Numerical Studies of a Superelastic Nickel-Titanium Rhombic Dodecahedron Structure Using the Finite Element Method

Ian Patrick Morrissey
Marquette University

NUMERICAL STUDIES OF A SUPERELASTIC NICKEL-TITANIUM RHOMBIC DODECAHEDRON STRUCTURE USING THE FINITE ELEMENT METHOD

By Ian P. Morrissey, B.S.

A Thesis submitted to the Faculty of the Graduate School, Marquette University, in Partial Fulfillment of the Requirements for the Degree of Master of Science in Mechanical Engineering

Milwaukee, WI

May 2022

ABSTRACT
NUMERICAL STUDIES OF A SUPERELASTIC NICKEL-TITANIUM RHOMBIC DODECAHEDRON STRUCTURE USING THE FINITE ELEMENT METHOD

Ian P. Morrissey, B.S.

Marquette University, 2022

Energy dissipation is an important material property for materials used in applications such as armor, airplane wings, and automotive vehicle crumple zones. superelastic Nickel-Titanium (NiTi) and compliant under-dense materials both have excellent energy dissipation properties. Current research suggests that compliant under-dense materials made of superelastic NiTi have desirable energy dissipation properties. A rhombic dodecahedron Lattice Structured Material (LSM) is an example of a compliant under-dense Material which has potential to exhibit desirable energy dissipation properties when manufactured from superelastic NiTi. In this work, finite element modeling of a superelastic NiTi rhombic dodecahedron Lattice Structured Material is performed and an optimum for energy dissipation based solely on geometric modification is found.

ACKNOWLEDGEMENTS

Ian P. Morrissey, B.S.

I would like to acknowledge the funding and support I received from Marquette University Graduate School and Dr. Moore that was used to complete this research. I would like to acknowledge my advisor Dr. Moore for his truly tested and seemingly unlimited patience and support, and for the advice and the challenge I was provided throughout my experience in academia thus far. I would like to thank Dr. Murray for her continued moral and academic support, advice, and many enjoyable conversations. I would like to acknowledge Dr. Zhou for taking the time out of his busy schedule and going out on a limb to serve on my committee.

I would like to acknowledge Dr. Rice and Dr. Schimmels for their patience, advice, and support. I am looking forward to working under your advisory.

I would like to acknowledge Jacob Rusch and Allison Goetz whose support has been immeasurable and whose friendship I cherish deeply. I would like to acknowledge my friend Collin Shale for listening to me when everything seemed impossible. I would like to acknowledge Dr. Bowman, Tim Fair, and Waqar Khan for their advice and many engaging conversations. I would like to acknowledge Dr. Park for saying hello and asking why I am still here every time I see him, despite him knowing exactly why I am still here.

I would like to acknowledge my father, Christopher, my mother, Kathryn, my sister, Hannah, my brother, Alexander, and my sister-in-law, Echo, for their continued support. I would like to acknowledge my Aunt Meghan and Uncle Bob for their support and for providing me with feedback. I would like to acknowledge Rocco, Chris, Tom, and Mike for being lifelong friends. I would like to acknowledge everyone else who has helped me along the way.

The support I have received from friends, family, faculty, and colleagues has been immeasurable. Immeasurable is a word I have used too many times to describe the support, and still not enough to describe how much I appreciate it. I hope to never take for granted all of these truly wonderful and patient individuals. I hope I have not forgotten anyone, but I know I will never forget the kindness that others have shown me. It takes a village.

DEDICATION

I would like to dedicate this thesis to the teachers that believed in me when I did not and who provided an ear when I needed it most:

Mr. Nieman - Math

Ms. Caduto - Physics

Mr. Dupuis - Chemistry

Mr. Cleland – Design

Dr. Myslinski - Scripture Class & Philosophy Club Coordinator

Mr. Sneed - Guidance Counselor

Paul Caldwell - Choir Director

I would also like to dedicate this to my late friend Max Marischen. We all miss you. You were my brother, and you always will be.

"I love deadlines, I love the whooshing noise they make as they go by." – Douglas Adams

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SMA – Shape Memory Alloy

SME – Shape Memory Effect

SEE – Superelastic Effect

UDM – Under-dense material

LSM – Lattice Structured Material

SLM – Selective Laser Melting

EBM – Electron Beam Melting

NiTi – Nickel-Titanium

DNS – Direct Numerical Simulation

INTRODUCTION

Under-dense material (UDM) describes material in which deliberate void space exists within a volume of a given material which may or may not be filled with other material. UDM take several forms and names like cellular materials, metal foams, and Lattice Structured Materials (LSM) [1], [2]. The use of UDM for impact absorption and energy dissipation is well documented with uses such as armors [3] and airplane wings [1]. A LSM is a porous material usually made up of periodic cells of structural members. In [2], Messner describes them as akin to the beams and trusses of bridges and skyscrapers, but at a reduced scale. These materials are advantageous as they reduce the mass density of a given volume of a material and allow for the modification and customization of the mechanical behavior. LSMs can be manufactured through a variety of processes such as wire-weaving [4] and additive manufacturing processes like Selective Laser Melting (SLM) [5] and Electron Beam Melting (EBM)[6]. Bending dominated LSMs are compliant structures whose structural members are dominated by bending behavior [7]. A rhombic dodecahedron lattice is a bending dominated lattice made up of 12 rhombic faces for which the edges are the structural members of the LSM [8].

A Shape-Memory Alloy (SMA) is a class of material characterized by its unique mechanical property the Shape Memory Effect (SME) [9]–[11]. SME is the property of shape memory alloys in which a specimen of a SMA appears to be plastically deformed and, upon heating, that deformation is recovered. The Superelastic Effect (SEE) is another mechanical property that is characteristic of certain SMAs and can be described as the large, elastically recoverable deformation that occurs during a solid-state, “stress-induced” [10, p. 176] phase change. Nickel-Titanium (NiTi) is the nearly equiatomic nickel and titanium shape memory alloy. NiTi is also commonly referred to as Nitinol, as it is a NiTi alloy that was researched at the Naval Ordnance Laboratory (NOL) [12]. Superelastic NiTi wire is known for its damping properties as shown in [13]. Superelastic NiTi can accommodate strains in excess of 8%, making it a considerably compliant material relative to other metal alloys as shown in [9],[13].

SMA UDMs have been shown to be capable of providing desirable energy dissipation properties given the strain accommodation that can be provided with UDMs and SMAs as illustrated in [4]. A superelastic NiTi rhombic dodecahedron LSM could provide considerable energy dissipation properties due to the compliant mechanical behavior of both superelastic NiTi and the rhombic dodecahedron lattice structure.

Unit cell modeling of a periodic UDM is a computationally efficient means of analysis for generalizing the behavior of a UDM to a larger global structure. However, there are some limitations to unit cell modeling, namely higher stiffness than modeling an entire structure. The geometry and base materials of LSMs can be modified to in order to optimized the LSMs to a chosen application as noted in [2]. Adjusting the cross section of the members parallel and perpendicular to loading while keeping the relative density and volume of the unit cell constant may allow for better material performance, such as increased capacity for energy dissipation while maintaining the periodicity of the LSM. Adjusting the relative density of the structure will show how the energy dissipated changes with an increase in mass density. Finding a geometric optimum for maximum energy dissipation using a finite element analysis of a SMA LSM unit cell can be computationally expensive and time intensive for large numbers of simulations. Utilizing simplified beam element models allows for reduced computational time so that an optimal geometry for the LSM unit cell can be found. This thesis will find such an optimum using simplified beam element models of a superelastic NiTi rhombic dodecahedron LSM. It will provide the background of the aforementioned subjects, the methodology of these studies, and the results of the studies with pertinent discussion of geometric optimums for energy dissipation of a superelastic NiTi rhombic dodecahedron LSM.

BACKGROUND

I. Shape Memory Alloy Background

As noted earlier, SMA is a classification of metal alloys that exhibit the shape memory effect, a thermo-mechanical property. The SMA NiTi has been the subject of numerous studies, some of which will be outlined here. NiTi exhibits the SME as well as the SEE depending on temperature and alloy composition. The SME is the property of shape memory alloys in which a specimen of a SMA appears to be plastically deformed and, upon heating, that deformation is recovered. SEE is the property of some SMAs in which large recoverable deformations, typically uncharacteristic of metals, occur due to a "stress-induced" [10, p. 176] phase change [9]–[11]. Figure 1 shows typical stress-strain curves of the (a) SME with loading in blue and unloading in red with heat induced strain recovery in green and (b) SEE with loading in blue and unloading in red. These plots were created with color and labels for clarity based on the plots in [8, Fig. 1].

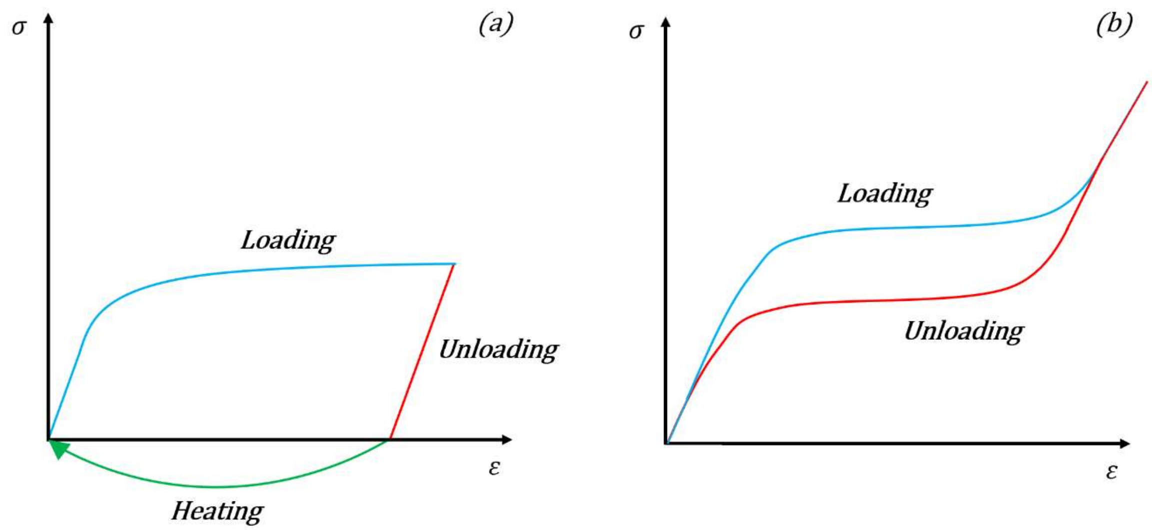

Figure 1: Typical stress-strain curves of the (a) SME and (b) SEE based on the plots in [8, Fig. 1].

