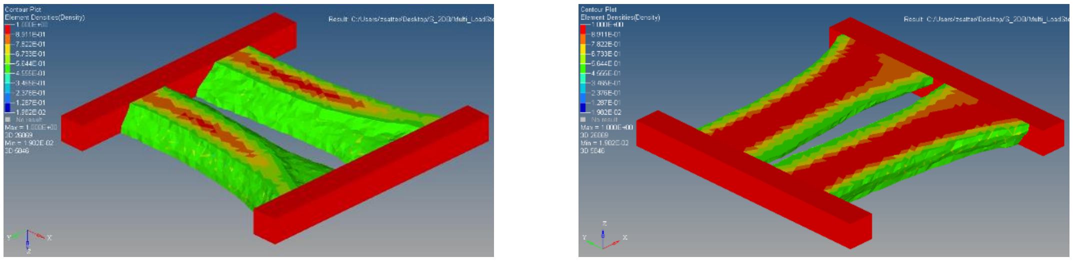


Figure 3.3. UC topology solution to single load case, (right) top, and (left) bottom

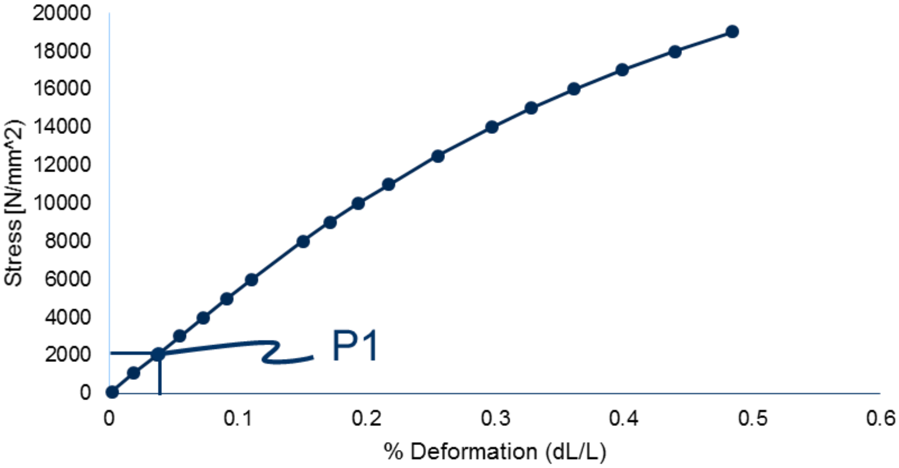


Figure 3.4. Uniaxial tension curve for UC with single targeted compliance

3.2.3. Multiple Load Case

The second test was constrained to target multiple compliances corresponding to multiple load cases. The two load cases were P_1 = 2,000 MPa and P_2 = 10,000 MPa with corresponding compliances of 3,685 MJ and 97,000 MJ, respectively. This second loading and compliance were chosen arbitrarily to determine if the deformation response of the first TO solution could be tuned, i.e. as shown in Figure 3.5. In this figure, the blue curve is the response of the first TO solution and the red curve is a hypothetical target response corresponding to the two compliances mentioned previously.

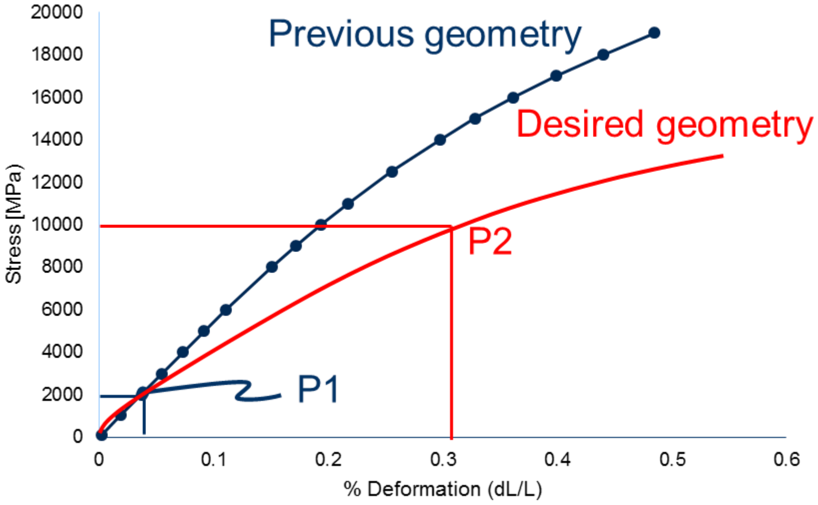


Figure 3.5. Hypothetical uniaxial tension curve with two targeted compliances

As in the first test, a geometric nonlinearity analysis was used to solve the equilibrium equations. The corresponding optimization setup can be seen in equation (3.4)

$$
\begin{aligned}
\min_{x}:&\ \ f(x) = \frac{V(x)}{V_0} \\
\textit{subject to}:&\ \ C(P_1) = 3685 \text{ MJ} \\
:&\ \ C(P_2) = \ 97000 \text{ MJ} \\
:&\ \ KU = F \\
:&\ \ 0 < x_{min} \le x \le 1
\end{aligned}
\qquad (3.4)
$$

Unfortunately, Optistruct was unable to converge to a solution for this second optimization with two targeted compliances and did not yield a resulting UC topology.

3.3. Discussion

Based on the inability for Optistruct to yield a solution to the multiple compliance problem, it was determined that the software was not suitable to solve the intended meta-material design problem. After closer examination of both the theoretical capabilities of the HM and current capabilities of the chosen software, several reasons surfaced to explain why this occurred. Three primary limitations were discovered, including geometric nonlinearity, periodic boundary conditions, and aspect ratio of the UC. These issues are described in detail below.

3.3.1. Geometric Nonlinearity

It is important to frame the objective of the overall design problem in question. As mentioned in section 1.3, the goal of replacing the rubber track pad with a meta-material with some elastic constitutive material is driven by the idea of replacing material nonlinearity with geometric nonlinearity. With this in mind, a TO tool must be able to account for geometric nonlinearity in both the homogenization formulation and in solving for the equilibrium.

It is important to note the difference between solving a nonlinear finite element analysis and accounting for geometric nonlinearity in the homogenization formulation of the TO algorithm. A nonlinear finite element analysis, which Optistruct has the capability to solve, is the iterative solver that steps through increased load steps or displacements until the intended load or displacement is reached [45]. This is a well-known and explored theoretical topic in the finite element method (FEM) that has been widely implemented in commercial FEM packages, including Optistruct. Meanwhile, accounting for geometric nonlinearity in the mathematic TO formulation is not as trivial and well explored.

3.1.3.1. Difference in Linear and Nonlinear TO Formulation

In the formulation for linear elasticity under the small deformation assumption, stress and strain can be simplified to the Cauchy stress tensor, σ, and the infinitesimal strain tensor, ε, respectively. The infinitesimal strain tensor can be denoted as

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial d_i}{\partial x_j} + \frac{\partial d_j}{\partial x_i}\right) \tag{3.5}$$

The foundation of the HM lies in the homogenized stiffness tensor, C_{ij}^H, which represents the global constitutive properties of the homogenized media in question. In the linear elastic case, this tensor is determined by the following equation

$$C_{ij}^H = \frac{1}{\Omega}\sum_{e\in\Omega}(d_0^{e(i)} - d^{e(i)})^T K^e(\phi)(d_0^{e(j)} - d^{e(J)}) \tag{3.6}$$

where $d_0^{e(i)}$ is the nodal displacements for the i^th element, K^e is the element stiffness matrix, and ϕ represents the design variable. Note that in (3.6) the displacements correspond to the infinitesimal strain tensor, ε [41].

In continuum mechanics, geometric nonlinearity can be defined by significant changes in shape and location between the initial, **X**, and final, **x**, configurations [46]. This can be appropriately accounted for by using the second Piola Kirchoff stress, S, and the

Green-Lagrangian strain tensor, E_{ij}. The Green-Lagrangian strain tensor is defined as E_{ij} [38, 43] below

$$E_{ij} = \frac{1}{2}\left(\frac{\partial d_i}{\partial X_j} + \frac{\partial d_j}{\partial X_i} + \frac{\partial d_k}{\partial X_i}\frac{\partial d_k}{\partial X_j} \right) \tag{3.7}$$

In geometric nonlinearity, the homogenized stiffness tensor C_{ijkl}^{H} must be calculated as a tangent to the current strain state using the Green-Lagrangian strain tensor at the currently deformed state as shown in the equation below

$$C_{ijkl}^{\tan} = Kg(E_e)C_{ijkl}^{0} + K\frac{1}{E_e}\frac{\partial g(E_e)}{\partial E_e}C_{ijkl}^{0}E_{kl}^{t}\frac{\partial E_e}{\partial E_{kl}^{t}} \tag{3.8}$$

which is derived from taking the derivative of the second Piola Stress tensor with respect to the Green-Lagrangian strain tensor [47]. The homogenized stiffness tensor, C_{ijkl}^{H}, is required to numerically represent the effective properties of the microscopic structure created by the design variables. It is then used to relate these properties in the finite element domain for solving of the equilibrium equations in the finite element analysis.

The nonlinear terms of stress and strain would further impact the optimization scheme in TO depending on how the objective function is formulated. Since the optimization algorithm is typically gradient-based, sensitivities of the design variables must be calculated by taking the partial derivative of the objective function with respect to the design variable. If the objective function is formulated to maximize some homogenized material properties, or C_{ijkl}^{H}, then linear or nonlinear stress and strain tensors matters in the complexity of the sensitivity derivation. The reviewed papers that have done this have used the adjunct method to derive the sensitivity equations [47,48]. In terms of the current meta-material design problem, the optimization can be set up by constraining the target material properties instead of optimizing for them. Thus, this issue can be circumvented in this case.

3.1.3.2. Consideration of Material Nonlinearity

The sources that consider and derive the equations for geometric nonlinearity simultaneously take material nonlinearity into account. The constitutive material will exceed yielding in many cases where geometric nonlinearity has a pronounced effect. In these cases, it would be critical to take material nonlinearity into consideration. However, in the intended track pad meta-material problem, it is required that the final structure should not yield in the range of deformation it undergoes as it should recover fully after each cyclic loading. Thus, material nonlinearity is not required in the TO formulation as the maximum stresses seen in the resulting structure will be below yielding of the constitutive material by design.

3.3.2. The Nonlinear Inverse Homogenization Problem

The IHP class of TO problems was introduced 2.1.1.3 as a way to target specific material properties instead of minimizing or maximizing them. However, closer inspection of the reviewed literature reveals that the IHP has only been used to target linearly elastic materials. For instance, Sigmund in [24] and Diaz and Bénard in [27] both targeted a single stiffness tensor under the assumption of linear elasticity. Most recently, Czech targeted a single shear modulus in [49] to design a meta-material UC under the assumption of linear elasticity even though strains of up to 10% were observed.

In order to solve the current problem, there exists several targeted stiffness tensors that represent the tangent to the stress-strain curve at different strain levels. Therefore, this problem represents a nonlinear IHP. A visual comparison of the target responses for a linear and nonlinear IHP can be found in Figure 3.6. The optimization setup for such a nonlinear IHP is not trivial as to how to implement multiple targeted stiffness tensors at different strain levels in terms of constraints. The complexity arises as the target stiffness tensor becomes strain-dependent in the nonlinear case where it is not so in the linear case. As of the time of this writing, a nonlinear IHP has not yet been formulated or solved in the literature.

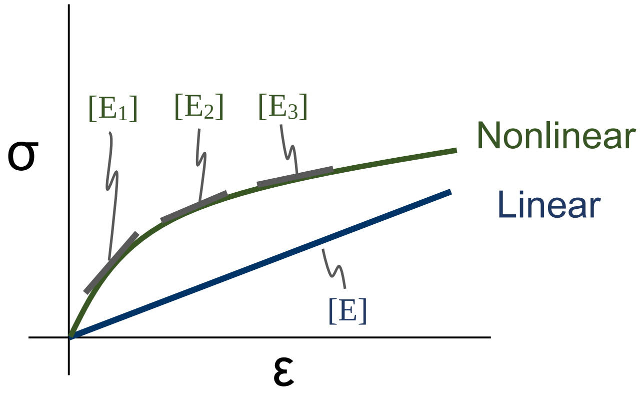


Figure 3.6 Illustrative Comparison of Target Responses in a Linear and Nonlinear IHP

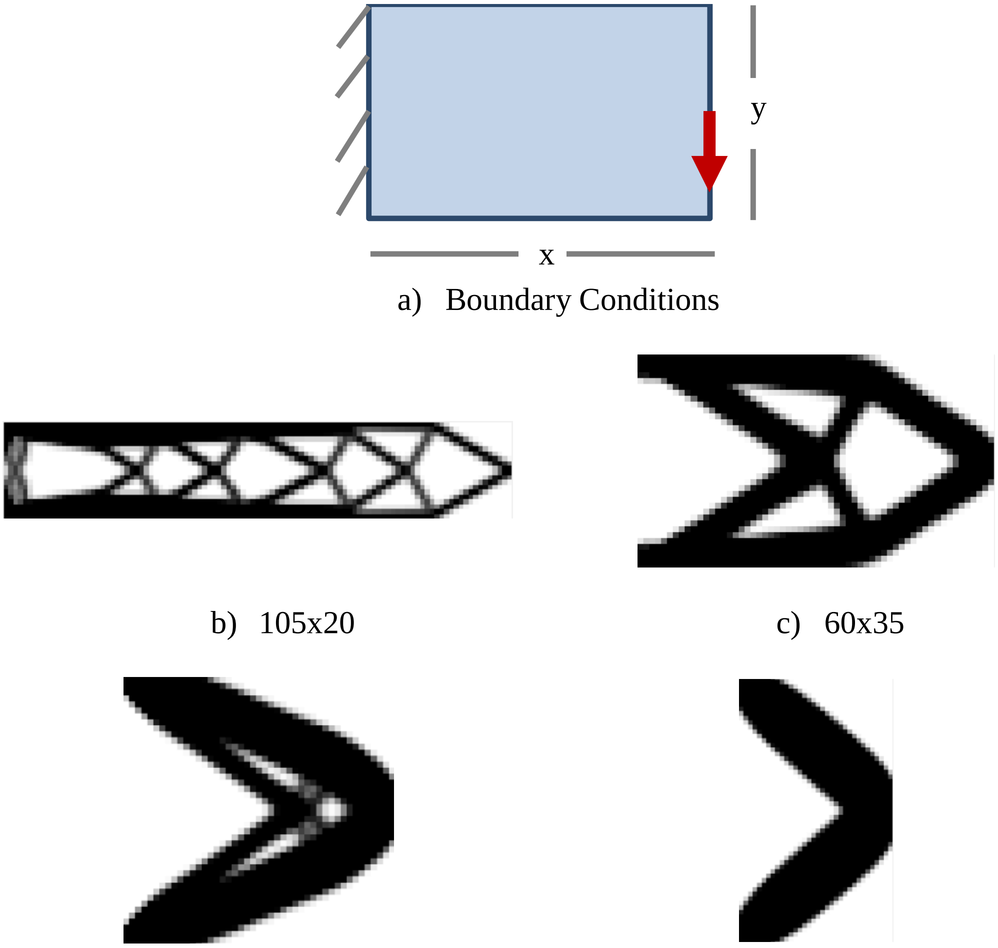
3.3.3. Periodic Boundary Conditions

As mentioned in section 2.1.1.4, one of the primary concerns in tailoring TO for the design of meta-materials is applying periodic boundary conditions to the UC. This is not a new or complex feature in the context of implementing in a finite element analysis, as it has been done in [27,39,50,51] among others. The applied boundary conditions presented in 3.2.1 were ill posed to effectively simulate periodic connectivity with other

UCs. Unfortunately, Optistruct does not currently have the ability to apply such boundary conditions and therefore limits the ability to design a meta-material UC accordingly.

3.3.4. Unit Cell Aspect Ratio

A more obscure limitation of TO in the design of meta-materials is its inability to consider aspect ratio of the UC, or design space, as a design variable. This is a downfall of the overall method of TO and not Optistruct specifically. Even in linear elasticity examples,



a) Boundary Conditions

b) 105x20

c) 60x35

d) 46x46

e) 30x65

Figure 3.7. Linear elastic TO a) boundary conditions with aspect ratio “x” by “y” and b-e) results with different aspect ratios

it can be seen that the aspect ratio of the design space can change the resulting topological design.

The results in Figure 3.7 were created via the 88-line TO Matlab code readily available in [52] where the author uses linear elasticity with an objective of minimizing strain energy. The top figure (a) shows the cantilever beam boundary conditions used. The following four images (b-e) show the various results due to changes in aspect ratio. Note that the same number of elements (2100), volume fraction (0.4), minimum radius (1.5), and penalization factor (3.0) were used in each example. This figure serves to illustrate the importance of considering aspect ratio of the design space when performing TO.

In most design problems not considering meta-materials, such as the one in Figure 3.7, changing the aspect ratio of the design space changes the physical problem, thus aspect ratio is not at issue. However, this is not the case in the design of a meta-material as the design space of a UC is arbitrary and unknown before a solution is obtained. Bénard and Diaz state in [26] that the choice in UC geometry can limit the solution set of achievable meta-materials. Since a change in the aspect ratio of a UC design space changes the inherent TO problem, the UC aspect ratio must be considered as another design parameter when performing TO in the design of meta-materials. As of the time of this writing, the author has not found any literature that explores the affect that changing the aspect ratio of the design space has on the existence and uniqueness of solutions using TO in the design of a meta-material UC.

3.4. Conclusions

The results from this chapter directly answer research question 2.a and associated hypothesis, reproduced below.

> R2. What method can be used to successfully develop a meta-material to meet the nonlinear deformation response of the current application?
>
> a. Is topology optimization a feasible method to accomplish this?
>
> H2. A meta-material can be developed using topology optimization to meet the material behavior requirements.

The literature review in Chapter 2 indicates that the only method to design meta-materials is TO. It is determined that it may be possible to use TO to solve the current meta-material design problem by appropriately addressing the issues discussed in the previous sections. This might be accomplished by developing the necessary TO formulation and developing a code accordingly. However, such a formulation and its associated code would be highly complex and outside the scope of the current work due to time limitations. Based on the results that Optistruct cannot be used in the design of the meta-material UC, it is determined that another tool or method must be used to obtain a solution. Thus, for the purposes of this work, the second hypothesis was proved false.

CHAPTER 4. DESIGNING VIA ENGINEERING PRINCIPLES

4.1. Objective

A new method is needed to design meta-materials based on some physical understanding of the base UC and its periodic interaction with adjacent UCs. Therefore, engineering principles were used to obtain this understanding and aid in the design of a meta-material UC that exhibits the same deformation as the targeted rubber response.

This chapter is organized to cover the following:

- An explanation of the evaluation method of a uniaxial meta-material response
- Description of the evolution of the initial “Brick” UC design
- Description of the evolution of the final “BrickOval” UC design
- Discussion of other considerations when designing and evaluating meta-material UCs
- Conclusions on the results and how they address the research questions

4.2. Method of Evaluation

When replacing a solid, homogenous material with an equivalent meta-material structure, the targeted equivalent properties and a method to evaluate such properties must be determined. In the case of the current elastomer, a 2nd order Ogden hyperelastic model can be used to fully define the homogenous mechanical properties in uniaxial tension and compression, equibiaxial tension, and pure shear depicted in Figure 4.1. The resulting stress-strain curves are representative of internal forces in the elastomer as a homogeneous continuum and can be related to the 2nd Piola Kirchoff stress tensor at different strain levels, as done by Dangeti in [3].

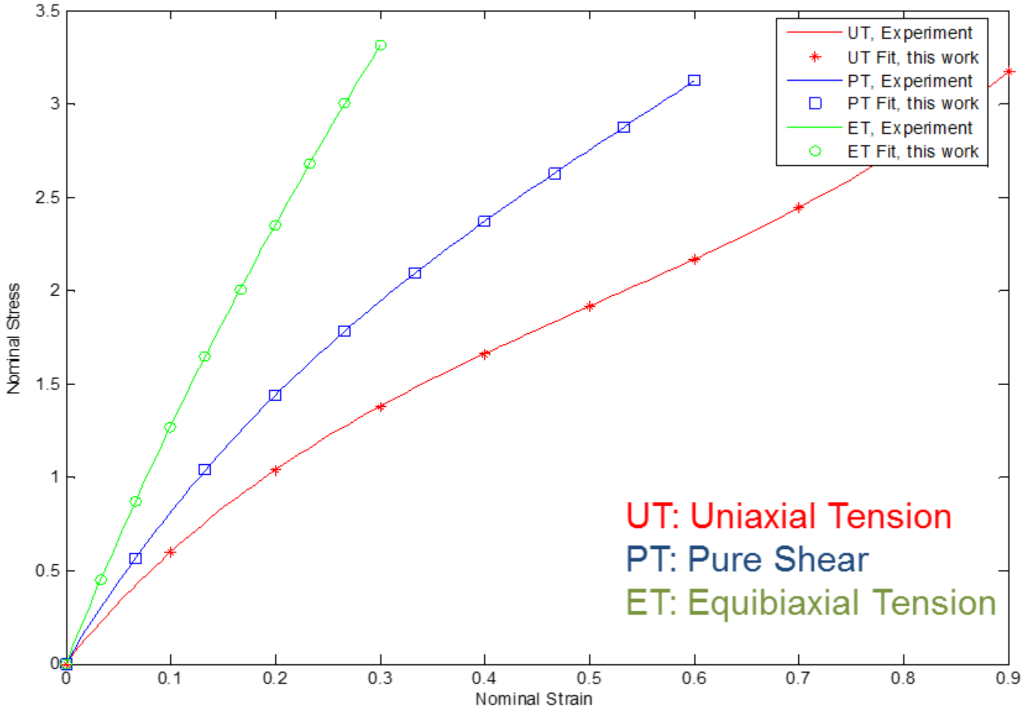


Figure 4.1 Current Elastomer Stress-Strain Response [3]

These stresses and strains correspond to the infinitesimal element of the rubber pad indicated in Figure 4.2 (a) after deformation with the appropriate load condition. Therefore, an equivalent replacement to this rubber pad might experience the same response characterized by these infinitesimal stresses and strains. However, the stress-strain relationship in a meta-material is not valid, as shown in Figure 4.2 (b) after deformation. This is invalid for the meta-material due to its combination of material and void at the infinitesimal level. Ultimately, a meta-material is a structure combined in a patterned continuum and its properties must be analyzed on a larger scale than classical elasticity. Thus, a different method must be chosen to compare the rubber properties to that of the designed meta-material.

