Computational Design of 2D-Mechanical Metamaterials

Kiara L. McMillan

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

Pinar Acar, Chair

Gary D. Seidel

Robert L. West

Giti A. Khodaparast

May 5, 2022

Blacksburg, Virginia

Keywords: Mechanical metamaterials, computational design tools, data-driven modeling, uncertainty, inverse design

Computational Design of 2D-Mechanical Metamaterials

Kiara L. McMillan

ABSTRACT

Mechanical metamaterials are novel materials that display unique properties from their underlying microstructure topology rather than the constituent material they are made from. Their effective properties displayed at macroscale depend on the design of their microstructural topology. In this work, two classes of mechanical metamaterials are studied within the 2D-space. The first class is made of trusses, referred to as truss-based mechanical metamaterials. These materials are studied through their application to a beam component, where finite element analysis is performed to determine how truss-based microstructures affect the displacement behavior of the beam. This analysis is further subsidized with the development of a graphical user interface, where users can design a beam made of truss-based microstructures to see how their design affects the beam's behavior. The second class of mechanical metamaterial investigated is made of self-assembled structures, called spinodoids. Their smooth topology makes them less prone to high stress concentrations present in truss-based mechanical metamaterials. A large database of spinodoids is generated in this study. Through data-driven modeling the geometry of the spinodoids is coupled with their Young's modulus value to approach inverse design under uncertainty. To see mechanical metamaterials applied to industry they need to be better understood and thoroughly characterized. Furthermore, more tools that specifically help push the ease in the design of these metamaterials are needed. This work aims to improve the understanding of mechanical metamaterials and develop efficient computational design strategies catered solely for them.

Computational Design of 2D-Mechanical Metamaterials

Kiara L. McMillan

GENERAL AUDIENCE ABSTRACT

Mechanical metamaterials are hierarchical materials involving periodically or aperiodically repeating unit cell arrangements in the microscale. The design of the unit cells allows these materials to display unique properties that are not usually found in traditionally manufactured materials. This will enable their use in a multitude of potential engineering applications. The presented study seeks to explore two classes of mechanical metamaterials within the 2D-space, including truss-based architectures and spinodoids. Truss-based mechanical metamaterials are made of trusses arranged in a lattice-like framework, where spinodoids are unit cells that contain smooth structures resulting from mimicking the two phases that coexist in a phase separation process called spinodal decomposition. In this research, computational design strategies are applied to efficiently model and further understand these sub-classes of mechanical metamaterials.

Dedicated to my parents who always provided encouraging words of wisdom and support throughout my education. I am forever grateful for you both and forever striving to be as thoughtful and kind as you both.

Acknowledgments

Thank you to my advisor, Dr. Pinar Acar, whose considerate and selfless guidance helped me throughout my research. Thank you to Dr. Robert West, Dr. Gary Seidel, and Dr. Giti Khodaparast for serving on my graduate committee. I would also like to thank Dr. Doğacan Su Öztürk, a research assistant professor at the Geophysical Institute of the University of Alaska Fairbanks, who helped me with applying the technique of principal component analysis (PCA). Thank you to PhD candidates in the Advanced STRuctures and Optimization (ASTRO) Lab, Arulmurugan Senthilnathan and Sheng Liu. Thank you to Arulmurugan for helping me understand the concept of shape moment invariants and Sheng for guiding me in how to utilize Abaqus. Finally, I would like to thank the New Horizon Graduate Scholars Fellowship Program at Virginia Tech for providing financial support.

Contents

List of Figures

List of Tables

List of Abbreviations

$[\mathbf{K}]$, $[\mathbf{K}_*]$, $[\mathbf{K}_{**}]$	Correlation matrices that make up $[\bar{\mathbf{K}}]$
$E[\mathbf{Y}_*]$	Predicted Young's modulus
$E\,[Y_*]$	Target Young's modulus
$[\bar{\mathbf{K}}]$	Multi-fidelity covariance matrix
$[\mathbf{X}]$, $[\mathbf{Y}]$	Vectors of training and testing data
β	Constant wave number
$\gamma_i \sim \mathcal{U}([0, 2\pi])$	Phase angle of ith wave vector
$[D]$, $[K^*]$	Elastic material properties and stiffness of the microstructure
$[F]$	Force vector
$[K]$	Global stiffness matrix
$[u]$	Vector of nodal displacements
$[\Sigma_*]$	Covariance of predicted values
σ, α, l	Hyper parameters of kernel function K
$\sigma[E]^2$	Young's modulus unit variance
σ^2	Variance
θ_1, θ_2, θ_3	Parameters that control spinodoid topology

$C_{1,1}$, $C_{1,2}$, $C_{1,3}$, $C_{3,3}$, and $C_{4,4}$	Stiffness properties
K	Kernel function
L	Length value
N	Number of waves
n	Value representing the number of a variable
$\mathbf{n_i} \sim \mathcal{U}(S^2)$	Direction of ith wave vector
E_{0}	Young's modulus solid material
E_e	Young's modulus of element
E_{min}	Young's modulus of the void material
p	Penalization factor
ρ_e	Density value of pixel (0,1)
φ	Phase field
FEA	Finite Element Analysis
GPR	Gaussian Process Regression
GRFs	Gaussian Random Fields
GUI	Graphical User Interface
PCA	Principal Component Analysis

Chapter 1

Introduction

1.1 Motivation and Objectives

Mechanical metamaterials are novel materials that provide a multitude of new and fascinating applications to many fields of technology. They can elicit unique properties such as high strength-to-weight ratios, negative Poisson's ratios, high energy absorption and multifunctional capabilities [1], [2]. Their effective properties depend primarily on their inner architecture rather than the base materials they are formed from [1]. Given their unique properties, they have a wide range of potential applications. However, in order to see them utilized in industry, these materials need to be further studied. Like traditional materials, in order to see them applied within industry, they need to be thoroughly understood by fully characterizing their properties and behavior [3]. Furthermore, this understanding can also be driven by the establishment of new efficient design and modeling techniques catered specifically towards mechanical metamaterials [3].

Within this research two subsets of mechanical materials are studied within the 2D-space. The first type is based on truss-like structures. These truss-based mechanical metamaterials are explored through the construction of a mechanical beam component, where the effect of applied external forces was determined through finite element analysis. A graphical user interface is created and implemented such that a user could easily understand how differing geometries of truss-based unit cells can affect the resultant properties of the beam component. Mechanical metamaterials with spinodal topologies, also known as spinodoids, are the second type of mechanical metamaterials explored within this research. These materials are investigated by generating a large database of 2D-spinodoid unit cells which are represented through monochrome images. By using surrogate modeling techniques a relationship between their calculated mechanical property and parameters controlling the geometry of the unit cell is found. Through this established relationship, the design of spinodoids is further investigated.

Through the study of these mechanical metamaterials the following contributions are made to this particular field of material research:

- Developing a computational methodology and a graphical user interface to explore the design of truss-based mechanical metamaterials.
- Considering the effects of uncertainty on spinodoids with data-driven modeling.
- Investigating the inverse design of spinodoids under uncertainty using data-driven modeling.
- Forwarding the progress toward the realization of mechanical metamaterials within industry by exploring efficient design and modeling techniques that have helped further the understanding and characterization of these materials.

1.2 Mechanical Metamaterials

Metamaterials are a relatively new class of artificially developed materials that exhibit unique properties and multifunctional capabilities. The properties they demonstrate are not frequently found in nature, furthermore, these properties can even be tailored through the design of the material itself [1]. These materials were first recognized as “metamaterials” in 1999 by the Defense Advanced Research Projects Agency Symposium on Meta-Materials [4]. The prefix “meta” is a Greek loanword meaning “beyond”, this prefix was chosen due to these materials’ abilities to demonstrate very unique properties compared to the average, naturally derived materials [4]. These unique properties most commonly fall under the field classifications of mechanical, acoustic, electromagnetic, optical, and thermal [3]. The effective properties of mechanical metamaterials are dependent on the underlying microstructural topology rather than the constituent material it is constructed from. These microstructures usually consist of repeated geometries called unit cells. Altering the unit cell geometry and varying how the unit cell is repeated and patterned control the material properties on the macroscopic scale [3], [4]. Patterning of the unit cell can be completed by distributing the unit cells periodically or aperiodically, while also altering the specific gradients of cell thicknesses throughout the entire material [3]. The shape, size, and the overall topology of the unit cell architectures can be designed in a manner so that the material displays specific mechanical properties as well [3]. There are several popular categories of mechanical metamaterials that are more notable which include truss-based, chiral, and origami, visualized in Figure 1.1 [5]. The images used in Figure 1.1 are derived from literature, examples of truss-based mechanical metamaterials courtesy of [1], [6], origami-based examples courtesy of [7], [8], chiral-based examples courtesy of [9], and kirigami-based examples courtesy of [10].

Mechanical metamaterials have many theorized applications, including to the fields of biomedical, robotics, aerospace, and civil [1]. Some of these theorized applications include foldable solar arrays based on origami structures, custom shoe soles inspired from truss-like structures, and medical implants formed by mimicking honeycomb microstructures [1], [11], [12]. Theorized applications of mechanical metamaterials are summarized in Table 1.1.

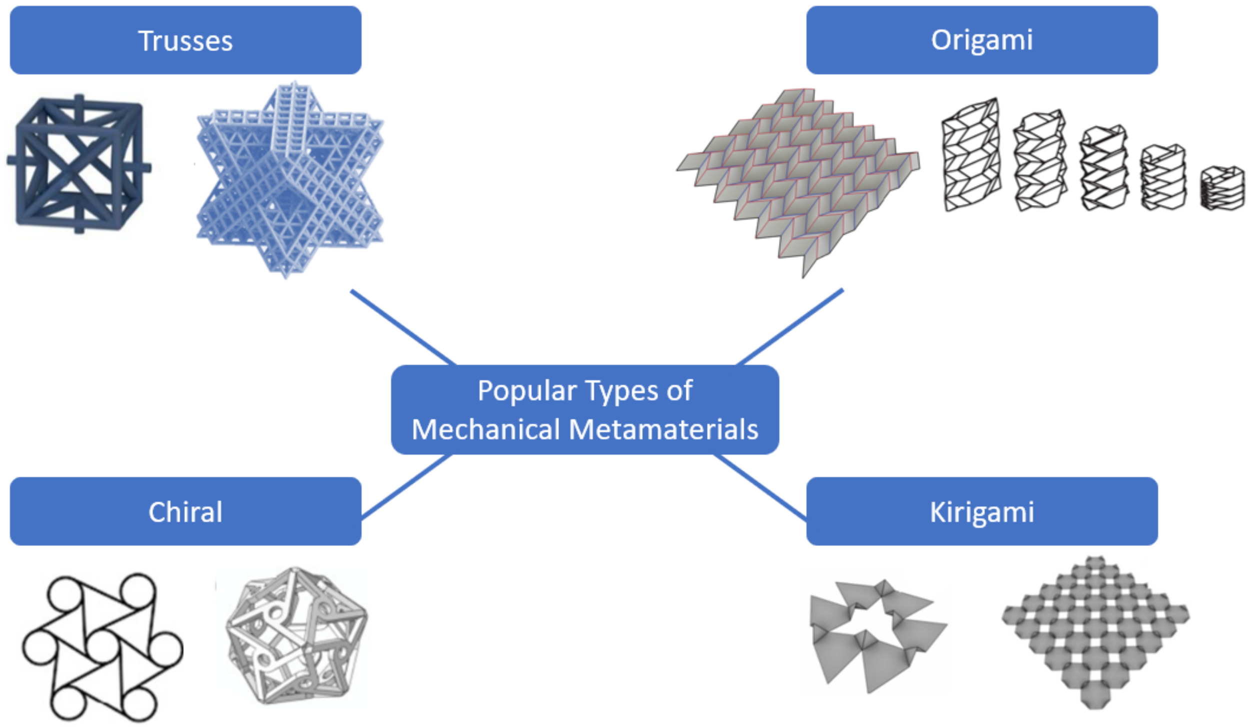


Figure 1.1: Popular types of mechanical metamaterials. examples of truss-based mechanical metamaterials. The images utilized in this figure are derived from literature, examples of truss-based mechanical metamaterials courtesy of [1], [6], origami-based examples courtesy of [7], [8], chiral-based examples courtesy of [9], and kirigami-based examples courtesy of [10].

Much of the research within this field is driven by the multitude of potential properties these materials may elicit [3]. Although there are some instances of mechanical metamaterials utilized in industry, given their potential applications and capabilities, they still are not widely seen within the commercial industry. This may be due to the lack of research in proving their reliability and resiliency compared to traditionally manufactured materials [3]. Investigating these materials' strength, toughness, and fatigue behavior is crucial in determining how these materials can be properly applied in industry [3]. Understanding the performance of mechanical metamaterials requires further experimental testing, as well as the development of high-fidelity computational models and design tools. As with any

Table 1.1: Examples of the theorized applications of mechanical metamaterials

Potential Application Fields		
Aerospace	Biomedical	Robotics
• Foldable solar arrays [13]	• Bone implants [5], [14]	• Soft robotics [15]
• Materials within airplanes or satellites [16]	• Orthopedic support (shoe soles) [12]	• Mechanical actuators [17], [18]
• Morphing airfoils [9]	• Medicine delivery [19]	

material, mechanical metamaterials may not perform to their original design and predicted properties [3]. Therefore, determining the reasoning as to why mechanical metamaterials may not perform as predicted or determining the uncertainty associated in their fabrication is also necessary for characterizing them. The stray in predicted behavior can be quantified in terms of uncertainty due to manufacturing or human interference in their eventual assembly [20]. Part of this research focuses on quantifying the uncertainty and its effects on the design of mechanical metamaterials by specifically focusing on spinodoids.

1.2.1 Truss-Based Mechanical Metamaterials

The first studied class of mechanical metamaterials in this research is the truss-based mechanical metamaterials. These materials are made of trusses purposefully repeated or arranged in a lattice-like framework [21]. Truss-based mechanical metamaterials can remain relatively simple, but they can also be arranged in configurations that increase their complexity. Such arrangements include octet-truss or cuboctahedron [22]. They can achieve high strength and stiffness while remaining lightweight and they also have been found to demonstrate high-energy absorption capabilities [23], [24].

Given their ability to demonstrate high strength and stiffness while remaining lightweight, they have the potential to be utilized within aerospace applications to aid in the ongoing goal of weight reduction [25], [26]. Other applications that have been explored include heat transferring and vibration control [24], [27], [28]. This idea of vibration control has also been further expanded to include piezoelectric properties, which would allow the production of electricity through vibrational harvesting [29]. Wang *et. al* have studied mechanical metamaterials with beam-like structures with theorized applications in the field of construction or materials with collapsible and flexible features [30].

1.2.2 Mechanical Metamaterials with Spinodal Topologies

Mechanical metamaterials with spinodal topologies form by following the process of spinodal decomposition. Kumar and coworkers coined these materials as *spinodoids*, by utilizing a simplified formulation of this decomposition process [5], [31]. They are a particular subset of mechanical metamaterials that are formed through a phase separation process called spinodal decomposition. Compared to truss-based or origami-based mechanical metamaterials, their design allows non-periodically repeating unit cell arrangements consisting of smooth and bi-continuous structures, as shown in Figure 1.2. The phase separation process that they are formed through occurs when two liquids or solids separate from each other, from one thermodynamic phase to two separate phases that exist in cohesion [32]. This process can be modeled analytically by the Cahn-Hilliard equation [33]. Yet, the numerical solution of the Cahn-Hilliard equation is computationally expensive due to the time-marching scheme. Therefore, Kumar and co-workers efficiently model the phase separation process through the application of Gaussian random fields (GRFs). The approximated spinodal topologies with the GRF model are defined as spinodoids [5], [31]. Unlike the more popularly known truss- or origami-based mechanical metamaterials, spinodoids are less prone to high stress concentrations that are more prevalent in the sharp corners existent in these metamaterials. Additionally, because of their self-assembled structure they are also less susceptible to symmetry-breaking defects which can cause unexpected behavior in more periodic geometries such as truss-based materials.

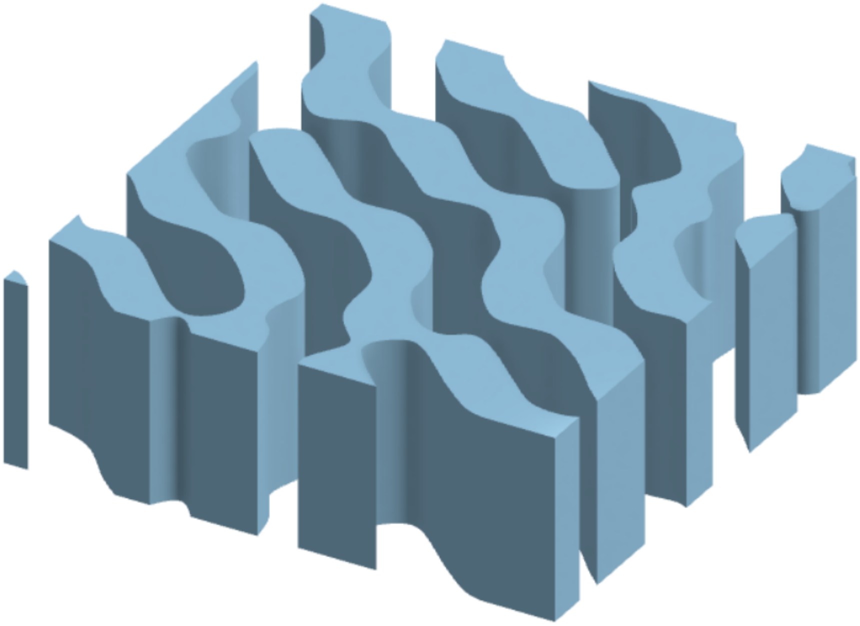


Figure 1.2: The geometry of spinodoids is made of smoother structures lacking sharp corners, this 3D-model was extruded from a generated 2D-spinodoid. Figure derived from [60].

Kumar and co-workers theorize spinodoids as replacements for bone implants as they were able to match target properties of bones through the tailored design of 3D-spinodoids [5]. Kochmann *et. al* investigated the ease of manufacturing spinodoids through another method aside from additive manufacturing using self-assembling polymers, allowing for easier scalability of spinodoids [34].

1.3 Design of Mechanical Metamaterials

Compared to traditionally manufactured, the microstructure topology of mechanical metamaterials can be designed to meet specific targets on mechanical performance. The ability to have such freedom in design results in vast design spaces for mechanical metamaterials [35]. In addition to this, the macro-scale properties are controlled by the microscale architecture involving the repeating patterns of unit cells [1]. This creates a hierarchy in the material, that can cause the property determination of the material to be difficult and computationally expensive, due to many complex microstructure topologies. Therefore methods to characterize mechanical metamaterials are usually based on homogenization and reduced-order modeling [36], [37]. With advancements in 3D-printing technology, the physical fabrication of mechanical metamaterials with complex geometries can also be achieved. Modeling difficulty further increases when these materials are represented in three dimensions, therefore there is a need to establish high-fidelity two-dimensional characterization methods to decrease this computational complexity [38], [39].

1.3.1 Inverse Design of Mechanical Metamaterials

As stated, the unit cells of mechanical metamaterials can be designed to elicit specific mechanical properties. This gives the opportunity to approach the design of these materials in the inverse manner. Meaning that the material can be tailored to the desired property, as opposed to a choosing a material to satisfy the desired property. With mechanical metamaterials providing such a large design space, efficient optimization techniques are needed to help approach the inverse design of these materials. Topology optimization is one method that has been applied to optimize different types of mechanical metamaterials to meet specific material properties [40], [41], [42]. Other methods of optimization that have been applied include stochastic optimization, linear-response optimization, and global optimization techniques such as differential evolution algorithm [35], [43], [44]. Another method used in approaching the inverse design of mechanical metamaterials include utilizing machine learning [45], [46], [47], [48]. In the inverse design approach of spinodoids by Kumar *et. al*, deep neural networks were applied to help relate the parameters controlling the spinodoid design and the mechanical properties [5]. Through a data-driven model, this research explores the inverse design of 2D-spinodoids with the consideration of fabrication-related uncertainty.

1.3.2 Design of Mechanical Metamaterials under Uncertainty

Although a lot of research in the field of mechanical metamaterials is focused on the discovery of new geometries that produce novel capabilities and properties, in order to see them utilized in real-world applications, they need to be studied to the same lengths as traditionally manufactured materials. Meaning that they need to be fully characterized in terms of how they behave under mechanical defects, repeated, cyclic loading, and other more detailed, extended mechanical tests [3]. One property of materials that affects the expected behavior is the effect of uncertainty on the design. This uncertainty stems from the fluctuations associated with manufacturing methods or any human interference in the development of materials [20]. The effects of uncertainty on mechanical metamaterials have been explored in recent years through robust optimization and with the consideration of hybrid uncertainties [49], [50]. In this study, the numerical methodology is extended to design metamaterials under the effects of the inherent uncertainty arising from processing and computational modeling. This is achieved by developing a data-driven framework using Gaussian process regression (GPR) to predict the expected behavior and variations of the mechanical properties as a function of the microstructural topology.

Chapter 2

Investigating Truss-based Mechanical Metamaterials

2.1 Research on Truss-based Mechanical Metamaterials

The main goal of the research on truss-based mechanical metamaterials is focused on determining the best way to model a mechanical beam component made of 2D-truss based microstructures. This mechanical component represents a smaller unit that would be repeated throughout a larger component to create a hierarchical structure. This larger component would then contain a hierarchy of structures resulting in a mechanical metamaterial. Although this research is based in the 2D-space, the modeling techniques established help better understand how this could be expanded to the 3D-space. With emerging additive manufacturing technology, the hierarchy that begins on the nano- to microscale can now be achieved through advanced 3D-printing techniques. This study aims to develop a modeling method to understand how external forces affect a 2D-mechanical beam made of truss-based microstructures. This is completed by developing a database of microstructures of differing truss-based geometry, constructing the beam component from these created microstructures, and modeling the effect of external forces on the beam's displacement behavior through finite element analysis (FEA). An initial goal of optimizing the microstructure topology of the beam eventually led to development of a graphical user interface (GUI). This GUI allows users to design their own beam component and understand the resultant deformation behavior subject to forces given by the user. Throughout this research an efficient modeling technique is developed to represent the connecting interfaces between the truss-based microstructures so that FEA can be efficiently performed on a mechanical component. This modeling technique allowed for the eventual creation of an easy-to-use GUI that could serve as a template for future design tools specific to the design and development of mechanical metamaterials. Some methods and techniques proposed in this chapter are published as a journal article on AIAA Journal: K. McMillan and P. Acar, “Database Development and Component Design with Two-Dimensional Trusslike Microstructures”, AIAA Journal, published 2021 [51].

2.2 Development of a Database Made of 2D Truss-Based Microstructures

The truss-based microstructures were created by first establishing a square border with equal side lengths. Within this square border, trusses were connected in several different arrangements that would produce a variety in structural layouts. These microstructures are represented by black and white images, where the black lines represent trusses and white is interpreted as void. See Figure 2.1 to view the 12 different structures composing the 2D-database. These 12 microstructures make-up the selection of structures that can construct the mechanical beam component to be modeled.