Solid-state phase change are responsible for the SME and SEE in NiTi. Martensite is the lower temperature phase of NiTi and austenite is the higher temperature, "parent" [9, p. 230] phase of NiTi [9]. Martensite has the monoclinic crystalline structure $B19'$ [14]. Austenite has the crystalline structure $B2$ [14]. Unstressed low-temperature martensitic NiTi has several variants of martensite whose "crystallographic equivalence" [9, p. 230] induces twinning of related martensite variants which accommodates any transformation strain caused by phase transformation from austenite to martensite upon cooling[9, p. 230].

When stressed sufficiently, all the twinned martensite variants detwin, accommodating strain inelastically. This provides the appearance of plastic deformation. The inelastically deformed, low-temperature detwinned martensitic NiTi transforms to austenite upon heating and the inelastic deformation is recovered. When the NiTi cools, the austenite transforms to the low-temperature martensite phase and the martensite retwins [9]. This is the basis for the SME for which Shape Memory Alloys are named, however, this effect will not be considered in this thesis.

If NiTi is loaded in the high temperature austenite phase, NiTi transitions to a single-variant martensite due to transformation strain induced by stress. When a critical stress is reached during loading, the austenite begins to transform to single-variant martensite. This transformation during loading can be seen in the plateau during loading in Figure 1(b). During unloading, the “stress-induced” [10, p. 176] single variant martensite transforms to austenite. This can be seen in Figure 1(b), represented by the lower plateau during unloading. Single-variant martensite transforms entirely to austenite when unloaded entirely. This is the basis for the SEE [9]. The stress-strain response of the SEE of NiTi is utilized in this thesis.

II. Superelasticity Material Model

From the Abaqus Theory Manual on Superelasticity [15], the Abaqus Material Model Library contains a superelastic material model that is based on the uniaxial stress-strain response of superelastic NiTi. The superelasticity material model requires the use of the elastic material model in conjunction to define the material properties. The material model works similarly to an elastoplastic material model using a Drucker-Prager based flow rule to calculate the transformation strain increment. The plateau in the stress-strain curve of superelastic NiTi, as shown in Figure 2, is representative of this transformation flow. An effective modulus of elasticity and Poisson’s ratio are calculated by the material model at every increment based on the fraction of martensite that has transformed from austenite and the elastic properties of the austenite and martensite phases. From [15], the elastic properties are represented by

$$E = E_A + \zeta(E_M - E_A) \tag{1}$$

$$\nu = \nu_A + \zeta(\nu_M - \nu_A) \tag{2}$$

where E is the calculated modulus of elasticity, ζ is fraction of martensite, E_A is the elastic modulus of austenite, E_M is the elastic modulus of martensite, ν is the calculated Poisson's ratio, ν_A is the Poisson's ratio of austenite, and ν_M is the Poisson's ratio of martensite. Figure 3 illustrates NiTi under load before and after its full transformation from austenite to martensite. The unstressed cubic austenite phase, represented by vertical lines, which is loaded until the stress is greater than σ_{tL}^{E} and all the austenite is transformed into single-variant martensite as represented by slanted lines. This figure was adapted from [8, Fig. 2]. From [15], the total strain increment, $\Delta\boldsymbol{\varepsilon}$, is given by equation (3) below

$$\Delta\boldsymbol{\varepsilon} = \Delta\boldsymbol{\varepsilon}^{el} + \Delta\boldsymbol{\varepsilon}^{tr} \tag{3}$$

where $\Delta\boldsymbol{\varepsilon}^{el}$ is the elastic component of strain increment and $\Delta\boldsymbol{\varepsilon}^{tr}$ is the transformation component of the strain increment.

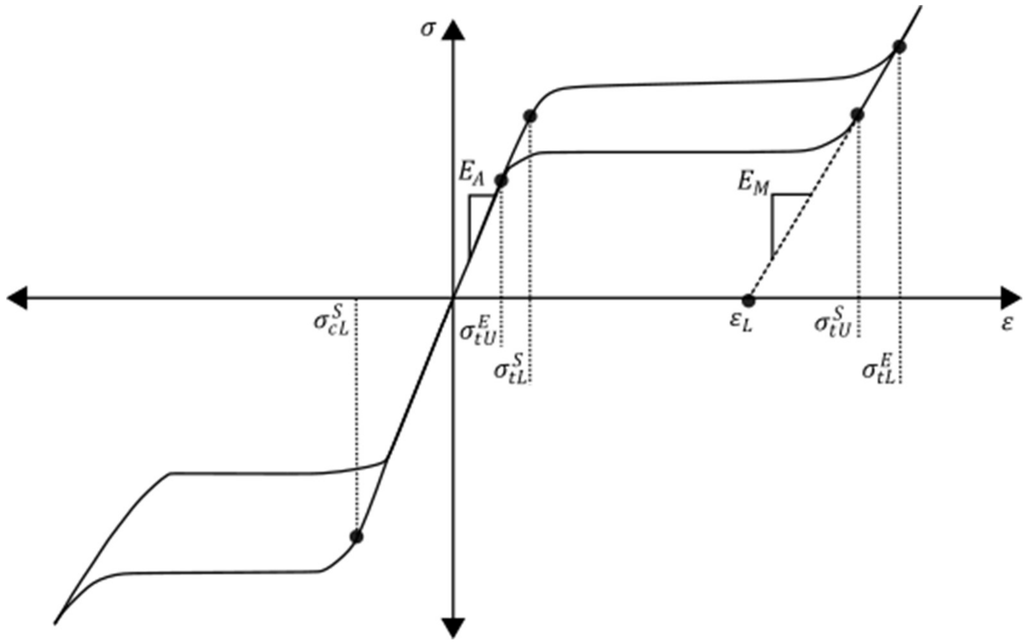

Figure 2: Example of a typical superelasticity stress-strain curve indicating material parameters adapted from [15, Fig. 1].

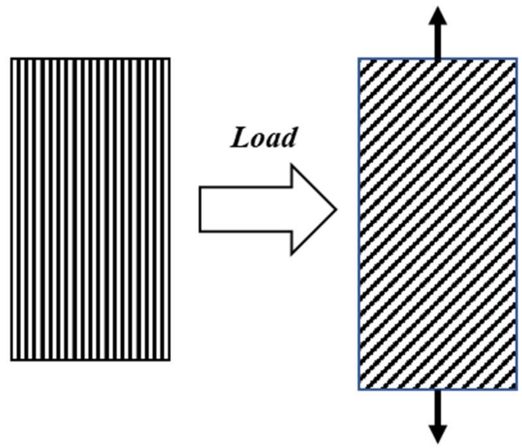

Figure 3: Depiction of superelastic NiTi stress-induced phase transformation with straight and slanted lines indicating austenite and martensite phases, respectively, adapted from [9, Fig. 2].

The material input parameters of the material model in [15] will now be described. ε^L is the transformation strain. σ_{tL}^S is the stress where austenite begins transformation to single-variant martensite during tensile loading. $\sigma_{tL}^E$is the stress where all austenite has transformed into single-variant martensite during tensile loading where $\zeta = 1$. σ_{tU}^S is the stress during tensile unloading where single-variant martensite begins transformation into austenite. σ_{tU}^E is the stress during tensile unloading where single-variant martensite finishes transformation into austenite where $\zeta = 0$. σ_{cL}^S is the stress where austenite begins transformation to single-variant martensite during compressive loading and is different from σ_{tL}^S [15]. Figure 2 shows these material parameters in relation to the stress-strain curve of the SMA.

The parameter α defined in Equation (4) describes the relationship between the initial values of σ_{cL}^{S} and σ_{tL}^{S} [11, Eq. 82], shown as.

$$\alpha = \sqrt{\frac{2}{3}} \frac{\sigma_{cL}^{S} - \sigma_{tL}^{S}}{\sigma_{cL}^{S} + \sigma_{tL}^{S}} \tag{4}$$

$\alpha = 0$ was used for the analyses in this thesis, as the initial compressive and tensile austenite to martensite starting transformation stresses during loading are assumed to be equal. The material parameters in [11] were used for evaluations of the material model in [11, pp. 33–34] and for the analyses in this thesis.

Table 1: Material Parameters from [11, pp. 33–34] used in these analyses.

$E_M\ (MPa)$	60000	$T_0(°C)$	0
ν_M	0.3	$\left(\frac{\delta\sigma}{\delta T}\right)_L \left(\frac{MPa}{°C}\right)$	0
ε_L	0.075	$\left(\frac{\delta\sigma}{\delta T}\right)_U \left(\frac{MPa}{°C}\right)$	0
$\sigma_{tL}^{S}\ (MPa)$	520	Shape Set Parameter	0
$\sigma_{tL}^{E}\ (MPa)$	600	$E_A\ (MPa)$	60000
$\sigma_{tU}^{S}\ (MPa)$	300	ν_a	0.3
$\sigma_{tU}^{E}\ (MPa)$	200	$\rho\ \left(\frac{kg}{mm^3}\right)$	$6.50(10^{-6})$
$\sigma_{cL}^{S}\ (MPa)$	520		

III. Under-Dense Material Background

As noted earlier, under-dense material (UDM) describes material in which deliberate void space exists within a volume of a given material which may or may not be filled with other material. This can be periodic or non-periodic in nature. These materials reduce the relative density of a given volume of material and modify the mechanical behavior. The relative density, $\bar{\rho}$, is a term used to relate the density of a bulk material and the space that is occupied by that material in a finite volume or a unit cell of the UDM. Relative density is generally given by

$$\bar{\rho} = \frac{V_{Bulk\ Material}}{V_{Total}} \tag{5}$$

Where $V_{Bulk\ Material}$ is the volume of the bulk material and V_{Total} is the total volume of occupied space. Relative density for a unit cell is given by

$$\bar{\rho}_{Unit\ Cell} = \frac{V_{Bulk\ Material}}{V_{Unit\ Cell}} \tag{6}$$

where $\bar{\rho}_{Unit\ Cell}$ represents the relative density, $V_{Bulk\ Material}$ is the volume of bulk material within the unit cell, and $V_{Unit\ Cell}$ is the volume of the entire unit cell [8].