The ultimate goal in the current design problem is not to replace the rubber pad with a meta-material that exhibits the same equibiaxial tension or shear characteristics. Instead, it is to replace the pad with a meta-material that has the same behavior in the primary loading condition,

compression. Thus, the evaluation method can be relaxed to focus only on a comparison of compression response. Therefore, a uniaxial deformation calculation, or % vertical deformation, can then be defined as a so-called "meta-strain" as shown below

$$\text{Meta} - S\,\text{train} = \%\ \text{Vertical Deformation} = \frac{\delta}{H}(100) \qquad (4.1)$$

where H represents the total height of the rubber or meta-material pad and δ represents the vertical displacement after deformation, as shown in Figure 4.2 (a) and (b). The meta-strain will be calculated for every load case corresponding to the range of the target response and will allow for direct comparison between the rubber and meta-material.

In order to compare the meta-material, the % vertical deformation target values must be determined for the rubber pad. The analysis setup is shown in Figure 4.3 with the rubber pad on top of a rigid body. Only half of the rubber pad in the x-direction (67 x 21 mm instead of 134 x 21 mm) was modeled to ensure the mesh was not biased about the center and to decrease simulation time. Therefore, boundary conditions were imposed on the right side of the pad to ensure no translation in the x-direction. Additionally, the rigid body was used to allow the rubber to translate in the negative x-direction without friction as it deformed. A radius of 2 mm was used on the bottom corner of the rubber pad to allow

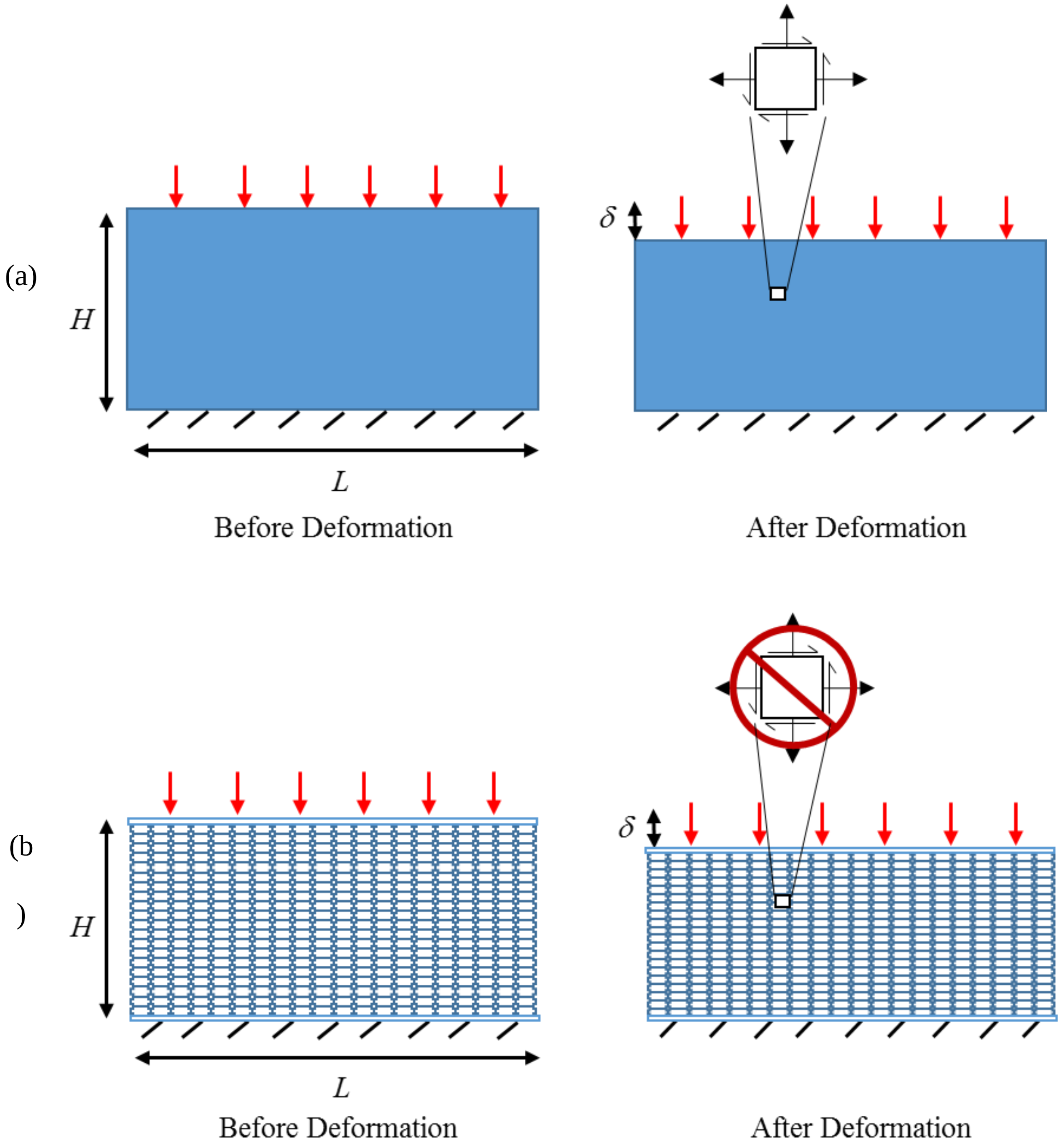


Figure 4.2 Example Methods of Determining Material Properties of (a) Rubber and (b) Meta-Material

deformation at higher strains without excessive element-level deformation at the corner of the pad and prevent the need for adaptive meshing.

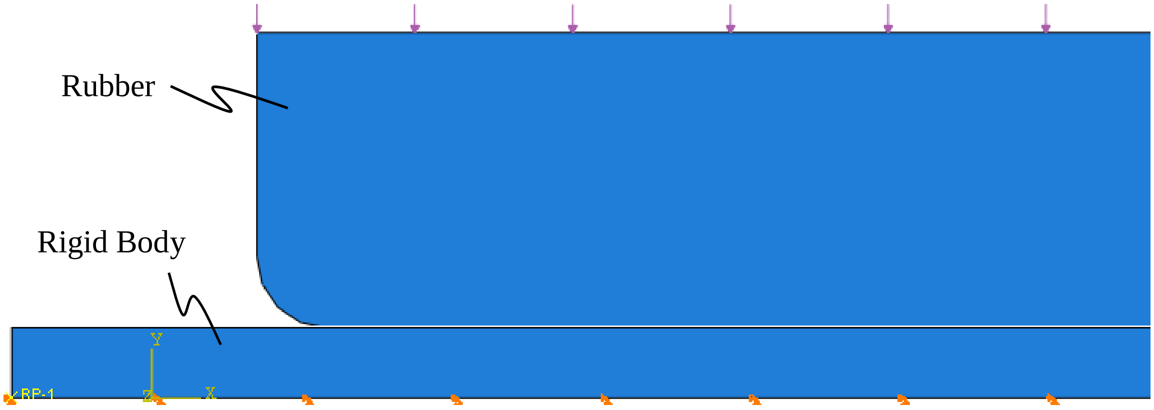


Figure 4.3 Load and Boundary Conditions for Rubber Pad

Pressures were applied in compression to the rubber pad corresponding to the 4 stresses listed in Table 4.1. The infinitesimal strain values of the rubber material model corresponding to the given stresses are shown in comparison with the % vertical deformation at the center of the rubber pad from this analysis. A comparison in terms of % difference between infinitesimal strain and % vertical deformation shows that there is little to no difference between the values of these two types of target responses (within 2%) despite the methods used to achieve them. This small variation can be attributed to numerical anomalies of running the finite element simulation. Due to the small difference between these two sets of values, the author has chosen to use the original strain values as

Table 4.1 Target % Vertical Deformation Values

Stress [Mpa]	Strain [-]	% Vertical Deformation [-]	% Difference
-0.3817	5.000%	5.073%	-1.459%
-0.8384	10.000%	9.841%	1.587%
-2.0632	20.000%	20.135%	-0.677%
-3.9327	30.000%	29.495%	1.682%

targets for comparison with that of the meta-material designs. These target % vertical deformation values can be seen as a response to the applied stresses in Figure 4.4.

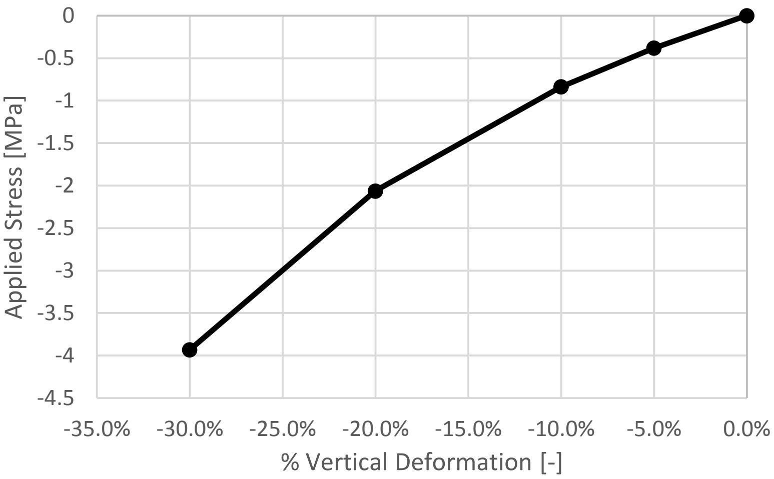


Figure 4.4 Uniaxial Compression Target Response

4.3. Design of the Brick Unit Cell

The following subs-sections explain the process in designing the initial concept UC based on engineering principles.

4.3.1. Element Geometry

Since the objective of this application is to replace a nonlinear material with a nonlinear geometry, it is appropriate to understand this geometric nonlinearity in structures. One such structure is the Fixed-Fixed Beam (FFB) with its free body diagram shown in the top left of Figure 4.5. This figure also shows how the geometric nonlinearity is affected based on the aspect ratio, L/h, of the beam. As this ratio increases, the structure's geometric nonlinearity becomes increasingly more pronounced. Note that this figure intentionally does not show numerical values as this nonlinear behavior is true across different elastic materials, though the force-displacement values will change. In order to accomplish the desired large deformations needed to replace the elastomer, it is proposed to use this FFB in a UC and achieve large deformations via bending.

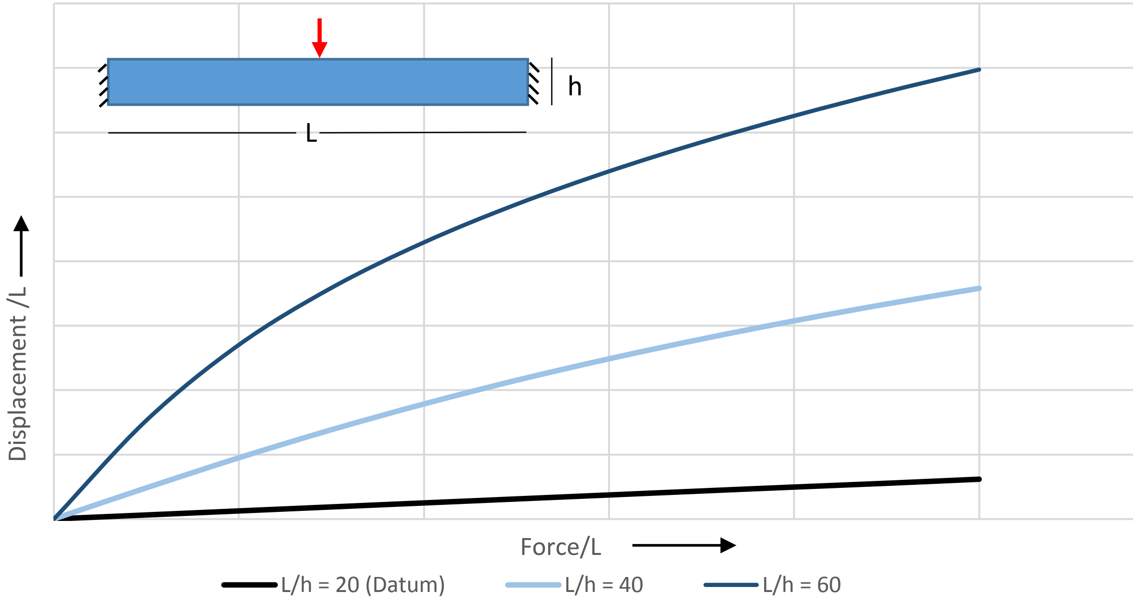


Figure 4.5 FFB Free Body Diagram and Geometric Nonlinearity based on Aspect Ratio

4.3.2. Integration into a UC

A UC is now designed with a focus on integrating the FFB. In order for a UC to contain a FFB, the boundary conditions of this element geometry must be preserved. For the purposes of a UC, the applied point load can be approximated as a localized distributed load. The fixed ends and applied load can then be enforced via the same member, or strut. The FFB with boundary conditions as enforced by three struts can be seen in Figure 4.6. The FFB is shown here with a high aspect ratio to allow for geometric nonlinearity. The strut should correspondingly have negligible deformation as compared to the FFB. This will enable the FFB to act as the primary mode of deformation in the UC and allow the designer to isolate and modify its parameters appropriately.

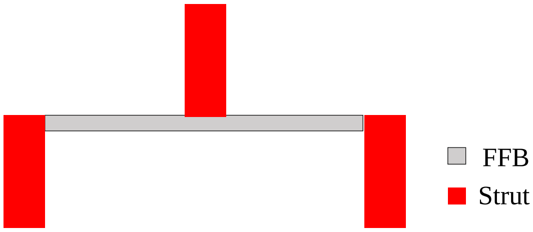


Figure 4.6 FFB with Boundary Conditions Preserved

Once boundary conditions were enforced, considerations were given to UC periodicity and connectivity. The strut was therefore modified as shown in Figure 4.7. A repeating UC now emerges from this periodicity, denoted as the "Brick" concept. This concept uses a meta-material layer shift of half-UC length to impose the load from layer 1, via strut a, to cause bending in the FFB, b, which is constrained to both struts c_1 and c_2 in layer 2. These layer 2 struts would then impose the load on the next layer. In this way, the load path continues throughout the meta-material where the strut acts as the boundary condition and load source for the current and successive layer, respectively.

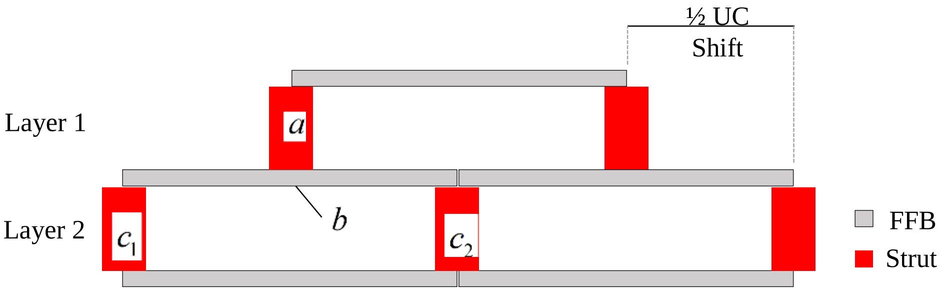


Figure 4.7 Modified Strut to Allow UC Periodicity and Connectivity

4.3.3. Evaluation of Brick UC Concept

The “Brick” UC concept must now be evaluated. The “Brick” UC is shown in Figure 4.8 with its four design parameters. Parameters t_1 and L are most closely tied with the FFB while t_2 and H modify the strut.

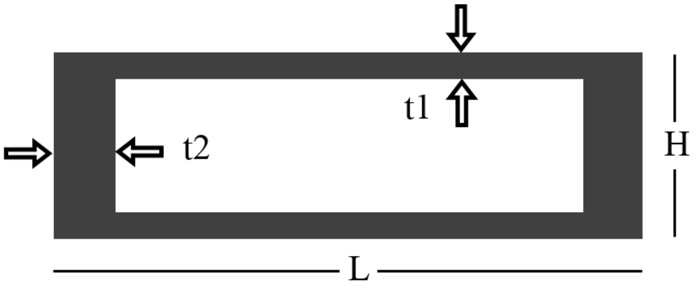


Figure 4.8 "Brick" UC with Parameters

4.3.3.1. <u>UC Tessellation</u>

The “Brick” UC was then tessellated into a 4x4 meta-material array with loading and boundary conditions as shown in Figure 4.9. The existing track pad has approximate dimensions of 136 x 21 mm in the two-dimensional cross-section of interest. The number of UCs chosen in the tessellation was based on the initial dimensions chosen for the parameters and the available design space of the current application. It was acknowledged that the number of UCs required to fill this design space would change as the parameters of the UC changed. Thus, iteration would be carried out with respect to optimal UC parameters and tessellation as needed. The boundary conditions were imposed such that the bottom layer was fixed and the vertical sides of the exterior UCs could only translate in the y-direction. These boundary conditions on the sides of the meta-material were imposed to prevent collapse of the overhanging UCs.

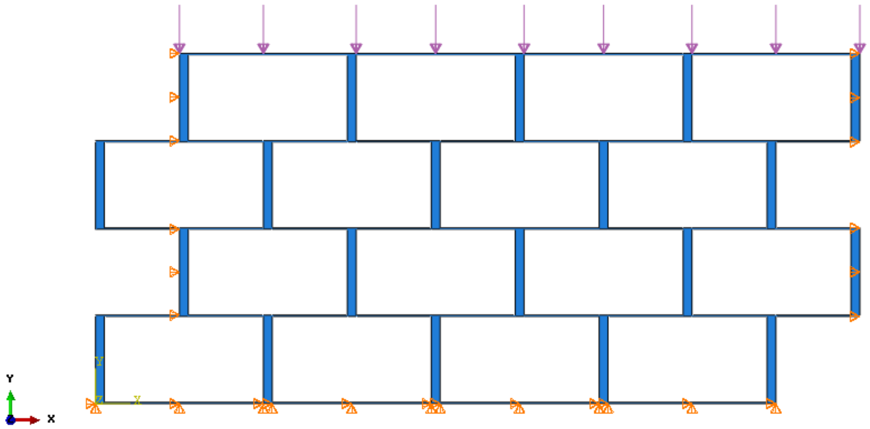


Figure 4.9 "Brick" UC in a 4x4 Tessellation

4.3.3.2. Sensitivity Analysis

A sensitivity analysis was conducted of the 4x4 “Brick” UC tessellation to determine the feasibility of modifying the UC parameters such that the vertical deformation response converges towards that of the target. The sensitivity analysis only analyzed three of the four “Brick” concept parameters. The strut thickness, t_2, was not analyzed since the strut was designed to not have an effect on vertical deformation. The strut will only displace due to rigid body motion. Even though overall height of the UC, H, modifies the strut, it is analyzed due to its contribution to the total height of the tessellated meta-material. This contribution effects the vertical meta-material deformation properties defined in (4.1). The contribution of UC height, H_{UC}, in the total height of the 4x4 tessellation is shown below as

$$H_{Total} = 4(H_{UC}) - 3(t_1) \tag{4.2}$$

where H_{Total} is the total height of the meta-material. The height is not simply four times UC height since there are overlaps in the UCs as they share the same thickness, t_1. The sensitivity analysis performed was a Full Factorial. The parameters are shown in Table 4.2 with their high and low values. All dimensions are shown in millimeters. The Python script used to perform this analysis is included in Chapter 7.