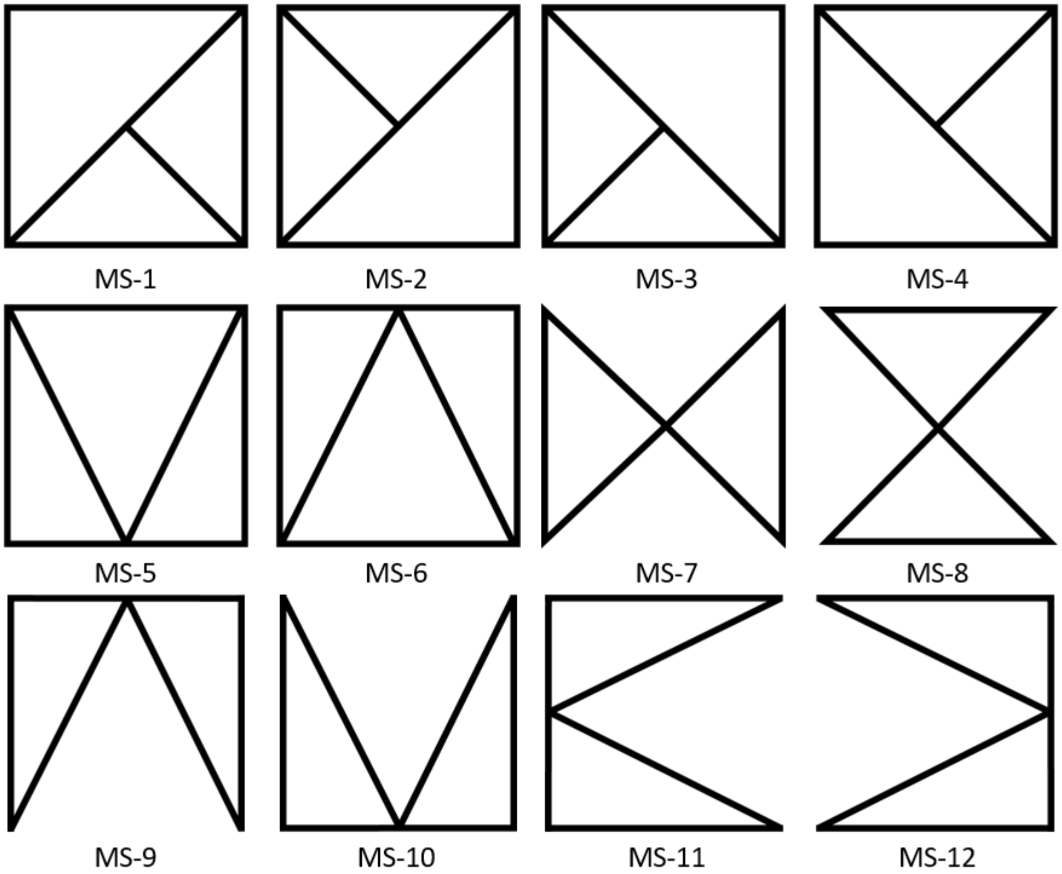


Figure 2.1: The database of microstructures contains 12 truss-based structures, each referenced as a number, for example the first microstructure would be referenced as "Microstructure 1" or "MS-1" [53].

2.3 Modeling the Beam Component Constructed of Microstructures

Within this research, an optimization problem is set to determine the optimum layout of the truss-based microstructures constructing the beam to minimize its horizontal displacement when subject to external tensile forces. The mechanical beam component considered in this optimization problem is modeled as a simple cantilever beam. The beam is considered to be split up into equal length, square sections, where each section is comprised of one of the twelve microstructures in the database. The external tensile force acts throughout the beam by pulling on the right corners of each microstructure section. The length and number of sections within the beam can be arbitrarily chosen, therefore the size of the microstructure contained within each section is determined as the entire beam's length divided by the total number of sections. A diagram of the described beam in the initial optimization problem is displayed in Figure 2.2.

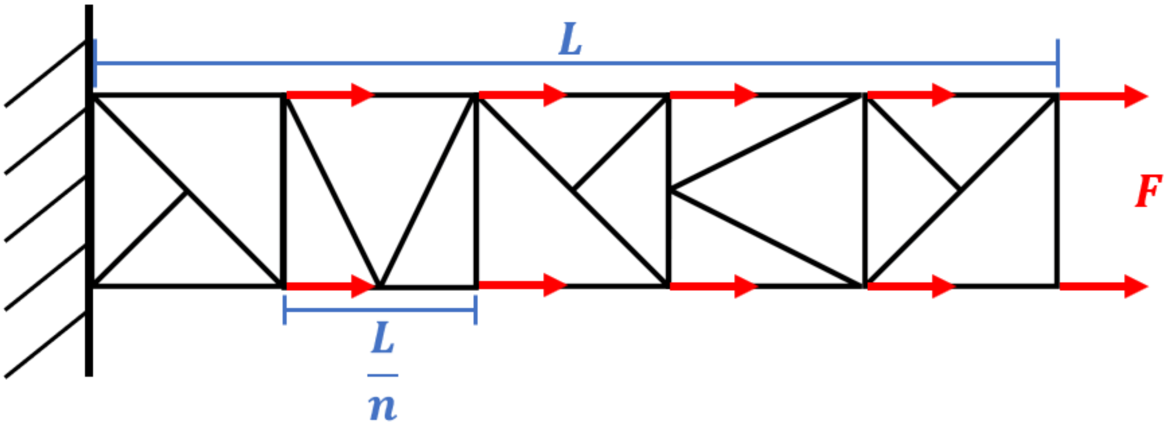


Figure 2.2: Representation of the cantilever beam made of five microstructures, with an external tensile force acting on the two outer corners of each microstructure section [51].

2.3.1 Finite Element Analysis of Beam Component

After establishing the database of microstructures and the representation of the mechanical beam component, the next task was determining how to model the component so that finite element analysis (FEA) could be applied. FEA would need to be completed to determine the beam's deformation behavior when subject to the external forces. The formulation used to perform FEA on the beam made of microstructures is described by,

$$[D]\,[K^*]\,[u] = [F]\,, \tag{2.1}$$

where $[D]$ and $[K^*]$ relate to the elastic material properties and stiffness of the microstructures (where the global stiffness matrix $[K]$ is given by $[K] = [D]\,[K^*]$), $[u]$ is the vector of the beam's nodal displacements, and $[F]$ is the applied force vector. To obtain the displacement of the beam, $[u]$ is solved for given the force, $[F]$, and material properties of the beam, $[D]$ and $[K^*]$.

The FEA code utilized is a numerically, efficient code developed through MATLAB, that is based on modeling the elements as trusses. The structure that is analyzed within this code is constructed through the connections of the nodes which define the elements within the structure. This takes into consideration the total number of elements, the amount of nodes contained in each of these elements, and the total number of nodes in the entire structure. The connections are established through the definition of one node connection to the next, which results in the formation of one element. For example, if one element contains the first and second nodes then this connection is defined as "(1,2)". These connections are defined until all elements of the structure have been described. Following the definition of the structure, the properties of the structure are given by matrices $[D]$ and $[K^*]$. Matrix $[D]$ is first defined by the stiffness properties given by $C_{1,1}$, $C_{1,2}$, $C_{1,3}$, $C_{3,3}$, and $C_{4,4}$. From the definition of these stiffness values the elastic moduli and Poisson's ratios of the material can be calculated. Which together form the needed material property matrix, $[D]$. The matrix, $[K^*]$, is defined by the length of each truss element, in which case of the beam are the bars making up each structure. The length of each element is equal to the total length of the beam divided by the total number of sections in the beam. Once the material properties are defined through matrices, $[D]$ and $[K^*]$, the local stiffness matrices can be established. These local stiffness matrices are formed by also taking into account the element connections and compiled to form the global stiffness matrix which represents the entire structure. This global stiffness matrix along with the defined force vector is then used in Equation 2.1 to solve for the displacements of the defined nodes in the analyzed component.

Since the design of the beam can be arbitrarily altered given by the database of 12 varying truss-based microstructures, the main part of the FEA code described above that considers the changing microstructure topology, is the definition of the element and nodal connections. Therefore, two different methods in modeling the individual microstructures for the application of the FEA formulation were executed. Each method was derived using MATLAB. The first method was applied to the initial optimization problem in determining the optimum microstructure layout to minimize the horizontal displacement of the beam subject to tensile force. The results from this completed optimization problem would command for the derivation of the second modeling method. This second method allowed for a more comprehensive approach by modeling the entire beam as an assembly of underlying microstructural members for the application of FEA. This second method of modeling the changing topology would eventually be utilized in the development of the graphical user interface (GUI).

Method 1: FEA Performed on Each Microstructure Section

The first method used to establish the element and node connections of the beam for FEA involved analyzing each separate section made of a microstructure. This involved creating individual MATLAB functions, each belonging to one of the twelve microstructures within the database. Depending on the microstructure assigned to the section of the beam, that function would be called on to perform a FEA on that particular microstructure. This function would then provide an output providing the summed displacement of all of the nodes contained within that microstructure. The resultant displacement of each section would then be stored and summed, once all of the sections were evaluated, to determine the total displacement of the beam. As stated previously in order to perform the FEA on the changing microstructures within the beam, the nodes and elements of each created microstructure would need to be assigned. These assignments were manually defined for each microstructure and set at the beginning of each of its respective functions. Several of these manual assignments are displayed as examples in Figure 2.3. As the FEA code is completed within each individual function, each function requires the material properties and force vector acting on the structure. Therefore, each function requires the input of the material properties, which are defined by matrices $[D]$ and $[K^*]$, followed by the input of the force vector, which indicates that the external force is acting on the right corners of each section. The material properties are assumed to be consistent throughout the entire beam so the property matrices are consistent throughout each function. Finally, another aspect to consider in the FEA is the differing boundary conditions of each section within the beam. In the initial optimization problem, the only differing boundary condition is the first section attached to the fixed end of the cantilever beam. This boundary condition fixes the displacement of all nodes connected to that fixed end, while all other nodes are free to displace in both the x- and y-directions. This boundary condition is another input within each function that is defined by a “0” or “1”. This binary condition indicates whether the microstructure is connected to the fixed end of the beam, indicated by “1”, or if it is free, indicated by “0”. Once all the functions were constructed for each microstructure within the database, the initial optimization problem was analyzed.

An optimization problem was conducted to determine the best layout of microstructures to form a beam component that minimizes the horizontal displacement. Therefore, the design goal was to determine the optimum placement of 2D-microstructures within each section of the beam that would minimize the displacement resulting from a constant external, horizontal force acting throughout the beam. A visual diagram of the defined optimization problem is shown in Figure 2.4.

In order to determine the optimum microstructure topology, a central code was developed that utilizes the 12 FEA functions defined in the first modeling method. As the beam

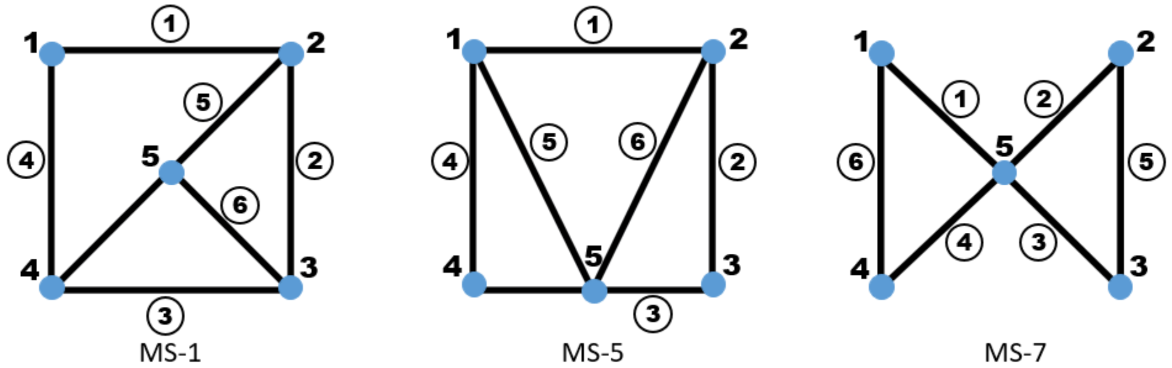


Figure 2.3: Node and element assignments manually defined for the first method in modeling for FEA. Microstructures 1, 5, and 7 are displayed. Node assignments are indicated by the blue circles, where element assignments are indicated by the numbers inscribed by circles [51].

properties can be arbitrarily chosen, the developed central code asks the user to input several important factors that define the construction of the beam. This information is then utilized in the FEA code within each function. The main code asks the user to input the material properties of the beam, defined by $C_{1,1}$, $C_{1,2}$, $C_{1,3}$, $C_{3,3}$, and $C_{4,4}$, the length of the entire beam, and the number of sections within the beam, once these properties are established matrices, $[D]$ and $[K^*]$, can be assembled. After the beam properties are established, the magnitude of the force acting on throughout the beam is also defined by the user. Once these values are entered, the central code finds the optimum topology of the beam by looping through each of the 12 microstructures within the database. This loop accesses each of the defined functions, developed in the first modeling method, which runs a FEA on each microstructure. The main code then stores the microstructure that produces the smallest horizontal displacement. Once a microstructure for each section of the beam is determined, the displacements are summed at the end to determine the final overall displacement of the beam.

The developed code for the optimization problem was tested by defining an arbitrary beam made of Ti-6Al-4V, which is a common Titanium-Aluminum alloy used in aerospace materi-

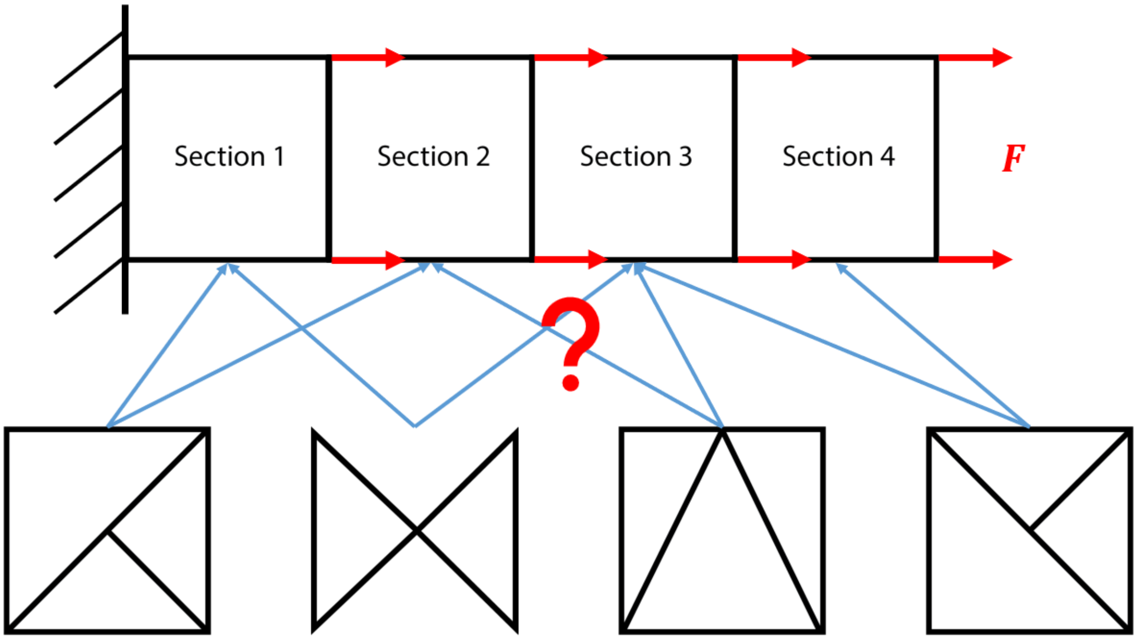
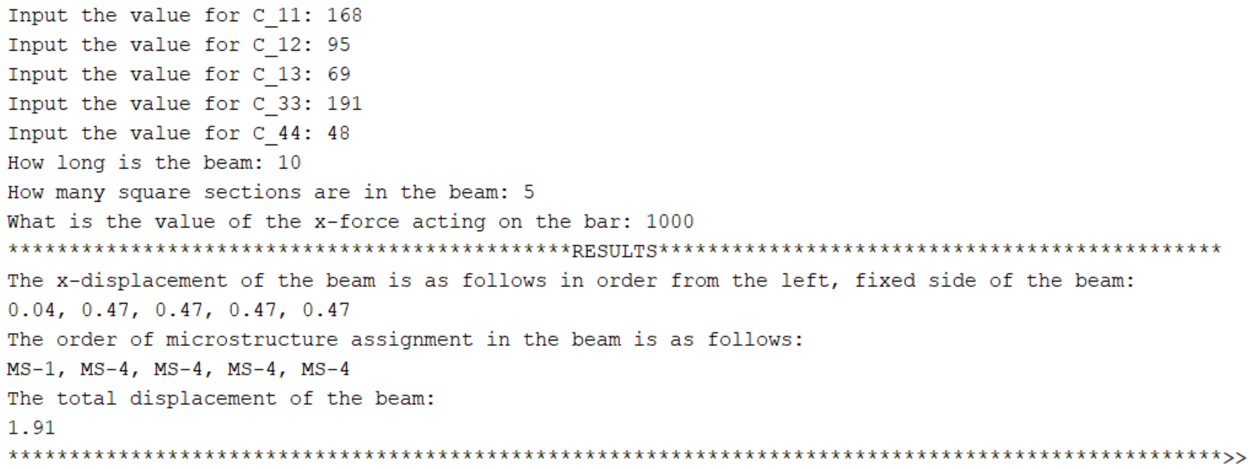


Figure 2.4: Diagram of the defined optimization problem, where the goal is to find the optimum layout of microstructures to minimize displacement due to an external horizontal force [53].

als. The properties defining this material given to the code are 168, 95, 69, 191, and 48 GPa for $C_{1,1}$, $C_{1,2}$, $C_{1,3}$, $C_{3,3}$, and $C_{4,4}$, respectively [52]. The length of the test beam was defined as 10 mm with a total of five sections and a force with a magnitude of 1000 N acting on the appropriate nodes was applied. Through the exhaustive search performed in the main code, this resulted in the beam having the layout of Microstructure 1 being attached to the fixed end of the beam, and all remaining sections set as Microstructure 4 [51]. This led to the sectional displacement of 0.04 mm for Microstructure 1 and sectional displacements of 0.47 mm for the remaining sections made of Microstructure 4. Resulting in a total horizontal displacement of 1.91 mm [53]. The output of the completed code and its results are displayed in Figure 2.5. This resultant layout was consistent with altering the customizable factors of the beam such as the length, number of sections, and magnitude of external force. With changing magnitude of the force, the resultant displacement behavior changed accordingly, meaning a force with higher magnitude led to the same displacement, but with higher magnitude [53]. To verify that the code would always result in the same layout of microstructures, each FEA function was tested individually with the same boundary conditions to determine if Microstructures 1 and 4 would always result in the smallest displacement. This analysis led to the realization that this first method used in modeling the beam for the application of FEA may not be taking into account the structural connections between each section of the beam. Meaning that although this method of modeling indicates the sensitivity of each section contained in the beam, it may not be the most accurate method in that it does not consider how the force is acting throughout the entire beam as a whole. More specifically, it does not provide information regarding how the displacement behavior of one section of the beam affects the next section. This led to the need of determining another method that would allow FEA to be applied to the beam as an entirety as opposed to a conglomerate of separate sections. The main part that would need to be altered in the first established modeling technique is how the element and node connections of the beam are defined. Instead of defining the node and element connections manually, a method that determines these connections automatically is needed. This led to the development of second modeling technique defined in the following section.

Method 2: FEA Performed on the Beam as a Whole

The analysis of running the initial optimization problem with the first method in modeling the beam indicated that this initial method is not taking into account the behavior of the interfaces between each section. Utilizing the first method of manually assigning the node and element connections would require a high amount of human input. For just one microstructure, the process of assigning the proper nodes and elements was extensive. Manually assigning the nodes and elements for a beam made of more than one section would

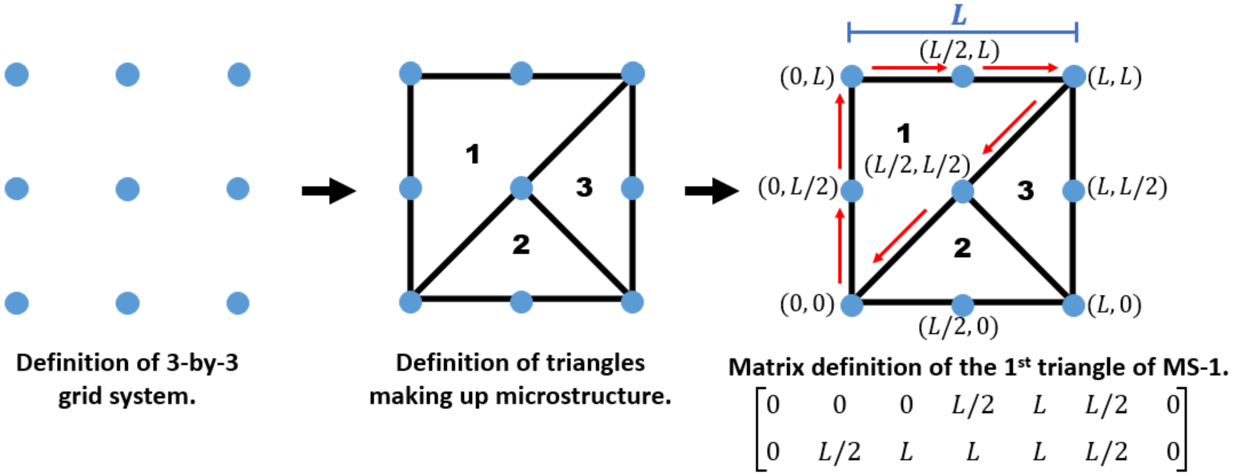


```
Input the value for C_11: 168
Input the value for C_12: 95
Input the value for C_13: 69
Input the value for C_33: 191
Input the value for C_44: 48
How long is the beam: 10
How many square sections are in the beam: 5
What is the value of the x-force acting on the bar: 1000
**********************************************RESULTS**********************************************
The x-displacement of the beam is as follows in order from the left, fixed side of the beam:
0.04, 0.47, 0.47, 0.47, 0.47
The order of microstructure assignment in the beam is as follows:
MS-1, MS-4, MS-4, MS-4, MS-4
The total displacement of the beam:
1.91
***************************************************************************************************>>
```

Figure 2.5: Input and result of the optimization code applied to the arbitrary beam made of Ti-6Al-4V [51].

be inefficient and most likely lead to human error. Therefore, the second method in modeling the beam for the FEA code, involved determining a technique that would allow for the automatic assignment of the node and element connections.

After observing the geometry of the 12 microstructures, each structure could be defined as a set of two or three triangles. Within MATLAB, using the plotting command, these triangles can be traced by following a path of predefined x- and y- coordinates. The vertices of each triangle can be considered as nodes within the entire microstructure. With the idea that these nodes can be defined by x- and y- coordinates, in conjunction with the plotting command in MATLAB, the microstructures can be "drawn" by following a path of coordinates defining each triangle in the microstructure. This was accomplished by defining a three-by-three grid system that each microstructure can be placed on. The triangles making up each of the 12 microstructures can be drawn using this grid system. The triangles within each microstructure are defined by a path dictated by defining the smallest coordinate first, considering the x-coordinate before the y-coordinate, this guideline is followed until the triangle is closed by repeating the first pair of coordinates on its path. These base coordinates make up each of the elements in the microstructure, which hold length values defined by the section length. The “pathway” of coordinates are the only parts of this method that require manual input and these coordinates would be stored in a database or reference “library”, which is in the form of a MATLAB function. This database is then called upon in later code to construct the entire beam. See Figure 2.6 that shows how this process of defining each microstructure through a list of coordinates was applied to Microstructure 1 (MS-1).


Definition of 3-by-3 grid system.
1
2
3
Definition of triangles making up microstructure.
L
(L/2, L)
(0, L)
(L, L)
1
(L/2, L/2)
(0, L/2)
3
(L, L/2)
2
(0, 0)
(L, 0)
(L/2, 0)
Matrix definition of the 1st triangle of MS-1.

$$\begin{bmatrix} 0 & 0 & 0 & L/2 & L & L/2 & 0 \\ 0 & L/2 & L & L & L & L/2 & 0 \end{bmatrix}$$

Figure 2.6: Method of defining each microstructure with a set of coordinates making up the triangles contained within each structure. This method is applied to Microstructure 1 (MS-1). Figure derived from [53].

The reference library contains a total of 12 matrices, each matrix defining the triangles making up each microstructure. The size of each matrix is 2 x n, where the first row contains the x-coordinates and the second row contains the y-coordinates, and n depends on number of total coordinates to construct each triangle, as some microstructures contain three triangles while others only contain two. These 12 matrices are the only values that are manually defined in this code and by using the standard three-by-three grid that was used to define each structure, any structure can be defined using this grid system. The automatic assignment of nodes and elements for the entire beam component is completed in the main code, outside of the reference library of matrices.