LSMs are periodic UDMs made up of beam-like or truss-like members. LSMs achieve modification of the mechanical behavior of a given base material such as the stiffness, density, and isotropy by the deliberate arrangement of the structural members [2], [7]. LSM can be characterized by the dominating mechanical behavior of the lattice members as “bending-dominated” or “stretching-dominated” [7, p. 1035]. The macroscale mechanical behavior of bending-dominated lattices is relatively compliant while the mechanical behavior of stretching-dominated lattices is relatively stiff. The dominating mechanical behavior of these lattices is a spectrum of these as noted in [7]. The "Maxwell Criterion" [7, p. 1036] describes the dominating behavior of a LSM based on an algebraic rule. For 3D space from [7, Eq. 2], it is represented by

$$M = s - 3n + 6 \tag{7}$$

where M is the Maxwell number, s is the number of struts, and n is the number of joints. Maxwell numbers less than 0 are considered bending-dominated. Maxwell numbers equal to zero are considered stretching-dominated. Maxwell numbers greater than zero are considered mostly stretch-dominated [5].

Rhombic dodecahedron lattices are bending-dominated structures. It is made up of 12 faces of rhombuses whose edges are the members of the unit cell. [8] describes the mechanical properties of the rhombic dodecahedron unit cell as orthotropic with equivalent elastic moduli in the *y*-direction and *z*-direction and a more compliant elastic modulus in the *x*-direction. Figure 4 shows the rhombic dodecahedron structure based off of [8, Fig. 1]. Figure 4(a) shows the structural members in blue and the vertices as red dots. Figure 4(b) shows the rhombic dodecahedron face has side lengths of L and internal angles of 2α and 2Θ. Figure 4(c) shows top, side, and front views of the rhombic dodecahedron unit cell with overall unit cell dimensions. Figure 5 shows 3-Dimensional renderings of the rhombic dodecahedron structure created in Abaqus CAE.

Evaluation of the rhombic dodecahedron unit cell by Maxwell's Criterion finds the structure to be bending-dominated. There are 14 joints of the rhombic dodecahedron Unit Cell and 24 members, and therefore the Maxwell number is -12. The rhombic dodecahedron structure is a bending-dominated lattice by this rule. A circular-cross section was selected for the analyses in this thesis.

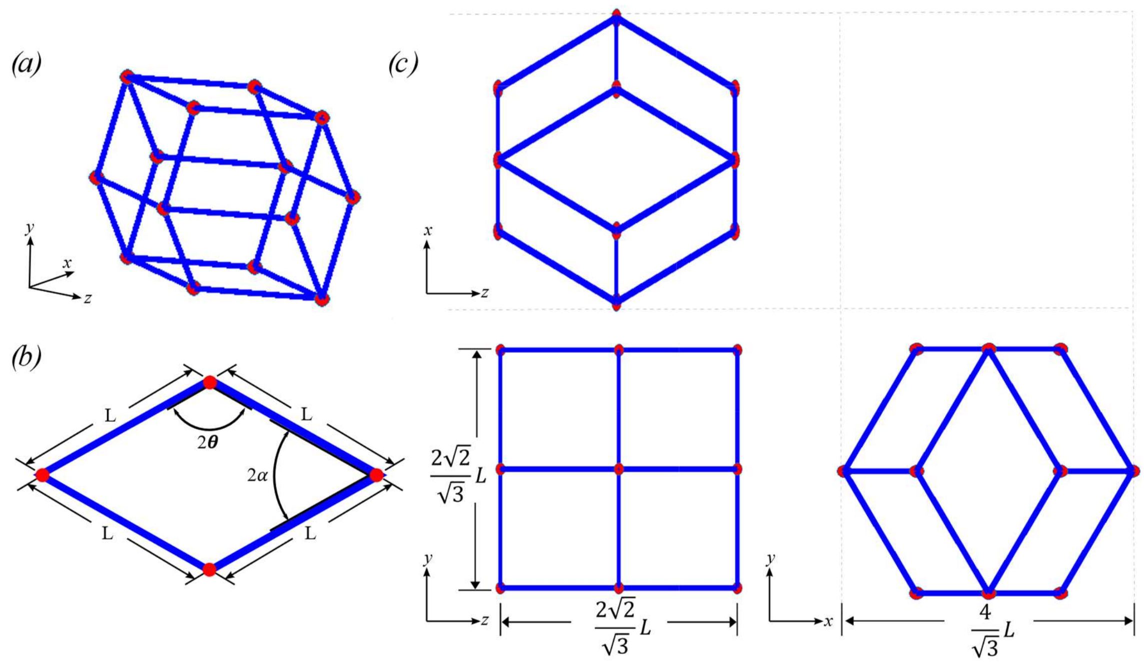

Figure 4: Open-cell rhombic dodecahedron structure with dimensions adapted from [8, Fig. 1]

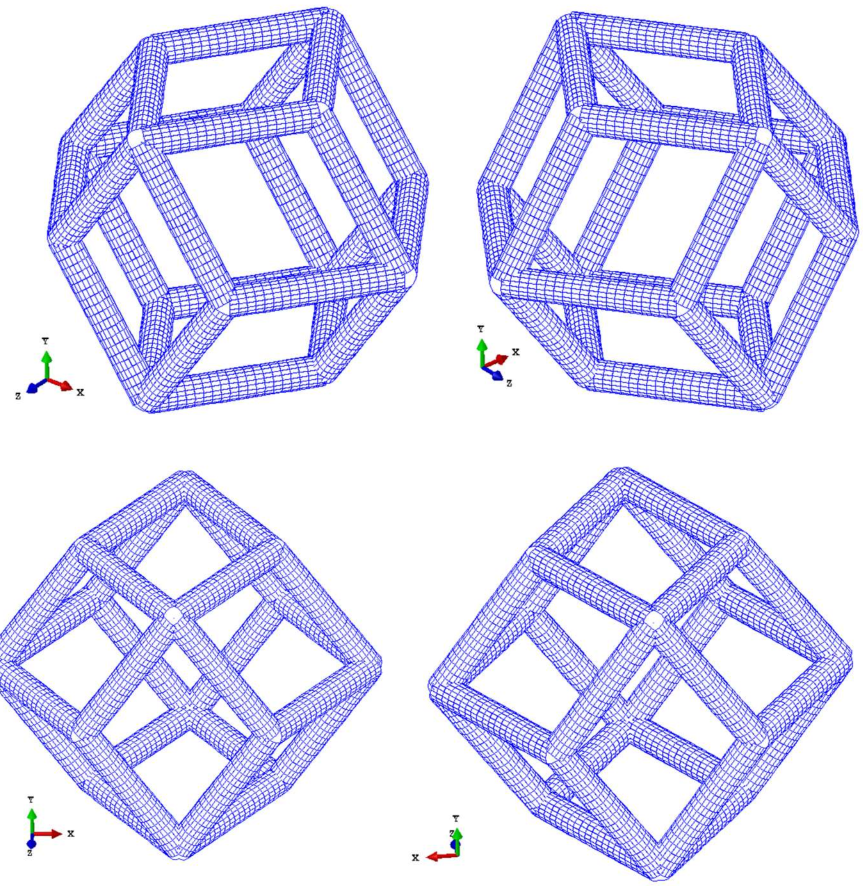

Figure 5: The rhombic dodecahedron unit cell rendered in Abaqus CAE from multiple viewpoints.

The relative density of the rhombic dodecahedron unit cell with a circular cross-section for the structural members is represented by

$$\bar{\rho}_{RhD} = \frac{27\sqrt{3}\pi r^2}{4L^2} \tag{8}$$

where $\bar{\rho}_{RhD}$ is the relative density of the rhombic dodecahedron unit cell, r is the radius of the cross-section, and L is the side length of the rhombus.

Manufacturing of a superelastic NiTi rhombic dodecahedron LSM could be manufactured via a variety of methods. Wire-woven methods were used to produce a

SMA truss as shown in [4]. These structures could also be created by an additive manufacturing method such as 4D printing [16].

IV. Damping and Energy Dissipation

Materials capable of high energy dissipation are of interest for their damping properties when considering a need to absorb and control high energy impacts and intense vibrational loading [1], [3], [4], [13]. Energy dissipation is shown to be desirable for applications such as composite armor in [1] and for an impact absorber on an airplane wing in [3]. Superelastic NiTi has desirable energy dissipation properties based on the work in [13]. Energy is dissipated as a result of the “stress-induced” [10, p. 176] solid state phase change as the unloading path has lower energy than the loading path [4], [13].