Table 4.2 Full Factorial Parameters and Values

	High (1)	Low (0)
L [mm]	15.0	10.0
H [mm]	5.0	3.0
t1 [mm]	0.100	0.075

The results of the Full Factorial analysis are shown in Figure 4.10 and compared to the target curve. The legend indicates the UC parameters as high or low with the order of "L, H, t_1." In other words, curve "101" refers to the "Brick" UC with a length of 15.0 mm,

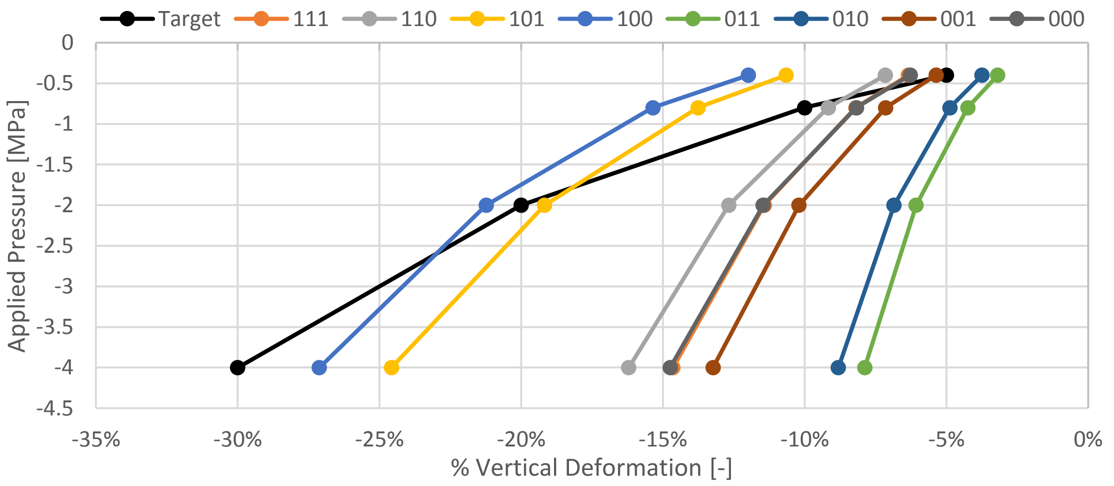


Figure 4.10 Full Factorial Analysis of 4x4 "Brick" Tessellation

height of 3.0 mm, and t_1 thickness of 0.100 mm. A constant strut thickness, t_2, of 0.5 mm was maintained through all tests. All simulations were evaluated with a constitutive material of generic steel with a linear elastic material model (E = 210 GPa, ρ = 7850 kg/m^3, μ = 0.30). The results in Figure 4.10 show that increasing UC width and decreasing thickness t_1 both increase the % vertical deformation of the 4x4 meta-material. Increasing the height of the UC decreases the overall % vertical deformation.

4.3.3.3. Conclusion on Feasibility

The results of Full Factorial analysis of the "Brick" UC parameters indicate that no combination of parameters will enable convergence of the nonlinear response towards that of the target. Each factorial response exhibits higher stiffening as additional load is applied than that of the target curve. There is no evidence from the factorial responses that this rate of stiffening can be modified. Therefore, the "Brick" UC concept is deemed infeasible in matching the target response.

4.4. Design of the BrickOval Unit Cell

Since the "Brick" UC concept could not converge to the target response, a design iteration in terms of a geometric modification of this concept is proposed. This modification and its results are described in the following sub-sections.

4.4.1. Additional Element Geometry

The basis of the "Brick" UC concept was the FFB and its deformation. However, as this geometry alone did not enable tuning towards the target response, another geometry is proposed to be combined with the existing "Brick" geometry. This proposed geometry is the oval, or constant-thickness curved beam. The oval has three parameters which control its behavior. These parameters, along with their sensitivities to geometric nonlinearity under a compressive loading, can be seen in Figure 4.11. This figure shows that either decreasing the oval thickness or increasing the ratio of R1/R2 both increase geometric nonlinearity of the geometry.

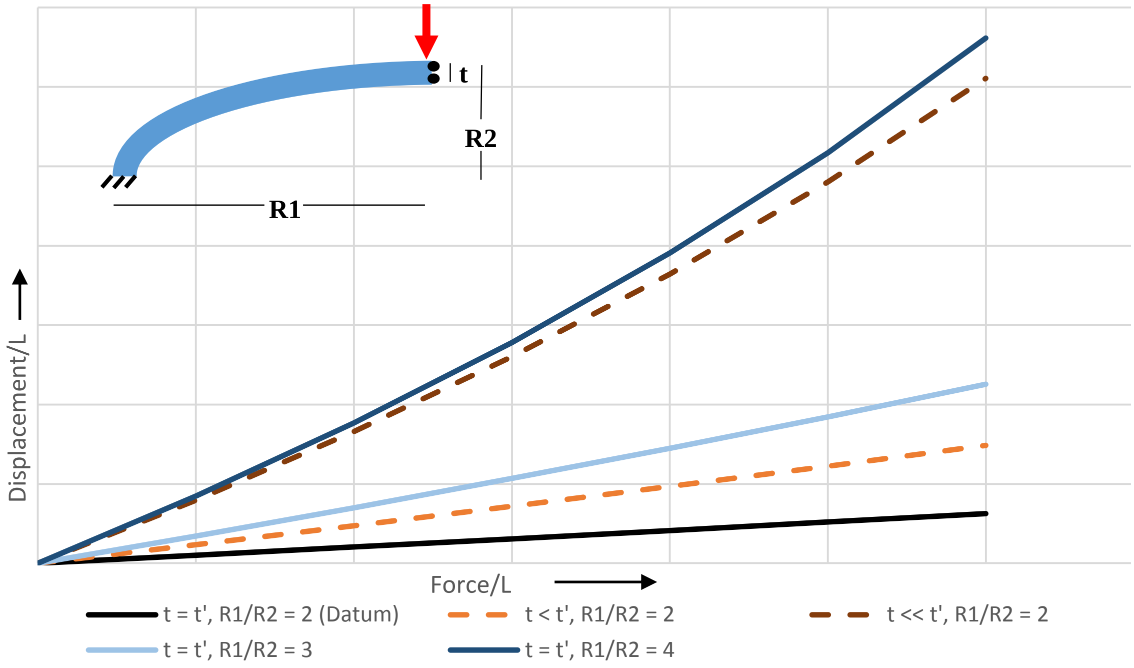


Figure 4.11 Oval Geometric Free Body Diagram Nonlinearity based on Parameter Sensitivities

The primary difference between the FFB and the oval under compression is that they have inverse stiffening properties. In other words, the FFB stiffens as additional load is applied while the oval under compression softens as additional load applies. Therefore, by combining these two geometries in the same UC, an effective UC stiffness may occur similar to that of adding springs in parallel, defined in the equation below

$$K_{eff} = K_1 + K_2 \qquad (4.3)$$

where K_{eff} is the effective stiffness, K_1 represents the stiffness of the FFB, and K_2 represents the stiffness of the oval. This combination of geometries with inverse stiffness responses is predicted to allow for a UC geometry with parameters that can be adjusted to match the target response. This combination of each individual stiffness to match the target response is shown in Figure 4.12.

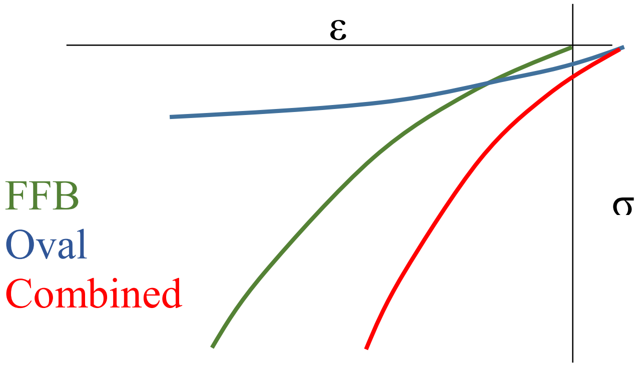


Figure 4.12 Predicted Effect Of Combining Oval with FFB in a UC

4.4.2. Integration into Brick UC Concept

The oval geometry was added to the "Brick" UC design as shown in Figure 4.13 along with UC design parameters. Since the radii of the oval geometry was constrained to the right and left strut and to the top and bottom FFB, the only additional design parameter in the new, "BrickOval", UC concept is the oval thickness, t3.

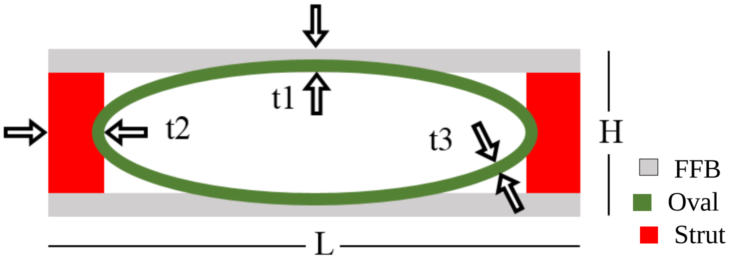


Figure 4.13 Integration of Oval with FFB in UC

The “BrickOval” UC combines the FFB in oval geometry in a manner consistent with springs in parallel. This can be proven by analyzing the load path through the geometry as shown in Figure 4.14. The total force, F_{Tot}, is applied to the springs in parallel.

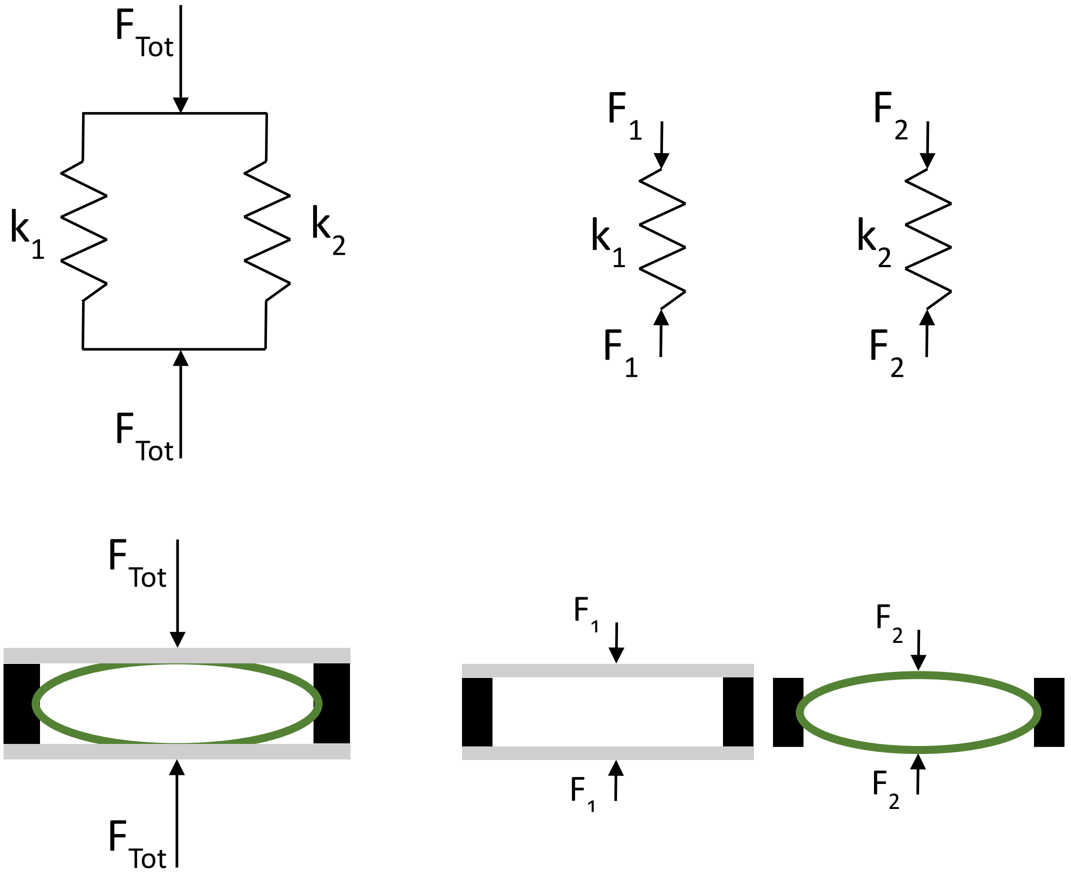


Figure 4.14 Load Path of (top) Springs and (bottom) Geometries in Parallel

Thus, each spring experiences only a portion of the total force, i.e. $F_{Tot} = F_1 + F_2$. Similarly, the "BrickOval" geometry combines the FFB and oval in parallel as each geometry sees a portion of the total applied force.

4.4.3. Evaluation of BrickOval UC Concept

This new BrickOval UC concept was evaluated in a similar manner to the previous concept. The UC was tessellated into a 4x4 array as before. An example tessellation is shown in Figure 4.15 with the same material properties, loading, and boundary conditions as the previous concept.

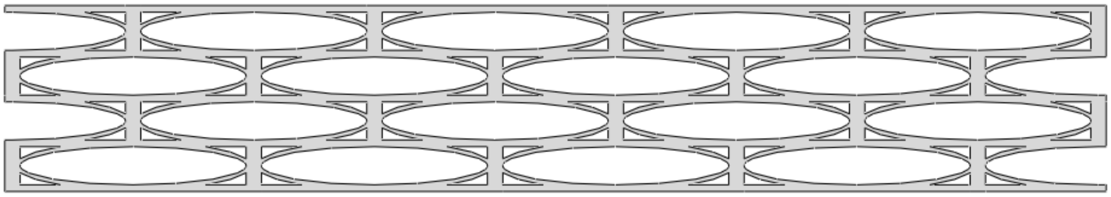


Figure 4.15 Example 4x4 Tessellation of BrickOval UC

A Full Factorial was not required with the "BrickOval" concept. A sensitivity analysis was done only by varying the oval thickness, t_3, for a high and low value while maintaining all other variables the same. As can be seen from Figure 4.16, as the oval thickness increases, the rate at which the vertical response stiffens decreases. In other words, the curvature of the response curve decreases.

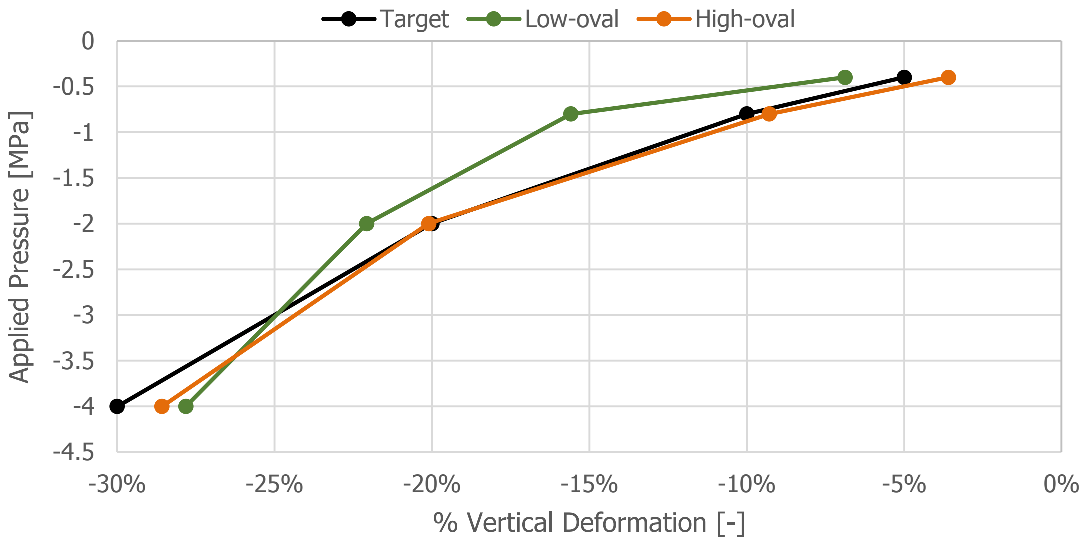


Figure 4.16 Sensitivity of Oval Thickness in BrickOval Design

Note that for the lower oval thickness value, the “BrickOval” concept has a deformation response similar to that of the original “Brick” UC design. This result supports the analogy of springs in parallel in equation (4.3) as shown in the equation below

$$\lim_{K_2 \to 0} K_{eff} = K_1 \tag{4.4}$$

where the effective stiffness becomes that of the FFB as the stiffness of the oval approaches zero.

Since the sensitivity analysis results indicate that the “BrickOval” UC parameters can be modified to adjust the nonlinear response towards the target curve, this design is deemed a feasible candidate.

4.4.4. Size Optimization of UC

Since the “BrickOval” UC is deemed feasible, the concept will undergo a Size Optimization (SO) of its parameters to converge the vertical deformation response towards that of the target curve.

4.4.4.1. Optimization Setup and Procedure

The SO has an objective to minimize the difference in the obtained curve and the target curve via the sum of least squares as shown below

$$\min_{f} : f = \sum_{i=1}^{N} \left(\varepsilon_i^t - \varepsilon_i^c \right)^2 \tag{4.5}$$

where ε_i^t and ε_i^c are the target and concept % vertical deformation, respectively, for load case i. For the given application, the four load cases and their respective target % vertical deformation values are given in Table 4.1. The optimization routine was carried out in ModeFrontier to optimize the input UC parameters. These UC parameters were input to a Python script that assembled the appropriate 4x4 BrickOval meta-material, applied the load and boundary conditions, generated a mesh, and performed a finite element analysis for all four load cases in Abaqus.

Three variables that directly affected the FFB and oval were considered as optimization variables – L, t_1, and t_3. The UC height, H, and strut thickness, t_2, were held constant at 5.0 mm and 1.0 mm, respectively. A constraint was placed on the “BrickOval” UC geometry such that oval thickness, t_3, could not exceed the thickness of the FFB, t_1, due to geometric considerations in assembling the UC via the Python script. This constraint was enforced in the optimizer by creating a variable α such that

$$t_3 = \alpha t_1 \tag{4.6}$$

where 0<α<1. A representation of the graphical optimization setup in ModeFrontier can be seen in Figure 4.17. A genetic algorithm, specifically the Non-dominated Sorting Genetic Algorithm II (NSGA-II), was chosen to explore the design space with a generation size of 15 and default mutation parameters (crossover probability of 0.9 and mutation

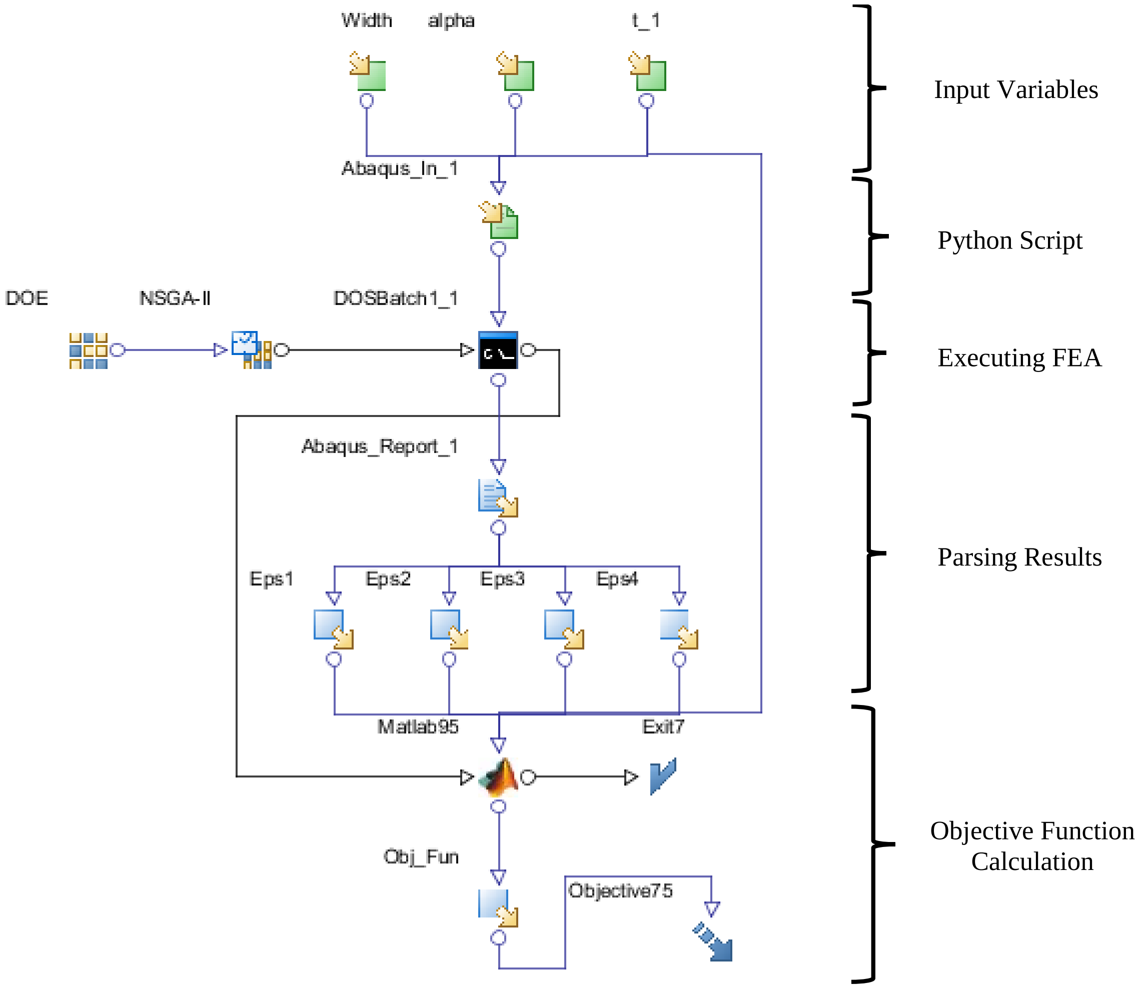


Figure 4.17 Optimization Setup in ModeFrontier

probability of 1.0). The initial generation was selected via a design of experiments of a Uniform Latin Hypercube to evenly sample the design space. The Python script used in this optimization to automate the geometry generation based on input variables, execute the four load cases, and then output the desired values is included in Appendix B.