This main code no longer considers the initial optimization problem, instead the basis of this main code builds a beam of microstructures that is defined by the user and solves how the displacement behavior of the constructed beam depends on the external force applied. The automatic assignment of nodes and connections is completed in this main code by first gathering the vector that defines the make-up of the beam. This vector is a list of microstructures that can be any combination of the 12 microstructures in the database. This vector tells the library function, containing the definition of each microstructure, which matrices of coordinates to pull out in order to construct the beam within the main code. The construction of the entire beam is defined as a matrix containing a list of all of the coordinates pulled from this library. When this matrix is defined, it ensures that the coordinates of each node within each microstructure is based on its current position of the beam, instead of its base coordinates from the library. Therefore each new section or microstructure contains coordinates that are based on $n \times L$, where n is the number of sections that precede that microstructure and L is the length of each section of the beam which is still based on the total length of the beam divided by the total number of sections contained in the beam. This matrix of coordinates defining the entire beam structure is then filtered by eliminating repeated coordinates, as each, unique coordinate needs to define a node only once. This filtered matrix is then converted into node numbers, from 1 to the total length of the matrix, which also defines the total number of nodes in the beam. The conversion of coordinates to node numbers is completed by using the same guidelines when defining the triangles of each microstructure, where the smallest x-coordinate is defined first. After each, unique coordinate is defined a node number, the pre-filtered matrix containing all of the original coordinates that define the construction of the entire beam is called back. Each coordinate within this matrix is then assigned its respective node number. The node connections that define the finite elements are found by first separating this pre-filtered matrix into the triangles making up the structure. This can be done by finding where adjacent, same-valued nodes are located in the matrix, since this would indicate the end and beginning of a new triangle. Once the number of triangles in the beam is established the connections between each node or the definition of elements can be found. This is done by following the same tracing technique used to define the triangles of each microstructure, as the nodes are in the original order of the coordinates defining each of these triangles. Since some connections will be repeated in the connecting interfaces of each section, these repeated connections are identified and eliminated. See Figure 2.7 to view the visual output that demonstrates an example of the automatic definition of node numbers and their connections, defined as the elements, for the entire beam of microstructures.

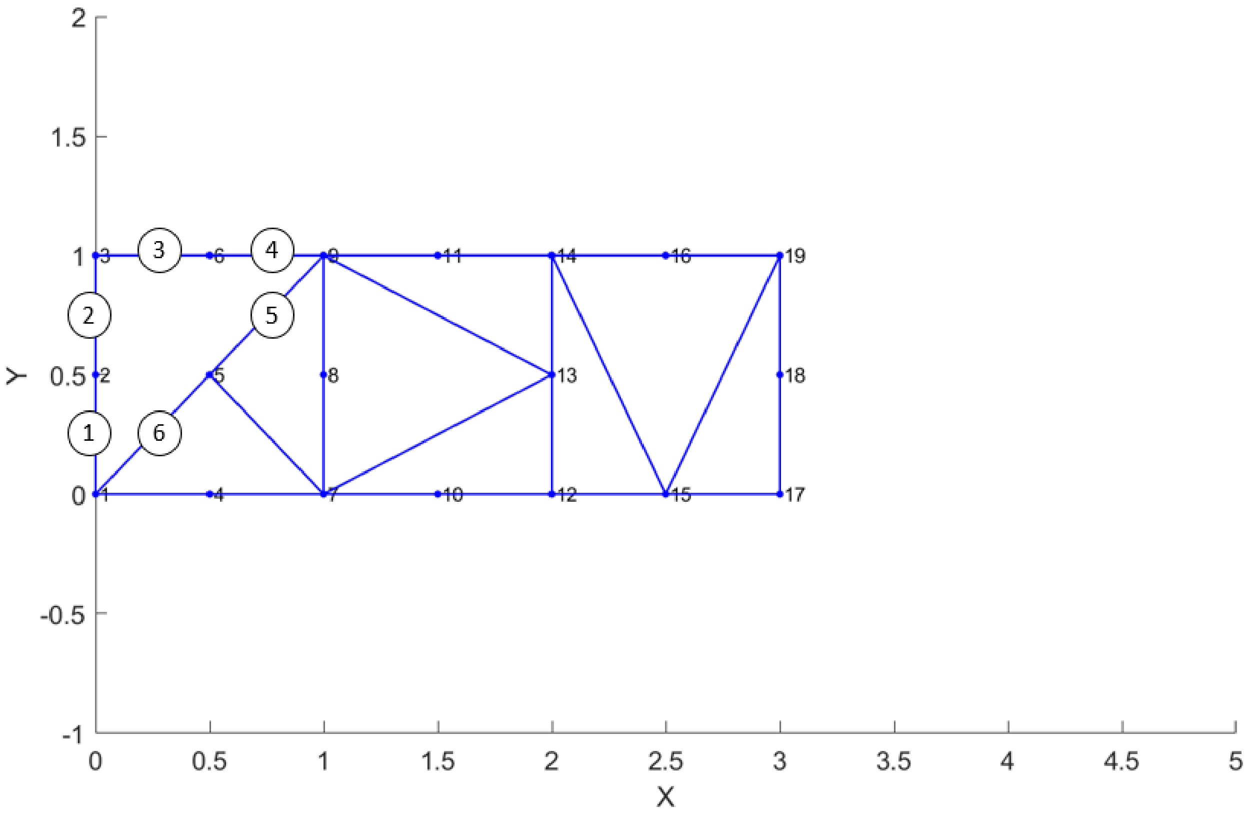


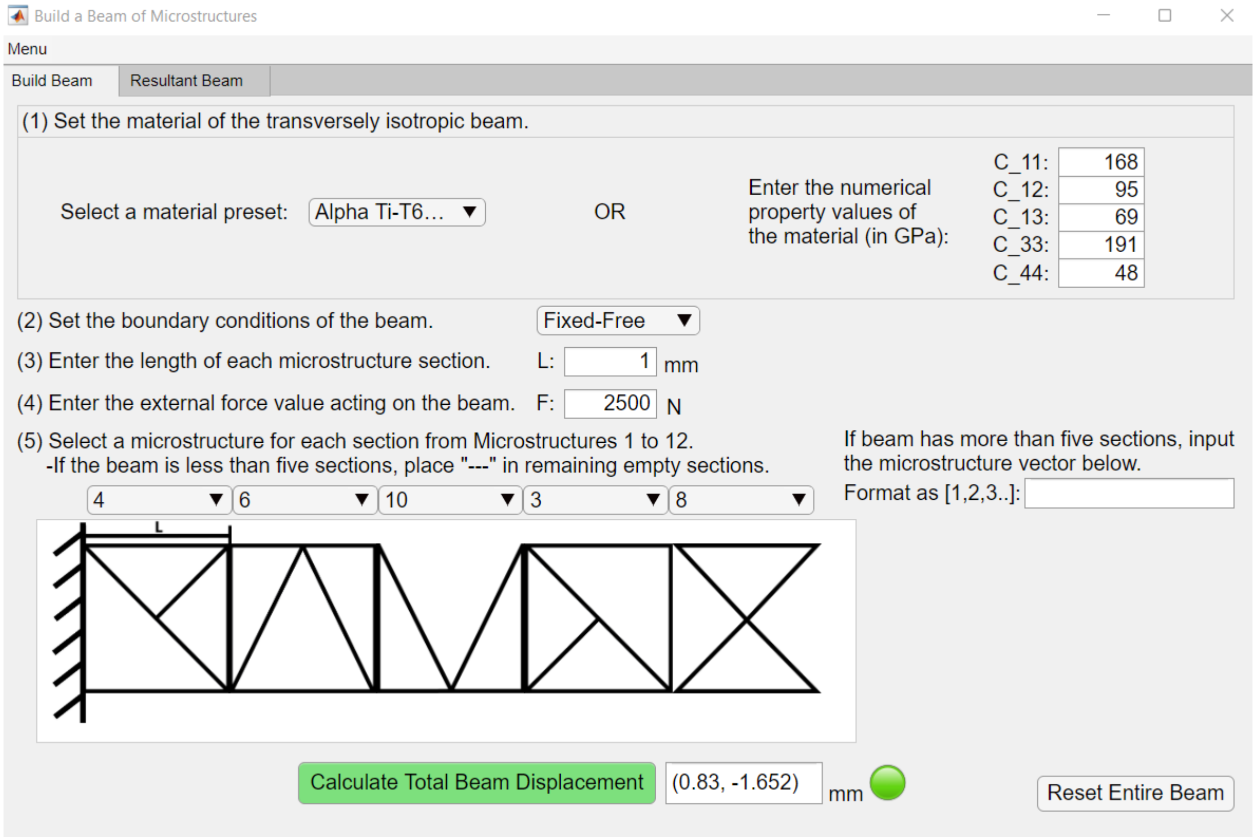
Figure 2.7: Automatic assignment of nodes and subsequent elements by using the second method, for a beam made of Microstructures 1, 12, and 5 in that order. Figure derived from [51].

This method allowed for the length of each element to be automatically defined as well as the automatic generation of the force vector affecting the entire beam. Through this second method of modeling the beam, the FEA of any length beam, made of any combination of microstructures within the database, can be performed. As a result of the freedom of design this modeling technique provides, the initial optimization problem was not applied. As if the same exhaustive search method was used in completing this problem, it would be computationally expensive to search through 12^n different combinations, where n is the number of sections making up the beam. Although this optimization was not analyzed, this method led to the development of a graphical user interface (GUI) that allows users to build any kind of beam made of the truss-based microstructures and subsequently, understand how differing materials and external forces may affect the displacement behavior of their built beam through FEA.

2.4 Development of Graphical User Interface

From literature review, the lack of design tools specifically for mechanical metamaterials was listed as one of the factors that contributes to the delay in the application of these kinds of materials in industry [3]. The developed graphical user interface (GUI), that allows FEA to be performed on a beam made of truss-based microstructures, was made possible by the development of the second modeling method previously described. This GUI could serve as a template for future GUIs and design tools specifically meant to target users working with truss-based or any kind mechanical metamaterial.

The GUI was created using MATLAB's App Designer Tool. The main panel of this GUI, shown in Figure 2.8, displays the different options a user can select from which allows them to customize and construct their beam made of microstructures. The first property that the user needs to define is the material the beam is made of. They have the option of selecting material presets of α-Ti-6Al-4V or β-Ti-6Al-4V, or they can select the "Manual Input" option and enter the property values of $C_{1,1}$, $C_{1,2}$, $C_{1,3}$, $C_{3,3}$, and $C_{4,4}$ in order to build a beam with any material of their choice. The following step is defining the boundary conditions of the beam. The ability to define different kinds of boundary conditions of the beam was not previously possible with the first method in modeling the beam, but with the automatic assignment of the nodes and elements these boundary conditions can be adjusted for. The beam's boundary conditions that the user can select from include "Fixed-Free", "Fixed-Fixed", "Fixed-Roller", or "Roller-Roller". The length of each section of the beam is then defined by numerical input, this length value is consistent across each section of the beam. The user then needs to input the external, horizontal force that is distributed across all nodes of the beam, which can be defined as either tension or compression. When these values are defined, the user can then design the microstructure topology of the beam. They have the option of creating a beam of 1 to n number of sections. If their beam contains five or fewer sections they can select from a drop-down menu of microstructures for each section and see their constructed beam in real time. Otherwise they can type in a vector that defines the microstructure make-up of the beam in a separate input. This allows the user to exceed a beam made of five sections. Additionally, if the user has a pre-defined beam make-up they can upload an Excel file into the program as well. This freedom of choice in the design of the microstructures making up each section of the beam, allows a user to choose from 12^n combinations to define the microstructure topology of the beam, where n is the number of sections they decide to have in their beam. The user does not need to input the beam's properties in the particular order described, to indicate when the beam is completely defined the "Calculate Total Beam Displacement" button will turn from red to green. This shift indicates that a FEA can then be applied to their constructed beam. Once this button is pressed the total x- and y-displacement caused by the set external force will be displayed in the adjacent box. While this number describes a simple summary of the beam's resultant behavior, there is another tab the user can view that will provide a more detailed overview of the resultant beam.


Build a Beam of Microstructures
Menu
Build Beam
Resultant Beam
L
Calculate Total Beam Displacement
Reset Entire Beam

(1) Set the material of the transversely isotropic beam.

Select a material preset: Alpha Ti-T6... ▼

OR

Enter the numerical property values of the material (in GPa):

C_11:	168
C_12:	95
C_13:	69
C_33:	191
C_44:	48

(2) Set the boundary conditions of the beam. Fixed-Free ▼

(3) Enter the length of each microstructure section. L: 1 mm

(4) Enter the external force value acting on the beam. F: 2500 N

(5) Select a microstructure for each section from Microstructures 1 to 12.
-If the beam is less than five sections, place "---" in remaining empty sections.

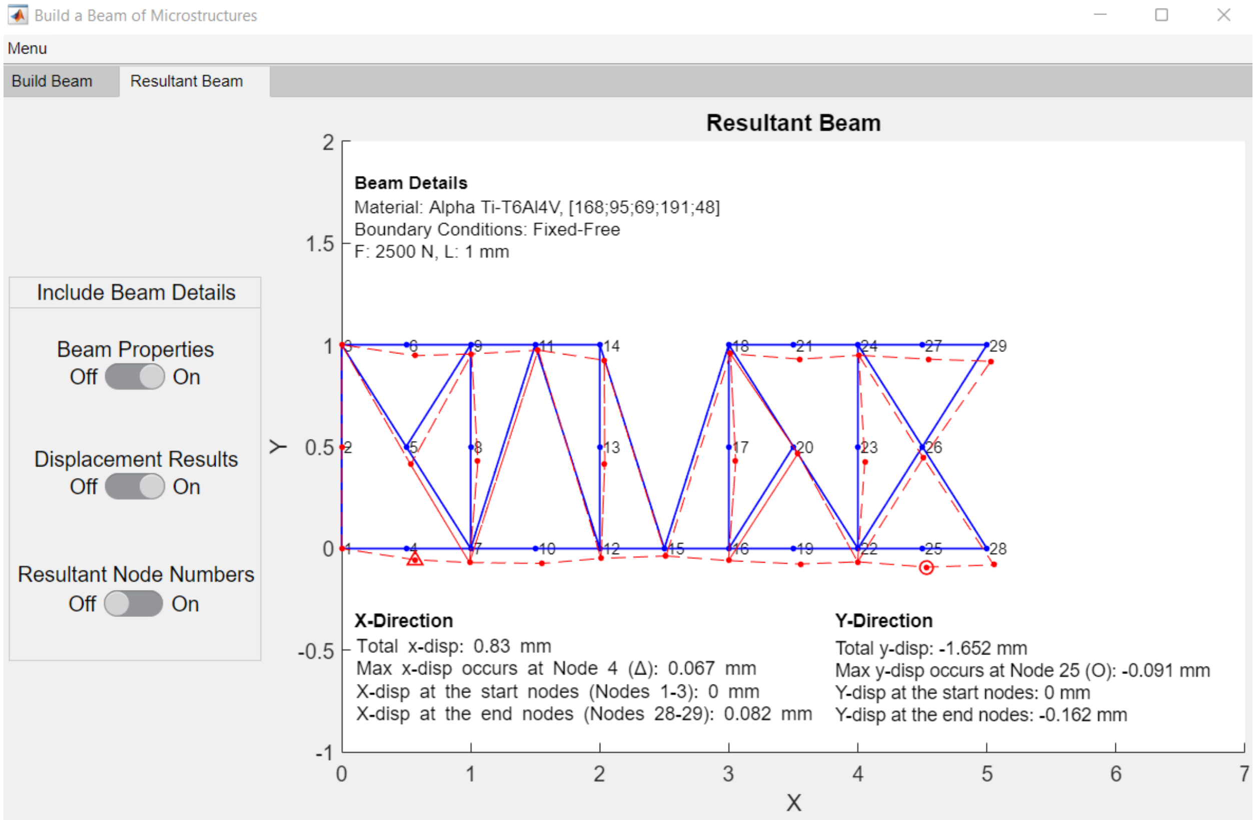
4 ▼ 6 ▼ 10 ▼ 3 ▼ 8 ▼

If beam has more than five sections, input the microstructure vector below.
Format as [1,2,3..]:

(0.83, -1.652) mm ●

Figure 2.8: The main panel of the GUI that allows the user to build a beam of their choice [51].

Once the user is done defining and running the FEA on their beam they can select the "Resultant Beam" tab to switch to another panel that displays a more detailed account of the beam's displacement behavior, shown in Figure 2.9. Within this panel the user can view the individual displacement of each node of the beam. More detailed information, aside from the total displacement of the beam is also included, such as the x- and y-displacement at the start and end nodes, along with indicators that select the nodes of maximum x- and y-displacement. By preferences of the user, this amount of detail can be toggled on and off by the knobs located on the left panel and saved as a photo. A video demoing this GUI is provided, https://youtu.be/YF1iXNcSl20. This developed GUI has been published on the open-source platform of GitHub, available at https://github.com/kmcmi486/buildabeamofmicrostructures/. By having this code available to the public, anyone has the opportunity to interact with this interface and understand how truss-based microstructures can affect the displacement behavior a mechanical beam component.



Include Beam Details

Beam Properties
Off On

Displacement Results
Off On

Resultant Node Numbers
Off On

Beam Details
Material: Alpha Ti-T6Al4V, [168;95;69;191;48]
Boundary Conditions: Fixed-Free
F: 2500 N, L: 1 mm

X-Direction
Total x-disp: 0.83 mm
Max x-disp occurs at Node 4 (Δ): 0.067 mm
X-disp at the start nodes (Nodes 1-3): 0 mm
X-disp at the end nodes (Nodes 28-29): 0.082 mm

Y-Direction
Total y-disp: -1.652 mm
Max y-disp occurs at Node 25 (O): -0.091 mm
Y-disp at the start nodes: 0 mm
Y-disp at the end nodes: -0.162 mm

Figure 2.9: The resultant panel of the GUI that displays a more detailed account of the beam's displacement behavior. This beam is the result of the design and inputs displayed in Figure 2.8 [51].

To ensure that the developed GUI was working properly, several varied beam configurations were tested and analyzed. Two configurations were directly compared to one another in order to determine the direct effect that different inputs would have on the resultant beam output. The first configuration that will be discussed is a beam made of α-Ti-6Al-4V. This beam contains a total of four sections and a fixed-free boundary condition. The horizontal force acting on the beam is a tensile force of 2500 N. The second configuration is made of β-Ti-6Al-4V with four sections and a fixed-roller boundary condition. In this configuration, a compression force double in magnitude, of 5000 N, is acting on the beam. Both configurations have section lengths of 1 mm. Each configuration and their resultant x- and y-displacement values are summarized in Table 2.1, while Figure 2.10 displays the resultant first and second beam configurations respectively from left to right, respectively.

Table 2.1: The varied parameters for each beam configuration and their resultant x- and y-displacements [51]

Input Parameters	**Configuration 1**	**Configuration 2**
Material	α-phase Ti-6Al-4V	β-phase Ti-6Al-4V
Applied Force (N)	2500	-5000
Section Length (mm)	1	1
Boundary Conditions	Fixed-Free	Fixed-Roller
Microstructure Design	[1,6,5,3]	[12,7,10,11]
Resultant Total Displacement Values		
X-Displacement (mm)	0.837	-1.859
Y-Displacement (mm)	-1.582	3.403

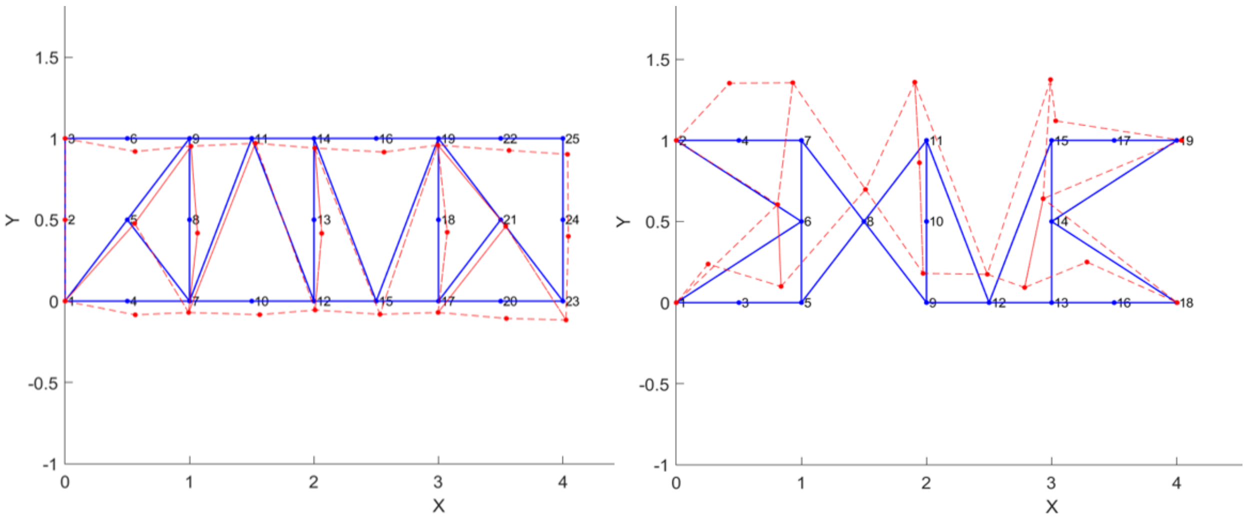


Figure 2.10: The beams resulting from the defined first and second configurations [51].

When analyzing the first configuration that experiences a smaller, tensile force the resultant direction of the beam is the southeast direction, with displacement values smaller in value when compared to the second configuration which experiences a compression force twice higher in magnitude. The resultant beam of this second configuration is also displaced in the opposite direction of northwest. After analyzing the numerical and visual results of the configurations, it was determined that the GUI executes properly, but in order to verify the results more extensively, external FEA software can be utilized [53].

2.5 Possible Improvements to Developed Graphical User Interface and Future Work on Truss-Based Mechanical Metamaterials

The development of the GUI led to a generation of new ideas in how this GUI could be improved for actual industry. As the microstructure for mechanical metamaterials can be selected and designed to exhibit certain properties, instead of having the GUI display the resultant behavior of a predefined mechanical beam the user could instead input design constraints or goals which could then be interpreted as objectives in the design of beam. This objective would be to determine an optimum layout of microstructures that satisfies the user's inputs. These kind of inverse design problems could be approached as optimization problems, similar to the initial optimization problem discussed during the development of the first method in modeling the beam of microstructures. Yet, instead of utilizing an exhaustive search, a time-efficient, gradient-based optimization algorithm can be applied so that an optimum solution could be obtained more efficiently. Extending on this idea, developing a machine learning algorithm that can learn from the solutions of the inverse design problems, could possibly lead to the generation of more kinds of truss-based geometries that are based on the defined grid system discussed in the second modeling method. While the original developed database of truss-based microstructures does provide a variety in topology, along with idea of adding solutions by applying machine learning, the database could also be improved by having the user input their own geometry based on the same grid system. To expand outside of the option of analyzing transversely isotropic materials, more kinds of materials could also be considered by expanding the manual input material properties. To verify the results from the GUI, several 2D-beam configurations within the GUI were tested and analyzed. In order to further verify these results from the GUI, the 2D-beam component could be translated in external FEA software or to an actual 3D-component through additive manufacturing. Once 3D-components are fabricated, the 2D-results can be analyzed and compared to corresponding tension and compression tests.

Aside from possible improvements to the developed GUI, other ideas on how to further study these truss-based mechanical metamaterials is also considered. To further characterize the effect of these truss-based microstructures on the mechanical beam component, other properties beside displacement behavior could be investigated. This could include observing failure behavior of these microstructures such as buckling and how long term force applications, such as cyclic loading, affects these structures as well.

2.6 Summary

In this research, truss-based mechanical metamaterials were studied through their application to a 2D-mechanical beam component. This was completed by developing a database made of 12 truss-based microstructures that make up the topology of the beam component. In order to understand how the beam component would behave with these kinds of structures a computational modeling strategy would need to be employed in order to conduct FEA on the beam. As the beam is split into square sections that can be filled with the database of truss-based microstructures, the first method developed to model the beam analyzed each section individually. Using an exhaustive search method an optimization problem was approached in which the optimum layout of microstructures was determined to minimize the displacement of the beam subject to an external, horizontal force. From the analysis of the results of this optimization problem, the interfaces of each beam section were not accurately being accounted for, therefore the second method in modeling the beam for FEA resulted. This method led to standardizing the make-up of each microstructure on a grid system. This standardization allowed for the automatic assignment of nodes and elements, and their subsequent connections. Therefore, no matter the microstructure topology of the beam, the beam could be analyzed in its entirety as opposed to a conglomerate of individual microstructure sections. This second modeling method led to the development of an easy-to-use GUI, which helps users easily understand how truss-based microstructures can affect the displacement behavior of a mechanical component. This GUI allows a user to construct a beam of any combination of the 12 microstructures from the database and understand the deformation behavior of their designed beam. It is aimed that the GUI contributes to the development of computational tools for the design of mechanical metamaterials in industry. Although there are promising applications for truss-based mechanical metamaterials, such as to fields of biomedical and aerospace, literature has indicated that these materials are prone to high stress concentrations in their corners and susceptible to symmetry breaking defects when manufactured [5], [25], [54]. This is also implied by the displacement results of Figure 2.10. With the continued study of different kinds of mechanical metamaterials, a self-assembled kind of mechanical metamaterial was discovered. Contrast to trusses and other periodic mechanical metamaterials, the geometry of the unit cells is smooth and less susceptible to high stress concentrations. These found mechanical materials are called spinodoids and the study completed with these materials will be discussed in Chapter 3.