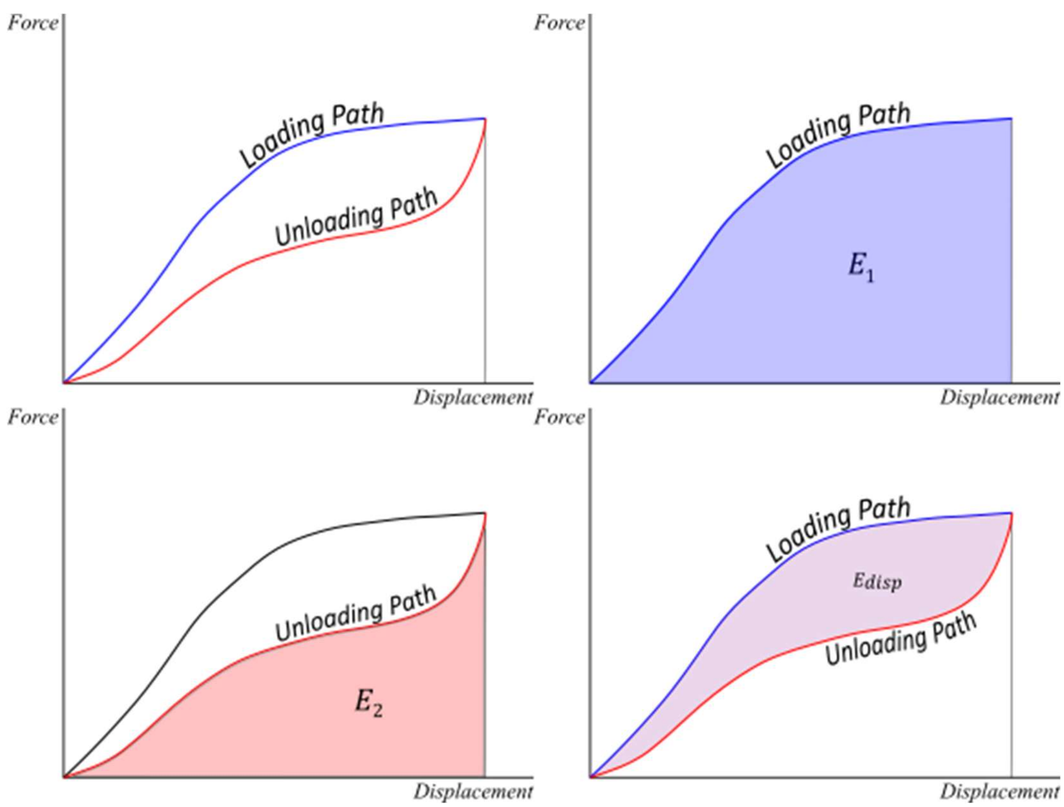

Figure 6: Energy dissipation shown as the difference between the energy of the loading path in blue and the unloading path in blue on a typical compression curve of an SMA adapted from [4, Fig. 11].

Energy dissipation can be quantified by the difference in potential energy input during loading and the potential energy output during unloading. [4] calculated the dissipated energy from the difference of energy in the loading and unloading paths of wire-woven SMA trusses in compression. From [4, Eq. 1], the energy dissipated is calculated by

$$E_{disp} = E_1 - E_2 \tag{9}$$

where loading energy, the unloading energy, and energy dissipated are represented by E_1, E_2, and E_{disp}, respectively. Figure 6 shows E_1, E_2, and E_{disp} with respect to the loading paths. From [4, Eq. 2] and [4, Eq. 3], the energy dissipation coefficient, η, which for the purposes of this thesis represents a normalization of the energy dissipated, is calculated by

$$\eta = \frac{E_{disp}}{\pi(E_1 - \frac{E_{disp}}{2})} \tag{10}$$

where η is the energy dissipation coefficient [4, p. 2288].

SIMULATION SET-UP AND STUDIES

I. Overview

The superelastic NiTi rhombic dodecahedron structure unit cell was modeled using two methods. The first model, which will be referred to as the Direct Numerical Simulation (DNS) model, uses a mesh of tetrahedral elements. The second model, which will be referred to as the Beam model, was created using beam finite elements [17, pp. 535–596] for computational efficiency. The DNS is the more precise model of behavior at the expense of computational efficiency. A study to evaluate the mechanical behavior of the unit cell was carried out in which the unit cell was displaced in tension and compression along the x-direction and y-direction. This study was carried out using both the DNS model and the Beam model to evaluate how well the Beam model can model the mechanical behavior of the unit cell. The Beam model was then used for two studies to evaluate the unit cell for its capacity for energy dissipation given changes in geometry and relative density, respectively. A third study was then carried out to evaluate the unit cell for its capacity for energy dissipation when combining the changes in geometry and relative density while also scaling up the unit cell.

II. Meshing

Both the DNS model and the Beam Model used elements from the Abaqus Element Library [18][19]. The DNS model was meshed using *C3D10* tetrahedral elements. The unit cell was modeled in PTC CREO Parametric 4.0 Academic and meshed in Abaqus CAE. The DNS Model used 21683 elements and 38303 nodes. The Beam model was meshed using *B31* beam elements. The unit cell was also modeled in Abaqus CAE with line segments between vertex points on which *B31* beam elements were meshed on. *B31* beam elements model beam bending behavior based on Timoshenko beam theory[17, pp. 535–596]. *B31* elements have a single-integration point and were used for analyses in this thesis. The Beam model had 480 elements and 1430 nodes. Figure 7 and Figure 8 show the meshes of the rhombic dodecahedron unit cell for the Beam model and the DNS model, respectively.

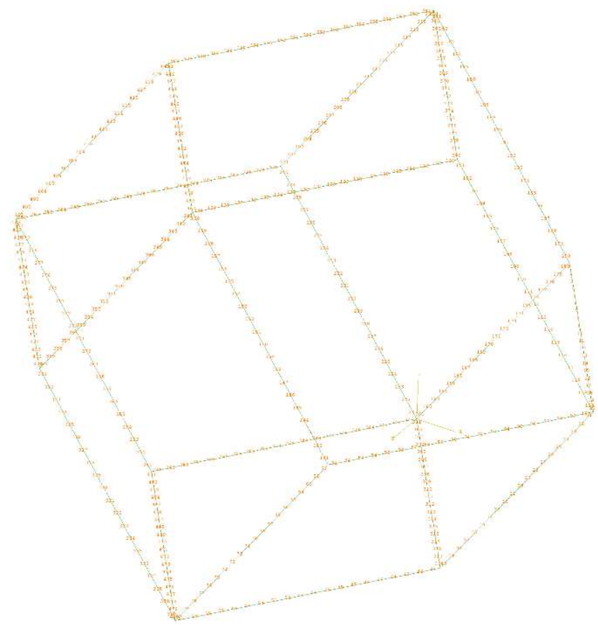

Figure 7: Beam model mesh using B31 beam elements with 480 elements and 1430 nodes

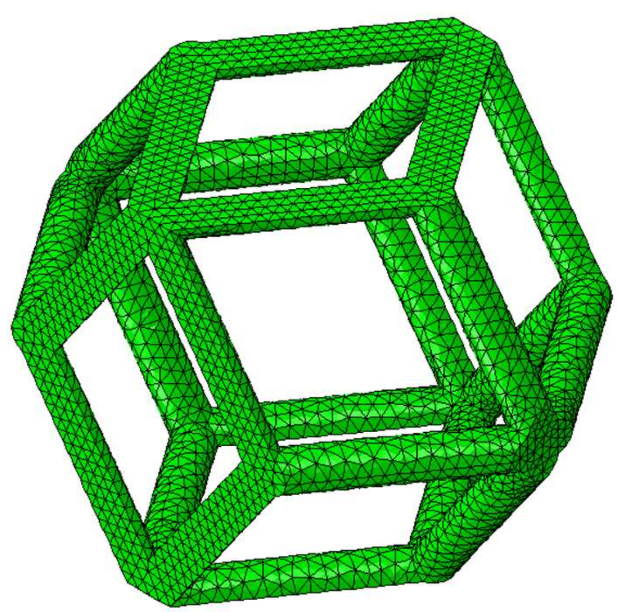

Figure 8: DNS model mesh of the rhombic dodecahedron unit cell using C3D10 elements with 21683 elements and 38303 nodes

III. Mechanical Behavior Study

A study on the mechanical behavior of the superelastic NiTi rhombic dodecahedron unit cell was carried out. The force-displacement responses of the unit cell in the *x*-direction and the *y*-direction were evaluated. The structural members of the unit cell had a length of 1 mm and a radius of 0.087 mm. Figure 9 shows the boundary conditions used for this analysis for displacement-controlled loading in tension and compression along the *x*-direction and *y*-direction, respectively. Figure 9(a) and Figure 9(b) shows the boundary conditions used on the DNS model and the Beam model, respectively. Both show the top, front, and side views of the unit cell. The applied displacements in the *x*-direction and *y*-direction are shown in green and orange, respectively. The DNS model is shown with the displacements on the highlighted areas of application. The Beam model is shown with the displacements applied at the nodes. Both the DNS model and the Beam Model were used for this study. The edge planes of the unit cell were displacement constrained in the direction perpendicular to the direction of loading as shown by the rollers. The unit cell was displaced along the *x*-direction and *y*-direction as denoted in green and orange, respectively, at the opposite edge planes edges to 2%, 4%, 6%, and 8% of the side length, and then unloaded to 0% displacement. Force and displacement parallel to the loading direction was extracted from the nodes of the loading plane. The mean average of the force and displacement of these nodes was calculated at every increment. It should be noted that the displacement-controlled loading in the *x*-direction and *y*-direction occurred as two independent loading conditions; displacement-controlled loading in the *x*-direction and *y*-direction did not occur simultaneously.

Quasi-static loading conditions were assumed for both models. No contact analysis or post-buckling analysis was performed in both models. Both the DNS Model and the Beam Model used a full Newton-Raphson solution and nonlinear geometry was accounted for. The Beam model had an initial step size (unitless) of 0.01, a maximum step size of 0.01, and a minimum step-size of 0.00001 over an interval of 1. The DNS model had an initial step size of 0.001, a maximum step size of 0.05, and a minimum step-size of 0.00001 over an interval of 1.