4.4.4.2. Results from Optimization

The algorithm converged as indicated by the design history shown in Figure 4.18. This figure shows the convergence of the objective function over the designs. Note that the objective values are plotted on a logarithmic scale.

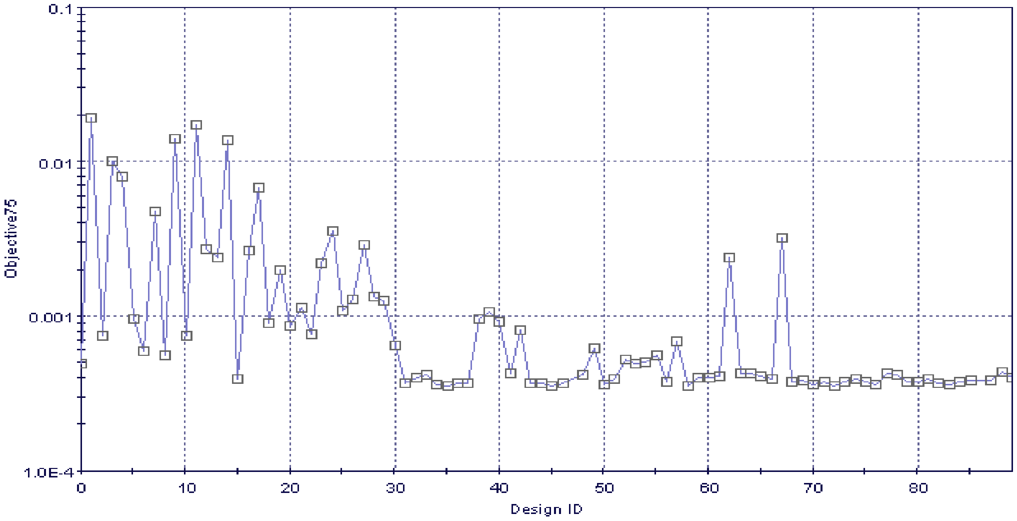


Figure 4.18 Optimization Design History and Objective Convergence

The design with the lowest objective function value (at 3.5534 E-04), and therefore most closely matching the target curve, is shown in Figure 4.19. This response can be seen to match the target curve very closely as the objective function value correlates to an average difference of 0.94% difference between the target and optimized responses.

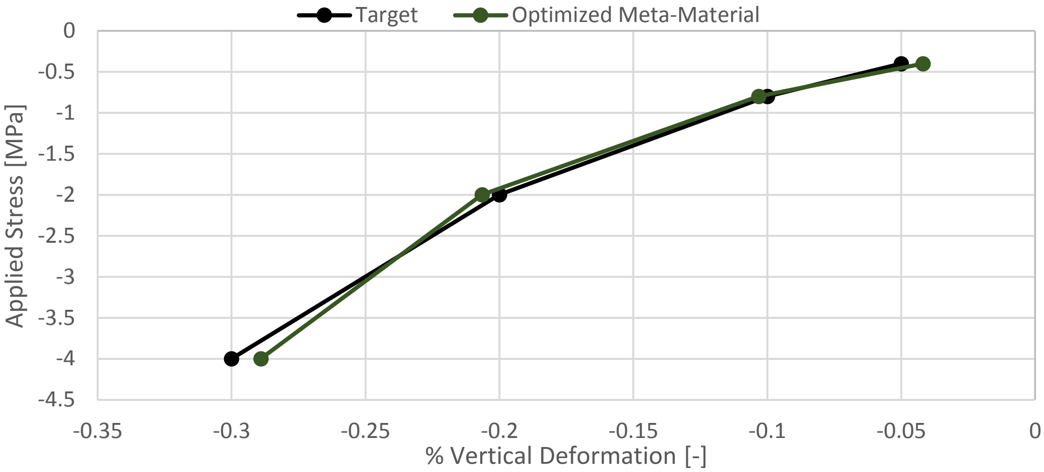


Figure 4.19 Optimized Meta-Material Deformation Response

The optimized design corresponding with this curve can be seen in Figure 4.20 in an assembled 4x4 array. The dimensions of this array are 138 mm by 18.815 mm which fits within the application design space mentioned previously. The optimized UC dimensions corresponding to this design are listed in Table 4.3.

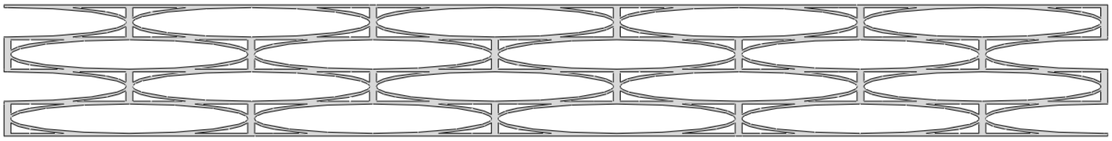


Figure 4.20 Optimized BrickOval Design in a 4x4 Array

Table 4.3 Dimensions of Optimized UC

	Dimension [mm]
L	36.000
H	5.000
t_1	0.395
t_2	1.000
t_3	0.352

4.4.5. Conclusion on BrickOval UC Design

These results are very promising since the “BrickOval” UC was able to be systematically designed to match the targeted nonlinear deformation curve. Therefore, the primary objective of this work in replacing the material nonlinearity of the current rubber pad with geometric nonlinearity of a designed meta-material was achieved. However, there are two constraints that are not met with this design: stress and manufacturability. The most important of these constraints is that the maximum stress observed in the 4x4 structure was 400% above yield stress of the constitutive material, steel. This excessive stress prevents the nonlinearity observed from the linear elastic simulation from being experimentally viable in a physical prototype. Additionally, the manufacturability is limited as current additive manufacturing processes have a limit of manufacturing members with a minimum size of 0.4 mm. This issue can be addressed by either improved manufacturing techniques to further decrease this constraint or by scaling the UC, as will be discussed later in this chapter. It should also be noted that the height and strut thickness were constrained in this optimization procedure. It may be possible to further modify the “BrickOval” UC parameters to further decrease the stress to within permissible limits.

4.5. Design Considerations

There exists numerous considerations in the design of meta-materials. Exploring these design considerations can lead to more concrete design principles. Three such considerations will be discussed in detail along with their implications on the current application.

4.5.1. Homogenization Limit

The homogenization limit can be determined to obtain the vertical deformation properties of an infinitely-tessellated meta-material. Since an infinite continuum of UCs cannot be simulated, a convergence study can be performed to find the asymptote in which these homogenized properties theoretically lie. This study will also reveal the relative effect of boundary conditions in the context of determining bulk material properties of meta-materials. However, it is important to note that different boundary conditions and UC geometries may have a different effect on convergence towards the homogenization limit.

A uniaxial deformation response homogenization limit can be defined as the limit in which increasing the number of UCs in the meta-material no longer has any effect on the vertical properties obtained. The reason differences in these vertical properties exist is primarily due to boundary conditions. As the number of UCs increases, the less overall effect the boundary conditions have. Thus, a homogenization limit for these vertical properties can be determined.

A test was run on the optimized BrickOval solution to determine the homogenization limit of this specific UC geometry. This test was run for seven different n-by-n UC tessellations across the same four load cases (shown in Table 4.1) used to evaluate the meta-material. Only even numbers were used for the n-by-n tessellations due to odd numbers changing how the boundary conditions are applied to the meta-material because of the half-UC shift between layers in the y-direction. These homogenization limit results are shown in Figure 4.21.

In this figure, it can be seen that as the tessellation increases, the change in vertical deformation continuously decreases. This can be further seen in Table 4.4 where the percent change in deformation is shown between similar tessellations across all load cases. Note that the difference between the 8x8 and 10x10 is about the same as that between the 20x20 and 30x30. This indicates convergence of properties.

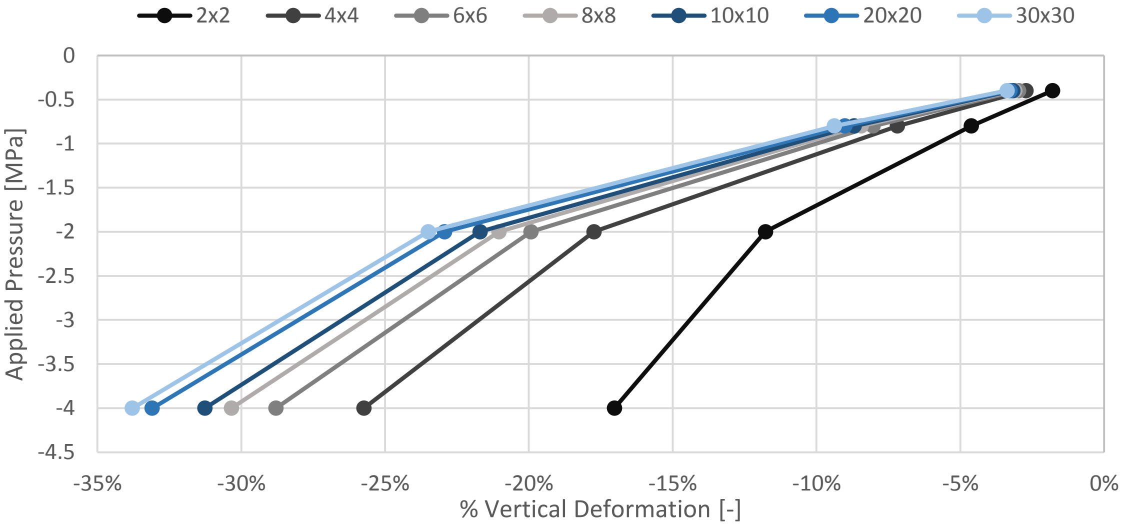


Figure 4.21 % Deformation Homogenization Limit Results for BrickOval UC

Table 4.4 % Difference in Vertical Deformation Comparison Across Load Cases

	% Difference in Vertical Deformation					
Applied Pressure	2x2 / 4x4	4x4 / 6x6	6x6 / 8x8	8x8 / 10x10	10x10 / 20x20	20x20 / 30x30
-0.3817	-34.227%	-8.739%	-4.036%	-2.228%	-3.000%	-3.036%
-0.8384	-35.700%	-10.514%	-4.953%	-2.751%	-3.601%	-3.889%
-2.0632	-33.619%	-10.976%	-5.272%	-3.053%	-5.374%	-2.425%
-3.9327	-33.860%	-10.624%	-5.097%	-2.957%	-5.525%	-2.042%

Perhaps the clearest way to see this convergence is in Figure 4.22 where the % vertical deformation is plotted across the various meta-material tessellations for each load case. In every load case, convergence towards some property is visibly seen through exponential decay towards an asymptote. For the BrickOval UC, approximate convergence is seen at a tessellation of 20x20. This differs from the results of Czech in [49] where convergence was expressed to exist at a tessellation of 10x10. This difference most likely exists due to the varying aspect ratio of the UC, the UC geometry, and the boundary conditions applied. The UC aspect ratio in that work was 1:1 while the optimized BrickOval was 7.2:1. The internal geometries were vastly different and Czech applied symmetric boundary conditions to simulate infinite UC tessellation.

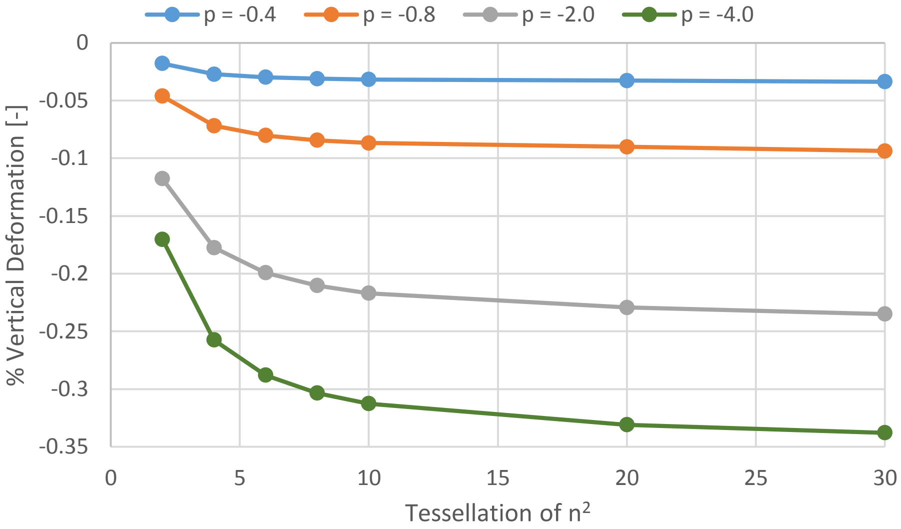


Figure 4.22 % Deformation Convergence towards Homogeneous Properties

4.5.2. Unit Cell Scaling

Another consideration in the design for meta-material UCs is the concept of UC scaling. Understanding how global properties and internal stresses change as a result of scaling can be useful in guiding the design of the UC. A study was performed on the optimized BrickOval solution by scaling the UC dimensions to determine both of these trends while maintaining a 4x4 UC tessellation. Scaling in this case refers to multiplying all dimensions by a scalar value. The comparison of % vertical deformation between the scaled meta-materials are shown in Table 4.5.

Table 4.5 Comparison of Vertical Deformation between Scaled Unit Cells

	Scale						
	0.1	0.25	0.5	1	2	4	10
Applied Pressure [MPa]	% Difference from Datum [-]						
-0.3817	0.020%	-0.062%	0.009%	-	0.000%	-0.009%	-0.023%
-0.8384	0.016%	-0.064%	0.006%	-	0.000%	-0.009%	-0.027%
-2.0632	0.017%	-0.026%	0.010%	-	0.000%	-0.001%	0.008%
-3.9327	0.014%	-0.010%	0.004%	-	0.000%	0.000%	0.009%

This table clearly shows that scaling the meta-material has virtually no effect on the bulk deformation properties. Additionally, Figure 4.23 shows that the stress distribution and max stresses exhibited in the UCs at the datum, 0.1x scaled, and 10x scaled UC’s are all equivalent. The max stresses in each are about 2100 MPa. These results are promising for the designer in terms of enabling freedom to modify the UC size to fit the design space, allow for a more homogeneous meta-material continuum, or meet specific manufacturing constraints once the UC has been designed to match the target response. The designer can scale the UC without increasing the maximum observed stresses. However, if the max stress in the optimized UC is above the permissible value, then the designer cannot scale the UC to reduce the stresses to the desired level. It should also be noted that if the designer decreases the UC size to enable additional UCs to fill the design space, this change in tessellation size will change the % vertical deformation properties as indicated in 4.5.1. Therefore, the UC parameters may have to be optimized again to match the target curve.

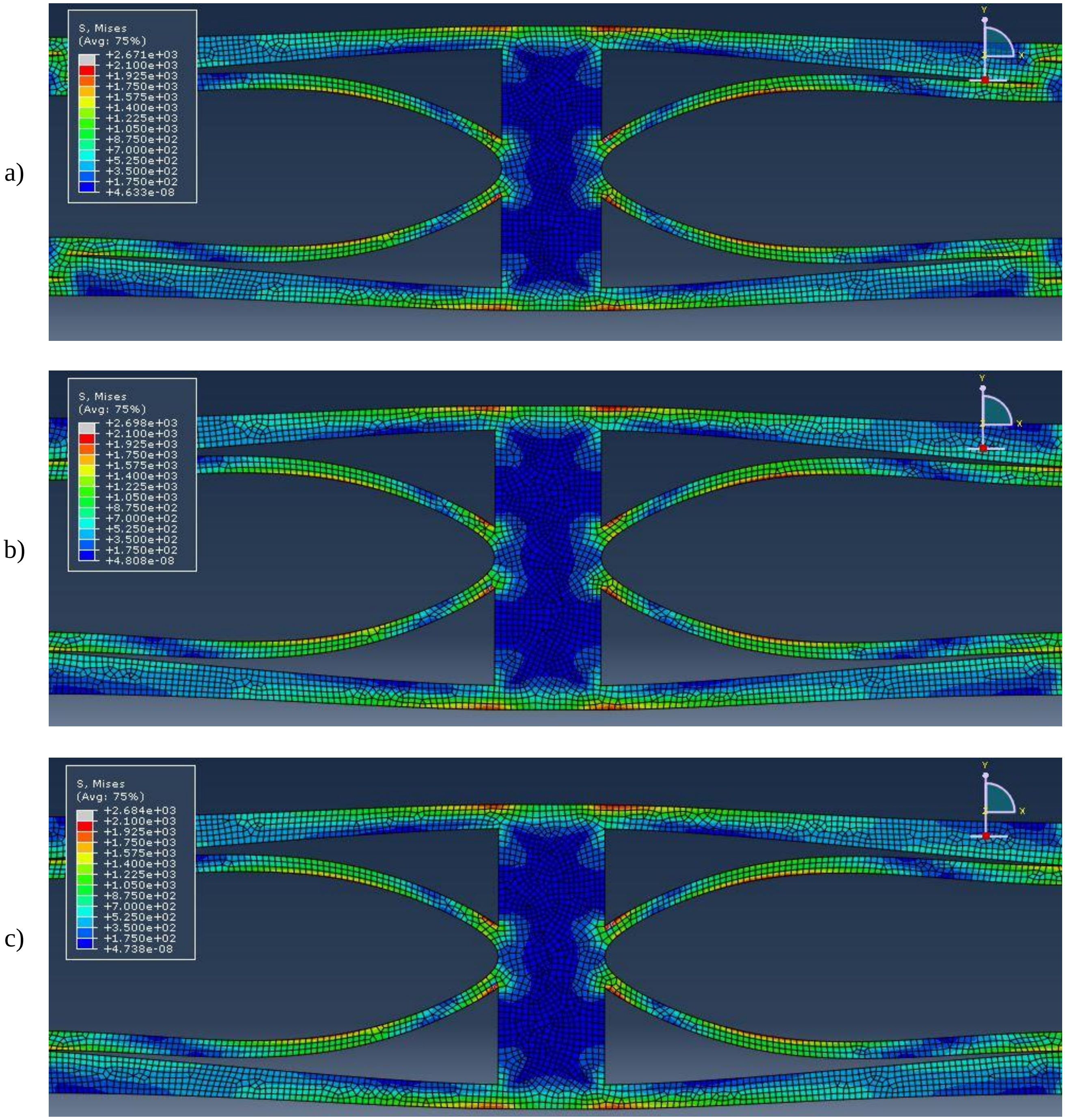


Figure 4.23 Max Stress and Stress Distribution Comparison of a) Datum, b) 0.1 Scale, and c) 10.0 Scale UC's

4.5.3. Material Selection

The choice of material in design is important for the component's performance, cost, manufacturability, and other factors [9].

4.3.5.1. Theoretical Objective

For the current application, the working principle of the "BrickOval" UC design is to obtain large global deformation via bending without yielding of the constitutive material, i.e. low strains at the local level. However, the choice in this constitutive material inherently determines the maximum global deformation before yielding given a geometric configuration.

In the analyses presented in 4.4 for the "BrickOval" design, the constitutive material was chosen to be steel. However, an optimal material for the current application would be one in which the difference between yield stress and elastic modulus is minimized to allow maximum deformation before yielding. In other words, an optimal constitutive material of the UC should minimize the ratio

$$E : \sigma_Y \tag{4.7}$$

where E is elastic modulus and σ_Y is yield stress of the material. An Ashby chart comparing these two properties can be found in Figure 4.24. In this diagram, elastomers can be found to have an exceptionally desirable ratio in the range of 1:1-10:1. Of course, the current elastomer is being replaced by a linear elastic metal. By contrast, the steel used in the simulations has a ratio of 360:1. The linear elastic metal with the lowest ratio is that of titanium alloy Ti 3Al-8V-6Cr-4Mo-4Zr-0.05Pd (E = 102 GPa, σ_Y = 1103 MPa, ν =

0.32) at 90:1. Therefore, simulations should be run with this titanium alloy to analyze the feasibility of any meta-material UC design that aims to achieve maximum deformation before yielding.

4.3.5.2. Initial Results

Based on the selection of the titanium alloy as the best material choice to achieve large deformations before yielding, several simulations were run with titanium to determine the feasibility of the “BrickOval” geometry to be redesigned to both match the nonlinear target response and fall within the permissible stress limits of the alloy. To determine this initial feasibility, the “BrickOval” parameters were manually tuned to get a close approximation of the target vertical deformation response as shown in Figure 4.25.