Chapter 3

Investigating Mechanical Metamaterials with Spinodal Topologies, Spinodoids

3.1 Research on Spinodoids

The beginning of this research involved studying mechanical metamaterials with truss-based geometry. These truss-based structures were studied within the 2D-space through the development of a database which was applied to the construction of a mechanical beam component. The research was finalized through the development of a graphical user interface which allows users to understand the effect these truss-based microstrucures have on a mechanical component. To further extend the study of mechanical metamaterials, more research was completed on a novel kind of mechanical metamaterial called spinodoids. Spinodoids form by replicating the process of spinodal decomposition. Spinodal decomposition is a phase separation process that results in the formation of two smooth phases [55]. In the formation of spinodoids, one phase from the process is classified as void, while the other is classified as solid [5]. The combination of these phases produces the smooth, bi-continuous structures, seen in Figure 3.1. The smoothness of these structures makes these materials less susceptible to high stress concentrations present in truss-based mechanical metamaterials. In addition, the spinodal topologies are aperiodic compared to more commonly known origami-, plate-, or truss-based mechanical metamaterials [5]. Periodic geometries are more prone to symmetry breaking defects that lead to unpredictable mechanical behavior [5]. With the perceived benefits over commonly known mechanical metamaterials, these spinodal topology-based mechanical metamaterials, known as spinodoids, were further studied within the 2D-space. Within this study, 2D-spinodoids were generated and characterized and through the use of data-driven modeling techniques the inverse design of these structures was performed. Some methods and techniques proposed in this chapter have been submitted as a journal article to AIAA Journal: K. McMillan and P. Acar, "Inverse Design Optimization of 2D-Mechanical Metamaterials under Uncertainty", AIAA Journal, under review.

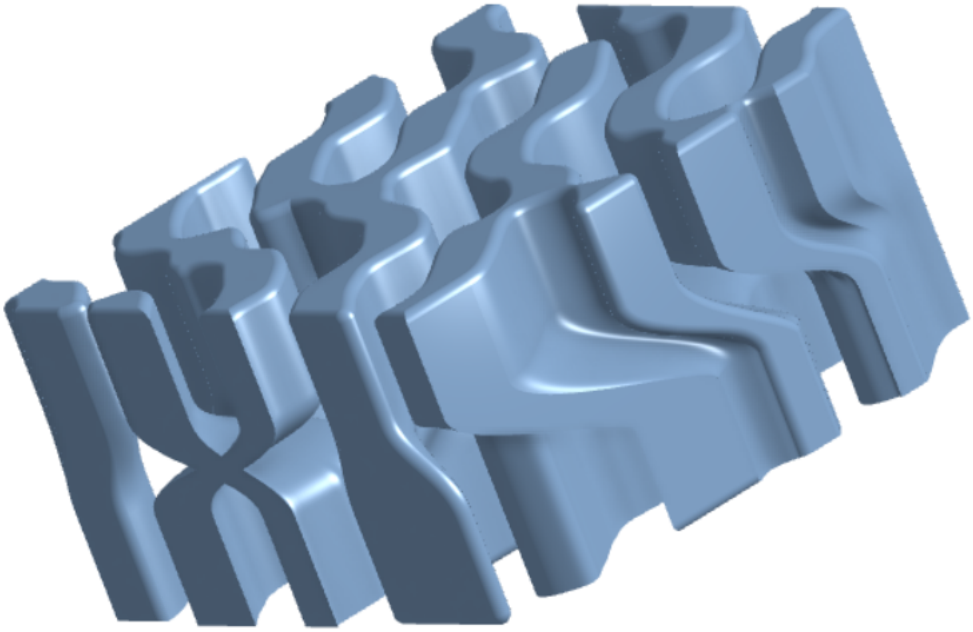


Figure 3.1: Example of a spinodoid-like structure created from 3D-extrusions of generated images representing spinodoids. These extrusions were meshed together to form the spinodal-like geometry seen in this structure. Figure derived from [60].

3.2 Mechanical Metamaterials Formed from Spinodal Decomposition: Spinodoids

The materials studied take form by simulating the phase separation process called spinodal decomposition. This is a natural process that is taken advantage of in the manufacturing of polymeric membranes, binary alloys, and micro-emulsions [5], [33], [56], [57]. Spinodal decomposition occurs in a thermodynamically unstable solid solution [32], [58]. The process first appears as a solution of finely mixed precipitates, as it eventually evolves, two-distinct phases form, which then coexist in equilibrium [32]. Compared to other phase separation processes, the dynamics of spinodal decomposition can be modeled through an approximate analytical solution known as the Cahn-Hilliard Equation [55], [59]. Although this phase separation can be modeled through this analytical solution, it is computationally expensive to model it numerically. Therefore Kumar *et. al* utilized an efficient model of the equation by ridding of its time dependency and applying Gaussian random fields (GRFs) [5], [31]. This formulation can be described by defining a phase field of φ that represents the fluctuations of the differing concentrations of the two separating phases. When considering the formation of spinodoids, one phase is defined as solid, while the other is defined as void. Through GRFs, this phase field can be decomposed with Fourier analysis. With this, the concentration φ can describe the solid or void phase at $x \in V$, where V is described by $V \in \mathbb{R}^3$ [5],

$$\begin{gathered}\varphi(X) = \sqrt{\frac{2}{N}} \sum_{i=1}^{N>>1} \cos(\beta \mathbf{n_i} \cdot \mathbf{x} + \gamma_i), \text{with} \\ \mathbf{n_i} \sim \mathcal{U}(S^2), \gamma_i \sim \mathcal{U}([0, 2\pi])\end{gathered} \tag{3.1}$$

where $S^2 = \{\mathbf{k} \in \mathbb{R}^2 : ||\mathbf{k}|| = 1\}$ represents the unit circle in two dimensions. Within this equation N is the number of waves, $\beta > 0$ is a constant wave number, $\mathbf{n_i} \sim \mathcal{U}(S^2)$ is the direction of the wave vector, and $\gamma_i \sim \mathcal{U}([0, 2\pi])$ is the phase angle of the ith wave vector, both the direction and phase angle of the wave vector are uniformly-distributed. Equation 3.1 represents the GRF that produces isotropic topologies, where anisotropic topologies are resultant of the approximation produced from a non-uniform orientation distribution function (ODF) described by [5],

$$\mathbf{n_i} \sim \mathcal{U}(\{\mathbf{k} \in S^2 : (|\mathbf{k} \cdot \widehat{\mathbf{e_1}}| > \cos\theta_1) \oplus (|\mathbf{k} \cdot \widehat{\mathbf{e_2}}| > \cos\theta_2) \oplus (|\mathbf{k} \cdot \widehat{\mathbf{e_3}}| > \cos\theta_3)\}), \tag{3.2}$$

where the θ values control the resultant formed spinodoid topologies. To determine the assignment of solid and void phases in the structure the level sets of the phase field were computed. Through a binary indicator function, solid or void is assigned at X by [5],

$$\text{solid network} : \chi(X) = \begin{cases} 1 \text{ if } \varphi(X) \leq \varphi_0 \\ 0 \text{ if } \varphi(X) > \varphi_0 \end{cases} \tag{3.3}$$

where φ_0 is the threshold at the average relative density of the solid phase. This is described by [5],

$$\varphi_0 = \sqrt{2}\text{erf}^{-1}(2\rho - 1) \tag{3.4}$$

where ρ is defined by $\rho = \mathbb{E}[\chi]$. To prevent disjoint solid domains ρ is contained within the range of $\rho \in [0.3, 1]$, in this research a value of $\rho = 0.64$ was used. To summarize, these equations allow for the generation of spinodal-like topologies without the need to undergo computationally expensive simulations of the Cahn-Hilliard equation [5], [31]. In this study, these equations are considered in the 2D-domain as opposed to the 3D-domain considered in Kumar *et al.*'s work.

3.3 Generating Spinodoids

Like the research on truss-based mechanical metamaterials, MATLAB was utilized in the generation of the 2D-spinodoids in this research. The geometries are produced by following the formulations defined in Equations 3.1 and 3.2. The geometries are considered as unit cells that can be repeated or distributed throughout a material to create a hierarchy that defines a mechanical metamaterial. The spinodoid geometries in this research are represented by 2D-monochrome images, where black is considered solid material and white is considered as void. These images are defined by square arrays, whose size depends on the pixel resolution of the image. These arrays consist of zeros and ones, where zero indicates black and one indicates white. The geometries are controlled by the θ values displayed in Equation 3.2. The range of these values lie between 0 and $\pi/2$, as symmetric values would produce duplicate geometries. Although there is not a set number of θ parameters that control the resultant geometry of the spinodoid, the same number of θ parameters set in Kumar *et. al*'s research were used in this study. To study the effect of increasing the number of θ parameters on the spinodoid geometry, the Young's modulus, which is the mechanical property that is later used to characterize the generated spinodoids, was compared for spinodoids controlled by four and five θ parameters. The ranges of the Young's modulus values resulting from spinodoids controlled by four and five θ values were determined through a genetic algorithm where the highest and lowest values were determined. Figure 3.2 demonstrates that the range of the Young's modulus values did not differ significantly between spinodoid geometry controlled by three, four, or five θ parameters [60]. This indifference to increasing the amount of controlling parameters justified using only three θ parameters. In addition, increasing the amount of θ parameters increases the dimensionality of the system, which would eventually lead to a system that is not as efficient to model as one controlled by fewer parameters. A lower amount of parameters was not utilized, as it would not allow for a greater control over the geometry compared to three. Therefore, due to the idea of reducing the amount of parameters that would later be defined as design variables and to also maintain a greater control over the resultant geometry, three θ values were selected to be used in generating the data set of spinodoids. Depending on the pixel resolution of the desired image, the spinodoid can be generated in under a few minutes, but as the pixel resolution of the image increases so does the required time to produce the image within MATLAB, see Figure 3.3 [60]. Within this research the idea of computational efficiency was kept in mind, therefore a large database of lower resolution images were generated to be utilized later in the data-driven modeling techniques.

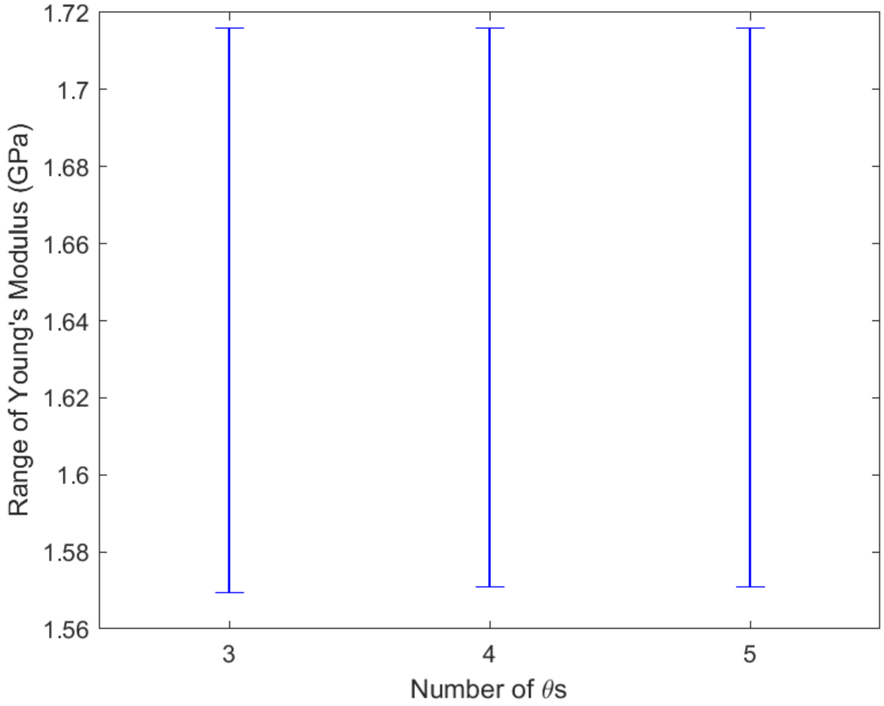


Figure 3.2: The value of Young's modulus dependent on the number of θ-parameters controlling the resultant spinodoid [60].

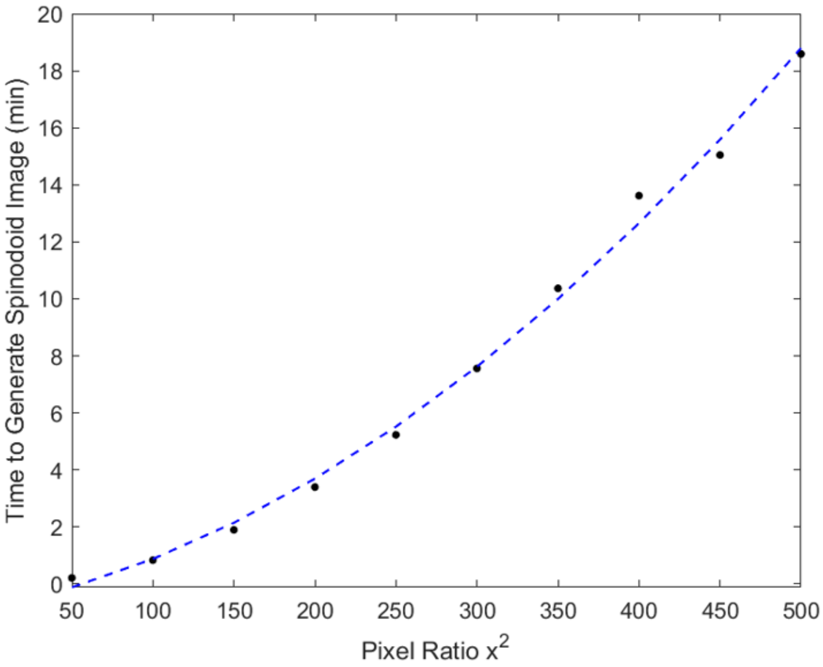
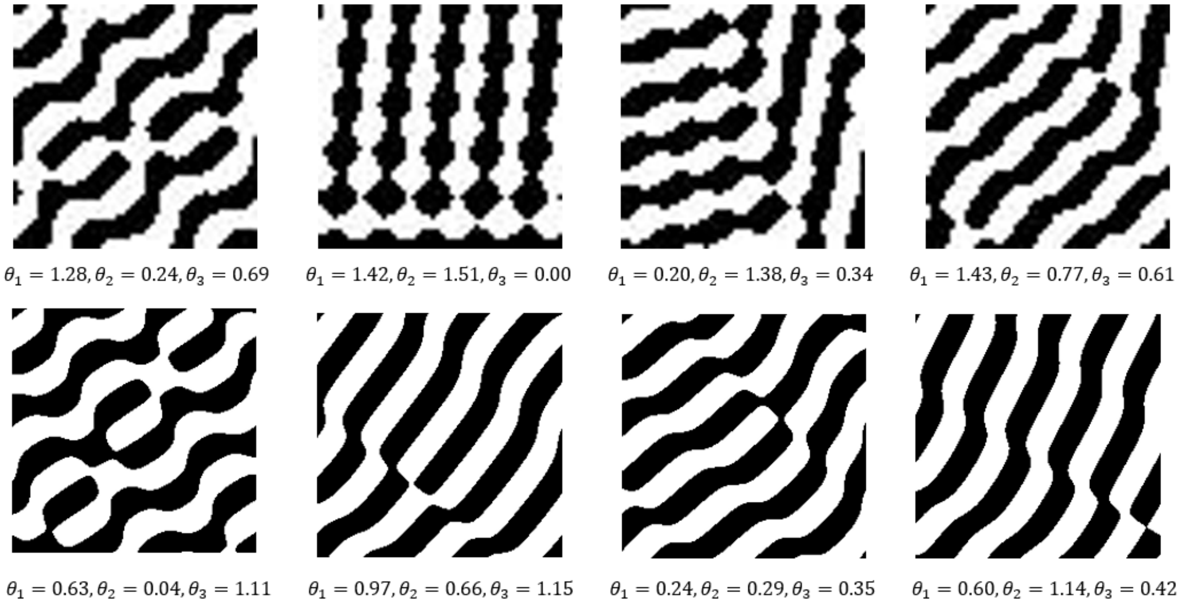


Figure 3.3: The time it takes to generate a 2D-spinodoid image depends on the pixel resolution of the image [60].

Examples of generated spinodoids and the θ values that produced them are displayed in Figure 3.4, the top images have a 50 by 50 pixel resolution, while the remaining bottom images have a 200 by 200 pixel resolution. To generate an extensive training data set, a total of 10,000 50 by 50 images were generated as the initial set of images representing different spinodoid unit cells. This data set was later extended for a specific data-driven modeling technique by the addition of 100 200 by 200 pixel images.

3.4 Characterizing Spinodoids

As the potential applications for this kind of mechanical metamaterials have been theorized to serve as replacements for bone implants, the mechanical properties of these generated spinodoids need to be determined [5]. Calculating the Young's modulus values of the gener-



$\theta_1 = 1.28, \theta_2 = 0.24, \theta_3 = 0.69$ $\theta_1 = 1.42, \theta_2 = 1.51, \theta_3 = 0.00$ $\theta_1 = 0.20, \theta_2 = 1.38, \theta_3 = 0.34$ $\theta_1 = 1.43, \theta_2 = 0.77, \theta_3 = 0.61$

$\theta_1 = 0.63, \theta_2 = 0.04, \theta_3 = 1.11$ $\theta_1 = 0.97, \theta_2 = 0.66, \theta_3 = 1.15$ $\theta_1 = 0.24, \theta_2 = 0.29, \theta_3 = 0.35$ $\theta_1 = 0.60, \theta_2 = 1.14, \theta_3 = 0.42$

Figure 3.4: Examples of the generated spinodoid monochrome images, the top images have a resolution of 50x50 pixels, while the bottom images have a resolution of 200x200 pixels. Figure derived from [60].

ated 2D-spinodoids would allow for the determination of the design space of these materials. Since spinodoids are represented in the form of 2D monochrome images, their Young's modulus is calculated through density-based pixel calculations. These calculations follow the formulation presented by Wang *et. al*, who used a modified solid isotropic material with penalization (SIMP) to calculate the Young's modulus of porous 2D-mechanical metamaterials while optimizing their designs [61], [62]. As this formulation calculates the Young's modulus of each element within the unit cell, the generated spinodoid images needed to be discretized based on their pixel resolution. Therefore the total number of elements whose Young's modulus would be calculated would be n^2, where n is the size of the square pixel resolution. The Young's modulus of the unit cell is then taken as the average of the Young's modulus of each element. Which is found by summing each modulus value and dividing the sum by the total number of elements in the image. The formulation applied to these images is provided [61], [62],

$$E_e(\rho_e) = E_{min} + \rho_e^p(E_0 - E_{min}), \tag{3.5}$$

where E_e, is the Young's modulus of each element, E_{min} is the Young's modulus of the void material to prevent singularity of the stiffness matrix (this was set to 10^{-9} GPa), ρ_e is assigned a value of 0 or 1 depending on if the pixel is black or white, p is the penalization factor if ρ_e was not a discrete value of 1 or 0 (to further drive the density to a 0 to 1 solution), and E_0 is the Young's modulus of the material that makes up the solid spinodoid. In this study, black pixels are considered solid material and white pixels are void, therefore ρ_e is assigned a value of 1 if the element is a black pixel and assigned a value of 0 if the element is a white pixel. Additionally, this density-based pixel formulation is not sensitive to direction. The solid material assigned to the spinodoids was arbitrarily chosen as polymethyl methacrylate (PMMA), leaving E_0 to be 3.3 GPa, but any material could be assigned if the Young's modulus is known [63]. The final Young's modulus of the spinodoid would then be found through,

$$E_{\text{spinodoid}} = \frac{\sum E_{e,\text{ black pixels}} + \sum E_{e,\text{ white pixels}}}{n}, \tag{3.6}$$

where $E_{\text{spinodoid}}$ is the Young's modulus of the spinodoid, $\sum E_e$, is the summation of the Young's modulus values for all of the black and white pixels respectively, and n is the total number of elements (or pixels) in the image. Using this formulation, the Young's modulus value of all generated spinodoids was calculated. Figure 3.5 demonstrates the entire design space of the 10,100 generated spinodoids, where the range of Young's modulus values lies between 1.55 and 1.71 GPa. This figure also indicates that the calculated Young's modulus value does not depend on the pixel resolution as similar modulus ranges arise between the 50 by 50 and the 200 by 200 resolution images. Some spinodoids and their corresponding Young's modulus values are displayed in Figure 3.6. With these calculated values, a relationship between the θ parameters controlling the resultant geometry of the spindodoids and the Young's modulus value could be explored through data-driven modeling techniques. By defining the relationship between the input, the controlling θ parameters, and the output, the Young's modulus value, an inverse design approach can be used to drive the unique design of a spinodoid that is created from a desired material property, as opposed to selecting a material satisfying a desired material property.

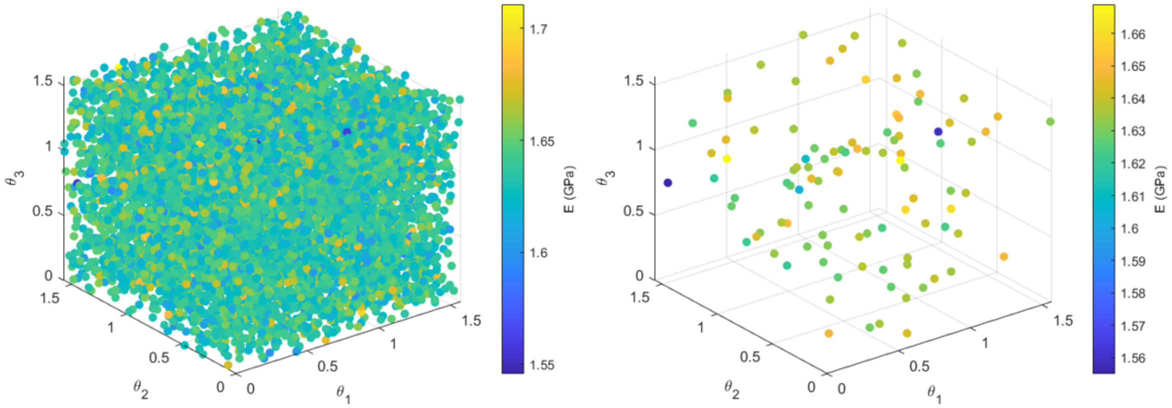
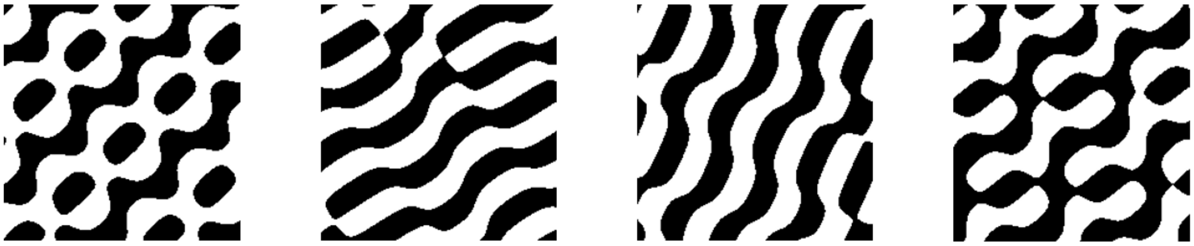


Figure 3.5: The calculated Young's modulus depending on θ_1, θ_2, and θ_3 for all generated spinodoids is displayed on the left and on the right, the same relationship is displayed for the generated 200 by 200 resolution images. Figure derived from [60].