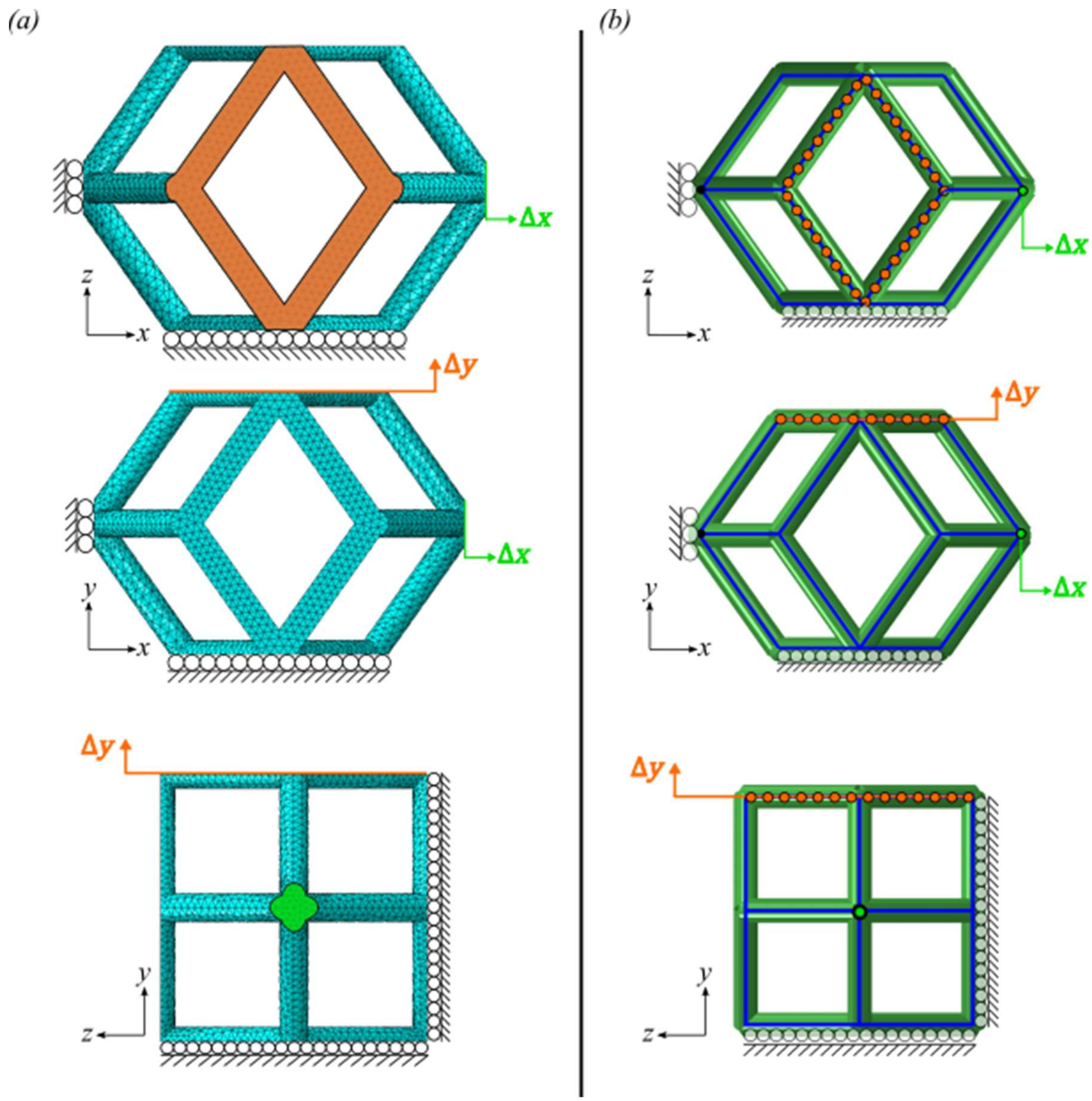

Figure 9: Boundary conditions for the unit cell analyses performed for the (a) DNS model and (b) Beam model.

IV. Energy Dissipation Studies

Three studies were carried out to find an energy dissipation optimum using the Beam model. Only the beam model was used for these studies due to its computational efficiency. Only the compression loading regime was used to calculate the energy dissipated and the energy dissipation coefficient. The trapezoid rule was used to calculate the energy of loading and unloading from the force-displacement curves. This was used to calculate the energy dissipated and the energy dissipation coefficient. The loading and boundary conditions of the Beam model in compression used in the mechanical behavior study were used for the energy dissipation studies.

For the first study, the relative density was kept constant while the radii of the members lying parallel to the *xz*-plane and *xy*-plane were varied. The members parallel to the *xz*-plane and *xy*-plane are referred to as the horizontal group and the vertical group, respectively. The radius of the horizontal group was changed while the unit cell volume remained constant, so the vertical group radii changed with it. The choice of varying the radii of members in perpendicular planes was deliberate as periodicity can be maintained while the geometry is modified. Figure 10 shows the unit cell with the horizontal radius at its upper and lower bounds of 0.039 mm and 0.120 mm. A step-size of .003 mm was used. When loading the unit cell as if it were in a “sandwich structure” [4, p. 2285] or in a “Sandwich Panel Configuration” [8, p. 2881], structural members of the rhombic dodecahedron unit cell perpendicular to loading behave primarily in uniaxial tension and compression. The energy dissipated and the energy dissipation coefficient are plotted against the horizontal radius.

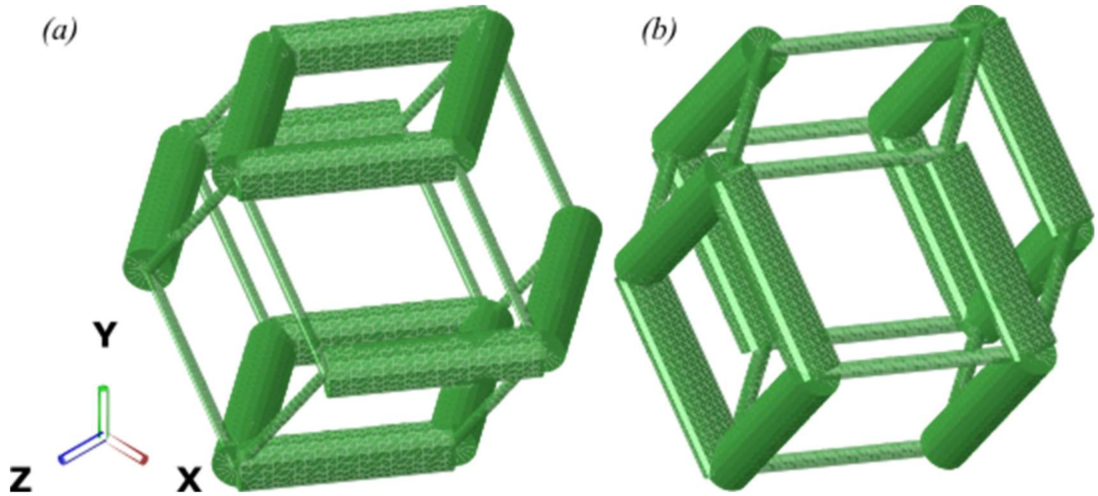

Figure 10: (a) Upper and (b) lower bound of the Horizontal Radius

For the second study, the relative density, $\bar{\rho}_u$, (unitless) was varied between 0.0372 and 0.3526. The relative density was calculated from radius values of 0.039 mm to 0.12 mm with a step-size of 0.03. The energy dissipated and the energy dissipation coefficient are plotted against the relative density. Figure 11 shows the unit cell of the rhombic dodecahedron with the upper and lower bounds of the relative density.

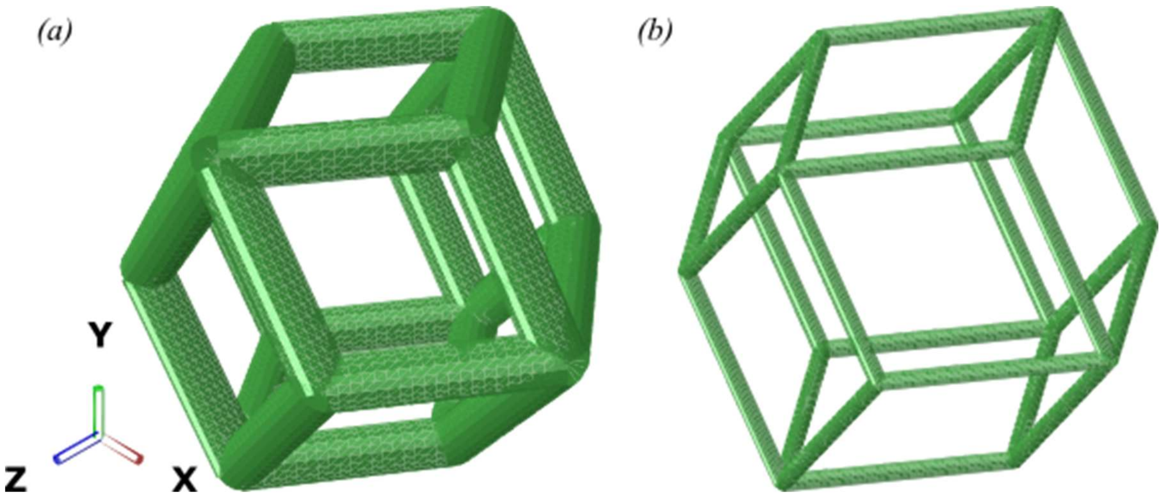

Figure 11: (a) Upper and (b) lower bound of the Relative Density

The third study combined the concepts of the first and second study. The length of the strut members was changed to 5 mm. The relative density, $\bar{\rho}_u$, was varied between 0.03 and 0.045 at a resolution of 0.03. The horizontal radius was changed by a factor named R_{var} between 0.5-1.25 at a resolution of 0.05. The load was varied between 1-8% of the unit cell length in the x-direction and y-direction. Energy Dissipation and the Energy Dissipation Coefficient were calculated. A peak von-Mises stress limit of 700 MPa was chosen using engineering judgement based on the upper limits of stresses of numerical studies and experimental data in [4], [10], [11] as to not include any simulations with stresses where fracture may occur. Any simulation exceeding this von-Mises stress or that provided a warning indicating buckling behavior was not included in plots. The third study was done to be thorough and to demonstrate that $\bar{\rho}_u$ and R_{var} are independent.

RESULTS AND DISCUSSION

I. Overview

This chapter will show and discuss the results of the three studies. Python scripts were used to extract force, displacement, and stress values from the ABAQUS .odb (output database) files and warning messages from the ABAQUS .msg (message) files. These values and warning messages were then post-processed and plots were created using MATLAB. Stress contour plots were generated in the Visualization Module of Abaqus CAE then processed and arranged for clarity and conciseness using Inkscape. Samples of the source code utilized in this work can be found in the appendices.