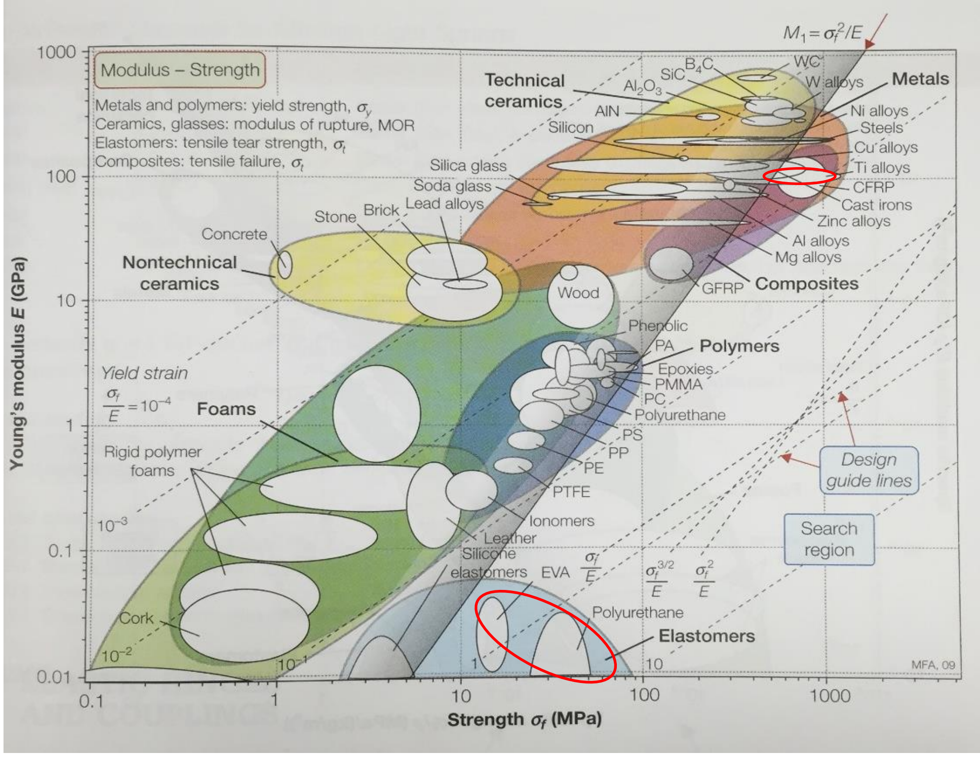


Figure 4.24 Ashby Chart Comparing Material Strength and Elastic Modulus [9]

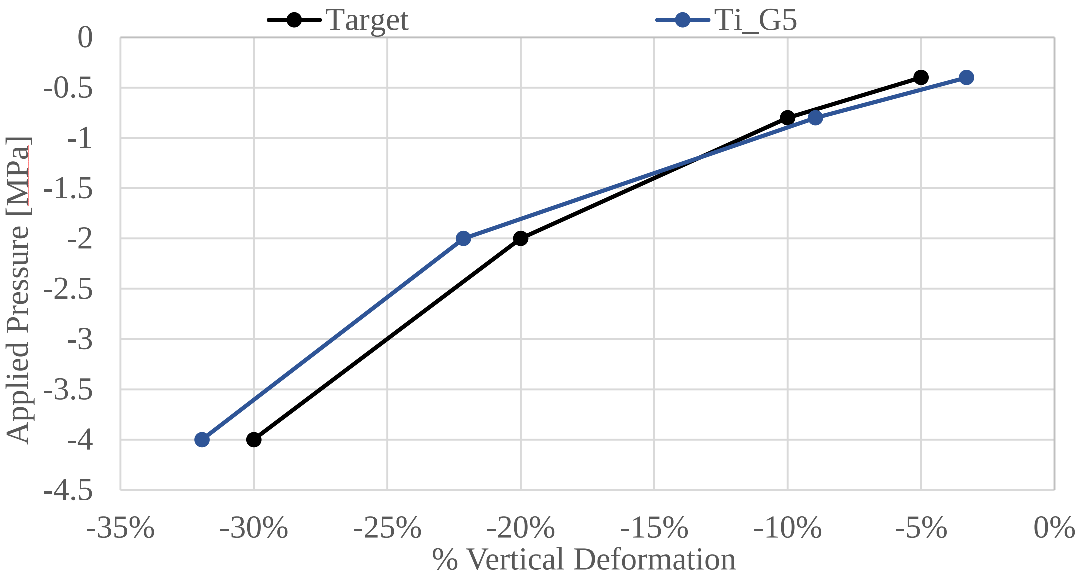


Figure 4.25 Initial Titanium Results via Manual Tuning

These initial results showed that to achieve the 20% vertical deformation, the internal stresses fell from 400% yield stress with steel down to 130% yield stress with the titanium alloy. This shows that an appropriate material selection marks significant improvement in decreasing the internal stresses, but still does not meet the constraint.

4.6. Conclusions

The results from this chapter directly answer research questions 1 and 2.b as well as the first hypothesis, reproduced below.

R1. Can a meta-material be developed in which the global behavior of the resultant medium exhibits nonlinear compressive behavior similar to that of the current elastomer?

H1. A meta-material can be designed whose global behavior in compression is of similar nonlinearity to that of the current elastomer.

The results of the “BrickOval” geometry clearly show that a meta-material can in-fact be developed in which the global behavior, in this case meta-strain, exhibits nonlinear compressive behavior similar to that of the current elastomer. Therefore, the first research question has been answered by validating the first hypothesis. Since successful results yielded from combining only two elemental geometries, it may be possible that other solutions exist to this same problem that combine other elemental geometries not yet explored.

R2. What method can be used to successfully develop a meta-material to meet the nonlinear deformation response of the current application?

b. Can a method be developed to accomplish this?

The approach using engineering principles to design the meta-material UC ultimately allowed the target nonlinear response to be met via the “BrickOval” UC design and simultaneously answered this research question. Unfortunately, the stress observed in the “BrickOval” UC was found to exceed the yield stress, even after a material selection analysis was completed. Therefore, the “BrickOval” UC design is not a feasible solution to replace the rubber pad in the current application.

Important conclusions were drawn regarding the general meta-material UC design. There is an effect of the number of UCs in the meta-material in regards to its resulting %

vertical deformation response that must be considered by the designer. For the “BrickOval” UC design, the homogenization limit is seen to be found at a 20x20 UC tessellation wherein significant change in the deformation response is not seen by further increasing the number of UCs. Additionally, the designer can linearly scale the UC without changing the deformation response or the stresses experienced in the structure. This allows the designer to easily scale a solution that matches the target response to meet manufacturing constraints. Lastly, a material selection was completed for the given application under the premise of achieving large UC deformation via bending without yielding the constitutive material. The analysis showed that the best material for this application has a minimum value of elastic modulus to yield stress. Therefore, for any application under the given premise of maximizing bending without yielding the constitutive material, titanium alloy Ti 3Al-8V-6Cr-4Mo-4Zr-0.05Pd is the optimal material. Though this material is expensive, initial feasibility of UC designs towards matching the target response should be determined with this alloy before further material selection continues with cost consideration.

CHAPTER 5. A UNIT CELL SYNTHESIS METHOD FOR META-MATERIAL DESIGN

The logic and process used in the previous chapter to design and develop the "Brick" and "BrickOval" UCs was systematic. This systematic process can be abstracted into a design framework aimed at designing meta-materials from a UC level to match a targeted nonlinear response. This chapter proposes such a design method and details its systematic and logical progression.

5.1. Method Introduction

5.1.1. Scope

The scope of the Synthesis method is to aid designers in developing meta-materials to meet targeted nonlinear deformation responses. These meta-materials are constrained to be two-dimensional geometries that are extruded in the third dimension. Therefore, development of a three-dimensional lattice structure, which might also be considered a meta-material, is outside the scope of this design method.

5.1.2. Intellectual Basis

The intellectual basis this method operates on is the designer's fundamental understanding of the geometric nonlinearities of element geometries. This fundamental understanding includes the sensitivity of geometric nonlinearity to geometry parameters including aspect ratios, thicknesses, radii, etc… This understanding precedes the design process and should therefore be known before implementation. Once implemented, geometric parameters can be adjusted to tune the response towards that of the target.

5.1.3. Method Overview

The Synthesis design method is a systematic procedure to develop a UC. The main steps of this procedure is visually summarized in the flowchart in Figure 5.1. There exists six steps in the method with a focus on building the UC around one or more known

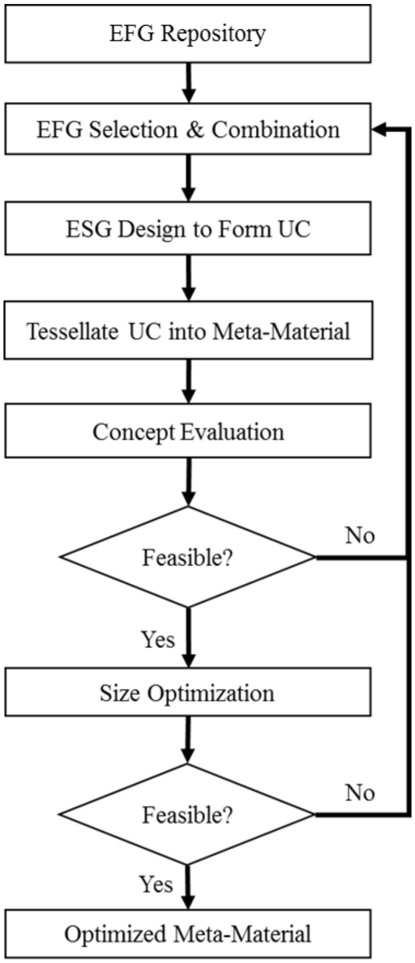


Figure 5.1 Synthesis UC Design Method Flowchart

geometric element and its corresponding geometric nonlinearity. These steps will be explained in detail in the next section.

5.2. Method Description

5.2.1. Step 1: EFG Repository

An *Elemental Functional Geometry* (EFG) is defined as a geometry whose deformation response is used to meet the target response. As mentioned previously, this method is predicated on the designer's understanding of geometry nonlinearity of different shapes. Thus, a repository containing EFGs is essential. This repository should contain several geometries with pre-determined information including the parameters that control the size and shape of each geometry and the sensitivity of these parameters affecting

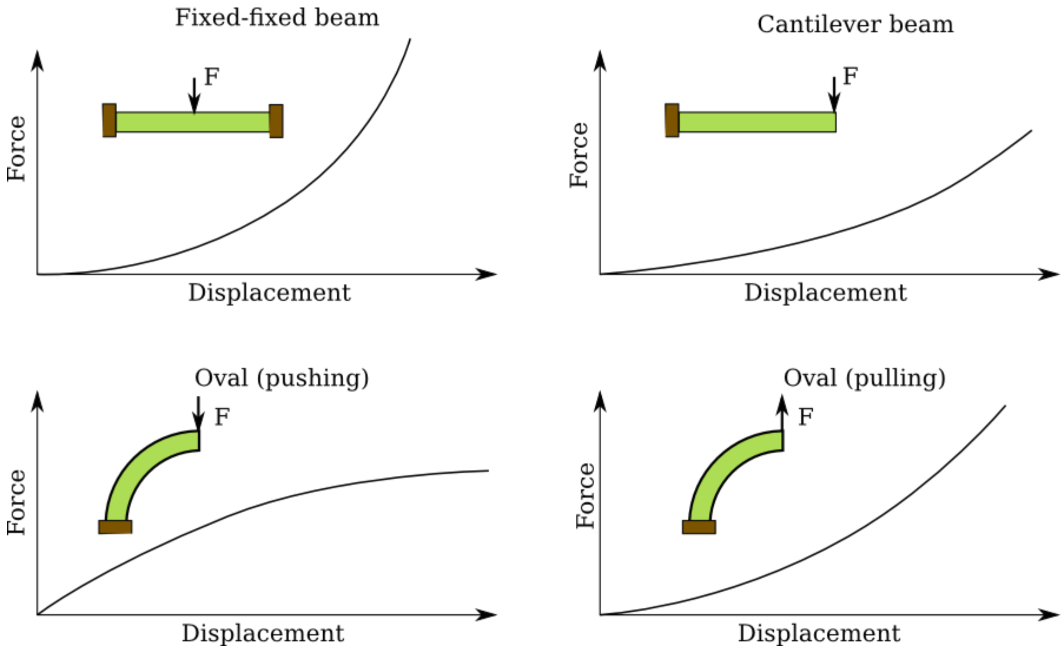


Figure 5.2 Example EFGs and their General Nonlinear Behavior

geometric nonlinearity. The repository will serve as a starting point in which a designer can choose from in Step 2. A set of four EFGs and their associated geometric nonlinearities is plotted in Figure 5.2 after being subjected to concentrated loads and undergoing large deformation. Three of these nonlinear responses experience stiffening with respect to displacement, but the oval geometry subjected to a pushing load has an inverse behavior. Thus, the designer can combine these differing and complimentary EFG deformation behaviors in a single UC to tune the overall meta-material behavior towards that of the target curve. This combination of EFGs and their stiffnesses can be considered in the same way springs can be combined in series or in parallel.

5.2.2. Step 2: EFG Selection and Combination

There are many ways in which EFGs can be combined. For the purpose of this work, these combinations will be categorized in the manner shown in Table 5.1 to represent different configurations that can be applied to a UC.

Table 5.1 Possible Connection Configurations

Connection Configuration	Description
0th Order	Single EFG
1st Order	Combination of two 0th order configurations (series or parallel)
2nd Order	Combination of two 1st order configurations (series or parallel)

Based on these definitions, the EFGs shown in Figure 5.2 represent four different 0th order configurations. The combination of 0th order EFGs into different 1st order configurations can be seen in Figure 5.3 while combinations of two 1st order configurations into different 2nd order configurations can be seen in Figure 5.4. The nonlinear 0th order stiffnesses of the EFGs combined to create a 1st order effective stiffness shown in series ($K_{eff,s}$) and parallel ($K_{eff,\mathrm{p}}$) below as

$$K_{eff,s} = \left(\frac{1}{k_1} + \frac{1}{k_2} \right)^{-1} \tag{4.1}$$

$$K_{eff,\mathrm{p}} = k_1 + k_2 \tag{4.2}$$

where k_1 and k_2 are the 0th order EFG stiffnesses, respectively. The effective stiffnesses of these connection configurations are shown in Figure 5.3 and Figure 5.4 as red and blue

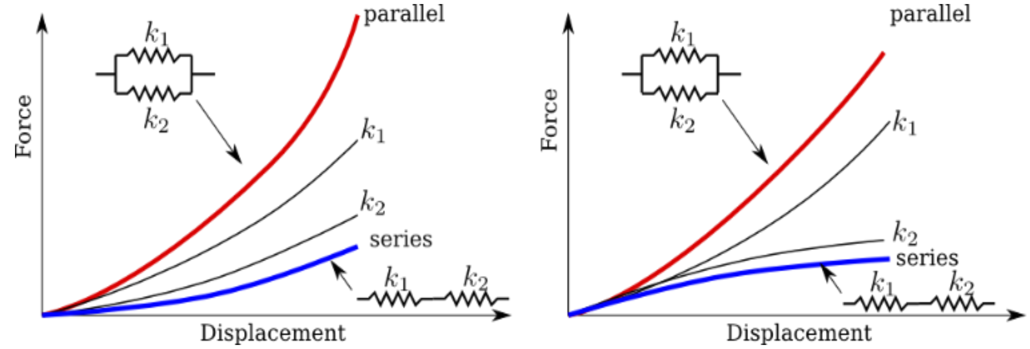


Figure 5.3 1st Order Connection Configuration

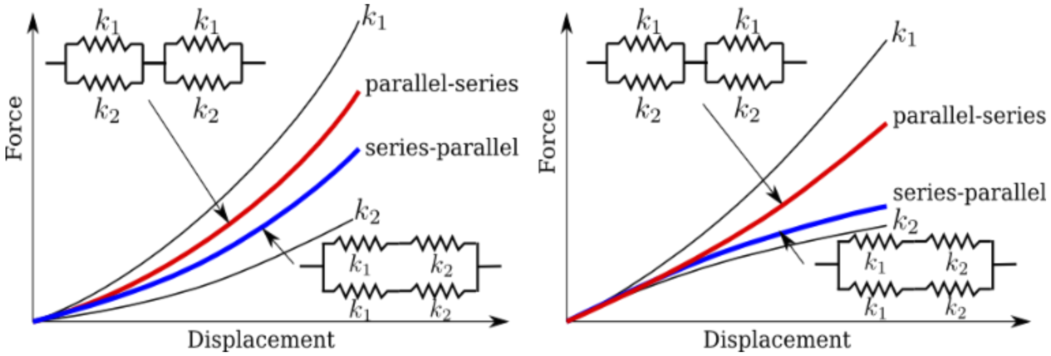


Figure 5.4 2nd Order Connection Configuration

curves, respectively. It can be seen from the possible configurations shown that a wide range of nonlinear deformation is possible and, upon tuning geometric parameters, many target responses can be met in this manner. Thus, the designer can use these connection configurations to synthesize a UC geometry.

5.2.3. Step 3: ESG Design to Form UC

Along with EFGs, the other required element to synthesize the UC is the *Elemental Structural Geometry* (ESG). The ESGs act as the structural components in a UC and serve as the rigid support or connection of the EFGs and adjacent UCs. Therefore, they typically have higher stiffness and do not interfere with the deformation of EFGs. Thus, the designer must design or select ESGs to form the UC that adhere to the following requirements:

1. An ESG must exhibit high stiffness and low deformation compared to the EFG(s)
2. An ESG must complete the topology of the UC by connecting the EFGs between UCs

The first requirement of the ESG serves to isolate the tunable nonlinear properties of the EFG while the second requirement serves to complete the UC in order to allow tessellation into a meta-material. Based on the fact that ESG deformation must be low, it is unnecessary to determine their deformation behavior before integration with the UC design.

5.2.4. Step 4: Tessellate of UC into a Meta-Material

Once a UC geometry has been designed, the meta-material can be formed by tessellating the UC several times in the x- and y- directions. For computational analysis and optimization purposes, a representative volume element (RVE) of the meta-material is constructed through tessellation of the UC. The number of UCs in the RVE depends on several factors. It is well known that the fewer UCs that exist in each direction, the more prominent the effect of boundary conditions. When the meta-material to be designed is much larger than the size of a UC, the target meta-material can be considered homogeneous. In this case, many UCs are required in the RVE and a convergence study is necessary to validate the homogeneous behavior. However, for applications with a restrictive design space, the dimension of the RVE can ultimately be determined by the size of the target structure and this size becomes the driving factor in the allowable number of UCs in the tessellation. Figure 5.5 shows how an example RVE with many UCs can be chosen to represent a portion of the meta-material. Likewise, the UC can be shown tessellated into both a RVE and the meta-material.

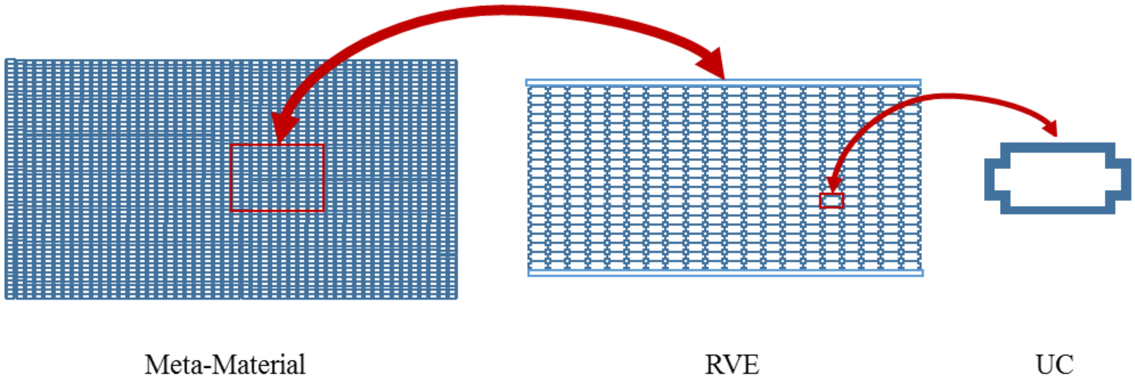


Figure 5.5 Decomposition of a Meta-Material into RVE and Tessellation of UC

5.2.5. Step 5: Perform Concept Evaluation

Since the meta-material design is to have a target deformation behavior which is different from that of its constitutive material, a means of determining the effective mechanical properties of the meta-material must be determined. For a meta-material RVE tessellated with a large number of UCs, the meta-material is evaluated based on the RVE's deformation characteristics. For a given target deformation behavior, typically described by one or multiple stress-strain curves, proper finite element analyses are performed on the RVE to obtain the force-displacement behavior of the meta-material. A so-called meta-strain can then be defined as the percentage of uniaxial deformation (i.e. average displacement) of the meta-material defined by

$$\text{meta-strain} = \%\ \text{Uniaxial Deformation} \ = \frac{\delta}{H}(100) \tag{4.3}$$

where δ is the displacement and H is the original height of the meta-material, as shown in Figure 5.6. The meta-material is subjected to a series of load cases corresponding to the range of the target curve. The meta-strain is then calculated at each load case to determine

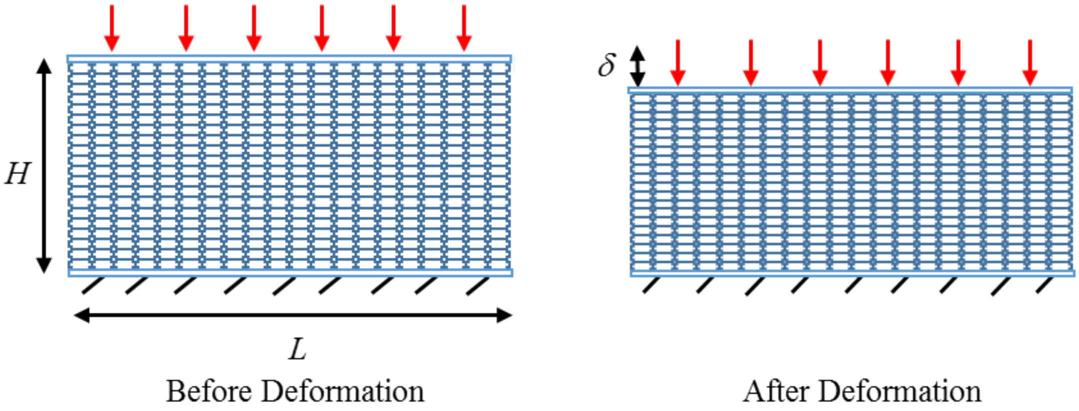


Figure 5.6 Example Meta-Material with Uniaxial Loading (left) and after Deformation (right)

the RVE deformation response which can then be compared to the target curve for evaluation.