E = 1.555 GPa
$\theta_1 = 0.032, \theta_2 = 1.485, \theta_3 = 0.773$

E = 1.622 GPa
$\theta_1 = 0.196, \theta_2 = 0.985, \theta_3 = 0.490$

E = 1.632 GPa
$\theta_1 = 1.168, \theta_2 = 0.709, \theta_3 = 1.567$

E = 1.669 GPa
$\theta_1 = 0.565, \theta_2 = 0.071, \theta_3 = 1.407$

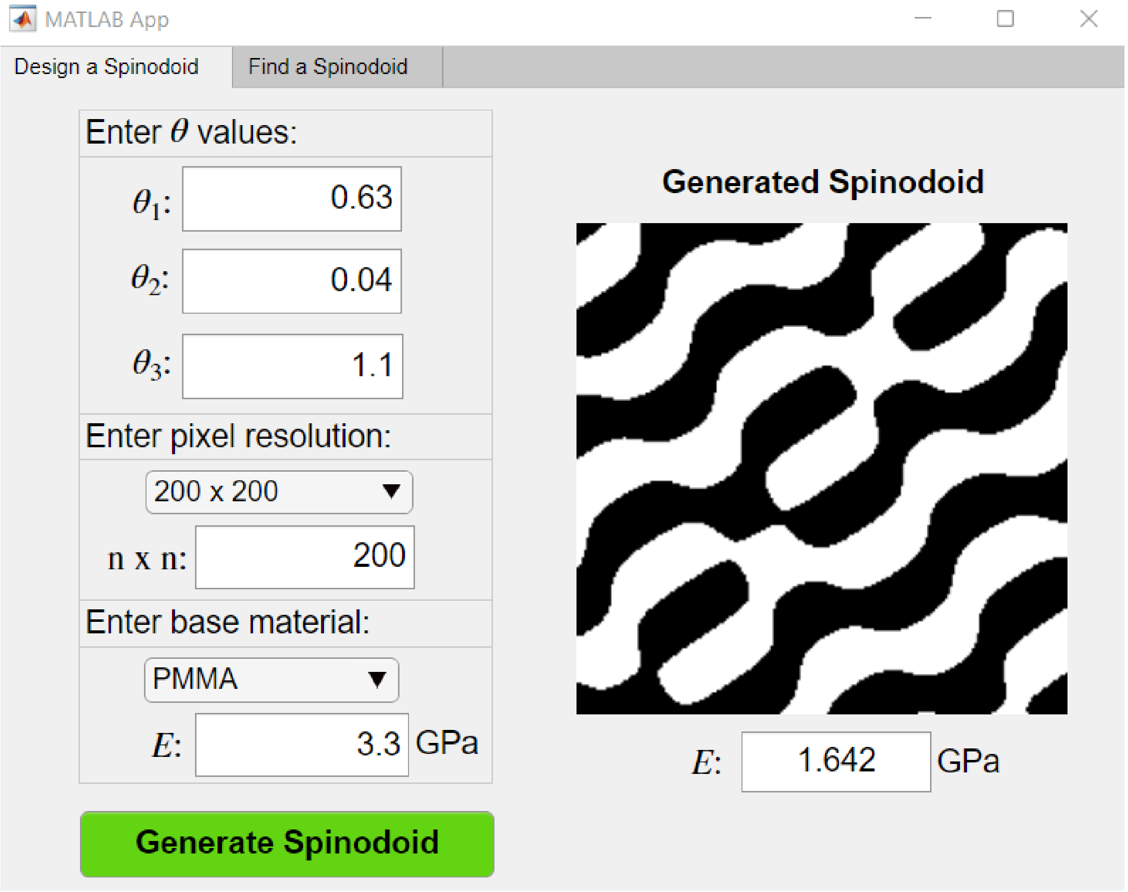
Figure 3.6: Examples of spinodoids and their calculated Young's modulus values found through density-based pixel calculations. Figure derived from [60].

3.5 Graphical User Interface to Design Spinodoids

With the capability to generate a 2D-monochrome image to represent the spinodoid and calculate the Young's modulus of the generated spinodoid, another graphical user interface (GUI) was created. This GUI allows a user to enter in three θ values, which give them control of the resultant geometry of the spinodoid. Along with these parameters, the user must enter the pixel resolution of their image and the material that their spinodoid unit cell is be made from. Once these values are entered into the GUI, seen in Figure 3.7, the user would then press the "Generate Spinodoid" button. This prompts the program to begin generating the spinodoid image. Once the spinodoid is generated, the Young's modulus value is calculated using the same density-based pixel formulations described above and the calculated value is displayed under the resultant spinodoid image. The first version of the developed GUI is published on GitHub and available at https://github.com/kmcmi486/designaspinodoid. A video demoing this GUI is provided, https://youtu.be/YF1iXNcSl20. This GUI is another design tool created to aid in the same goal described in the development of the GUI in Chapter 2.4, that allows users to construct a beam made of truss-based microstructures. These GUIs are based solely around working with and designing mechanical metamaterials, with the common goal of understanding the specific mechanical metamaterials studied in this research.

3.6 Data-driven Modeling of Spinodoids

Data-driven modeling is a popular method used in many areas of research particularly in the materials field [64], [65], [66]. This kind of modeling is utilized in materials research to estimate the non-explicit relationship between the materials and properties [67]. Machine



θ_1: 0.63

θ_2: 0.04

θ_3: 1.1

200 x 200

n x n: 200

PMMA

E: 3.3 GPa

E: 1.642 GPa

Figure 3.7: The main panel of the GUI created to design spinodoid unit cells. Figure derived from [60].

learning is a popular technique used to determine the relationship between the material and its property or more generally the input and output [31], [68], [69], [70]. In this research, principal component analysis (PCA) was first applied to the large database of spinodoid images. PCA was used to reduce the dimensionality of the image-based design space. However, the reduction in the problem dimension was not found to be satisfactory. Therefore, Gaussian process regression (GPR), was then applied to model the relationship between the controlling θ parameters and the resultant spinodoid geometry. With using this method comes the added benefit of considering uncertainty. The GPR model computes the expected values and the associated uncertainty of the material properties, without using a numerical uncertainty quantification algorithm [71].

3.6.1 Principal Component Analysis

Principal component analysis (PCA) is a machine learning method whose main goal is to reduce the dimensionality of a system, while also keeping the key characteristics of the system [72]. It is mainly applied to image-based data due to the high amount of dimensions contained within a color or monochrome image. PCA reduces the dimensions of the data by projecting the data onto lower dimensions defined as principal components (PCs). In order to reduce the dimensions while also covering the design space of the system the projected points are found by maximizing the variance of each PC, while also ensuring that each PC is uncorrelated. Therefore, each PC is orthogonal to each other in its projection to a lower dimension. These PCs are considered numerical representations of the data that describe the features or patterns of the high-dimensional data [73].

Initially a large database of 10,000, 50 by 50 monochrome images were generated. With each spinodoid being represented by a 50 by 50 pixel image the number of dimensions describing this data is 2500, derived from the resolution of the image. The high problem dimensionality increases the required computing times exponentially. Once the dimensions of the system are reduced, the PCs could be summed together with linear coefficients to describe each spinodoid. Ideally if the 2500 dimensions could be reduced to a reasonable amount of PCs, the resultant linear coefficients could be used as design variables to later optimize or derive a spinodoid with a specific modulus value. Using Scikit-learn library of Python, PCA was applied to the data set of 10,000 images [74]. With the initial application of PCA, a total of 450 PCs were needed to cover 90% of the data and a total of 938 PCs were needed to cover 95% of the data. See Figure 3.8 to view the explained variance based on the number of dimensions utilized, the figure demonstrates the number of dimensions needed to cover a certain percentage of the image-based data. After filtering was applied to the images, a total of 18 PCs were still needed to cover 80% of the data. Although this is significantly smaller than the pre-filtered data, it is still a high amount of PCs to work with, especially if the subsequent coefficients of these terms are to be considered in future optimization problems.

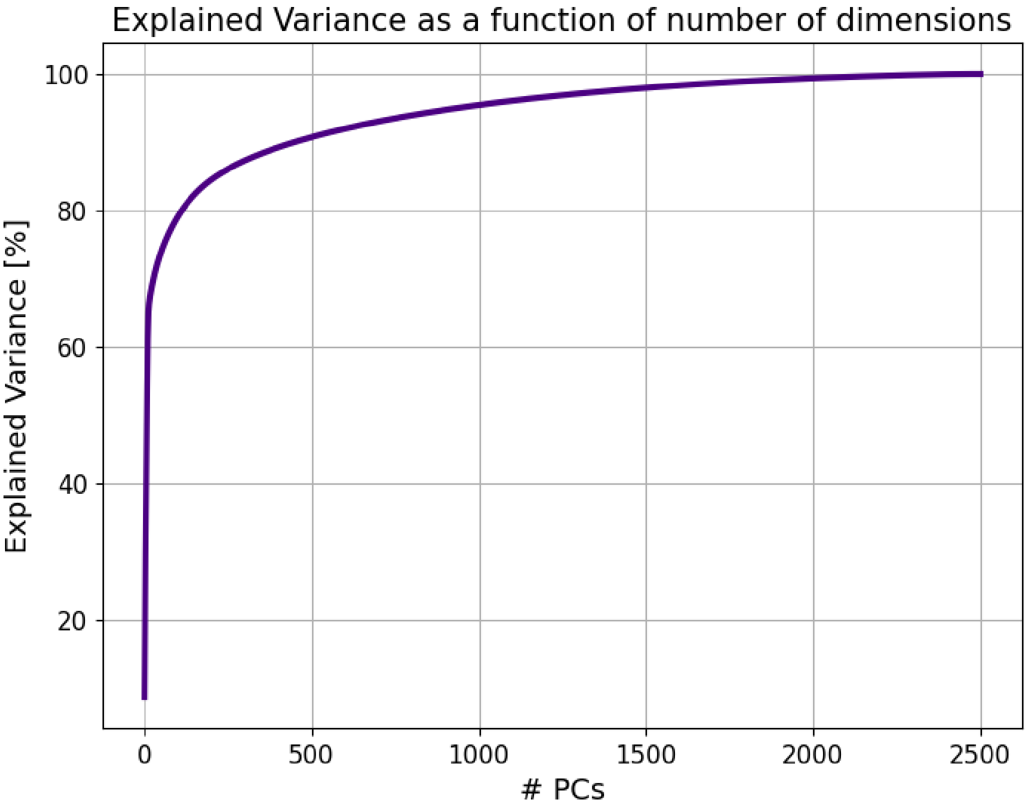


Figure 3.8: Explained variance of the database of 10,000 spinodoids based on the number of principal components utilized, as the percentage increases so does the number of required principal components [60].

Since this attempt with PCA led to a high number of required PCs to accurately represent the data, the goal of utilizing PCA as a potential modeling technique could not be proceeded with. Instead, another efficient data-driven modeling technique would need to be applied. This led to the application of GPR as another method to model the potential relationship between the spinodoids and their resultant Young's modulus value.

3.6.2 Gaussian Process Regression

Gaussian process regression (GPR) is a data-driven technique that is used to model and explore unknown relationships and functions [75]. This regression method has the same goal as other machine learning techniques of establishing a connection between the inputs and outputs of the system model. The added benefit of GPR over image-based modeling techniques, including PCA, is its ability to simultaneously calculate uncertainty associated with the expected value predictions. The analysis of uncertainty will also provide a better understanding on the effects of the fabrication-related uncertainty on spinodoid design. GPR works by predicting the probability distributions over functions that fit over a set of points. The form of these functions depends on the kernel function and its hyperparameters. The hyperparameters can be optimized to improve the fidelity of the GPR predictions [76], [77]. In this study, a multi-fidelity surrogate model is developed using GPR. This multi-fidelity model is built from low- and high-fidelity data, where the high-fidelity data is considered to be a more accurate representation of the problem. The high-fidelity data was created by generating higher resolution images of 200 by 200 pixels, that represent larger microstructural solution domains. This is characterized as high-fidelity data as the homogenized material property predictions over large microstructure domains are expected to be more convergent than the predictions over smaller domains. Accordingly, the low-fidelity data was assigned as the lower resolution images of 50 by 50 pixels.

To apply GPR, the input and output data needed to be established. The input data in this system are the three θ parameters that control the geometry of each of the spinodoid and the output data is the Young's modulus value of each spinodoid. The GPR model attempts to find a correlation between this data, while also predicting the Young's modulus values based on the θ parameters. In addition, GPR does not need a high amount of data to establish a model, therefore only 10% of the original database of 10,100 samples were utilized. In order to build and test the model, the data needed to be separated into training and testing for both the low- and high-fidelity data. A total of 80% of low-fidelity data and 70% of high-fidelity data was assigned as training data. The remaining percentage from each data set was assigned for testing. The application of GPR was completed in MATLAB. The following formulation describes the process in which GPR was applied to this system. Since calculations would be performed in matrix form, the input and output data of both training and testing are represented by matrices $[\mathbf{X}]$ and $[\mathbf{Y}]$,

$$[\mathbf{X_r}] = \begin{bmatrix} [\mathbf{X}_H] \\ [\mathbf{X}_L] \end{bmatrix}, [Y_r] = \begin{bmatrix} [\mathbf{Y}_H] \\ [\mathbf{Y}_L] \end{bmatrix} \qquad [\mathbf{X_t}] = \begin{bmatrix} [\mathbf{X}_H] \\ [\mathbf{X}_L] \end{bmatrix}, [Y_t] = \begin{bmatrix} [\mathbf{Y}_H] \\ [\mathbf{Y}_L] \end{bmatrix},$$

where $[\mathbf{X_r}]$ is a r by 3 matrix, where r is the number of training samples and the value, 3, represents the three θ values that control the geometry of each spinodoid, and $[\mathbf{Y_r}]$, with size of r by 1. The test samples are represented similarly in that $[\mathbf{X_t}]$ is a t by 3 matrix and $[\mathbf{Y_t}]$ is a t by 1 matrix, where t is the number of test samples. Within the GPR formulation a multi-fidelity covariance needs to be calculated to determine the variance within the data points considered as inputs, these inputs are the θ values that generated each image. The covariance matrix is constructed by plugging the input data into a kernel function. As mentioned, any kernel function can be used, but a rational quadratic kernel function was used in this model. To see other types of kernel functions, see Appendix A. The rational quadratic kernel function takes the form below,

$$K(x,\bar{x}) = \left(1 + \frac{(x-\bar{x})}{2\alpha l^2}\right)^{-\alpha}, \tag{3.8}$$

where α and l are the hyperparameters to be optimized and the x-values are the input values to be used to calculate the covariance [78]. The covariance matrix that holds the calculated values from the kernel function is represented by matrix, $[\bar{\mathbf{K}}]$,

$$[\bar{\mathbf{K}}] = \begin{bmatrix} [\mathbf{K}] & [\mathbf{K}_*] \\ [\mathbf{K}_*]^T & [\mathbf{K}_{**}] \end{bmatrix}, \tag{3.9}$$

where $[\mathbf{K}]$ compares the correlation within the training data of θ values with a size of r x r x p, where p represents the number of input parameters, which takes the value of 3, as there are three θ values that are being varied to generate each spinodoid. $[\mathbf{K}_*]$ compares the correlation between the training and test data with a size of r x t x p, $[\mathbf{K}_*]^T$ is the preceding matrix transposed, and $[\mathbf{K}_{**}]$ compares the correlation between the test data of θ values with a size of t x t x p. Once this covariance matrix is found, the expected values of the output data, which are the Young's modulus values, can be found through,

$$E[\mathbf{Y}_*] = [\mathbf{K}_*]^T[[\mathbf{K}] + \sigma^2[\mathbf{I}]_{rxr}]^{-1}[\mathbf{Y}], \tag{3.10}$$

where σ^2 is the initial estimation of the variance of the system, $E[\mathbf{Y}_*]$ is the expected value of Young's modulus and the matrix, $[\mathbf{Y}]$ includes both the low- and high-fidelity training modulus values. The GPR model is calibrated by optimizing the hyperparameters in Equation A.1 which is completed by minimizing the objective function,

$$f = \left\|\mathbf{E}\left[\mathbf{Y}_*\right] - \mathbf{E}_H\left[\mathbf{Y}_*\right]\right\| + \frac{1}{2}\left\|\mathbf{STD} - \mathbf{STD}_H\right\|, \tag{3.11}$$

where the amount of error in the prediction of the modulus values and associated standard deviation is minimized in comparison to the high-fidelity test points, which portray a more accurate representation of the system. This optimization problem was completed using a genetic algorithm, where the optimum values of σ, α, and l, were found to be as 0.0661, 0.0406, and 4.4396, respectively [60]. The uncertainty of the predictions can also be found simultaneously by,

$$[\Sigma_*] = [\mathbf{K}_{**}] - [\mathbf{K}_*]^T \left[[\mathbf{K}] + \sigma^2 [\mathbf{I}_{nxn}]\right]^{-1} [\mathbf{K}_*] + \sigma^2 [\mathbf{I}_{txt}] \, , \tag{3.12}$$

with this, variance of the output or Young's modulus can be found, from which unit variance can be calculated.

With the above formulation, MATLAB was used to derive the GPR model to represent the relationship between the input of the θ parameters and output of Young's modulus values. Figure 3.9 demonstrates the GPR model predictions plotted with the test data to verify the model [60]. The range of values expanding the predictions represent the uncertainty of the predictions. This uncertainty may arise from pre- and post-processing-related variability, inherent material uncertainty, variations arising from computational models, and any human involvement in the manufacturing. Since the training values of Young's modulus were dominated by values around 1.63 GPa, specifically with an average of 1.635 GPa, the GPR predictions also fall within this range of hovering around 1.63 GPa. When calculating a value to multiply by σ to cover 95% of the data, a value of 3.1σ was found, see Figure 3.10 [60]. Since this value falls within a range of 3σ to 3.5σ it helps further validate that the predictions also follow a Gaussian distribution as the test values follow a Gaussian distribution, shown by Figure 3.11. Figure 3.11 also demonstrates how closely the distribution of the GPR predictions replicates the test data. This help support the idea that the developed

GPR serves as an accurate model to represent the system. With the establishment of the GPR model that is able to link the θ parameters and Young's modulus, inverse design under uncertainty with spinodoids could be approached and is discussed in the following chapter.

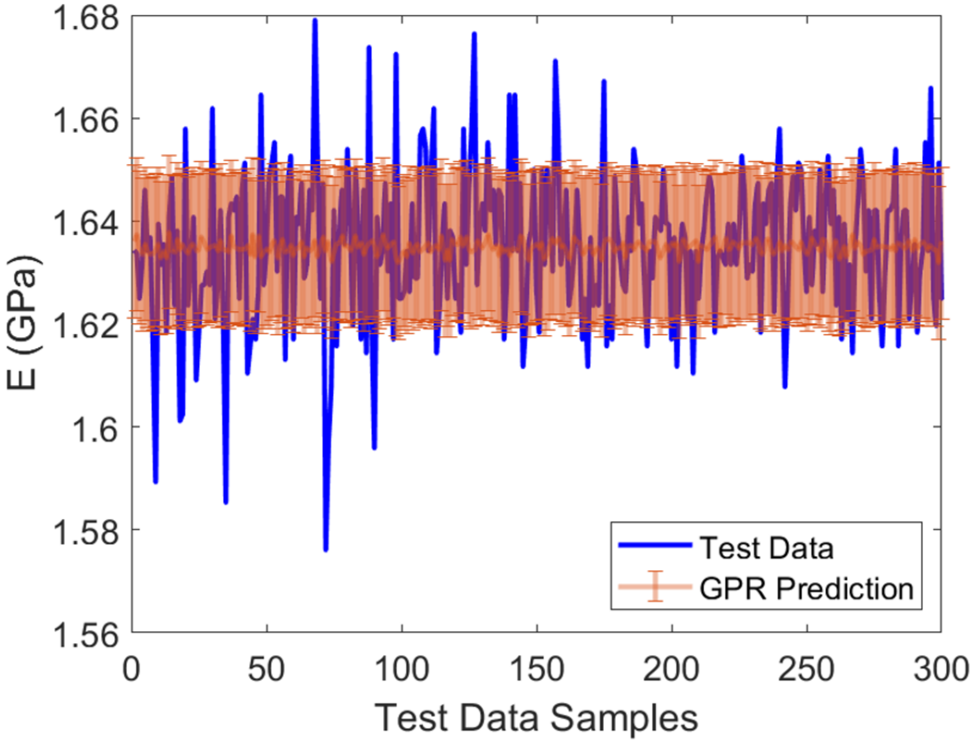


Figure 3.9: Test data compared to the GPR predictions for expected values, this also demonstrates the uncertainty associated with the GPR predictions [60].

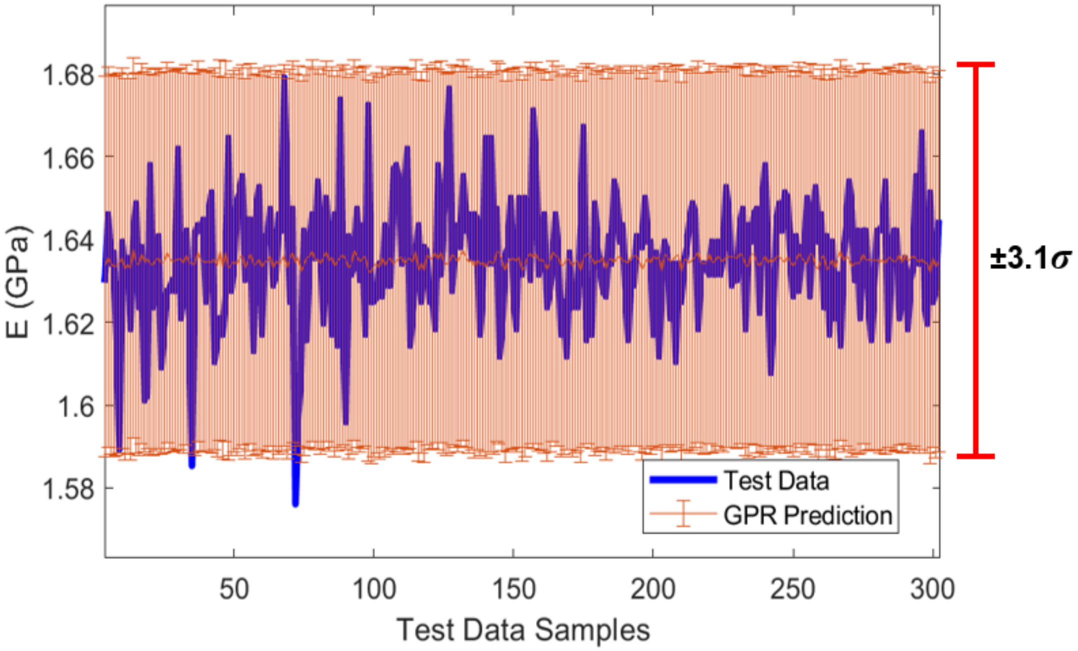


Figure 3.10: A value of ±3.1σ with the GPR predictions covers 95% of the test data [60].

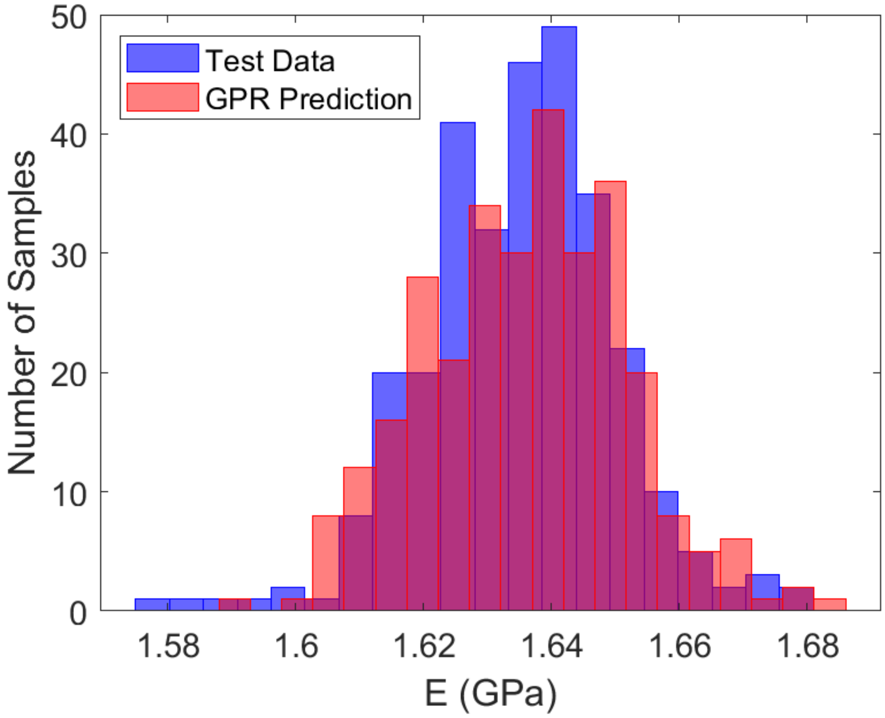


Figure 3.11: Probability distribution functions of the predictions from the GPR model and test data [60].