II. Mechanical Behavior Study Results

Figure 12 and Figure 13 show the force-displacement curve of the superelastic rhombic dodecahedron unit cell in the *x*-direction and the *y*-direction, respectively. The DNS model and the beam model are shown plotted together with varying displacements. The qualitative shape of the force-displacement plots show that the models exhibit similar mechanical behavior, although the response of the beam model is stiffer. The unit cell of the DNS model only includes volume of the material within the boundary of the unit cell; the cross-section is a semi-circle for the structural members which lie on the border of the unit cell, whereas the cross-section of the beam model is a full circle. The Beam model does not model the behavior of stress-concentrations from the geometry at the vertices of the structure. The force-displacement response of the *y*-direction is stiffer than the *x*-direction. This is consistent with the mechanical behavior of the rhombic dodecahedron structure in [8] that showed a stiffer modulus of elasticity in the *y*-direction than in the *x*-direction.

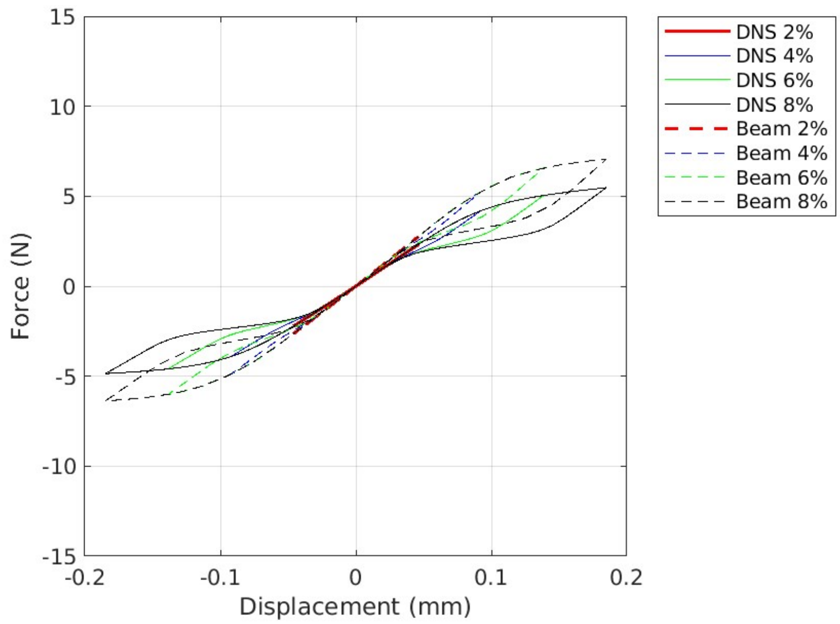

Figure 12: Force-Displacement Curve of Beam and DNS Model in x-Direction

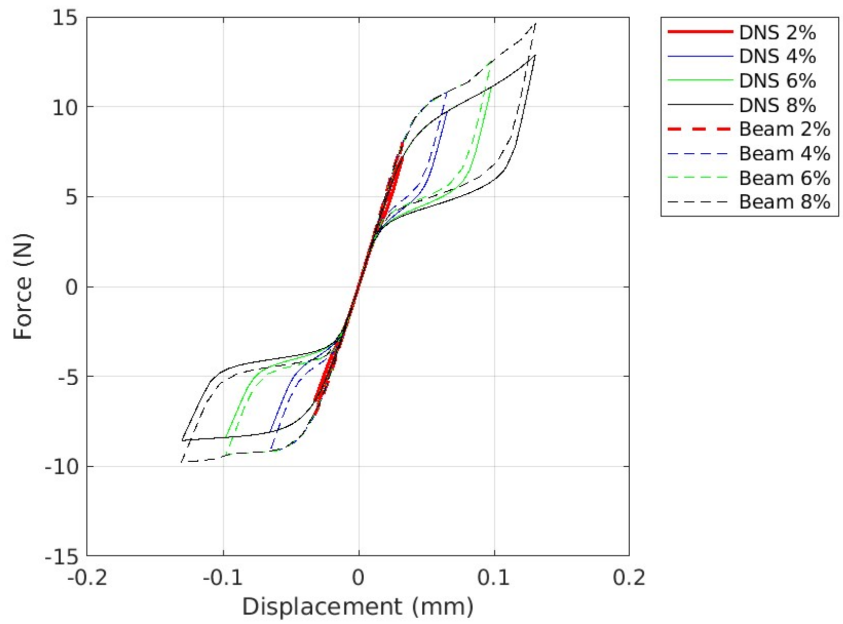

Figure 13: Force-Displacement Curve of Beam and DNS Model for y-direction

Figure 14-17 show the equivalent von-Mises Stress contour plots of the rhombic dodecahedron unit cell in compression in the x-direction, tension in the x-direction, compression in the y-direction, and tension in the y-direction, respectively, on the deformed model. In both models and for all displacements, stresses are similarly distributed along the structural members with higher stresses closer to the vertices of the LSM. The DNS model shows higher stresses at the vertices than the Beam model. The elements used in the Beam model only model beam bending behavior of the and do not model the complex geometry at the vertices where stress concentrations occur. The stress of the outermost structural members is higher in the DNS model, as the full cross section of each member was not used in the unit cell of the DNS model.

Some of the stresses in the models would exceed the stress where fracture would occur are shown for the 6% and 8% displacement loading conditions of the displacement.

These results show that the predicted stresses are consistent between the DNS and Beam models.

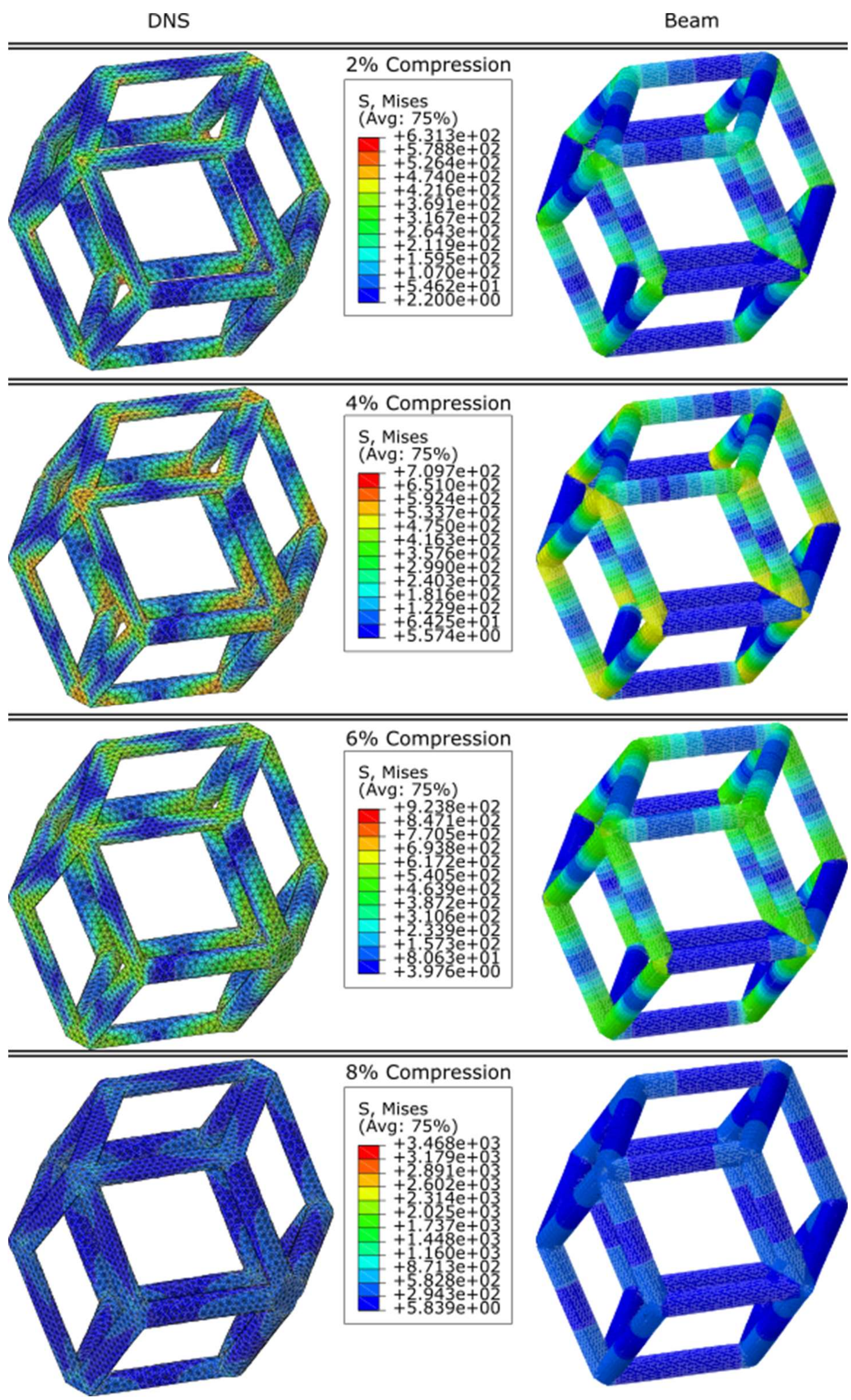

Figure 14: Equivalent Von-Mises Stress (MPa) contour plots for compression in the x-direction for the DNS (left) and Beam Model (right). The colored legend changes with each displacement.