The complete deformation behavior of a nonlinear material can be defined by a material tensor which contains multiple nonlinear variables. While the nonlinear material tensor can be determined by the material's stress-strain responses under a variety of loading conditions and deformation modes, it is often the case that one or two deformation modes dominate the deformation of the target material in a given application. Therefore, in most cases, it is sufficient to only take the stress-strain response of the target material in its dominant deformation mode(s) and find a meta-material solution to match the dominant deformation behavior. With the identified target stress-strain response, the ability to tune UC parameters to match the desired response is paramount before moving on to the next step. Determining this feasibility can be done by carrying out and analyzing a design of experiments study. This is a necessary intermediate step between the formation of the concept UC and optimizing UC parameters to meet the desired behavior.

If the concept UC with the selected EFG configuration is found to have a deformation behavior close to the desired material response during the concept evaluation stage, this concept UC is regarded as a "feasible" design. Otherwise, a different EFG configuration of the same or a higher order is selected and Steps 2-5 are repeated with the new conceptual UC until feasibility is obtained. Note that, higher order EFG configurations typically lead to an increase in the design parameters of the UC which may impart more tuning ability to match the target behavior. While there may be multiple ways of combining the EFGs to achieve the desired deformation behavior, as shown in Figure 5.3 and Figure

5.4, it is logical that one starts with the lowest order configurations for the simplicity of the unit cell.

5.2.6. Step 6: Perform Size Optimization

An optimization of the dimensions of the EFGs and ESGs in the UC is conducted once the UC concept is deemed feasible. The optimization procedure will converge the deformation response of the meta-material towards that of the target response. The optimization setup can be mathematically written as

$$\min_{f} f = \sum_{i=1}^{N} \left(\varepsilon_i^t - \varepsilon_i^c \right)^2 \tag{4.4}$$

where ε_i^t and ε_i^c are the target strain and pseudo-strain (i.e. % vertical deformation) of the meta-material RVE, respectively, at the *i*-th load level in a total of N load cases. The optimization algorithm should be chosen based on considerations of convergence properties and ability to handle the number of UC design parameters. Once the optimization is converged, the resulting meta-material should have a deformation response equivalent to that of the target. Note that, the solution of such an optimization problem is typically not unique. Whether the result of an optimization run is acceptable also depends on an evaluation against the application-specific design constraints. After a converged solution is obtained, the design constraints are analyzed to further rule out a potentially infeasible design of the meta-material. Such design constraints include manufacturing feasibility, material-dependent stress allowance, and the requirement of non-contact within permissible deformation limits within the UC. If a meta-material design is deemed infeasible in this step, either the placement of the EFGs in the current configuration or their initial dimensions are modified and another SO is carried out with the new initial conditions until the desired deformation behavior is obtained and the design constraints are satisfied. However, if the SO iterations do not yield an acceptable optimal design, then the designer goes back to Step 2. Then, a different EFG configuration of the same or a higher order is selected and Steps 2-6 are repeated with the new conceptual UC. However, once Step 6 is completed and meets all design constraints, the resulting meta-material is a feasible solution.

5.3. Discussion

The four elemental geometries presented in this work (Figure 5.2) can be viewed as examples for future development. It is intended that many additional EFGs be studied and added to the repository. In order to achieve this, users of this method can add to the research knowledge by testing new EFGs and adding them to the repository by sharing them with the research community.

When determining and classifying new EFGs, it is important to note how the boundary conditions and direction of loading both affect nonlinearity of the same geometry. Boundary and loading conditions differentiate the fixed-fixed beam and the cantilever beam. The additional constraints on the fixed-fixed beam increases its geometric nonlinearity as shown in Figure 5.2. A second example is how the direction of loading changes the nonlinear response of the oval depicted in the same figure. The pushing load results in a softening response whereas the pulling load yields a stiffening response.

Therefore, geometries with different boundary or loading conditions that yield different geometric nonlinearities will result in different EFGs.

The optimization objective presented in this work is but one method to converge the concept design's response to that of the target. This objective function can and should be modified to meet the designer's specific needs. In the current work, the sum of least squares considers all load case responses to be of equal weighting. In specific applications, it may be useful to increase the weight of responses at critical load cases to ensure convergence on these values. This would be useful in the case of a highly nonlinear target response that is difficult to converge on within the required tolerance.

In the development of the "BrickOval" UC in Chapter 4, it is clear how a single EFG, the fixed-fixed beam, was integrated in the "Brick" UC by using a 0th order configuration. After testing for feasibility during concept evaluation (Step 5), it was determined that the concept's behavior could not converge to the target response. Thus, the design was iterated by repeating steps 2-5. In this second iteration, the oval was added to the fixed-fixed beam by means of a 1st order configuration in parallel with EFGs having inverse stiffnesses. The effective stiffness of this configuration can be visualized on the right image of Figure 5.3. In the tank track pad application, the number of UC tessellations was ultimately limited by the available design space. Based on the "BrickOval" UC design, the convergence study showed that a RVE of 20x20 UCs would be necessary to determine properties representative of a meta-material with a very large number of UCs.

5.4. Conclusion

The method proposed in this chapter expands on the answer to the research question 2.b presented in Chapter 4.

> R2. What method can be used to successfully develop a meta-material to meet the nonlinear deformation response of the current application?
>
> b. Can a method be developed to accomplish this?

The proposed design method represents a general design framework that was developed to design meta-materials to match a nonlinear target response. However, the abstraction of the approach developed in Chapter 4 into a general design framework adds to the knowledge base of the design community. Specifically, it can be used as a tool for designers seeking to develop meta-materials to match a nonlinear target response, especially given the absence of any other meta-material design methods for this application in the current literature.

It is important to note that the design framework presented in this work is preliminary. This method must be applied to many additional case studies to prove its validity. Many new EFGs should be explored and analyzed to build a substantial EFG library for designers to choose from. Additional EFG connection configurations can also be explored for targeting higher order and more complex nonlinear responses. Furthermore, a multi-objective optimization process can be implemented to minimize stress while also considering other manufacturability constraints to further reduce design time. The method presented in this work only considers a single target deformation curve. It would be useful to extend to this framework to simultaneously target multiple deformation modes such as equi-biaxial tension, compression, and shear.

CHAPTER 6. CONCLUSIONS AND FUTURE WORK

6.1. Conclusions

This research contained within this work was successful in answering all three research questions and was able to contribute to the knowledge in the research community regarding meta-material design. The first, and primary, research question was, “Can a meta-material be developed in which the global behavior of the resultant medium exhibits nonlinear compressive behavior similar to that of the current elastomer?” The “BrickOval” unit cell was developed based on engineering principles and was successful in matching a nonlinear deformation response of the given tank track pad application. This primary research objective was reached only after answering the second and third research questions.

The second research question was, “What method can be used to successfully develop a meta-material to meet the nonlinear deformation response of the current application?” This research question was answered by asking two separate sub-questions.

Research question 2.a was, “Is topology optimization a feasible method to accomplish this?” Through a literature review of contemporary research, it was first determined that topology optimization was the only existing method used to design meta-material topologies. After further investigation, it was determined that topology optimization is currently not a suitable method to design a meta-material to match a nonlinear deformation response. The current limitations in topology optimization include implementation of geometric nonlinearity in the numeric formulation of the optimization algorithm, determining a process to solve the nonlinear inverse homogenization problem, using periodic unit cell boundary conditions, and taking into account the unit cell aspect ratio. These four limitations should be addressed in future research to increase the domain of applications topology optimization can be successfully used as a design tool.

Research question 2.b was, “Can a method be developed to accomplish this?” A method involving engineering principles, a fundamental understanding of elemental geometry nonlinearity, and combining the stiffnesses of multiple elemental geometries was used in the design of the “BrickOval” unit cell. This method was then abstracted into the more general Unit Cell Synthesis Method that represents a framework to be used as a design tool for future meta-material designers targeting a nonlinear response. The proposed synthesis method answers sub-question 2.b as well as original research question 2 by showing that a meta-material with the desired response could be developed.

6.2. Broader Impact

This work yielded knowledge that will have several broader impacts. The four limitations in topology optimization can be used as gaps to be addressed in future research to expand the applications in which this technique can be successfully used as a design tool. The meta-material that was designed in Chapter 4 demonstrated it was possible to design a meta-material to match a nonlinear deformation response. This was a previously unexplored area and therefore expands the knowledge in meta-material design and its potential applications. Additionally, the logical and systematic design method depicted in Chapter 5 represents a framework that can aid future meta-material designers. Finally, users of this method can add to the research knowledge by testing new EFGs, adding them to the repository, and sharing them with the research community.

6.3. Future Work

6.3.1. Tank Track Pad Application

While this work proved that designing a meta-material to match the target response was feasible, several application-specific constraints were not met. The maximum stress experienced within the constitutive material must be lowered to well below that of yielding to not only eliminate the possibility of plastic deformation, but to also consider the high cycle life of the meta-material. Also, manufacturing constraints such as minimum feature size should be considered in greater detail. Therefore, the meta-material redesign should use the framework provided addressing these issues under static load conditions, and optimized to match the response curve.

Once this has been accomplished, dynamic simulations must be conducted to ensure the meta-material response matches that of the current rubber pad under these more complex conditions. At the meta-material level, aspects of strain rate and road wheel interaction should be compared to that of the current rubber pad. At the track system level, issues of vibration and dampening effects should also be considered. Before the meta-material can be implemented with the physical track system, one must determine how to replicate the boundary conditions in the simulations within the physical system. This is not a trivial task as a possible solution may introduce friction and other forces that were not considered in these models. The ultimate motivation of the meta-material design was to improve fatigue life of the current rubber pad. Thus, the fatigue properties should be determined by appropriate simulations and physical testing.

6.3.2. Synthesis Method

Currently, the proposed method has not been rigorously tested to prove its use as a design tool. Therefore, this method should be applied to many different case studies to prove validity. Also, the proposed meta-material design method only represents a framework and, as a result, a number of improvements can be made to increase its usefulness as a design tool. Many new EFGs should be explored and analyzed to build a substantial EFG repository for designers to choose from. Additional EFG connection configurations can also be explored for targeting higher order and more complex nonlinear responses. Furthermore, a multi-objective optimization process can be implemented to minimize stress while also considering other manufacturability constraints to further reduce design time. The method presented in this work only considers a single target deformation curve. It would be useful to extend this framework to simultaneously target multiple deformation modes such as equi-biaxial tension, compression, and shear.

CHAPTER 7. REFERENCES

[1] Katz, H. S., and Wittig, J. T., 1986, Development and Fabrication of Track Pads.

[2] Ostberg, D., and Bradford, B., 2009, “Impact of Loading Distribution of Abrams Suspension on Track Performance and Durability,” Ground Vehicle Systems Engineering and Technology Symposium, pp. 1–11.

[3] Dangeti, V. S., 2014, “Identifying Target Properties for the Design of Meta-Material Tank Track Pads,” Igarss 2014, (1), pp. 1–5.

[4] Bement, A. L., Bowen, H. K., Burlant, W. J., Burrows, J. C., Decker, R. F., Frost, B. R. T., Gratch, S., Holonyak, N., Isakoff, S. E., Jaumot, F. E., Jorgensen, P. J., Lawley, A., Mikesell, R. F., Okrent, D., Pipes, B., Rushton, B. M., Schanz, J. J., Simon, D. M., Tenebaum, M., Vogely, W. A., Wei, R. P., and Westwood, A. R. C., 1982, Elastomers for Tank-Track Pads, Washington, DC.

[5] Lesuer, D. R., Santor, S. D., Cornell, R. H., and Patt, J., 1983, Field Evaluation of Tank Track Pad Failures.

[6] Lesuer, D. R., Goldberg, A., and Patt, J., 1985, Computer Modeling of Tank Track Elastomers.

[7] Mars, W. V, and Ostberg, D., 2012, “Fatigue Damage Analysis of an Elastomeric Tank Track Component,” Simulia Community Conference, pp. 1–14.

[8] Thyagaraja, N. B., 2011, “Requirements Determination of a Novel Non-Pneumatic Wheel Shear Beam for Low Rolling Resistance.”

[9] Ashby, M. F., 2011, Material Selection in Mechanical Design, Elsevier, Burlington.

[10] Czech, C., Guarneri, P., Gibert, J., and Fadel, G., 2012, “On the accurate analysis of linear elastic meta-material properties for use in design optimization problems,” Compos. Sci. Technol., **72**(5), pp. 580–586.

[11] Walser, R. M., 2001, "Electromagnetic Metamaterials," International Symposium on Optical Science and Technology, A. Lakhtakia, W.S. Weiglhofer, and I.J. Hodgkinson, eds., International Society for Optics and Photonics, pp. 1–15.

[12] Cui, T. J., Smith, D. R., and Liu, R., 2009, Metamaterials: Theory, Design, and Applications, Springer Science & Business Media, New York.

[13] Babuska, I., 1976, Homogenization Approach in Engineering, Springer.

[14] Allaire, G., 2002, Shape Optimization by the Homogenization Method, Springer Science & Business Media.

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

[16] Hassani, B., and Hinton, E., 1998, "A review of homogenization and topology optimization I—homogenization theory for media with periodic structure," Comput. Struct., **69**, pp. 707–717.

[17] Hassani, B., and Hinton, E., 1998, "A review of homogenization and topology opimization II—analytical and numerical solution of homogenization equations," Comput. Struct., **69**(6), pp. 719–738.

[18] Hassani, B., and Hinton, E., 1998, "A review of homogenization and topology optimization III—topology optimization using optimality criteria," Comput. Struct., **69**(6), pp. 739–756.

[19] Nishiwaki, S., Frecker, M. I., Min, S., and Kikuchi, N., 1998, "Topology optimization of compliant mechanisms using the homogenization method," Int. J. Numer. Methods Eng., **42**(3), pp. 535–559.

[20] Sigmund, O., 2011, "On the usefulness of non-gradient approaches in topology optimization," Struct. Multidiscip. Optim., **43**(5), pp. 589–596.

[21] Sigmund, O., 2001, "A 99 line topology optimization code written in matlab," Struct. Multidiscip. Optim., **21**(2), pp. 120–127.

[22] Bendsøe, M. P., and Sigmund, O., 1999, “Material interpolation schemes in topology optimization,” Arch. Appl. Mech., **69**(9-10), pp. 635–654.

[23] Eschenauer, H. a, and Olhoff, N., 2001, “Topology optimization of continuum structures: A review,” Appl. Mech. Rev., **54**(4), p. 331.

[24] Sigmund, O., 1994, “Materials with Prescribed Constitutive Parameters: An Inverse Homogenization Problem,” Int. J. Solids Struct., **31**(17), pp. 2313–2329.

[25] Sigmund, O., 1995, “Tailoring materials with prescribed elastic properties,” Mech. Mater., **20**(4), pp. 351–368.

[26] Bénard, A., and Diaz, A. R., 2001, “On the discretization of problems involving periodic planar tilings,” Commun. Numer. Methods Eng., **17**(8), pp. 543–549.

[27] Diaz, A. R., and Bénard, A., 2003, “Designing materials with prescribed elastic properties using polygonal cells,” Int. J. Numer. Methods Eng., **57**(3), pp. 301–314.

[28] Sigmund, O., 2009, “Systematic Design of Metamaterials by Topology Optimization,” IUTAM Symp. Model. Nanomater. Nanosyst., pp. 151–159.

[29] Wang, M., Wang, X., and Guo, D., 2003, “A level set method for structural topology optimization,” Comput. methods Appl. Mech. …, **192**, pp. 227–246.

[30] van Dijk, N. P., Maute, K., Langelaar, M., and van Keulen, F., 2013, “Level-set methods for structural topology optimization: a review,” Struct. Multidiscip. Optim., **48**(3), pp. 437–472.

[31] Deaton, J. D., and Grandhi, R. V., 2014, “A survey of structural and multidisciplinary continuum topology optimization: Post 2000,” Struct. Multidiscip. Optim., **49**(1), pp. 1–38.

[32] Allaire, G., Jouve, F., and Toader, A., 2002, “A Level-set Method for Shape Optimization,” Comptes Rendus Math., **334**(12), pp. 1125–1130.

[33] Rao, S. S., 2009, Engineering Optimization: Theory and Practice, John Wiley & Sons.

[34] Hassan, R., Cohanim, B., Weck, O. De, and Venter, G., 2005, “A Comparison of Particle Swarm Optimization and the Genetic Algorithm,” Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference, American Institute of Aeronautics and Astronautics, pp. 1–13.

[35] Nguyen, J., Park, S., and Rosen, D., 2013, “Heuristic optimization method for cellular structure design of light weight components,” Int. J. Precis. …, **14**(6), pp. 1071–1078.

[36] Li, Y., Chen, Y., and Zhou, C., 2009, “Design of Flexible Skin for Target Displacements Based on Meso-Structures,” ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, San Diego, California, p. C2009/CIE–87137.

[37] Gary Wang, G., Dong, Z., and Aitchison, P., 2001, “Adaptive Response Surface Method - a Global Optimization Scheme for Approximation-Based Design Problems,” Eng. Optim., **33**(6), pp. 707–733.

[38] Box, G., and Wilson, K., 1951, “On the experimental attainment of optimum conditions,” J. R. Stat. Soc., **13**(1), pp. 1–45.

[39] Czech, C., Guarneri, P., and Fadel, G., 2012, “Meta-Material Design of the Shear Layer of a Non-Pneumatic Wheel Using Topology Optimization,” International Design Engineering Technical Conferences and Computers in Information in Engineering Conference, pp. 1–11.

[40] Gibiansky, L. V, and Sigmund, O., 2000, “Multiphase Composites with Extremal Bulk Modulus,” J. Mech. Phys. Solids, **48**(3), pp. 461–498.

[41] Carstensen, J. V., Guest, J. K., Lotfi, R., Chen, W., and Schroers, J., 2015, “Topology Optimization of Cellular Materials with Maximized Energy Absorption,” International Design Engineering Technical Conferences and Computers in Information in Engineering Conference, Boston, MA, pp. 1–10.

[42] Schultz, J., Griese, D., Ju, J., Shankar, P., Summers, J. D., and Thompson, L., 2012,

"Design of Honeycomb Mesostructures for Crushing Energy Absorption," J. Mech. Des., **134**(7), p. 071004.

[43] Mehta, V., 2010, "Design, Analysis, and Applications of Cellular Contact-Aided Compliant Mechanisms," Pennsylvania State University.

[44] Choi, S.-K., and Patel, J., 2011, "Optimal Design of Cellular Structures under Random Fields," J. Eng. Des., **22**(9), pp. 651–668.

[45] 2011, OptiStruct 11.0 User Guide, Altair Engineering, Inc.

[46] Sadd, M. H., 2005, Elasticity - Theory, Applications, and Numerics, Elsevier Butterworth-Heinemann, Burlington.

[47] Daeyoon, J., and Gea, H. chang, 2004, "Topology Optimization of Nonlinear Structures," Fish. Res., **70**(2-3), pp. 141–159.

[48] Buhl, T., Pedersen, C. B. W., and Sigmund, O., 2000, "Stiffness Design of Geometrically Nonlinear Structures using Topology Optimization," Struct. Multidiscip. Optim., **19**, pp. 93–104.

[49] Czech, C., 2012, "Design of Meta-Materials Outside the Homogenization Limit Using Multiscale Analysis and Topology Optimization."

[50] Paulino, G. H., Silva, E. C. N., and Le, C. H., 2009, "Optimal Design of Periodic Functionally Graded Composites with Prescribed Properties," Struct. Multidiscip. Optim., **38**(5), pp. 469–489.