Chapter 4

Inverse Design of Spinodoids Under Uncertainty

4.1 Inverse Design of Spinodoids using Gaussian Process Regression

The benefit of mechanical metamaterials over that of traditionally manufactured materials is the ability to design the underlying microstructure topology so that the material can elicit specific, desired properties. This capability provides the opportunity to approach the design of mechanical metamaterials inversely, where the properties can first be defined and the material is then constructed based on these set properties. In this study, inverse design is explored with 2D-spinodoid unit cells, with the additional analysis of uncertainty in the resultant design. In the inverse design problems explored in this research, the desired spinodoid is found by determining the three θ parameters that control the spinodoid geometry, as these variables are the inputs that the developed Gaussian process regression (GPR) model is based upon. The desired properties that can be targeted include a specific Young's modulus or the uncertainty in the design. The inverse design problems are approached through the definition of optimization problems. A total of three inverse design problems are defined and analyzed. These three optimization problems focus on generating a spinodoid that satisfies an arbitrarily selected Young's modulus value, minimizing the uncertainty or variance of Young's modulus in the spinodoid design, and a combination of the two where a desired Young's modulus value is set with the additional goal of minimizing the expected uncertainty.

4.2 Defining the Inverse Design Problems

The first design problem aims to find the three θ values that will generate a spinodoid with a desired Young's modulus value. The goal of this optimization problem is to minimize the error between the desired modulus value and the Young's modulus found through the calibrated GPR model created through the process detailed in Chapter 3.6. This optimization problem takes the form of,

$$\begin{aligned} \min \quad & \left\| E\left[Y_*\right] - E_{desired} \right\| \\ E\left[Y_*\right] = & f([\bar{\mathbf{K}}](l, \sigma, \alpha, \theta_1, \theta_2, \theta_3)) \\ \text{s.t.} \quad & 0 \leq \theta_1, \theta_2, \theta_3 \leq \frac{\pi}{2}, \end{aligned} \tag{4.1}$$

where $E[Y_*]$ is the target expected E value that is predicted by the calibrated GPR model whose predictions depend on the multi-fidelity covariance matrix $[\bar{\mathbf{K}}]$ and hyperparameters, l, σ, and α that were optimized to calibrate the model. The θ values selected must fall within the same range used to generate the spinodoid of $0 \leq \theta_1, \theta_2, \theta_3 \leq \pi/2$. These θ values are also the input values that are used as the test values in the formation of the GPR model, which also affect the multi-fidelity covariance matrix. The resultant expected Young's modulus value, E, is compared to the target Young's modulus value. Given that the range of the Young's modulus values used to train the model were dominated by values around 1.63 GPa and the GPR model resulted in a prediction range of 1.63 to about 1.64 GPa, the target Young's modulus value for this initial optimization problem was set to 1.636 GPa [60]. This target Young's modulus value would be utilized in the two remaining inverse design problems. The second inverse design problem focused on producing a spinodoid with a minimum amount of uncertainty associated with the design, therefore minimizing its subsequent unit variance associated with the Young's modulus. This unit variance is interpreted as the variance per number of test samples. This kind of optimization focused on minimizing uncertainty is also known as robust optimization. This optimization problem has a similar form as the first established optimization problem, excluding the objective function,

$$\begin{aligned} \min \quad & \sigma^2[E] \\ \sigma[E]^2 = & f([\bar{\mathbf{K}}](l, \sigma, \alpha, \theta_1, \theta_2, \theta_3)) \\ \text{s.t.} \quad & 0 \leq \theta_1, \theta_2, \theta_3 \leq \frac{\pi}{2}, \end{aligned} \tag{4.2}$$

where $\sigma^2[E]$ is the unit variance of the expected Young's modulus, E. The last inverse problem combines the design goals of the first two, resulting in the goal to find a target Young's modulus, while also minimizing the associated uncertainty in the spinodoid design. This led to the development of a multi-component objective function that is a sum of both previous objective functions. This optimization problem is defined as follows,

$$\begin{aligned} \min \quad & \|E\left[Y_*\right] - E_{desired}\| + \text{STD}[E] \\ E\left[Y_*\right], \sigma^2[E] = & f([\bar{\mathbf{K}}](l, \sigma, \alpha, \theta_1, \theta_2, \theta_3)) \\ \text{s.t.} \quad & 0 \leq \theta_1, \theta_2, \theta_3 \leq \frac{\pi}{2}, \end{aligned} \tag{4.3}$$

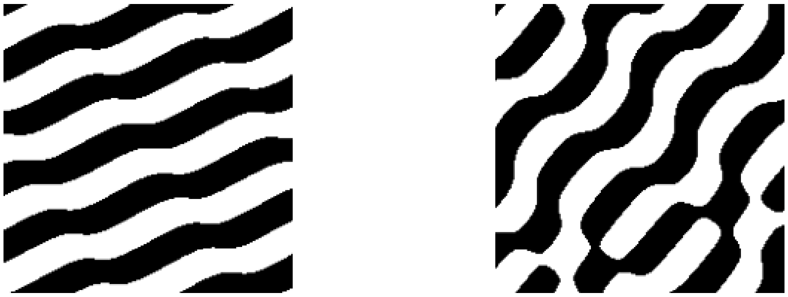
here both goals are considered through the composite objective function. Each of these optimization problems were executed within MATLAB using an interior-point optimization algorithm. In addition, to determine if a global minimum existed for each optimization problem, several initial guesses of θ values were tested for each problem. The same initial guesses were given to each formulated optimization problem. The results of each optimization problem are discussed in detail in the following section.

4.3 Analysis of Optimum Spinodoids Found with Inverse Design

After executing the optimization problems defined in the previous section, the first goal of finding a specific Young's modulus and third goal that is a combination of finding a specific Young's modulus value and minimizing the uncertainty in Young's modulus resulted in multiple possible solutions. The second optimization problem, which focused solely on minimizing the uncertainty of Young's modulus, initially resulted in only one solution, but with further analysis the resultant uncertainty or variance value was consistent with other optimized spinodoids.

The first problem focused on finding a target Young's modulus value of 1.636 GPa. By using an interior-point algorithm, a spinodoid with θ values of 0.850, 0.309, and 0.332 for θ_1, θ_2, and θ_3, respectively, satisfied the target Young's modulus value. After trying another initial guess for the θ parameters, another spinodoid, generated from θ values of 0.827, 1.321, and 0.459 also met the design target for Young's modulus. These spinodoids serve as two options that satisfy the first inverse design problem in targeting a specific Young's modulus value and are displayed in Figure 4.1 [60]. Other initial guesses were tried and the associated optimized spinodoids are summarized in Figure 4.4.

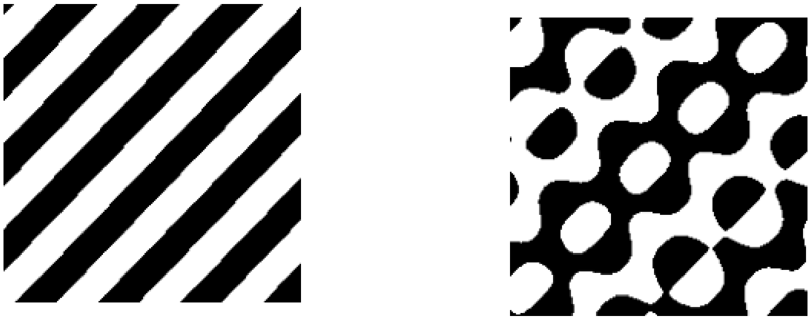
For the second design goal, focused on minimizing the uncertainty of Young's modulus, only



$\theta_1 = 0.850, \theta_2 = 0.309, \theta_3 = 0.332$ $\theta_1 = 0.827, \theta_2 = 1.321, \theta_3 = 0.459$

Figure 4.1: The two spinodoids and their θ values that are solutions to the first inverse design problem of targeting a specific Young's modulus of 1.636 GPa. Figure derived from [60].

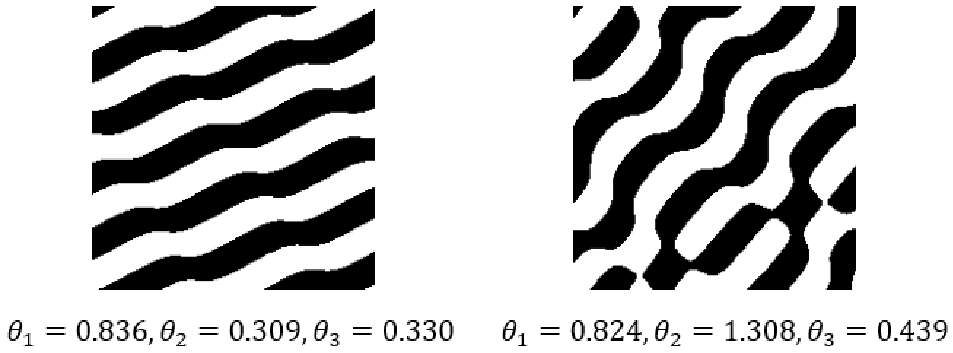
one solution within the bounds of 0 to $\pi/2$ was found after attempting the same initial guesses utilized in the first and last optimization problems. Optimized values for θ_1, θ_2, and θ_3 of 0.827, 0.790, and 0.787, produced a spinodoid with a Young's modulus of 1.636 GPa and a unit variance value of 0.0221 GPa^2 [60]. The resultant simple geometry of this spinodoid compared to other spinodoids, led to the initial conclusion that simple geometry was expected to decrease uncertainty. However, further investigation reveals that this conclusion did not withstand. Once other initial guesses were attempted, outside of the initial guesses used in the other optimization problems, a spinodoid with θ values of 0.794, 1.571, and 0, also satisfied the design goal. See Figure 4.2 to view both of the discussed solutions [60]. Continued closer inspection of the unit variance values given with each step taken during the optimization process, the value of the unit variance centered around 0.0221 GPa^2. This unit variance value was also consistent in the other optimized spinodoids as well. This consistency may indicate that the geometry of the spinodoid may have little effect on the uncertainty of properties. Another reason could be the prediction dependency on the 2D-domain and a future work is recommended to investigate the uncertainty behavior of spinodoids in large 3D-microstructural spaces.



$\theta_1 = 0.827, \theta_2 = 0.790, \theta_3 = 0.787$ $\theta_1 = 0.794, \theta_2 = 1.571, \theta_3 = 0$

Figure 4.2: The solutions found for the second inverse design problem focused on minimizing the uncertainty of the spinodoid. Figure derived from [60].

The last optimization problem also resulted in multiple solutions. With the same two initial guesses used for the initial optimization problem, the first optimum solution with θ values of 0.836, 0.309, and 0.330 satisfied the goal of targeting a Young's modulus value and minimizing uncertainty. The spinodoid forming from θ values of 0.824, 1.308, and 0.439 for θ_1, θ_2, and θ_3, was the second found optimum solution [51]. Both of these solutions are displayed in Figure4.3. Analyzing the multiple solutions that resulted from this last design problem, where the goals included targeting a Young's modulus value and minimizing uncertainty, the geometry of the optimum spinodoids were very similar to that of the solutions found for the initial optimization problem, where the focus was just on targeting a specific Young's modulus value. This may indicate that the first design goal in targeting a specific Young's modulus value is dominating the objective function in this multi-objective optimization problem, since the unit variance values do not change between solutions, given by the analysis of the second optimization problem. The average objective function values of each optimization problem was also analyzed. The first objective function had an average value of 4.9491e-12 GPa, while the second and third average values increased in magnitude of 0.0221 GPa^2 and 0.0111 GPa, respectively. Figure 4.4 demonstrates a summary of all of the optimized spinodoids given the trial of four different initial guesses for each inverse design problem, plus that of the second solution found for the second optimization problem, resulting from an initial guess outside of the other four guesses.



$\theta_1 = 0.836, \theta_2 = 0.309, \theta_3 = 0.330$ $\theta_1 = 0.824, \theta_2 = 1.308, \theta_3 = 0.439$

Figure 4.3: The two spinodoids and their θ values that are solutions to the last defined inverse design problem with the multi-objective goal in targeting a specific Young's modulus value and minimizing uncertainty. Figure derived from [60].

Figure 4.4 summarizes that more than one spinodoid can be chosen as solutions for each optimization problem. These multiple solutions may be explained by Figure 4.5, which demonstrates that the Young's modulus values used in training the GPR model lie within a range of 1.5484 to 1.6922 GPa, with many values lying within the range of about 1.62 to about 1.64 GPa, with an average of 1.6350 GPa. Furthermore the resultant expected Young's modulus values used in finding the optimum spinodoids lie in the reduced range of 1.6318 to 1.6379 GPa, with an average of 1.6349 GPa. With the target Young's modulus value used in the optimization problems of 1.636 GPa, this study aimed to demonstrate the capability of obtaining multiple optimum solutions, as shown in Figure 4.4. These multiple options provide a potential benefit, as it gives the flexibility to choose more than one design over the other if other considerations such as ease of manufacturing or possibly specific aesthetics come into play.

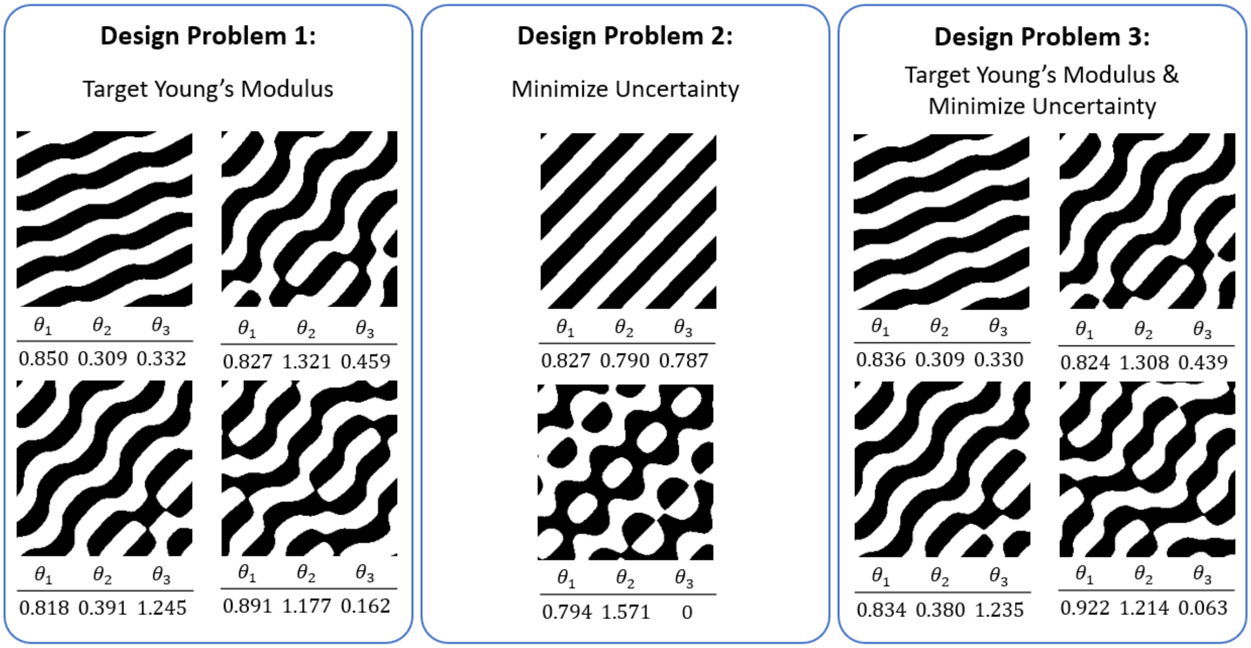


Figure 4.4: Summary of solutions from all the inverse design problems approached with optimization problems [60].

4.4 Summary

In this study, spinodoids, a subset of mechanical metamaterials were further investigated over that of truss-based mechanical metamaterials, due to their known benefits over this popularly known sub-group. Spinodoids are less susceptible to high stress concentrations due to their smoother topology and are less prone to symmetry breaking defects because of their aperiodicity [5]. These materials were studied within the 2D-space by first generating a large database of 2D-monochrome images representing individual spinodoid cells. Following the formulation defined by Equations 3.1 and 3.2, the images were generated within MATLAB. These spinodoids were then characterized by their Young's modulus through density-based pixel calculations. Once characterized by this mechanical property, data-driven modeling techniques were then applied to derive a relationship between the spinodoid geometry or the θ parameters controlling the geometry and their calculated Young's modulus value. Initially,

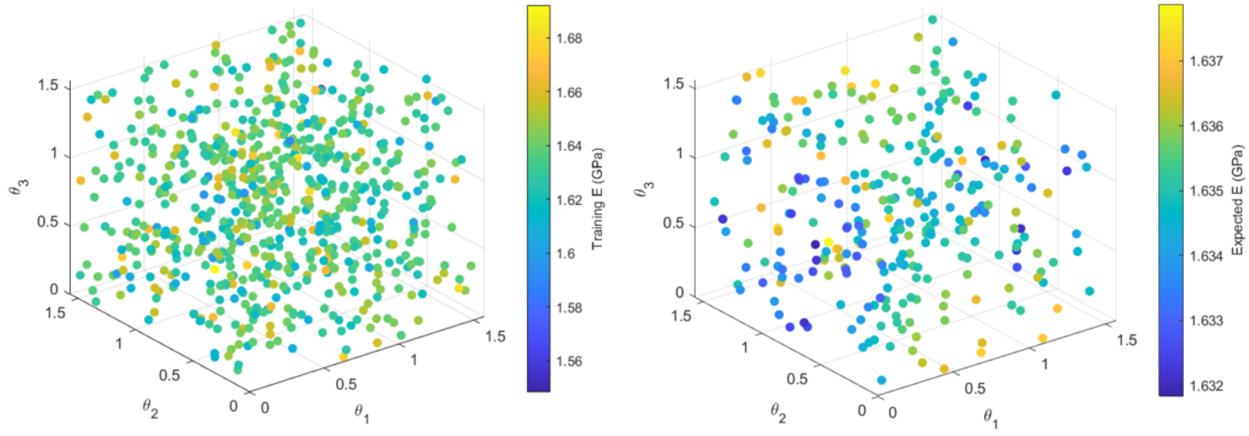


Figure 4.5: The θ parameters plotted against the Young's modulus values used to develop the GPR model for the optimization problems (left) and θ parameters plotted against the expected or predicted Young's modulus values resulting from the developed GPR model (right) [60].

a principal component analysis was applied in an attempt to reduce the dimensionality of the image-based data. Given that this method did not result in a significant enough decrease in the needed dimensions to represent the data, another data-driven modeling technique was applied. GPR was then applied given its added benefit of simultaneous uncertainty quantification with its calculation of predictions. A GPR model was derived by using a multi-fidelity approach and further utilized to approach several inverse design problems. These inverse design problems were defined as three optimization problems. One had the objective of finding a target Young's modulus value, the second focused on minimizing uncertainty with the design, and the last combined these goals to create a multi-objective optimization problem. What was critical about the results of these inverse design problems was result of multiple optimum spinodoid solutions [60]. Having a choice between which optimum design to select from prevents a potential designer from being locked into a certain design, which may not satisfy other manufacturing or cost constraints. The overall process in this study of spinodoids is summarized by Figure 4.6. After developing a method to perform inverse design with spinodoids, the graphical user interface developed to design spinodoids in Chapter 3.5 could be expanded upon by also offering the option for a user to derive a specific spinodoid that satisfies their design goals. These design goals can be defined by specific mechanical properties, such as a target Young's modulus value or uncertainty considerations. This consideration of design tools keeps in mind that more tools focused specifically on the design of mechanical metamaterials need to be developed in order for industry to understand and utilize their many potential capabilities [3].

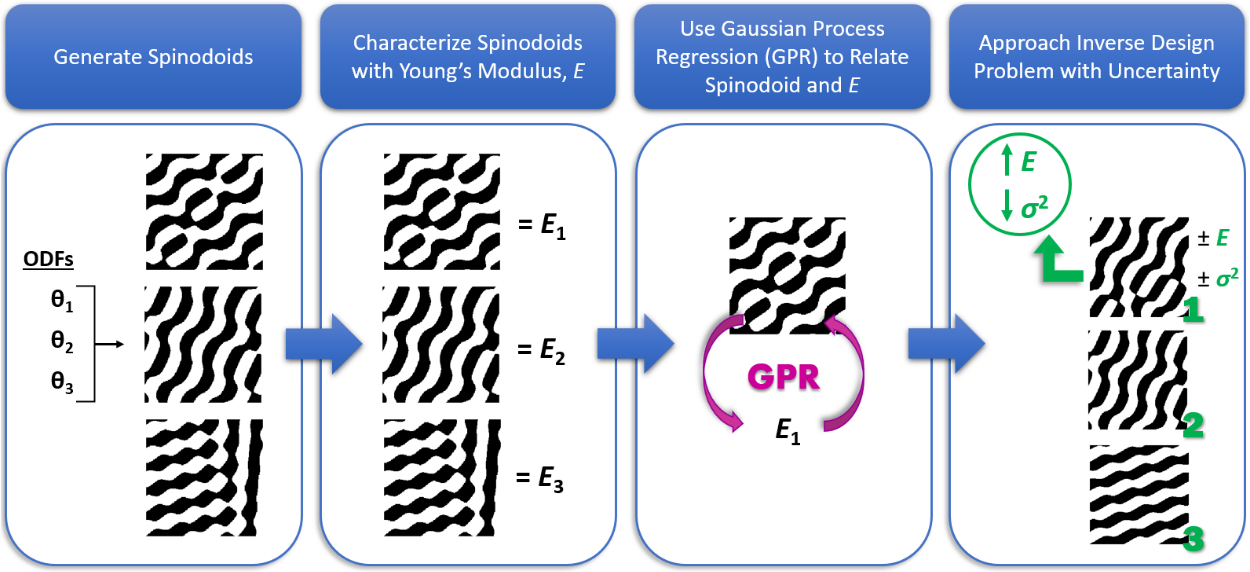

Generate Spinodoids
Characterize Spinodoids with Young's Modulus, E
Use Gaussian Process Regression (GPR) to Relate Spinodoid and E
Approach Inverse Design Problem with Uncertainty
ODFs
θ_1
θ_2
θ_3
$= E_1$
$= E_2$
$= E_3$
GPR
E_1
$\uparrow E$
$\downarrow \sigma^2$
$\pm E$
$\pm \sigma^2$
1
2
3

Figure 4.6: Summary demonstrating the research focused on spinodoids. Figure derived from [60].

Chapter 5

Conclusion

5.1 Future Work

Although two different kinds of mechanical metamaterials were studied in this research, truss-based and spinodoids, spinodoids were chosen to be studied in greater detail. Due to their smoother and aperiodic topologies, they are less prone to high stress concentrations and symmetry-breaking defects prevalent in truss-based mechanical metamaterials [5], [31]. Therefore, further interest in how this study of spinodoids can be expanded upon was explored, particularly in how the characterization of these spinodoids could be improved. In the characterization process, a density-based pixel calculation was used to obtain the Young's modulus of each generated spinodoid. Although, the presented formulation has been utilized in other peer-reviewed research [61], [62], the similarity in the resulting Young's modulus values between each image, could provide an indication that the characterization of these materials should also consider the effects of the underlying topology in addition to the volume fraction of the solid and void phases.