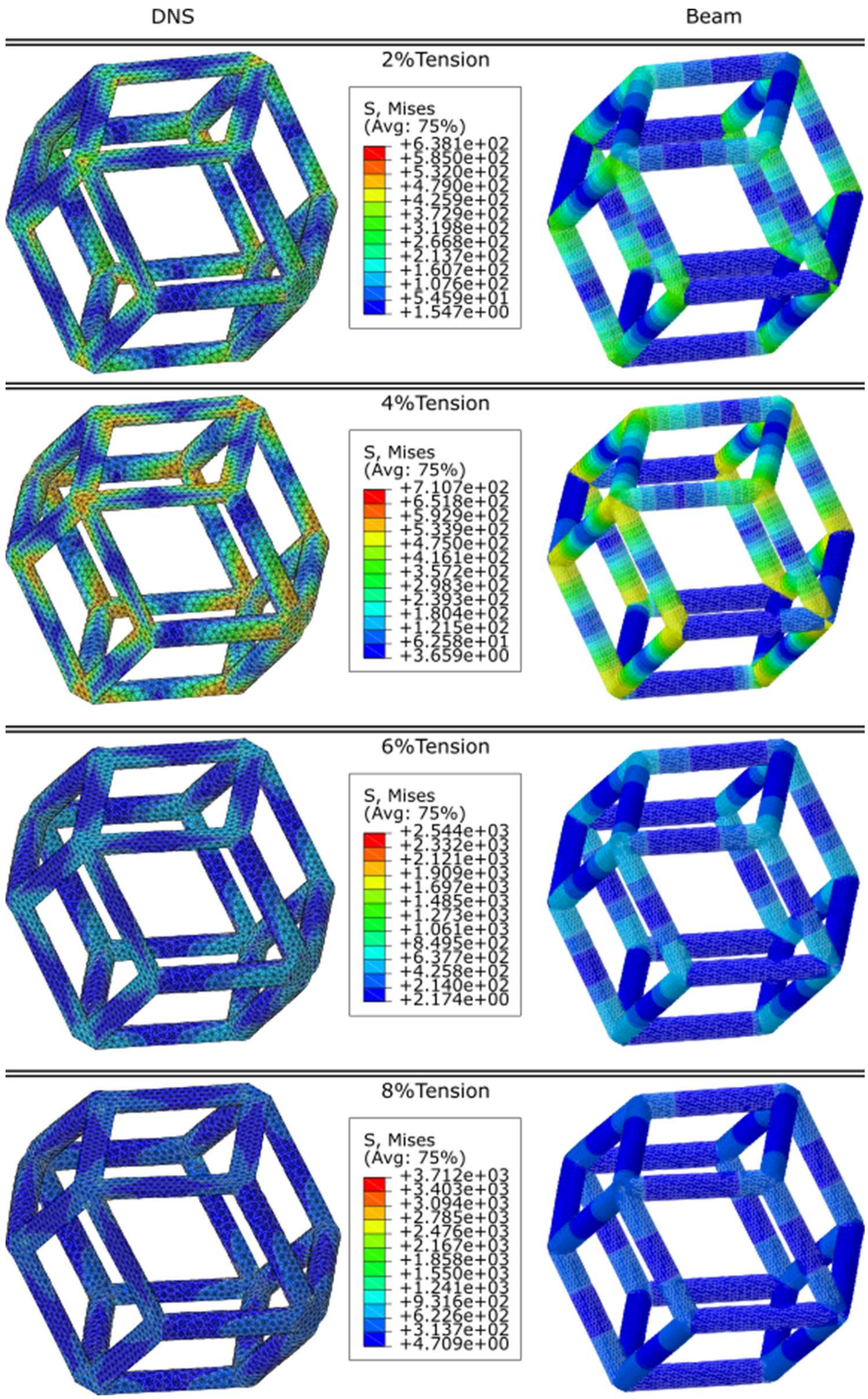

Figure 15: Equivalent Von-Mises Stress (MPa) contour plots for tension in the x-direction for the DNS (left) and Beam Model (right). The colored legend changes with each displacement.

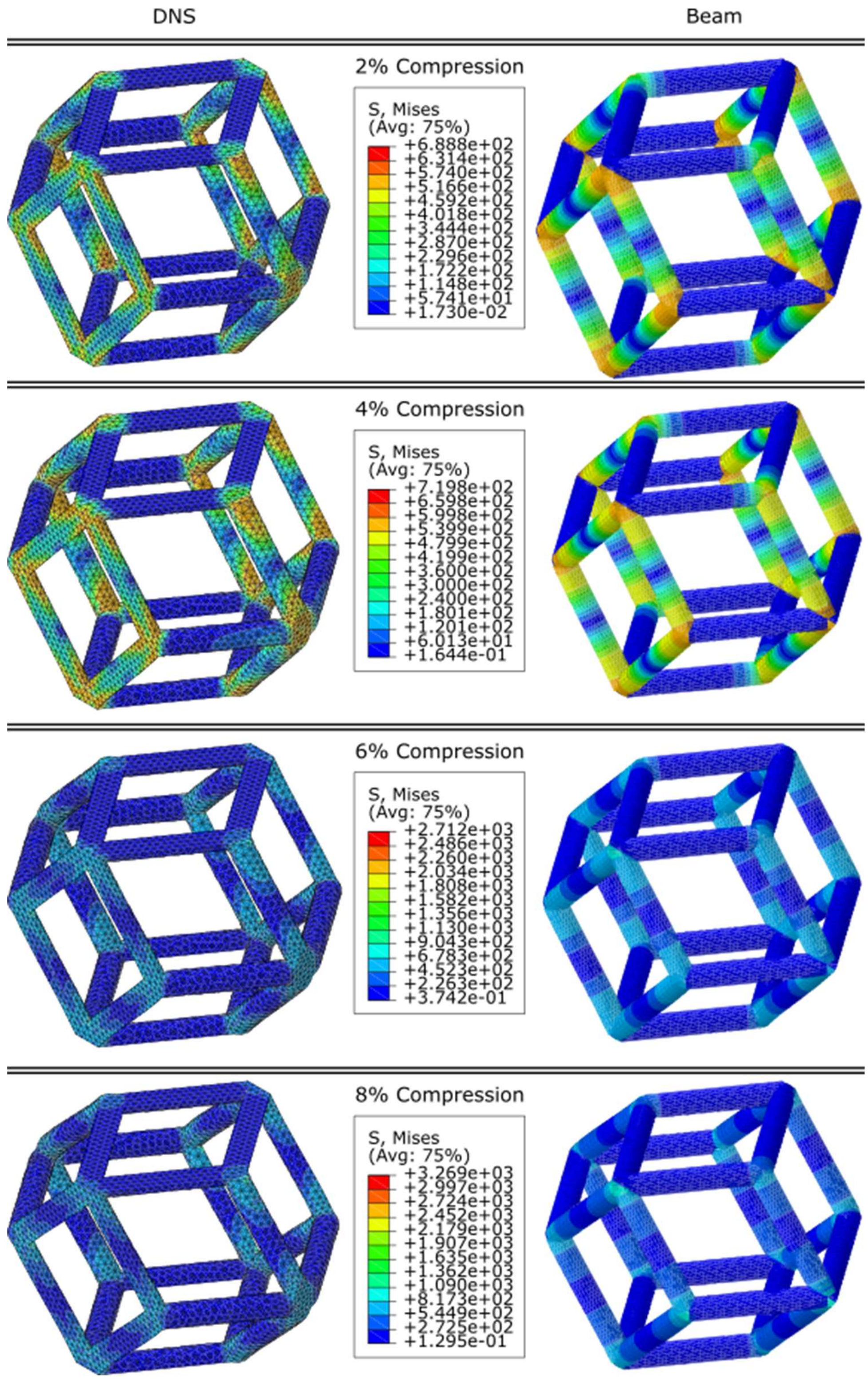

Figure 16: Equivalent Von-Mises Stress (MPa) contour plots for compression in y-direction for the DNS (left) and Beam Model (right). The colored legend changes with each displacement.

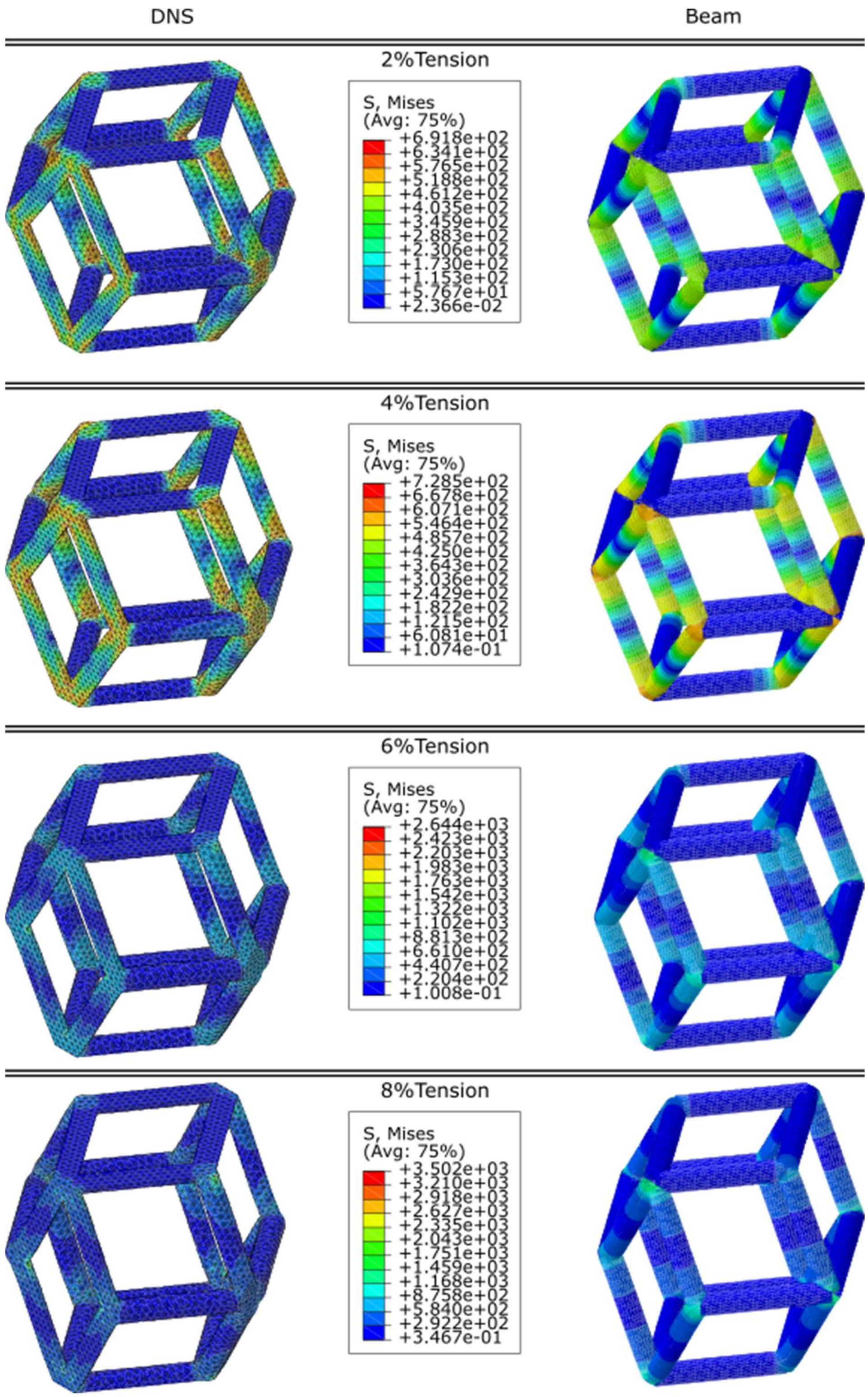

Figure 17: Equivalent Von-Mises Stress (MPa) contour plots for tension in the y-direction for the DNS (left) and Beam Model (right). The colored legend changes with each displacement.