[51] Lipperman, F., Fuchs, M. B., and Ryvkin, M., 2008, "Stress localization and strength optimization of frame material with periodic microstructure," Comput. Methods Appl. Mech. Eng., **197**(45-48), pp. 4016–4026.

[52] Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., and Sigmund, O., 2011, "Efficient topology optimization in MATLAB using 88 lines of code," Struct. Multidiscip. Optim., **43**(1), pp. 1–16.

APPENDICES

APPENDIX A. PYTHON SCRIPT FOR BRICK DESIGN

The following Python script was used to generate data for the sensitivity analysis of the “Brick” UC design. This script generated the UC geometry per dimensional inputs, tessellated the UC into a 4x4 meta-material, applied loading and boundary conditions, generated a mesh on the meta-material, iterated a static nonlinear analysis across four load cases, and exported results of each analysis to a report file. The code was generalized to allow inputs of “1” or “0” for each of the three variables of interest (L, H, and t_1) to indicate “high” and “low” values, respectively.

```
# # Brick UC Sensitivity Python Script
# # By: Zachary Satterfield
# # 3/14/2015
#

w=0
h=0
t=0
# ------
# ------ INPUT VARIABLES
# ------
# ------
# ------ Dimensions
# ------
if w==1:
        Width=15
else:
        Width=10
if h==1:
        Height=5.0
else:
        Height=3.0
if t==1:
        t_2=0.1
else:
        t_2=0.075
```

```
t_1=0.5
thick=1.0   # thickness
# ------
# ------ Material Properties
# ------
density=7.75e-06
E_mod=210000
Poisson=0.30
# ------
# ------ Names
# ------
rptName='brick_orthogonal_'+str(w)+str(h)+str(t)+'.rpt'
# ------
# ------ Load
# ------
pressure1=0.4
pressure2=0.8
pressure3=2.0
pressure4=4.0
# ------
# ------

from abaqus import *
from abaqusConstants import *
session.Viewport(name='Viewport: 1', origin=(0.0, 0.0), width=164.556259155273,
    height=151.574996948242)
session.viewports['Viewport: 1'].makeCurrent()
session.viewports['Viewport: 1'].maximize()
from caeModules import *
from driverUtils import executeOnCaeStartup
executeOnCaeStartup()
session.viewports['Viewport: 1'].partDisplay.geometryOptions.setValues(
    referenceRepresentation=ON)
s = mdb.models['Model-1'].ConstrainedSketch(name='__profile__',
    sheetSize=200.0)
g, v, d, c = s.geometry, s.vertices, s.dimensions, s.constraints
s.setPrimaryObject(option=STANDALONE)
s.Spot(point=(0.0, 0.0))
s.FixedConstraint(entity=v[0])
s.rectangle(point1=(0.0, 0.0), point2=(Width, Height))
s.rectangle(point1=(1.0, 1.0), point2=(Width-1, Height-1))
session.viewports['Viewport: 1'].view.setValues(nearPlane=184.137,
    farPlane=192.986, width=38.7365, height=18.6266, cameraPosition=(6.9825,
    3.412, 188.562), cameraTarget=(6.9825, 3.412, 0))
s.ObliqueDimension(vertex1=v[4], vertex2=v[5], textPoint=(3.32142686843872,
    -1.53938269615173), value=Width)
s.ObliqueDimension(vertex1=v[1], vertex2=v[2], textPoint=(-2.46543073654175,
    2.35099101066589), value=Height)
s.DistanceDimension(entity1=g[6], entity2=g[2], textPoint=(0.526414394378662,
    6.3199577331543), value=1.0)
s.DistanceDimension(entity1=g[8], entity2=g[4], textPoint=(9.81687927246094,
    6.51644229888916), value=1.0)
s.DistanceDimension(entity1=g[9], entity2=g[5], textPoint=(11.548999786377,
    -0.0461080074310303), value=1.0)
s.DistanceDimension(entity1=g[7], entity2=g[3], textPoint=(12.3363265991211,
    4.31582641601563), value=1.0)
s=mdb.models['Model-1'].sketches['__profile__']
s.Parameter(name='dimensions_0', path='dimensions[0]', expression=str(Width))
s.Parameter(name='dimensions_1', path='dimensions[1]', expression=str(Height),
    previousParameter='dimensions_0')
s.Parameter(name='t1', path='dimensions[3]', expression=str(t_1),
    previousParameter='dimensions_1')
s.Parameter(name='dimensions_2', path='dimensions[2]', expression='t1',
    previousParameter='t1')
s.Parameter(name='t2', path='dimensions[5]', expression=str(t_2),
    previousParameter='dimensions_2')
s.Parameter(name='dimensions_4', path='dimensions[4]', expression='t2',
    previousParameter='t2')
p = mdb.models['Model-1'].Part(name='Part-1', dimensionality=TWO_D_PLANAR,
    type=DEFORMABLE_BODY)
p = mdb.models['Model-1'].parts['Part-1']
p.BaseShell(sketch=s)
s.unsetPrimaryObject()
p = mdb.models['Model-1'].parts['Part-1']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
del mdb.models['Model-1'].sketches['__profile__']
session.viewports['Viewport: 1'].view.setValues(nearPlane=20.1852,
    farPlane=24.5361, width=18.724, height=8.47169, viewOffsetX=2.47589,
    viewOffsetY=0.814535)
session.viewports['Viewport: 1'].partDisplay.setValues(sectionAssignments=ON,
    engineeringFeatures=ON)
session.viewports['Viewport: 1'].partDisplay.geometryOptions.setValues(
    referenceRepresentation=OFF)
mdb.models['Model-1'].Material(name='Material-1')
mdb.models['Model-1'].materials['Material-1'].Density(table=((density, ), ))
mdb.models['Model-1'].materials['Material-1'].Elastic(table=((E_mod, Poisson),
    ))
```

```
mdb.models['Model-1'].HomogeneousSolidSection(name='Section-1',
   material='Material-1', thickness=thick)
p = mdb.models['Model-1'].parts['Part-1']
f = p.faces
faces = f.getSequenceFromMask(mask=('[#1 ]', ), )
region = p.Set(faces=faces, name='Set-1')
p = mdb.models['Model-1'].parts['Part-1']
p.SectionAssignment(region=region, sectionName='Section-1', offset=0.0,
   offsetType=MIDDLE_SURFACE, offsetField='',
   thicknessAssignment=FROM_SECTION)
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(
   optimizationTasks=OFF, geometricRestrictions=OFF, stopConditions=OFF)
a = mdb.models['Model-1'].rootAssembly
a.DatumCsysByDefault(CARTESIAN)
p = mdb.models['Model-1'].parts['Part-1']
a.Instance(name='Part-1-1', part=p, dependent=OFF)
session.viewports['Viewport: 1'].view.setValues(nearPlane=19.3892,
   farPlane=25.3322, width=26.8518, height=12.1491, viewOffsetX=5.51218,
   viewOffsetY=2.49527)
a = mdb.models['Model-1'].rootAssembly
a.LinearInstancePattern(instanceList=('Part-1-1', ), direction1=(1.0, 0.0,
   0.0), direction2=(0.0, 1.0, 0.0), number1=2, number2=1, spacing1=(Width-t_1)/2,
   spacing2=5.0)
a = mdb.models['Model-1'].rootAssembly
a.translate(instanceList=('Part-1-1-lin-2-1', ), vector=(0.0, Height-t_2, 0.0))
#: The instance Part-1-1-lin-2-1 was translated by 0., 4.5, 0. with respect to the assembly
coordinate system
a = mdb.models['Model-1'].rootAssembly
a.InstanceFromBooleanMerge(name='Part-2', instances=(a.instances['Part-1-1'],
   a.instances['Part-1-1-lin-2-1'], ), originalInstances=SUPPRESS,
   domain=GEOMETRY)
session.viewports['Viewport: 1'].view.setValues(nearPlane=16.9686,
   farPlane=27.7528, width=46.1837, height=20.8958, viewOffsetX=10.4106,
   viewOffsetY=5.00072)
a = mdb.models['Model-1'].rootAssembly
a.LinearInstancePattern(instanceList=('Part-2-1', ), direction1=(1.0, 0.0,
   0.0), direction2=(0.0, 1.0, 0.0), number1=3, number2=2, spacing1=(Width-t_1),
   spacing2=2*(Height-t_2))
a = mdb.models['Model-1'].rootAssembly
a.InstanceFromBooleanMerge(name='Part-3', instances=(a.instances['Part-2-1'],
   a.instances['Part-2-1-lin-1-2'], a.instances['Part-2-1-lin-2-1'],
   a.instances['Part-2-1-lin-2-2'], a.instances['Part-2-1-lin-3-1'],
    a.instances['Part-2-1-lin-3-2'], ), originalInstances=SUPPRESS,
    domain=GEOMETRY)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(
    adaptiveMeshConstraints=ON)
mdb.models['Model-1'].StaticStep(name='Step-1', previous='Initial', nlgeom=ON)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(step='Step-1')
mdb.models['Model-1'].fieldOutputRequests['F-Output-1'].setValues(variables=(
    'S', 'PE', 'PEEQ', 'PEMAG', 'LE', 'U'))
a = mdb.models['Model-1'].rootAssembly
v1 = a.instances['Part-3-1'].vertices
verts1 = v1.getSequenceFromMask(mask=('[#0 #20000 ]', ), )
a.Set(vertices=verts1, name='yDisp')
#: The set 'yDisp' has been created (1 vertex).
regionDef=mdb.models['Model-1'].rootAssembly.sets['yDisp']
mdb.models['Model-1'].historyOutputRequests['H-Output-1'].setValues(variables=(
    'U2', ), frequency=LAST_INCREMENT, region=regionDef, sectionPoints=DEFAULT,
    rebar=EXCLUDE)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(loads=ON, bcs=ON,
    predefinedFields=ON, connectors=ON, adaptiveMeshConstraints=OFF)
a = mdb.models['Model-1'].rootAssembly
s1 = a.instances['Part-3-1'].edges
side1Edges1 = s1.getSequenceFromMask(mask=('[#0 #3e000 ]', ), )
region = a.Surface(side1Edges=side1Edges1, name='Surf-1')
mdb.models['Model-1'].Pressure(name='Load-1', createStepName='Step-1',
    region=region, distributionType=UNIFORM, field='', magnitude=pressure1,
    amplitude=UNSET)
a = mdb.models['Model-1'].rootAssembly
e1 = a.instances['Part-3-1'].edges
edges1 = e1.getSequenceFromMask(mask=('[#0 #3e000000 ]', ), )
region = a.Set(edges=edges1, name='Set-2')
mdb.models['Model-1'].DisplacementBC(name='BC-1', createStepName='Step-1',
    region=region, u1=0.0, u2=0.0, ur3=UNSET, amplitude=UNSET, fixed=OFF,
    distributionType=UNIFORM, fieldName='', localCsys=None)
a = mdb.models['Model-1'].rootAssembly
e1 = a.instances['Part-3-1'].edges
edges1 = e1.getSequenceFromMask(mask=('[#0 #1101000 #1 ]', ), )
region = a.Set(edges=edges1, name='Set-3')
mdb.models['Model-1'].DisplacementBC(name='BC-2', createStepName='Step-1',
    region=region, u1=0.0, u2=UNSET, ur3=UNSET, amplitude=UNSET, fixed=OFF,
    distributionType=UNIFORM, fieldName='', localCsys=None)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(mesh=ON, loads=OFF,
    bcs=OFF, predefinedFields=OFF, connectors=OFF)
session.viewports['Viewport: 1'].assemblyDisplay.meshOptions.setValues(
    meshTechnique=ON)
```

```
a = mdb.models['Model-1'].rootAssembly
e1 = a.instances['Part-3-1'].edges
edges1 = e1.getSequenceFromMask(mask=('[#0 #441000 #1 ]', ), )
a.Set(edges=edges1, name='Set-3')
p = mdb.models['Model-1'].parts['Part-1']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
session.viewports['Viewport: 1'].partDisplay.setValues(sectionAssignments=OFF,
    engineeringFeatures=OFF, mesh=ON)
session.viewports['Viewport: 1'].partDisplay.meshOptions.setValues(
    meshTechnique=ON)
# ---
# --- Mesh ---
# ---
p = mdb.models['Model-1'].parts['Part-3']
e = p.edges
pickedEdges = e.getSequenceFromMask(mask=(
    '[#150448a4 #41541451 #a124510b #a ]', ), )
p.seedEdgeBySize(edges=pickedEdges, size=t_1*0.4, deviationFactor=0.1,
    constraint=FINER)
p = mdb.models['Model-1'].parts['Part-3']
e = p.edges
edges = e.getSequenceFromMask(mask=('[#150448a4 #41541451 #a124510b #a ]', ), )
p.Set(edges=edges, name='EdgeSeeds_ThickBeam')
p = mdb.models['Model-1'].parts['Part-3']
e = p.edges
pickedEdges = e.getSequenceFromMask(mask=(
    '[#eafbb75b #beabebae #5edbaef4 #35 ]', ), )
p.seedEdgeBySize(edges=pickedEdges, size=t_2*0.3, deviationFactor=0.1,
    constraint=FINER)
p = mdb.models['Model-1'].parts['Part-3']
e = p.edges
edges = e.getSequenceFromMask(mask=('[#eafbb75b #beabebae #5edbaef4 #35 ]', ),
    )
p.Set(edges=edges, name='EdgeSeeds_ThinBeam')
p = mdb.models['Model-1'].parts['Part-3']
p.generateMesh()
# ---
# ---
# ---
session.viewports['Viewport: 1'].view.setValues(nearPlane=69.5691,
    farPlane=80.017, width=44.8615, height=20.3804, viewOffsetX=2.63266,
    viewOffsetY=0.179799)
a1 = mdb.models['Model-1'].rootAssembly
a1.regenerate()
```

```
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=a)
a = mdb.models['Model-1'].rootAssembly
e1 = a.instances['Part-3-1'].edges
edges1 = e1.getSequenceFromMask(mask=('[#0 #10000 ]', ), )
a.Set(edges=edges1, name='Set-4')
session.viewports['Viewport: 1'].assemblyDisplay.setValues(loads=ON, bcs=ON,
   predefinedFields=ON, connectors=ON, adaptiveMeshConstraints=OFF)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(loads=OFF, bcs=OFF,
   predefinedFields=OFF, connectors=OFF, adaptiveMeshConstraints=ON)
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(
   adaptiveMeshConstraints=ON)
regionDef=mdb.models['Model-1'].rootAssembly.sets['Set-4']
mdb.models['Model-1'].FieldOutputRequest(name='F-Output-2',
   createStepName='Step-1', variables=('UT', ), frequency=LAST_INCREMENT,
   region=regionDef, sectionPoints=DEFAULT, rebar=EXCLUDE)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(
   adaptiveMeshConstraints=OFF)
session.viewports['Viewport: 1'].assemblyDisplay.meshOptions.setValues(
   meshTechnique=OFF)
mdb.Job(name='Job-1', model='Model-1', description='', type=ANALYSIS,
   atTime=None, waitMinutes=0, waitHours=0, queue=None, memory=90,
   memoryUnits=PERCENTAGE, getMemoryFromAnalysis=True,
   explicitPrecision=SINGLE, nodalOutputPrecision=SINGLE, echoPrint=OFF,
   modelPrint=OFF, contactPrint=OFF, historyPrint=OFF, userSubroutine='',
   scratch='', resultsFormat=ODB, multiprocessingMode=DEFAULT, numCpus=1,
   numGPUs=0)
mdb.jobs['Job-1'].submit(consistencyChecking=OFF)
#
# ----- Wait for job to complete
#
mdb.jobs['Job-1'].waitForCompletion()
#
# ----- Generate Report
#
o3 = session.openOdb(name='Job-1.odb')
lastFrame=o3.steps['Step-1'].frames[-1]
session.viewports['Viewport: 1'].setValues(displayedObject=o3)
odb = session.odbs['Job-1.odb']
session.fieldReportOptions.setValues(printXYData=OFF, printTotal=OFF)
session.writeFieldReport(fileName=rptName, append=OFF,
   sortItem='Node Label', odb=odb, step=0, frame=lastFrame, outputPosition=NODAL,
    variable=(('UT', NODAL, ((COMPONENT, 'UT2'), )), ))


 # ---
 # ---
 # --- Second Load Step ---
 # ---
 # ---
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(mesh=OFF, loads=ON,
     bcs=ON, predefinedFields=ON, connectors=ON)
mdb.models['Model-1'].loads['Load-1'].setValues(magnitude=pressure2)
mdb.jobs['Job-1'].submit(consistencyChecking=OFF)
 #
 # ----- Wait for job to complete
 #
mdb.jobs['Job-1'].waitForCompletion()
 #
 # ----- Generate Report
 #
o3 = session.openOdb(name='Job-1.odb')
lastFrame=o3.steps['Step-1'].frames[-1]
session.viewports['Viewport: 1'].setValues(displayedObject=o3)
odb = session.odbs['Job-1.odb']
session.fieldReportOptions.setValues(printXYData=OFF, printTotal=OFF)
session.writeFieldReport(fileName=rptName, append=ON,
     sortItem='Node Label', odb=odb, step=0, frame=lastFrame, outputPosition=NODAL,
     variable=(('UT', NODAL, ((COMPONENT, 'UT2'), )), ))


# ---
# ---
# --- Third Load Step ---
# ---
# ---
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(mesh=OFF, loads=ON,
    bcs=ON, predefinedFields=ON, connectors=ON)
mdb.models['Model-1'].loads['Load-1'].setValues(magnitude=pressure3)
mdb.jobs['Job-1'].submit(consistencyChecking=OFF)
#
# ----- Wait for job to complete
```

```
#
mdb.jobs['Job-1'].waitForCompletion()
#
# ----- Generate Report
#
o3 = session.openOdb(name='Job-1.odb')
lastFrame=o3.steps['Step-1'].frames[-1]
session.viewports['Viewport: 1'].setValues(displayedObject=o3)
odb = session.odbs['Job-1.odb']
session.fieldReportOptions.setValues(printXYData=OFF, printTotal=OFF)
session.writeFieldReport(fileName=rptName, append=ON,
    sortItem='Node Label', odb=odb, step=0, frame=lastFrame, outputPosition=NODAL,
    variable=(('UT', NODAL, ((COMPONENT, 'UT2'), )), ))


# ---
# ---
# --- Fourth Load Step ---
# ---
# ---
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(mesh=OFF, loads=ON,
    bcs=ON, predefinedFields=ON, connectors=ON)
mdb.models['Model-1'].loads['Load-1'].setValues(magnitude=pressure4)
mdb.jobs['Job-1'].submit(consistencyChecking=OFF)
#
# ----- Wait for job to complete
#
mdb.jobs['Job-1'].waitForCompletion()
#
# ----- Generate Report
#
o3 = session.openOdb(name='Job-1.odb')
lastFrame=o3.steps['Step-1'].frames[-1]
session.viewports['Viewport: 1'].setValues(displayedObject=o3)
odb = session.odbs['Job-1.odb']
session.fieldReportOptions.setValues(printXYData=OFF, printTotal=OFF)
session.writeFieldReport(fileName=rptName, append=ON,
    sortItem='Node Label', odb=odb, step=0, frame=lastFrame, outputPosition=NODAL,
    variable=(('UT', NODAL, ((COMPONENT, 'UT2'), )), ))
```