5.1.1 Analyzing Characterization Methods

Another factor to consider when characterizing each spinodoid could be the consideration of the shapes making up each spinodoid. This consideration may allow for a more unique and explicit characterization for each spinodoid within the data set. To aid in quantifying the shapes, the method of shape moment invariants could be applied. The calculation of shape moments allows an image to be quantified in terms of moment values that are invariant to rotation, scaling, or translation [79], [80]. The method of calculating moment invariants was first utilized in recognizing patterns [81]. These calculated moment invariants are defined as Hu moments [81]. Initial research on applying shape moment invariants to the generated spinodoids is detailed in Appendix B.

In addition to characterizing the shapes effects using moment invariants, the mechanical properties such as Young modulus should also be computed with a higher fidelity approach than the density-based model. Therefore the Young's modulus values calculated through the density-based pixel calculations need to be verified through finite element analysis (FEA) to determine if the found range of Young's modulus values through these calculations is sufficient in determining the Young's modulus of the spinodoids. As well as verifying the calculated Young's modulus, directional properties could also be found by performing FEA. These FEA simulations can be completed by transferring the 2D-geometry of a spinodoid to a 2D-solid part within Abaqus or other FEA software. Ideas on how the geometry can be transferred to Abaqus are visualized in Figure 5.1. These ideas show that the eventual part within Abaqus can be constructed as one solid part with a thin frame in order to simplify the application of loads and boundary conditions, or the part could be constructed of two phases, solid and void. In this second idea, the void phase is assigned material properties close to zero, such as 10^-9 GPa for Young's modulus, similarly done in the density-based pixel calculations for the void material. Refer to Appendix C for suggestions on how to utilize Abaqus to determine mechanical properties of a spinodoid unit cell.

Ideally, all of the Young's modulus values within the data set of images would be calculated through FEA simulations and compared with the density-based, pixel calculations,

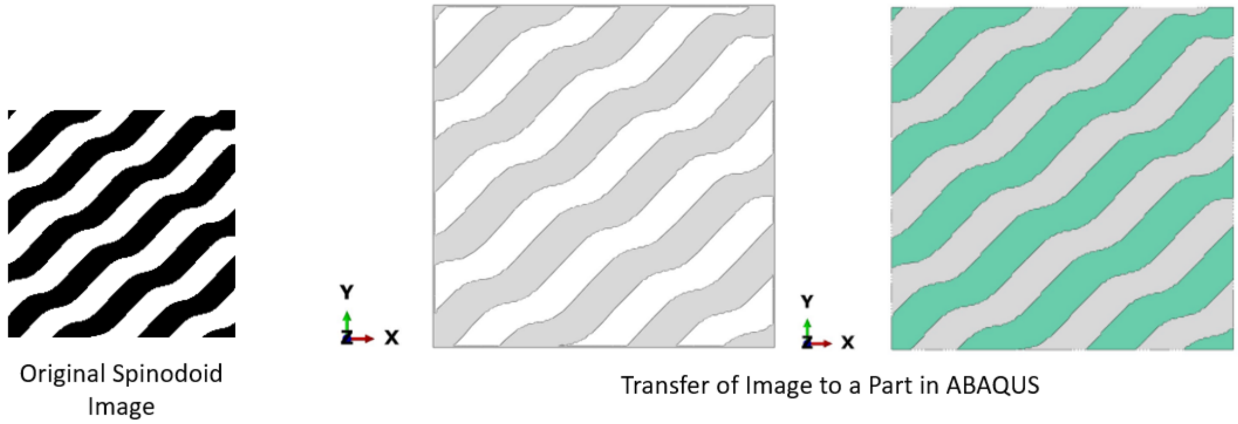


Figure 5.1: Ideas on how to transfer a spinodoid image to a part in Abaqus. The left part demonstrates a solid material with true void sections, whereas the right part is made of two materials, one material has solid property values (green) and the other material has very low material properties to closely represent true void (gray). Figure derived from [60].

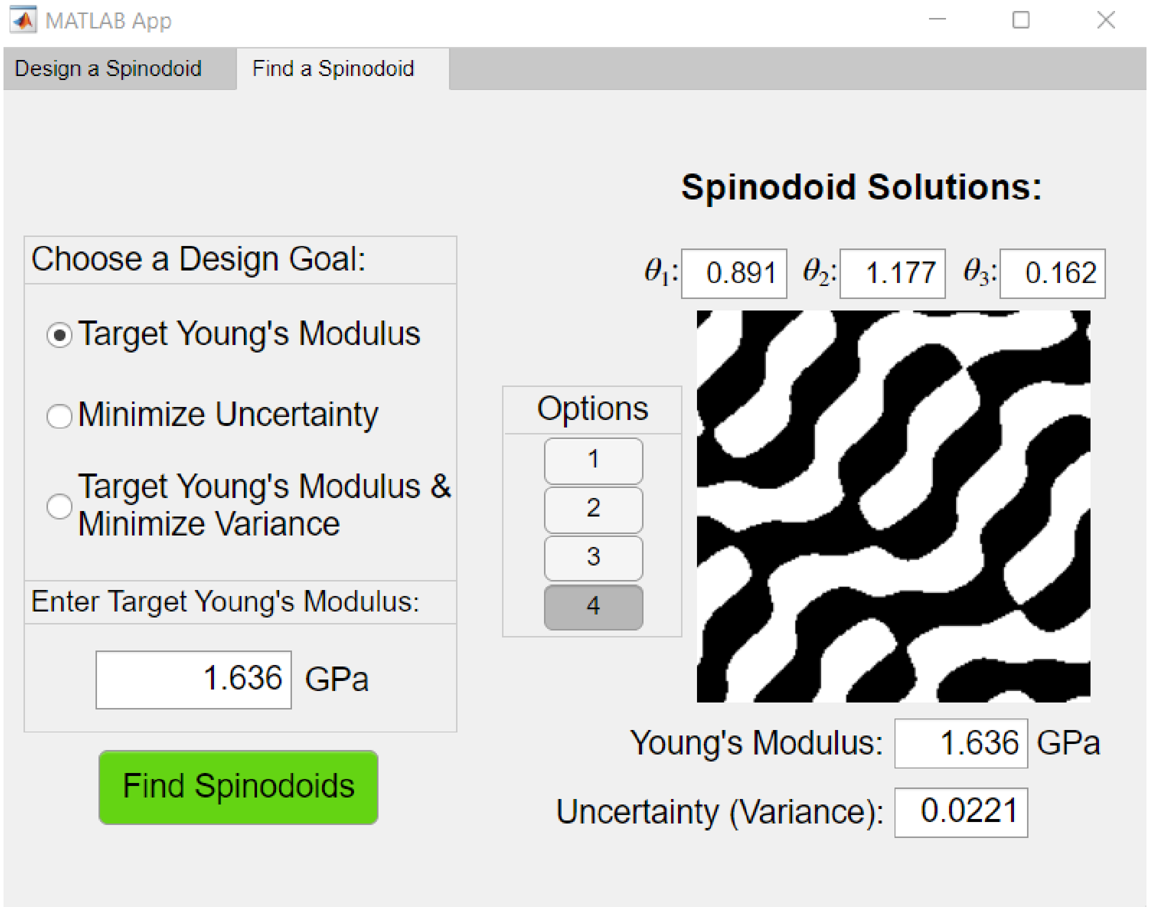
but performing the FEA of 10,100 spinodoids through Abaqus would be computationally expensive. Therefore, with a study by Chan *et. al*, only a small, yet diverse database of unit cells are needed to develop sufficient machine learning techniques to model relationships between unit cells and properties [46]. Therefore a machine learning model that relates the Young's modulus and the spinodoid could be developed by running FEA simulations on a smaller subset of spinodoids containing specifically chosen diverse geometries [46]. This chosen subset of spinodoids could be selected visually or through the analysis of their calculated shape moment values. The shape moment values would be compared against one another to determine which spinodoids have the most variation from one another. If the proposed machine learning model provides higher fidelity predictions for the mechanical properties, it can be used to replace the density-based model to compute the mechanical properties of spinodoids.

5.1.2 Development of a Graphical User Interface for Inverse Design of Spinodoids

Although a graphical user interface (GUI) was developed in Chapter 3.5 that allows a user to design a spinodoid, after establishing the process of approaching inverse design in Chapter 4, this GUI could be expanded upon by incorporating this inverse design process as another feature of the GUI. This inverse design tool would allow a user to input the desired properties they would like a spinodoid to elicit. In terms of this research, their design goals could be finding a specific Young's modulus, minimizing the uncertainty related to the design, or a combination of both. With their chosen design goal they would then need to enter their desired Young's modulus value and the GUI would then determine the three θ values that would generate a spinodoid that satisfies their design requirements. This inverse design problem would be approached using the same method presented in Chapter 4 where an optimization problem would be executed and multiple solutions would result using an interior-point algorithm. The spinodoids resulting from this GUI would then be displayed as potential options from which they could select from. See Figure 5.2 for a mock-up in how the addition to this GUI could be designed.

5.1.3 3D-Modeling of Mechanical Metamaterials

Although this research focused primarily on the design of two classes of mechanical metamaterials within the 2D-space, to provide a further accurate analysis of these materials they would need to be considered in three dimensions. See Figure 5.3 on the theorized 3D-models of one generated truss-based unit cell and one generated spinodoid unit cell from this research. This may allow for a more holistic idea on how specifically the studied mechanical metamaterials could be applied to different real-world applications. Given that 3D modeling



Choose a Design Goal:

● Target Young's Modulus

○ Minimize Uncertainty

○ Target Young's Modulus & Minimize Variance

Enter Target Young's Modulus: 1.636 GPa

θ_1: 0.891 θ_2: 1.177 θ_3: 0.162

Young's Modulus: 1.636 GPa

Uncertainty (Variance): 0.0221

Figure 5.2: Mock-up for a potential future addition to GUI developed in Chapter 3.5, this addition would add the capability of performing inverse design for spinodoids. Figure derived from [60].

significantly affects the computational time in modeling and calculations of properties, the work completed within the 2D-space could serve as a step towards establishing a multi-scale model that could be utilized in exploring the design and property space in 3D. Using a lower-fidelity model may also assist with efficient, further characterization of other aspects of these mechanical metamaterials as well. Further investigation of these materials could include observing their behavior under temperature effects, long-term loading, and determining their behavior under failure.

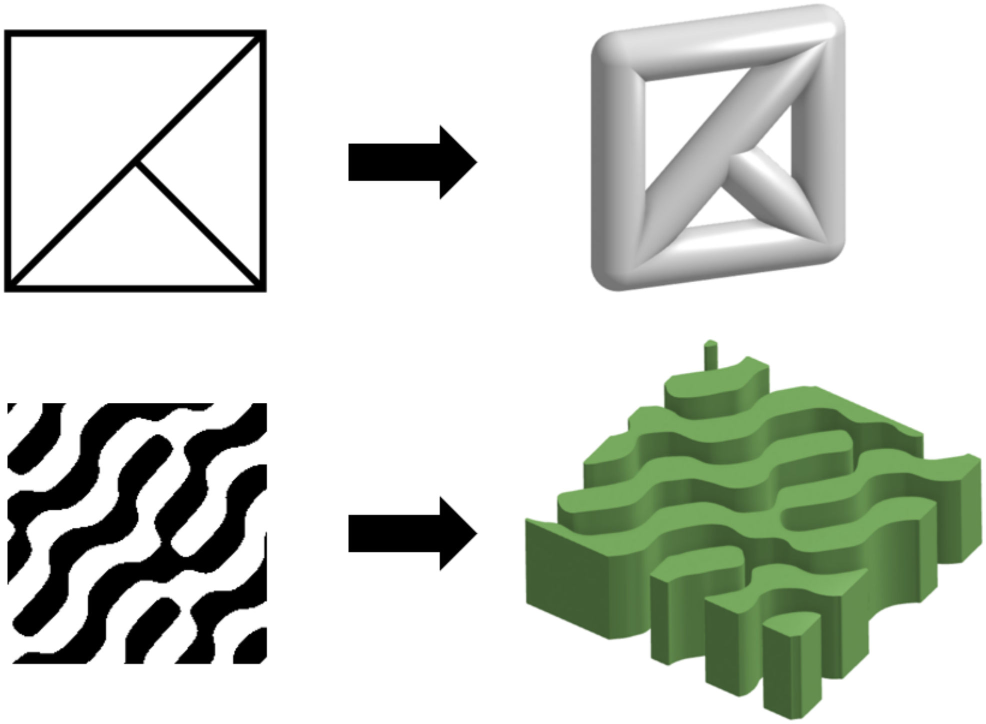


Figure 5.3: 3D-representations of one truss-based geometry from the database of 12 microstructures developed and one spinodoid unit cell from the generated data set of 10,100 spinodoid images. Figure derived from [51] and [60].

5.2 Concluding Remarks

In this research two kinds of mechanical metamaterials were investigated. The first type are truss-based, whose geometry is based on lattices made of truss-like structures. These mechanical metamaterials are well-known within this area of material research, with potential applications including aerospace, medical, and construction fields [23], [25], [30]. These materials were studied by investigating how the deformation behavior of a mechanical beam component changed based on the varying topology made of these microstructures. A database of 12 2D-truss-based structures was developed. To analyze how these structures would affect the displacement behavior of a simple cantilever beam, finite element analysis (FEA) was applied to the beam. In order to apply FEA, the nodes and elements of the structures and eventually the entire beam needed to be assigned. Initially, these assignments were manually determined for each of the 12 structures. With this method, an optimization problem was developed to determine the topology of the beam in order to minimize the displacement resulting from a force acting across the entire beam. The results of this problem indicated that the first method developed to model the beam for FEA, only allowed for a separate FEA of each individual microstructure within the beam. Although, this method considered the sensitivity of each structure, it did not consider how one section of the beam, containing one microstructure, would affect next section in the beam, containing another microstructure. Therefore, a new method in modeling the beam such that FEA could be performed on it as one entire component needed to be developed. In this method a standard grid was defined, using this grid every structure within the database could be assembled. Since each microstructure was made of a set of triangles, each microstructure was defined through a path of coordinates that trace each triangle in the structure. This allowed for a library containing the reference coordinates of each microstructure to be called upon when constructing the beam component made of multiple microstructures. This method allowed for the automatic assignment of nodes and elements, and their connections, which led to capability to run a FEA on the mechanical beam component as a whole, no matter the combination of the 12 microstructures used in its construction. This modeling led to the development of graphical user interface (GUI) that allows a user to build a beam component with any of the 12 microstructures and understand how any external tension or compression force can affect the overall displacement of the beam. The development of this GUI helps with the idea that new design tools need to be created to allow for a greater understanding in the behavior and capabilities of mechanical metamaterials [3]. With further development, the same optimization problem executed with the initial modeling method could be incorporated into the developed GUI, which could instead determine the topology of the beam to minimize its displacement given an external force. The incorporation of this inverse design problem would be an added benefit to the developed design tool. The research performed on truss-based mechanical metamaterials led to a better understanding of these materials, yet with the known factors that these materials contain sharp corners and their properties are highly dependent on their periodicity, making them susceptible to high stress concentrations and symmetry-breaking defects, a relatively new kind of mechanical metamaterial based on the process of spinodal decomposition was studied [5], [31].

The second kind of mechanical metamaterial investigated in this research is made of unit cells generated from following the phase separation process of spinodal decomposition. This thermodynamic, phase transformation process leads to two distinct phases that coexist together [32]. The microstructures resulting from following this process have smooth and aperiodic geometry, which makes them less prone to the discussed susceptibilities of truss-based mechanical metamaterials. Through research from Kumar and co-workers, the spinodoids are defined as geometries where one phase is assigned as solid and the other phase as void [5]. They modeled the process that these structures are simulated from through a simplification of the Cahn-Hilliard equation which serves as an analytical solution to spinodal decomposition [5], [31], [55]. Through three θ values, the geometry of the spinodoid can be controlled. In this research, this subset of mechanical metamaterials is studied within the 2D-space. To study these materials, spinodoids were generated by representing unit cells with monochrome images. In these images, black represented solid material and white represented void. A total of 10,000 50 by 50 images were developed and later subsidized by 100 200 by 200 images. In order to understand the behavior of the generated spinodoids, the Young's modulus for each spinodoid was found through density-based pixel calculations. In order to relate the spinodoid geometry to the calculated modulus values, two different kinds of data-driven modeling techniques were applied to the characterized data set. Principal component analysis (PCA) was the first method applied. With the application of PCA, the dimensionality of the 10,000 50 by 50 images (2500 dimensions) could only be reduced to 450 principal components (PCs) to represent 90% of the original image data. Through further filtering 80% of the data could be represented by 18 PCs. Although, the dimensions were significantly reduced from 2500, this high number of PCs would still be difficult to model and utilize in determining the relationship between the PCs and calculated Young's modulus values of each spinodoid. The second data-driven technique applied to the data set was Gaussian process regression (GPR). Compared to PCA, this method had the added benefit of simultaneously calculating uncertainty associated with its eventual predictions. In this research a multi-fidelity model was used, where the 10,000 50 by 50 images were considered as low-fidelity data and the 100 200 by 200 images were considered as high-fidelity data. This model would help define a relationship between the parameters controlling the geometry of the spinodoid, θ_1, θ_2, and θ_3, and the calculated Young's modulus values through the development of a covariance matrix constructed from the training data, a rational-quadratic kernel function, and optimized hyperparameters. Using this developed GPR model several inverse design problems were defined and solved through gradient-based optimization. Multiple solutions resulted for each defined problem, indicating that there is more than one solution in targeting a specific Young's modulus and minimizing uncertainty. These multiple options offer choice in the design, therefore the selection of a spinodoid to satisfy a certain design is not constrained to one specific geometry. To expand on this research, the characterization methods would like to be further investigated through further quantification methods such as shape quantifiers, like Hu moments [81]. Additionally, the calculated Young's modulus values would like to be compared to FEA simulations. With these additions, a more unique and verified model between the controlling parameters of θ and the Young's modulus could be established. This improved model could help further improve the inverse design problems to later be incorporated into the GUI that allows a user to develop spinodoids. This added capability would again serve as another design tool that allows others to understand

and design unique mechanical metamaterials. These kinds of design tools need to be developed in order to help further the realization of these materials in industry and real-world applications. Mechanical metamaterials have many potential applications, inverse design is a unique design approach that can be applied to develop mechanical metamaterials with outstanding mechanical properties. The methods developed in this research to study the design of mechanical metamaterials are aimed to contribute to the ongoing research for their design and development in different engineering fields.

Appendices

Appendix A

Types of Kernel Functions

A.1 Rational Quadratic Kernel Function

This rational quadratic kernel function was used in the development of the Gaussian process regression (GPR) model. The rational quadratic kernel function takes the form below,

$$K(x, \bar{x}) = \sigma^2 \left(1 + \frac{(x - \bar{x})}{2\alpha l^2}\right)^{-\alpha}, \tag{A.1}$$

where σ^2 the initial estimation of the variance of the system, α and l, are the hyperparameters to be optimized and the x-values are the input data [78].

A.2 Constant Value Kernel Function

A more simple kernel function where a constant value is solved for is,

$$K(x, \bar{x}) = \sigma^2 c, \tag{A.2}$$

where σ^2 is the variance of the training data or hyperparameter to be optimized and c is the constant value to be solved for, and the x-values are the input data [82].

A.3 Squared Exponential Kernel Function

Squared exponential kernel function is defined as,

$$K(x,\bar{x}) = \sigma^2 \text{exp}\left(-\frac{(x-\bar{x})^2}{2l^2}\right), \tag{A.3}$$

where σ^2 is the initial estimation for variance or hyperparameter to be optimized, l is another hyperparameter to be optimized, and the x-values are the input data [78].

A.4 Periodic Kernel Function

Periodic kernel function is defined as,

$$K(x,\bar{x}) = \sigma^2 \text{exp}(\frac{-2\text{sin}(\pi\left|x-\bar{x}\right|/\rho)}{l^2}), \tag{A.4}$$

where σ^2 is the initial estimation for variance or hyperparameter to be optimized, ρ and l are additional hyperparameters to be optimized, and the x-values are the input data [78].

Appendix B

Potential Application of Shape Moment Invariants

The topology of the spinodoids can be characterized using the shape moment invariants that are defined in terms of Hu moments. Although shape invariants involve seven different Hu moment functions, the first moment value is found to capture sufficient information of different microstructural topologies [83]. Therefore, in this analysis only the first moment, ϕ_1, is used to quantify the topology of spinodoids. Initially to determine whether each individual θ value had an effect on the resulting spinodoid, three sets of spinodoids were generated. The first set of 50 200 by 200 images was generated by varying θ_1, and keeping θ_2 and θ_3 constant. The remaining two sets were generated by varying θ_2 orθ_3 and keeping the remaining θ parameters constant. Figure B.1 demonstrates how the ϕ_1 values varied with each varying θ value, along with how the ϕ_1 values varied with the calculated Young's modulus of the generated spinodoids in the each set. These Young's modulus values were also found by using the density-based pixel calculations detailed in Chapter 3.4. Initially it appears that there could a relationship between these moment values and the θ parameters, but to verify this a larger data set of 1,000 50 by 50 images were generated that varies θ_1. Figure B.2 shows a random ϕ_1 distribution for varying θ_1 values. The figure also indicates that there is no explicit relationship between ϕ_1 and Young's modulus values. This analysis demonstrates that the potential relationships seen in Figure B.1 does not continue, and the

ϕ becomes more randomly distributed with varying θ_1 and that there is no clear relationship between ϕ_1 and Young's modulus.

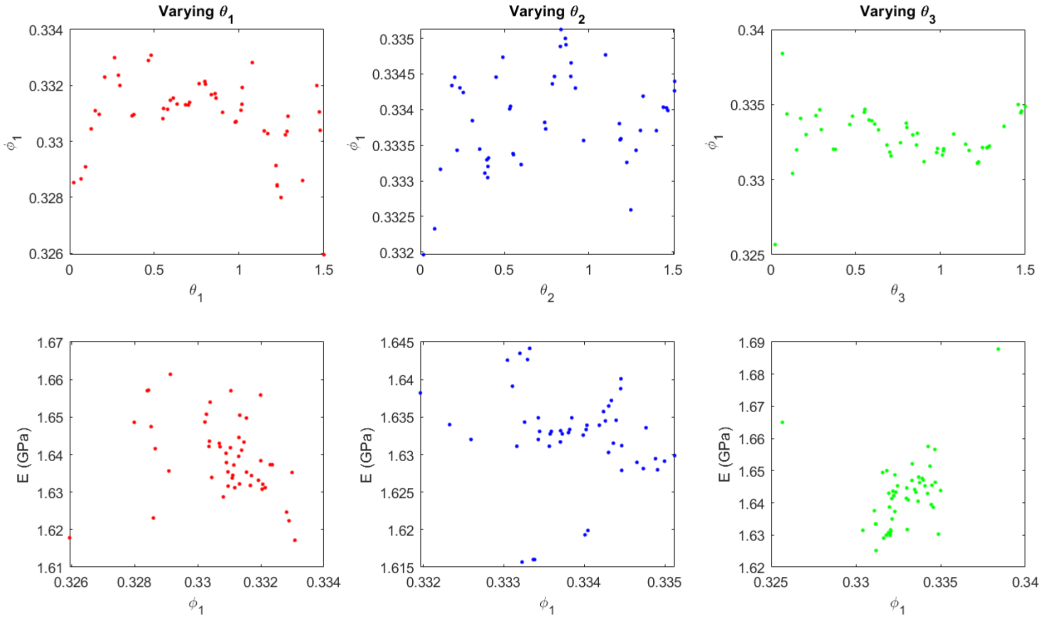


Figure B.1: Each varying θ parameter plotted against the calculated shape moment value of ϕ_1 (top). Young's modulus values for each spinodoid set are compared to the calculated ϕ_1 values (bottom).

The moment values were also calculated for all of the 50 by 50 pixel images of spinodoids generated, a total of 10,000 images. Within this data set none of the θ are kept constant, therefore all θ values are randomly varied. With this larger data set it appears that a large amount of spinodoids circles around one particular value. Figure B.3 demonstrates that with the entire data set of 10,000 images, the first moment value (ϕ_1) is centered around a Young's modulus value of about 1.63 GPa. Although these ϕ_1 values circle around one particular Young's modulus value and have a particular range within itself as well, incorporating the values of ϕ_1 along with the density-based pixel calculations may allow for a more unique characterization of each spinodoid. The similar ϕ_1 may also indicate that many of the spinodoids within the data set have similar geometries as well. One observation from this

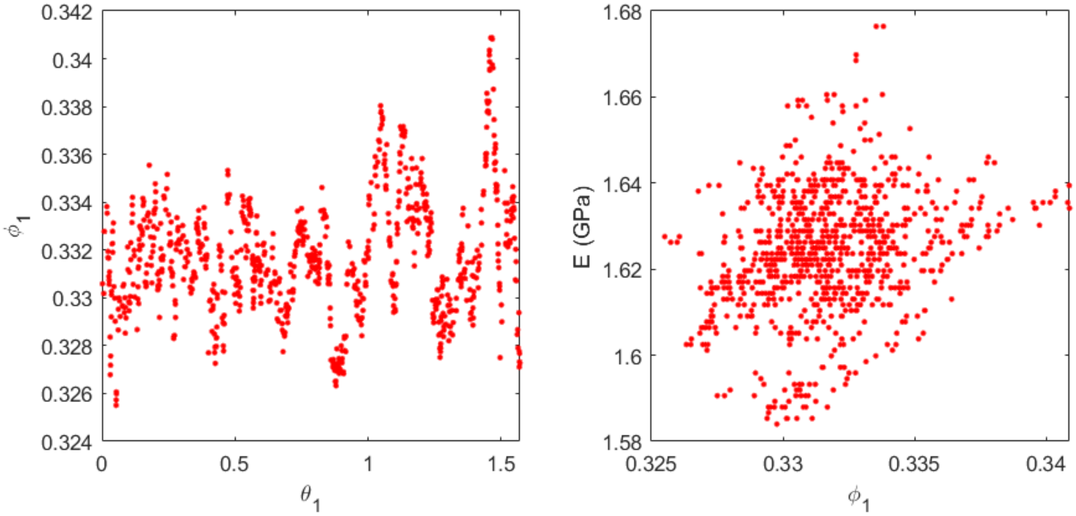


Figure B.2: Larger set of 1,000 50 by 50 images with spinodoids generated by varying θ_1, the left figure shows ϕ_1 plotted against θ_1 and the right figure demonstrates Young's modulus plotted against the calculated ϕ_1 values.

initial analysis of shape moment invariants, was the somewhat differing ranges of ϕ_1 from images with different resolutions. As the moment invariants should not be sensitive to image-scaling, this observation should be further investigated.

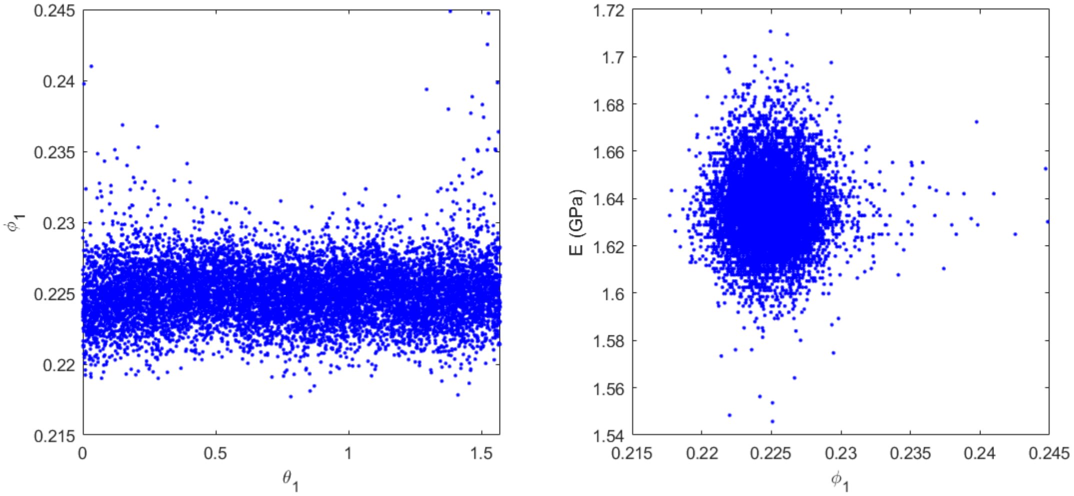


Figure B.3: Largest set of 10,000 50 by 50 images generated, the left figure shows ϕ_1 plotted against θ_1 and the right figure demonstrates Young's modulus plotted against the calculated ϕ_1 values.

Appendix C

Potential Application of Finite Element Analysis Software

Utilizing external finite element analysis (FEA) software such as Abaqus can help determine the validity or verify the Young's modulus calculations found through a density-based pixel formulation. Abaqus can run FEA simulations of this 2D-solid to determine the desired properties. By simulating a tensile test on the spinodoid unit cell, the average stress and strain can be found through Abaqus. Another method to find material properties could be through the application of an Abaqus plug-in, called EasyPBC [84]. An exemplary mesh generated in Abaqus that could be utilized in future simulations is shown in Figure C.1. By running such FEA simulations or utilizing other Abaqus tools, the density-based computations can be verified for the studied database of spinodoids. The FEA can also be used to develop surrogate models with the integration of the shape moment invariants.

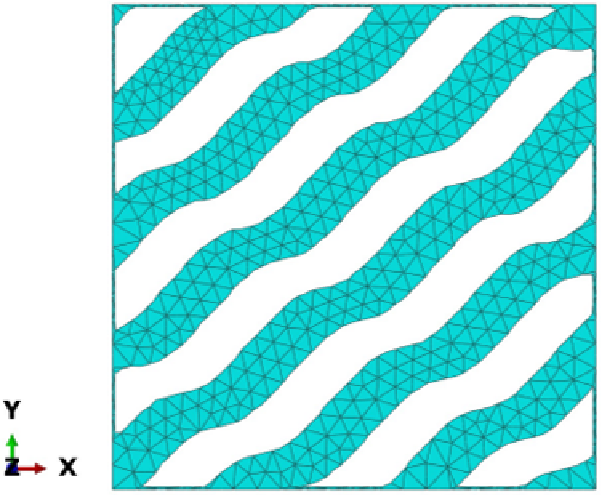


Figure C.1: The mesh file created for a spinodoid transferred to Abaqus.

Bibliography

[1] J. U. Surjadi, L. Gao, H. Du, X. Li, X. Xiong, N. X. Fang, and Y. Lu, "Mechanical metamaterials and their engineering applications," *Advanced Engineering Materials*, vol. 21, no. 3, p. 1800864, 2019.

[2] N. Yang, Y. Deng, S. Zhao, Y. Song, J. Huang, and N. Wu, "Mechanical metamaterials with discontinuous and tension/compression-dependent positive/negative poisson's ratio," *Advanced Engineering Materials*, p. 2100787, 2021.

[3] S. C. Fischer, L. Hillen, and C. Eberl, "Mechanical metamaterials on the way from laboratory scale to industrial applications: challenges for characterization and scalability," *Materials*, vol. 13, no. 16, p. 3605, 2020.

[4] A. Valipour, M. H. Kargozarfard, M. Rakhshi, A. Yaghootian, and H. M. Sedighi, "Metamaterials and their applications: an overview," *Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications*, p. 1464420721995858, 2021.

[5] S. Kumar, S. Tan, L. Zheng, and D. M. Kochmann, "Inverse-designed spinodoid metamaterials," *npj Computational Materials*, vol. 6, no. 1, pp. 1–10, 2020.

[6] M. Kadic, G. W. Milton, M. van Hecke, and M. Wegener, "3d metamaterials," *Nature Reviews Physics*, vol. 1, no. 3, pp. 198–210, 2019.

[7] Z. He and S. D. Guest, "On rigid origami ii: quadrilateral creased papers," *Proceedings of the Royal Society A*, vol. 476, no. 2237, p. 20200020, 2020.

[8] H. Yasuda and J. Yang, “Reentrant origami-based metamaterials with negative poisson’s ratio and bistability,” *Physical review letters*, vol. 114, no. 18, p. 185502, 2015.

[9] W. Wu, W. Hu, G. Qian, H. Liao, X. Xu, and F. Berto, “Mechanical design and multifunctional applications of chiral mechanical metamaterials: A review,” *Materials & Design*, vol. 180, p. 107950, 2019.

[10] Y. Tang, Y. Li, Y. Hong, S. Yang, and J. Yin, “Programmable active kirigami metasheets with more freedom of actuation,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 52, pp. 26407–26413, 2019.

[11] Z. Zhai, Y. Wang, and H. Jiang, “Origami-inspired, on-demand deployable and collapsible mechanical metamaterials with tunable stiffness,” *Proceedings of the National Academy of Sciences*, vol. 115, no. 9, pp. 2032–2037, 2018.

[12] D. J. N. Amorim, T. Nachtigall, and M. B. Alonso, “Exploring mechanical meta-material structures through personalised shoe sole design,” in *Proceedings of the ACM Symposium on Computational Fabrication*, pp. 1–8, 2019.

[13] E. Barchiesi, M. Spagnuolo, and L. Placidi, “Mechanical metamaterials: a state of the art,” *Mathematics and Mechanics of Solids*, vol. 24, no. 1, pp. 212–234, 2019.

[14] H. Kolken, K. Lietaert, T. van der Sloten, B. Pouran, A. Meynen, G. Van Loock, H. Weinans, L. Scheys, and A. A. Zadpoor, “Mechanical performance of auxetic meta-biomaterials,” *journal of the mechanical behavior of biomedical materials*, vol. 104, p. 103658, 2020.

[15] A. Rafsanjani, K. Bertoldi, and A. R. Studart, “Programming soft robots with flexible mechanical metamaterials,” *Science Robotics*, vol. 4, no. 29, p. eaav7874, 2019.

[16] Q. Liu, “Literature review: materials with negative poisson’s ratios and potential applications to aerospace and defence,” 2006.

[17] S. Li and K. Wang, “Fluidic origami: a plant-inspired adaptive structure with shape morphing and stiffness tuning,” *Smart Materials and Structures*, vol. 24, no. 10, p. 105031, 2015.

[18] P. Bhovad, J. Kaufmann, and S. Li, “Peristaltic locomotion without digital controllers: Exploiting multi-stability in origami to coordinate robotic motion,” *Extreme Mechanics Letters*, vol. 32, p. 100552, 2019.

[19] S. Babaee, S. Pajovic, A. R. Kirtane, J. Shi, E. Caffarel-Salvador, K. Hess, J. E. Collins, S. Tamang, A. V. Wahane, A. M. Hayward, *et al.*, “Temperature-responsive biometamaterials for gastrointestinal applications,” *Science translational medicine*, vol. 11, no. 488, p. eaau8581, 2019.

[20] Y. Wang and D. L. McDowell, “Uncertainty quantification in materials modeling,” in *Uncertainty Quantification in Multiscale Materials Modeling*, pp. 1–40, Elsevier, 2020.

[21] R. N. Glaesener, C. Lestringant, B. Telgen, and D. M. Kochmann, “Continuum models for stretching-and bending-dominated periodic trusses undergoing finite deformations,” *International Journal of Solids and Structures*, vol. 171, pp. 117–134, 2019.

[22] L. R. Meza, G. P. Phlipot, C. M. Portela, A. Maggi, L. C. Montemayor, A. Comella, D. M. Kochmann, and J. R. Greer, “Reexamining the mechanical property space of three-dimensional lattice architectures,” *Acta Materialia*, vol. 140, pp. 424–432, 2017.

[23] J.-H. Bastek, S. Kumar, B. Telgen, R. N. Glaesener, and D. M. Kochmann, “Inverting the structure–property map of truss metamaterials by deep learning,” *Proceedings of the National Academy of Sciences*, vol. 119, no. 1, 2022.

[24] C. Pan, Y. Han, and J. Lu, “Design and optimization of lattice structures: A review,” *Applied Sciences*, vol. 10, no. 18, p. 6374, 2020.

[25] Q. Liu, R. Xu, Y. Zhou, J. Ge, S. Yuan, Y. Long, and T. Shi, “Metamaterials mapped lightweight structures by principal stress lines and topology optimization: Methodology, additive manufacturing, ductile failure and tests,” *Materials & Design*, vol. 212, p. 110192, 2021.

[26] L. Mizzi and A. Spaggiari, “Lightweight mechanical metamaterials designed using hierarchical truss elements,” *Smart Materials and Structures*, vol. 29, no. 10, p. 105036, 2020.

[27] T. Kim, C. Zhao, T. Lu, and H. Hodson, “Convective heat dissipation with lattice-frame materials,” *Mechanics of Materials*, vol. 36, no. 8, pp. 767–780, 2004.

[28] D. M. Dykstra, S. Janbaz, and C. Coulais, “The extreme mechanics of viscoelastic metamaterials,” *arXiv preprint arXiv:2204.01375*, 2022.

[29] L. Shu, R. Liang, Y. Yu, T. Tian, Z. Rao, and Y. Wang, “Unique elastic, dielectric and piezoelectric properties of micro-architected metamaterials,” *Journal of Materials Chemistry C*, vol. 7, no. 9, pp. 2758–2765, 2019.

[30] Y.-B. Wang, H.-T. Liu, and T.-J. Li, “Novel beam-like mechanical metamaterials with different flexural rigidities in two directions,” *Composite Structures*, vol. 267, p. 113857, 2021.

[31] L. Zheng, S. Kumar, and D. M. Kochmann, “Data-driven topology optimization of spinodoid metamaterials with seamlessly tunable anisotropy,” *Computer Methods in Applied Mechanics and Engineering*, vol. 383, p. 113894, 2021.

[32] J. S. Langer, *Spinodal Decomposition*, pp. 19–42. Boston, MA: Springer US, 1975.

[33] V.-N. Tran Duc and P. K. Chan, “Using the cahn–hilliard theory in metastable binary solutions,” *ChemEngineering*, vol. 3, no. 3, 2019.

[34] C. M. Portela, A. Vidyasagar, S. Krödel, T. Weissenbach, D. W. Yee, J. R. Greer, and D. M. Kochmann, “Extreme mechanical resilience of self-assembled nanolabyrinthine materials,” *Proceedings of the National Academy of Sciences*, vol. 117, no. 11, pp. 5686–5693, 2020.

[35] G. Oliveri and J. T. Overvelde, “Inverse design of mechanical metamaterials that undergo buckling,” *Advanced Functional Materials*, vol. 30, no. 12, p. 1909033, 2020.

[36] O. Rokoš, M. M. Ameen, R. H. Peerlings, and M. G. Geers, “Micromorphic computational homogenization for mechanical metamaterials with patterning fluctuation fields,” *Journal of the Mechanics and Physics of Solids*, vol. 123, pp. 119–137, 2019.

[37] J. Morris, W. Wang, D. Shah, T. Plaisted, C. J. Hansen, and A. V. Amirkhizi, “Expanding the design space and optimizing stop bands for mechanical metamaterials,” *Materials & Design*, vol. 216, p. 110510, 2022.

[38] P. Callahan, J. Simmons, and M. De Graef, “A quantitative description of the morphological aspects of materials structures suitable for quantitative comparisons of 3d microstructures,” *Modelling and Simulation in Materials Science and Engineering*, vol. 21, no. 1, p. 015003, 2012.

[39] V. Sundararaghavan, “Reconstruction of three-dimensional anisotropic microstructures from two-dimensional micrographs imaged on orthogonal planes,” *Integrating Materials and Manufacturing Innovation*, vol. 3, no. 1, pp. 240–250, 2014.

[40] L. Ai and X.-L. Gao, “Topology optimization of 2-d mechanical metamaterials using a

parametric level set method combined with a meshfree algorithm," *Composite Structures*, vol. 229, p. 111318, 2019.

[41] Y. Wang, J. Gao, Z. Luo, T. Brown, and N. Zhang, "Level-set topology optimization for multimaterial and multifunctional mechanical metamaterials," *Engineering Optimization*, vol. 49, no. 1, pp. 22–42, 2017.

[42] M. Ye, L. Gao, F. Wang, and H. Li, "A novel design method for energy absorption property of chiral mechanical metamaterials," *Materials*, vol. 14, no. 18, p. 5386, 2021.

[43] H. Ronellenfitsch, N. Stoop, J. Yu, A. Forrow, and J. Dunkel, "Inverse design of discrete mechanical metamaterials," *Physical Review Materials*, vol. 3, no. 9, p. 095201, 2019.

[44] J. Yao, Y. Su, F. Scarpa, and Y. Li, "An optimization approach to design deformation patterns in perforated mechanical metamaterials using distributions of poisson's ratio-based unit cells," *Composite Structures*, vol. 281, p. 115015, 2022.

[45] A. Challapalli, D. Patel, and G. Li, "Inverse machine learning framework for optimizing lightweight metamaterials," *Materials & Design*, vol. 208, p. 109937, 2021.

[46] Y.-C. Chan, F. Ahmed, L. Wang, and W. Chen, "Metaset: Exploring shape and property spaces for data-driven metamaterials design," *Journal of Mechanical Design*, vol. 143, no. 3, 2021.

[47] R. Kulagin, Y. Beygelzimer, Y. Estrin, A. Schumilin, and P. Gumbsch, "Architectured lattice materials with tunable anisotropy: Design and analysis of the material property space with the aid of machine learning," *Advanced Engineering Materials*, vol. 22, no. 12, p. 2001069, 2020.

[48] L. Wu, L. Liu, Y. Wang, Z. Zhai, H. Zhuang, D. Krishnaraju, Q. Wang, and H. Jiang, "A

machine learning-based method to design modular metamaterials," *Extreme Mechanics Letters*, vol. 36, p. 100657, 2020.

[49] J. Zheng, Z. Luo, H. Li, and C. Jiang, "Robust topology optimization for cellular composites with hybrid uncertainties," *International Journal for Numerical Methods in Engineering*, vol. 115, no. 6, pp. 695–713, 2018.

[50] J. Wu, Z. Luo, H. Li, and N. Zhang, "Level-set topology optimization for mechanical metamaterials under hybrid uncertainties," *Computer Methods in Applied Mechanics and Engineering*, vol. 319, pp. 414–441, 2017.

[51] K. McMillan and P. Acar, "Database development and component design with two-dimensional trusslike microstructures," *AIAA Journal*, pp. 1–6, 2021.

[52] Q. Chen, L. Liu, C. Zhu, and K. Chen, "Mesomechanical modeling and numerical simulation of the diffraction elastic constants for ti6al4v polycrystalline alloy," *Metals*, vol. 8, no. 10, p. 822, 2018.

[53] K. L. McMillan and P. Acar, "Database development and topology optimization for 3d printed microstructures," in *AIAA Scitech 2021 Forum*, p. 1965, 2021.

[54] X. Zheng, X. Guo, and I. Watanabe, "A mathematically defined 3d auxetic metamaterial with tunable mechanical and conduction properties," *Materials & Design*, vol. 198, p. 109313, 2021.

[55] J. W. Cahn, "Phase separation by spinodal decomposition in isotropic systems," *The Journal of Chemical Physics*, vol. 42, no. 1, pp. 93–99, 1965.

[56] M. N. Lee and A. Mohraz, "Bicontinuous macroporous materials from bijel templates," *Advanced Materials*, vol. 22, no. 43, pp. 4836–4841, 2010.

[57] F. Findik, "Improvements in spinodal alloys from past to present," *Materials & Design*, vol. 42, pp. 131–146, 2012.

[58] S. Goryachev, "Theory of spinodal decomposition," *Physical review letters*, vol. 72, no. 12, p. 1850, 1994.

[59] C. P. Grant, "Spinodal decomposition for the cahn-hilliard equation," *Communications in Partial Differential Equations*, vol. 18, no. 3-4, pp. 453–490, 1993.

[60] K. L. McMillan, D. S. Öztürk, and P. Acar, "Inverse design of 2d-mechanical metamaterials with spinodal topologies under uncertainty," in *AIAA SCITECH 2022 Forum*, p. 0811, 2022.

[61] L. Wang, Y.-C. Chan, Z. Liu, P. Zhu, and W. Chen, "Data-driven metamaterial design with laplace-beltrami spectrum as "shape-dna"," *Structural and multidisciplinary optimization*, vol. 61, no. 6, pp. 2613–2628, 2020.

[62] L. Xia and P. Breitkopf, "Design of materials using topology optimization and energy-based homogenization approach in matlab," *Structural and multidisciplinary optimization*, vol. 52, no. 6, pp. 1229–1241, 2015.

[63] P. Christöfl, C. Czibula, M. Berer, G. Oreski, C. Teichert, and G. Pinter, "Comprehensive investigation of the viscoelastic properties of pmma by nanoindentation," *Polymer Testing*, vol. 93, p. 106978, 2021.

[64] F. J. Montáns, F. Chinesta, R. Gómez-Bombarelli, and J. N. Kutz, "Data-driven modeling and learning in science and engineering," *Comptes Rendus Mécanique*, vol. 347, no. 11, pp. 845–855, 2019.

[65] K. Choudhary, K. F. Garrity, A. C. Reid, B. DeCost, A. J. Biacchi, A. R. Hight Walker, Z. Trautt, J. Hattrick-Simpers, A. G. Kusne, A. Centrone, *et al.*, "The joint automated

repository for various integrated simulations (jarvis) for data-driven materials design," *npj Computational Materials*, vol. 6, no. 1, pp. 1–13, 2020.

[66] P. Acar, "Crystal plasticity model calibration for ti-7al alloy with a multi-fidelity computational scheme," *Integrating Materials and Manufacturing Innovation*, vol. 7, no. 4, pp. 186–194, 2018.

[67] M. Hasan and P. Acar, "Machine learning reinforced microstructure-sensitive prediction of material property closures," *Computational Materials Science*, p. 110930, 2021.

[68] T. Xue, A. Beatson, M. Chiaramonte, G. Roeder, J. T. Ash, Y. Menguc, S. Adriaenssens, R. P. Adams, and S. Mao, "A data-driven computational scheme for the nonlinear mechanical properties of cellular mechanical metamaterials under large deformation," *Soft matter*, vol. 16, no. 32, pp. 7524–7534, 2020.

[69] P. Acar, "Machine learning reinforced crystal plasticity modeling under experimental uncertainty," *AIAA Journal*, vol. 58, no. 8, pp. 3569–3576, 2020.

[70] A. Paul, P. Acar, W.-k. Liao, A. Choudhary, V. Sundararaghavan, and A. Agrawal, "Microstructure optimization with constrained design objectives using machine learning-based feedback-aware data-generation," *Computational Materials Science*, vol. 160, pp. 334–351, 2019.

[71] M. Hermkes, N. M. Kuehn, and C. Riggelsen, "Simultaneous quantification of epistemic and aleatory uncertainty in gmpes using gaussian process regression," *Bulletin of earthquake engineering*, vol. 12, no. 1, pp. 449–466, 2014.

[72] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[73] J. Lever, M. Krzywinski, and N. Altman, "Points of significance: Principal component analysis," *Nature methods*, vol. 14, no. 7, pp. 641–643, 2017.

[74] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[75] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of Mathematical Psychology*, vol. 85, pp. 1–16, 2018.

[76] J. Wang, "An intuitive tutorial to gaussian processes regression," *arXiv preprint arXiv:2009.10862*, 2020.

[77] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*, pp. 63–71, Springer, 2003.

[78] A. Wilson and R. Adams, "Gaussian process kernels for pattern discovery and extrapolation," in *International conference on machine learning*, pp. 1067–1075, PMLR, 2013.

[79] J. MacSleyne, J. Simmons, and M. De Graef, "On the use of 2-d moment invariants for the automated classification of particle shapes," *Acta Materialia*, vol. 56, no. 3, pp. 427–437, 2008.

[80] A. Senthilnathan and P. Acar, "Shape moment invariants as a new methodology for uncertainty quantification in microstructures," in *AIAA Scitech 2021 Forum*, p. 1697, 2021.

[81] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE transactions on information theory*, vol. 8, no. 2, pp. 179–187, 1962.

[82] T. Beckers, “An introduction to gaussian process models,” *arXiv preprint arXiv:2102.05497*, 2021.

[83] A. Senthilnathan and P. Acar, “Multi-scale modeling for texture and grain topology of polycrystalline microstructures under uncertainty,” in *AIAA SCITECH 2022 Forum*, p. 2106, 2022.

[84] S. L. Omairey, P. D. Dunning, and S. Sriramula, “Development of an abaqus plugin tool for periodic rve homogenisation,” *Engineering with Computers*, vol. 35, no. 2, pp. 567–577, 2019.