APPENDIX B. PYTHON SCRIPT FOR BRICKOVAL DESIGN

The following Python script was used in a ModeFrontier optimization routine to optimize the dimensions of the “BrickOval” UC design. This script generated the UC geometry per dimensional inputs, tessellated the UC into a meta-material, applied loading and boundary conditions, generated a mesh on the meta-material, iterated a static nonlinear analysis across four load cases, and exported results of each analysis to a report file. This script was generalized to modify all five “BrickOval” UC parameters, tessellate the UC a specified number of times in the x- and y-directions, respectively, and apply loading and boundary conditions accordingly.

```
# # BrickOval UC Python Script
# # By: Zachary Satterfield
# # 10/01/2015
#
# Number of UCs in x-direction
xdir = 4
# Number of UCs in y-direction (divided by two)
ydir = 2
jobName = 'BO_'+str(xdir)+'x'+str(2*ydir)
# # ------
# # ------ INPUT VARIABLES
# # ------
# # ------
# # ------ Dimensions
# # ------

Width=25.0
Height=5.0
t_2=0.65
t_3=0.5
t_1=1.5
thick=1.0          # thickness
ovalbig=Width/2-t_1
ovalsmall=Height/2-t_2
```

```
# ------
# ------ Material Properties
# ------
density=7.75e-06
E_mod=210000
Poisson=0.30
# ------
# ------ Names
# ------
rptName='BO_'+str(xdir)+'x'+str(2*ydir)+'.rpt'
# ------
# ------ Load
# ------
pressure1=0.4
pressure2=0.8
pressure3=2.0
pressure4=4.0
# ------
# ------

import math
from abaqus import *
from abaqusConstants import *
session.Viewport(name='Viewport: 1', origin=(0.0, 0.0), width=164.556259155273,
   height=151.574996948242)
session.viewports['Viewport: 1'].makeCurrent()
session.viewports['Viewport: 1'].maximize()
from caeModules import *
from driverUtils import executeOnCaeStartup
executeOnCaeStartup()
session.viewports['Viewport: 1'].partDisplay.geometryOptions.setValues(
   referenceRepresentation=ON)
s = mdb.models['Model-1'].ConstrainedSketch(name='__profile__',
   sheetSize=200.0)
g, v, d, c = s.geometry, s.vertices, s.dimensions, s.constraints
s.setPrimaryObject(option=STANDALONE)
s.Spot(point=(0.0, 0.0))
s.FixedConstraint(entity=v[0])
s.rectangle(point1=(0.0, 0.0), point2=(Width, Height))
s.rectangle(point1=(1.0, 1.0), point2=(Width-1, Height-1))

s.ObliqueDimension(vertex1=v[4], vertex2=v[5], textPoint=(3.32142686843872,
   -1.53938269615173), value=Width)
s.ObliqueDimension(vertex1=v[1], vertex2=v[2], textPoint=(-2.46543073654175,
   2.35099101066589), value=Height)
s.DistanceDimension(entity1=g[6], entity2=g[2], textPoint=(0.526414394378662,
   6.3199577331543), value=1.0)
s.DistanceDimension(entity1=g[8], entity2=g[4], textPoint=(9.81687927246094,
   6.51644229888916), value=1.0)
s.DistanceDimension(entity1=g[9], entity2=g[5], textPoint=(11.548999786377,
   -0.0461080074310303), value=1.0)
s.DistanceDimension(entity1=g[7], entity2=g[3], textPoint=(12.3363265991211,
   4.31582641601563), value=1.0)
s=mdb.models['Model-1'].sketches['__profile__']
s.Parameter(name='dimensions_0', path='dimensions[0]', expression=str(Width))
s.Parameter(name='dimensions_1', path='dimensions[1]', expression=str(Height),
   previousParameter='dimensions_0')
s.Parameter(name='t1', path='dimensions[3]', expression=str(t_1),
   previousParameter='dimensions_1')
s.Parameter(name='dimensions_2', path='dimensions[2]', expression='t1',
   previousParameter='t1')
s.Parameter(name='t2', path='dimensions[5]', expression=str(t_2),
   previousParameter='dimensions_2')
s.Parameter(name='dimensions_4', path='dimensions[4]', expression='t2',
   previousParameter='t2')
p = mdb.models['Model-1'].Part(name='Part-1', dimensionality=TWO_D_PLANAR,
   type=DEFORMABLE_BODY)

p = mdb.models['Model-1'].parts['Part-1']
p.BaseShell(sketch=s)
s.unsetPrimaryObject()
p = mdb.models['Model-1'].parts['Part-1']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
del mdb.models['Model-1'].sketches['__profile__']
s = mdb.models['Model-1'].ConstrainedSketch(name='__profile__',
   sheetSize=200.0)
g, v, d, c = s.geometry, s.vertices, s.dimensions, s.constraints
s.setPrimaryObject(option=STANDALONE)
s.Spot(point=(Width/2, Height/2))

s.FixedConstraint(entity=v[0])
session.viewports['Viewport: 1'].view.setValues(nearPlane=186.181,
   farPlane=190.942, width=21.4928, height=10.3349, cameraPosition=(3.72162,
   1.84548, 188.562), cameraTarget=(3.72162, 1.84548, 0))
s.EllipseByCenterPerimeter(center=(Width/2, Height/2), axisPoint1=(t_1, Height/2),
   axisPoint2=(Width/2, t_2))
s.EllipseByCenterPerimeter(center=(Width/2, Height/2), axisPoint1=(Width, Height/2),
   axisPoint2=(Width/2-t_1/2, Height/2+t_2))
```

```
s.autoDimension(objectList=(g[2], ))
#: 2 dimensions added

s.autoDimension(objectList=(g[4], ))
#: 2 dimensions added

s=mdb.models['Model-1'].sketches['__profile__']
s.Parameter(name='Bigdim', path='dimensions[0]', expression=str(ovalbig))
s.Parameter(name='dimensions_2', path='dimensions[2]', expression='Bigdim+'+str(t_3),
   previousParameter='Bigdim')
s.Parameter(name='Littledim', path='dimensions[1]', expression=str(ovalsmall),
   previousParameter='dimensions_2')
s.Parameter(name='dimensions_3', path='dimensions[3]',
   expression='Littledim+'+str(t_3), previousParameter='Littledim')

p = mdb.models['Model-1'].Part(name='Part-2', dimensionality=TWO_D_PLANAR,
   type=DEFORMABLE_BODY)
p = mdb.models['Model-1'].parts['Part-2']
p.BaseShell(sketch=s)
s.unsetPrimaryObject()
p = mdb.models['Model-1'].parts['Part-2']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
del mdb.models['Model-1'].sketches['__profile__']

session.viewports['Viewport: 1'].partDisplay.setValues(sectionAssignments=ON,
   engineeringFeatures=ON)
session.viewports['Viewport: 1'].partDisplay.geometryOptions.setValues(
   referenceRepresentation=OFF)
mdb.models['Model-1'].Material(name='Material-1')
mdb.models['Model-1'].materials['Material-1'].Density(table=((7.75e-06, ), ))
mdb.models['Model-1'].materials['Material-1'].Elastic(table=((210000.0, 0.3),
   ))
mdb.models['Model-1'].HomogeneousSolidSection(name='Section-1',
   material='Material-1', thickness=thick)
p = mdb.models['Model-1'].parts['Part-2']
f = p.faces
faces = f.getSequenceFromMask(mask=('[#1 ]', ), )
region = p.Set(faces=faces, name='Set-1')
p = mdb.models['Model-1'].parts['Part-2']
p.SectionAssignment(region=region, sectionName='Section-1', offset=0.0,
   offsetType=MIDDLE_SURFACE, offsetField='',
   thicknessAssignment=FROM_SECTION)
p = mdb.models['Model-1'].parts['Part-1']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
```

```
mdb.models['Model-1'].HomogeneousSolidSection(name='Section-2', 
    material='Material-1', thickness=thick)
mdb.models['Model-1'].HomogeneousSolidSection(name='Section-3', 
    material='Material-1', thickness=thick)
p = mdb.models['Model-1'].parts['Part-1']
f = p.faces
faces = f.getSequenceFromMask(mask=('[#1 ]', ), )
region = p.Set(faces=faces, name='Set-1')
p = mdb.models['Model-1'].parts['Part-1']
p.SectionAssignment(region=region, sectionName='Section-2', offset=0.0, 
    offsetType=MIDDLE_SURFACE, offsetField='', 
    thicknessAssignment=FROM_SECTION)
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(
    optimizationTasks=OFF, geometricRestrictions=OFF, stopConditions=OFF)
a = mdb.models['Model-1'].rootAssembly
a.DatumCsysByDefault(CARTESIAN)
p = mdb.models['Model-1'].parts['Part-1']
a.Instance(name='Part-1-1', part=p, dependent=OFF)
a = mdb.models['Model-1'].rootAssembly
p = mdb.models['Model-1'].parts['Part-2']
a.Instance(name='Part-2-1', part=p, dependent=OFF)

a = mdb.models['Model-1'].rootAssembly


a.InstanceFromBooleanMerge(name='Part-3', instances=(a.instances['Part-1-1'], 
    a.instances['Part-2-1'], ), originalInstances=SUPPRESS, domain=GEOMETRY)
p = mdb.models['Model-1'].parts['Part-1']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
p = mdb.models['Model-1'].parts['Part-3']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=a)


a = mdb.models['Model-1'].rootAssembly
a.LinearInstancePattern(instanceList=('Part-3-1', ), direction1=(1.0, 0.0, 
    0.0), direction2=(0.0, 1.0, 0.0), number1=1, number2=2, spacing1=10.0, 
    spacing2=Height-t_2)

a = mdb.models['Model-1'].rootAssembly
a.translate(instanceList=('Part-3-1-lin-1-2', ), vector=((Width-t_1)/2, 0.0, 0.0))
```

```
#: The instance Part-3-1-lin-1-2 was translated by 4.75, 0., 0. with respect to the assembly
coordinate system
a = mdb.models['Model-1'].rootAssembly
a.InstanceFromBooleanMerge(name='Part-4', instances=(a.instances['Part-3-1'],
    a.instances['Part-3-1-lin-1-2'], ), originalInstances=SUPPRESS,
    domain=GEOMETRY)


a = mdb.models['Model-1'].rootAssembly
a.LinearInstancePattern(instanceList=('Part-4-1', ), direction1=(1.0, 0.0,
    0.0), direction2=(0.0, 1.0, 0.0), number1=xdir, number2=ydir, spacing1=Width-t_1,
    spacing2=2*(Height-t_2))

final = list()

for i in range(xdir):
 for j in range(ydir):
  if (j == 0 and i == 0):
   continue
  newline = "mdb.models['Model-1'].rootAssembly.instances['Part-4-1-lin-%d-%d'],"
%(i+1,j+1)
  final.append(newline)

#print final
finalline = "".join(final)
#print finalline

lastline = "mdb.models['Model-
1'].rootAssembly.InstanceFromBooleanMerge(domain=GEOMETRY,instances=(mdb.m
odels['Model-1'].rootAssembly.instances['Part-4-1'],"+ finalline +"),name='Part-5',
originalInstances=SUPPRESS)"
exec(lastline)


session.viewports['Viewport: 1'].assemblyDisplay.setValues(
    adaptiveMeshConstraints=ON)
mdb.models['Model-1'].StaticStep(name='Step-1', previous='Initial',
    initialInc=0.0625, nlgeom=ON)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(step='Step-1')
session.viewports['Viewport: 1'].assemblyDisplay.setValues(loads=ON, bcs=ON,
    predefinedFields=ON, connectors=ON, adaptiveMeshConstraints=OFF)

#..........................................................................................#
#### Bottom face set creation ####
```

```
bottom = list()

for i in range(xdir):
 big_lines = "((Width/3+%d*(Width-t_1), 0, 0.0),),"%(i)
 bottom.append(big_lines)

for i in range(xdir-1):
 small_lines = "((t_1/3+%d*(Width-t_1), 0, 0.0),),"%(i+1)
 bottom.append(small_lines)


# print bottom
bottomline="".join(bottom)

#print bottomline
bottomfaceset = "mdb.models['Model-1'].parts['Part-5'].Set(edges=mdb.models['Model-
1'].parts['Part-5'].edges.findAt(" + bottomline + " ), name='Bottom face')"
exec(bottomfaceset)

#.........................................................................................#

#### Top surface set creation ####

top = list()

for i in range(xdir):
 big_lines_top = "(((Width-t_1)/2+Width/3+%d*(Width-t_1), Height*ydir*2-(2*ydir-
1)*t_2, 0.0),),"%(i)
 top.append(big_lines_top)

for i in range(xdir-1):
 small_lines_top = "(((Width-t_1)/2+t_1/3+%d*(Width-t_1), Height*ydir*2-(2*ydir-
1)*t_2, 0.0),),"%(i+1)
 top.append(small_lines_top)

topline = "".join(top)
#print topline

topfaceset = "mdb.models['Model-1'].parts['Part-5'].Surface(name='Top surface',
side1Edges= mdb.models['Model-1'].parts['Part-5'].edges.findAt(" + topline + "))"
exec(topfaceset)

#.........................................................................................#
```

```
#### Left face set creation ####

left = list()

for i in range(ydir):
 partline_left1 = "((0.0, Height/3 + %d*2*(Height-t_2), 0.0),),"%(i)
 partline_left2 = "(((Width-t_1)/2,Height-t_2+Height/3 + %d*2*(Height-t_2), 0.0),),"%(i)
# left.append(partline_left1)
 left.append(partline_left2)

#Print left

leftline = "".join(left)
#print leftline
leftfaceset = "mdb.models['Model-1'].parts['Part-5'].Set(edges=mdb.models['Model-
1'].parts['Part-5'].edges.findAt(" + leftline + " ), name='Left face') "
exec(leftfaceset)

#.................................................................................................#

#### Right face set creation ####

right = list()

for i in range(ydir):
 partline_right1 = "((xdir*Width-(xdir-1)*t_1, Height/3 + %d*2*(Height-t_2), 0.0),),"%(i)
 partline_right2 = "(((Width-t_1)/2 + xdir*Width -(xdir-1)*t_1,Height-t_2+Height/3 +
%d*2*(Height-t_2), 0.0),),"%(i)
# right.append(partline_right1)
 right.append(partline_right2)

#Print right

rightline = "".join(right)

#print rightline

rightfaceset = "mdb.models['Model-1'].parts['Part-5'].Set(edges=mdb.models['Model-
1'].parts['Part-5'].edges.findAt(" + rightline + " ), name='Right face') "
exec(rightfaceset)

#.................................................................................................#

a1 = mdb.models['Model-1'].rootAssembly
```

```
region = a1.instances['Part-5-1'].surfaces['Top surface']
mdb.models['Model-1'].Pressure(name='Load-1', createStepName='Step-1',
   region=region, distributionType=UNIFORM, field=", magnitude=pressure1,
   amplitude=UNSET)

a = mdb.models['Model-1'].rootAssembly
region = a.instances['Part-5-1'].sets['Bottom face']
mdb.models['Model-1'].DisplacementBC(name='BC-1', createStepName='Step-1',
   region=region, u1=0.0, u2=0.0, ur3=UNSET, amplitude=UNSET, fixed=OFF,
   distributionType=UNIFORM, fieldName=", localCsys=None)
session.viewports['Viewport: 1'].view.setValues(nearPlane=16.6264,
   farPlane=28.0949, width=48.3506, height=21.8762, viewOffsetX=17.4052,
   viewOffsetY=6.52403)

a = mdb.models['Model-1'].rootAssembly
region = a.instances['Part-5-1'].sets['Left face']
mdb.models['Model-1'].DisplacementBC(name='BC-2', createStepName='Step-1',
   region=region, u1=0.0, u2=UNSET, ur3=UNSET, amplitude=UNSET, fixed=OFF,
   distributionType=UNIFORM, fieldName=", localCsys=None)

a = mdb.models['Model-1'].rootAssembly
region = a.instances['Part-5-1'].sets['Right face']
mdb.models['Model-1'].DisplacementBC(name='BC-3', createStepName='Step-1',
   region=region, u1=0.0, u2=UNSET, ur3=UNSET, amplitude=UNSET, fixed=OFF,
   distributionType=UNIFORM, fieldName=", localCsys=None)

session.viewports['Viewport: 1'].assemblyDisplay.setValues(mesh=ON, loads=OFF,
   bcs=OFF, predefinedFields=OFF, connectors=OFF)
a4 = mdb.models['Model-1'].rootAssembly
e1 = a4.instances['Part-5-1'].edges
edges1 = e1.getSequenceFromMask(mask=('[#0 #2 ]', ), )
a4.Set(edges=edges1, name='Set-3')
regionDef=mdb.models['Model-1'].rootAssembly.sets['Set-3']


mdb.models['Model-1'].FieldOutputRequest(name='F-Output-2',
   createStepName='Step-1', variables=('UT', ), frequency=LAST_INCREMENT,
   region=regionDef, sectionPoints=DEFAULT, rebar=EXCLUDE)
session.viewports['Viewport: 1'].assemblyDisplay.meshOptions.setValues(
   meshTechnique=ON)
p = mdb.models['Model-1'].parts['Part-3']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
session.viewports['Viewport: 1'].partDisplay.setValues(sectionAssignments=OFF,
   engineeringFeatures=OFF, mesh=ON)
```

```
session.viewports['Viewport: 1'].partDisplay.meshOptions.setValues(
   meshTechnique=ON)
p = mdb.models['Model-1'].parts['Part-5']
session.viewports['Viewport: 1'].setValues(displayedObject=p)

# ------
# ------ Mesh ------
# ------
p = mdb.models['Model-1'].parts['Part-5']
p.seedPart(size=0.1, deviationFactor=0.1, minSizeFactor=0.1)
session.viewports['Viewport: 1'].view.setValues(nearPlane=320.662,
   farPlane=324.741, width=19.1738, height=9.20882, viewOffsetX=-13.1888,
   viewOffsetY=-7.51207)
p = mdb.models['Model-1'].parts['Part-5']
p.generateMesh()
session.viewports['Viewport: 1'].view.setValues(nearPlane=302.284,
   farPlane=343.119, width=191.322, height=91.8882, viewOffsetX=13.1103,
   viewOffsetY=25.1928)
a = mdb.models['Model-1'].rootAssembly
a.regenerate()

# ----- Job
mdb.Job(name=jobName, model='Model-1', description=", type=ANALYSIS,
   atTime=None, waitMinutes=0, waitHours=0, queue=None, memory=90,
   memoryUnits=PERCENTAGE, getMemoryFromAnalysis=True,
   explicitPrecision=SINGLE, nodalOutputPrecision=SINGLE, echoPrint=OFF,
   modelPrint=OFF, contactPrint=OFF, historyPrint=OFF, userSubroutine=",
   scratch=", resultsFormat=ODB, multiprocessingMode=DEFAULT, numCpus=1,
   numGPUs=0)
mdb.jobs[jobName].submit(consistencyChecking=OFF)
#
# ----- Wait for job to complete
#
mdb.jobs[jobName].waitForCompletion()
#
# ----- Generate Report
#
o3 = session.openOdb(name=jobName+'.odb')
lastFrame=o3.steps['Step-1'].frames[-1]
session.viewports['Viewport: 1'].setValues(displayedObject=o3)
odb = session.odbs[jobName+'.odb']
session.fieldReportOptions.setValues(printXYData=OFF, printTotal=OFF)
session.writeFieldReport(fileName=rptName, append=OFF,
   sortItem='Node Label', odb=odb, step=0, frame=lastFrame, outputPosition=NODAL,
   variable=(('UT', NODAL, ((COMPONENT, 'UT2'), )), ))

 # ---
 # ---
 # --- Second Load Step ---
 # ---
 # ---
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(mesh=OFF, loads=ON,
    bcs=ON, predefinedFields=ON, connectors=ON)
mdb.models['Model-1'].loads['Load-1'].setValues(magnitude=pressure2)
mdb.jobs[jobName].submit(consistencyChecking=OFF)
 #
 # ----- Wait for job to complete
 #
mdb.jobs[jobName].waitForCompletion()
 #
 # ----- Generate Report
 #
o3 = session.openOdb(name=jobName+'.odb')
lastFrame=o3.steps['Step-1'].frames[-1]
session.viewports['Viewport: 1'].setValues(displayedObject=o3)
odb = session.odbs[jobName+'.odb']
session.fieldReportOptions.setValues(printXYData=OFF, printTotal=OFF)
session.writeFieldReport(fileName=rptName, append=ON,
   sortItem='Node Label', odb=odb, step=0, frame=lastFrame, outputPosition=NODAL,
   variable=(('UT', NODAL, ((COMPONENT, 'UT2'), )), ))

 # ---
 # ---
 # --- Third Load Step ---
 # ---
 # ---
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(mesh=OFF, loads=ON,
    bcs=ON, predefinedFields=ON, connectors=ON)
mdb.models['Model-1'].loads['Load-1'].setValues(magnitude=pressure3)
mdb.jobs[jobName].submit(consistencyChecking=OFF)
 #
 # ----- Wait for job to complete
 #
mdb.jobs[jobName].waitForCompletion()
```

```
 #
 # ----- Generate Report
 #
o3 = session.openOdb(name=jobName+'.odb')
lastFrame=o3.steps['Step-1'].frames[-1]
session.viewports['Viewport: 1'].setValues(displayedObject=o3)
odb = session.odbs[jobName+'.odb']
session.fieldReportOptions.setValues(printXYData=OFF, printTotal=OFF)
session.writeFieldReport(fileName=rptName, append=ON,
    sortItem='Node Label', odb=odb, step=0, frame=lastFrame, outputPosition=NODAL,
    variable=(('UT', NODAL, ((COMPONENT, 'UT2'), )), ))

 # ---
 # ---
 # --- Fourth Load Step ---
 # ---
 # ---
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport: 1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(mesh=OFF, loads=ON,
     bcs=ON, predefinedFields=ON, connectors=ON)
mdb.models['Model-1'].loads['Load-1'].setValues(magnitude=pressure4)
mdb.jobs[jobName].submit(consistencyChecking=OFF)
 #
 # ----- Wait for job to complete
 #
mdb.jobs[jobName].waitForCompletion()
 #s
 #h ----- Generate Report
 #i
 #t
o3 = session.openOdb(name=jobName+'.odb')
lastFrame=o3.steps['Step-1'].frames[-1]
session.viewports['Viewport: 1'].setValues(displayedObject=o3)
odb = session.odbs[jobName+'.odb']
session.fieldReportOptions.setValues(printXYData=OFF, printTotal=OFF)
session.writeFieldReport(fileName=rptName, append=ON,
    sortItem='Node Label', odb=odb, step=0, frame=lastFrame, outputPosition=NODAL,
    variable=(('UT', NODAL, ((COMPONENT, 'UT2'), )), ))
```