III. Energy Dissipation and Radii Variation Study

Simulation results indicating buckling behavior were not included. These points are omitted from the plots in Figures 20-21. Figure 18 shows the energy dissipated versus horizontal radius for loading in the *x*-direction. Figure 19 shows the energy dissipation coefficient versus the horizontal radius for loading in the *x*-direction. The variation of geometry is symmetric relative to loading which may cause the bimodal response. There are local maximums centered about the nominal radius for the energy dissipated and the energy dissipation coefficient in both Figure 18 and Figure 19. There is a local minimum where the horizontal and vertical radii are equal at 0.087 mm in both Figure 18 and Figure 19. Figure 20 shows the energy dissipation versus horizontal radius for loading in the *y*-direction. Figure 21 shows the energy dissipation coefficient versus horizontal radius for loading in the *y*-direction. As the horizontal radius increases, the energy dissipation and energy dissipation coefficient decrease. The energy dissipated and energy dissipation coefficient increased with larger displacements for loading in the *x*-direction and the *y*-direction, which is consistent with the results found in [4].

The local maxima indicate that there is a geometric optimum for energy dissipation by varying the radii. The optimum occurs when the horizontal radii are less than the vertical radii at around 0.07mm. This can be used to find an energy dissipation optimum for a given relative density. For non-additively manufactured structures, this result could be used in conjunction with tabulated nominal wire sizes to find a wire-woven optimal structure using existing wire sizes.

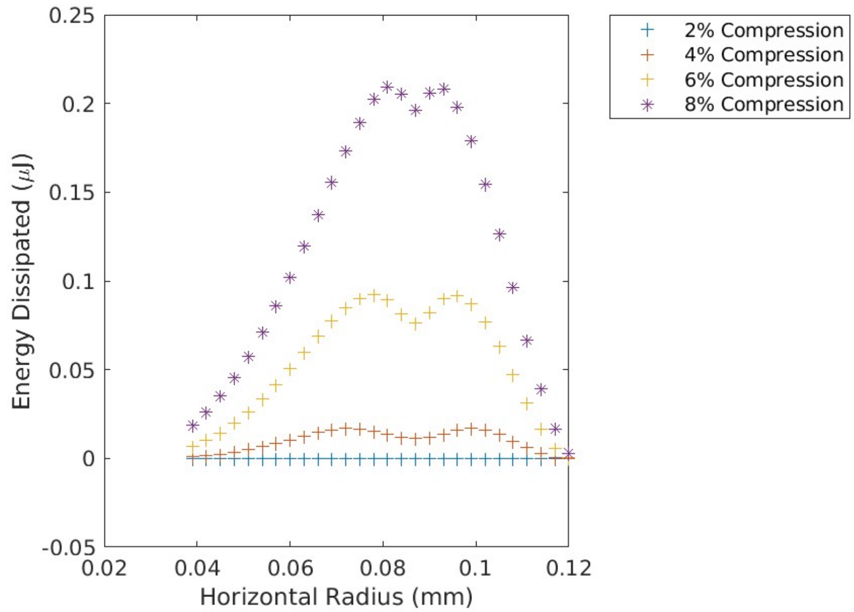

Figure 18: Energy dissipated vs horizontal radius for loading in x-direction.

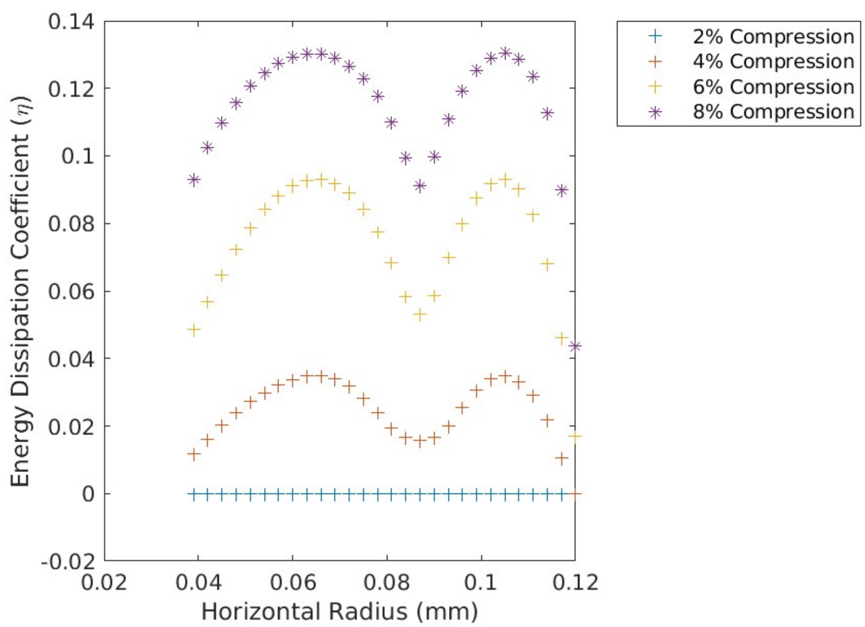

Figure 19: Energy dissipation coefficient vs horizontal radius for loading in x-direction.

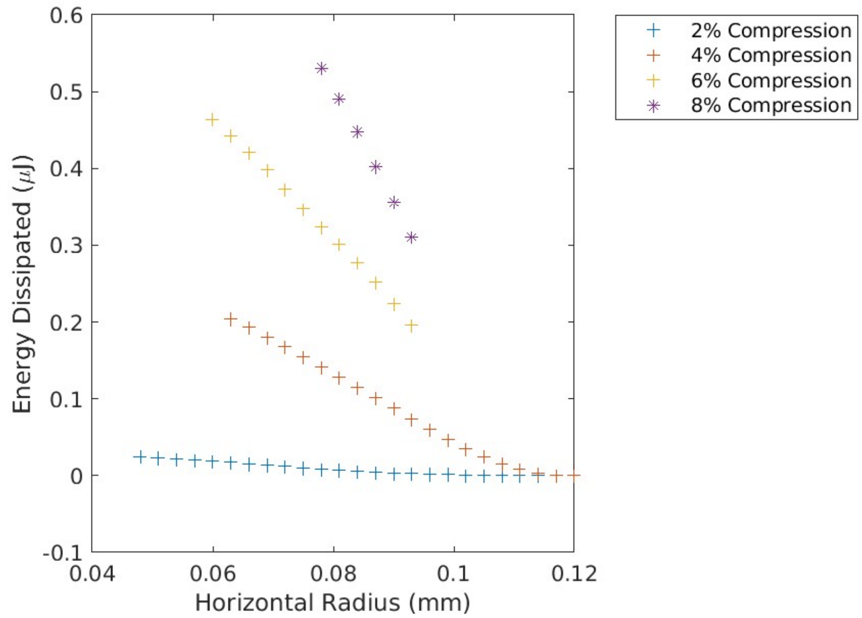

Figure 20: Energy dissipation vs horizontal radius for loading in y-direction. Simulations that indicated buckling were not included in this plot.

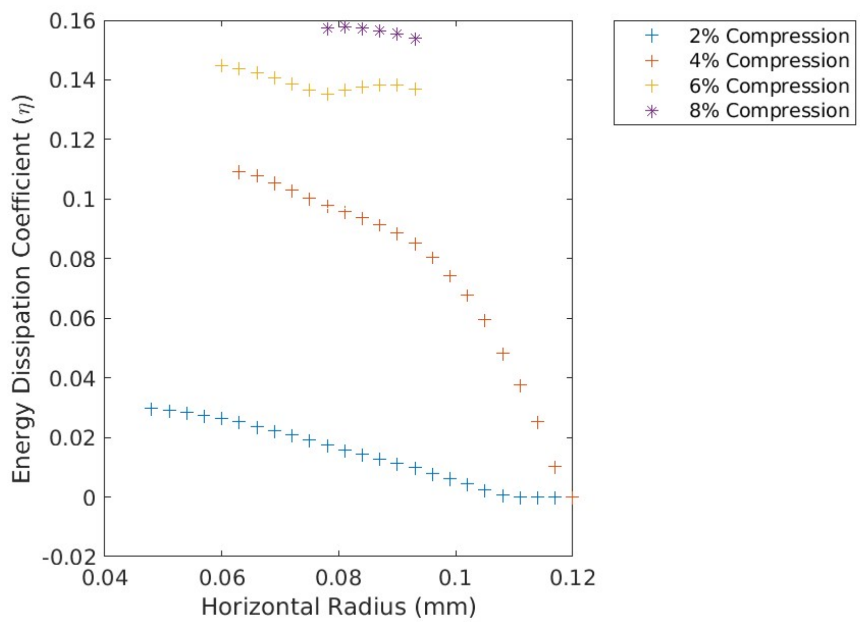

Figure 21: Energy dissipation coefficient vs horizontal radius for loading in y-direction. Simulations that indicated buckling were not included in this plot.

IV. Energy Dissipation and Relative Density Study

Figure 22 and Figure 24 shows the energy dissipated versus the relative density for loading in the *x*-direction and the *y*-direction, respectively. Figure 23 and Figure 25 shows the energy dissipation coefficient versus the relative density for loading in the *x*-direction and *y*-direction, respectively. All figures show that as the relative density increases, the energy dissipated and the energy dissipation coefficient increase. An increase in the energy dissipated and the energy dissipation coefficient with an increase in displacement remains consistent with the results in [4]. As the relative density increases, the energy dissipated appears to increase exponentially. As the relative density increases, the energy dissipation coefficient appears to increase logarithmically. In other words, there are diminishing returns as the density of the structure is increased with respect to using more material, namely an increase in mass density and financial cost.

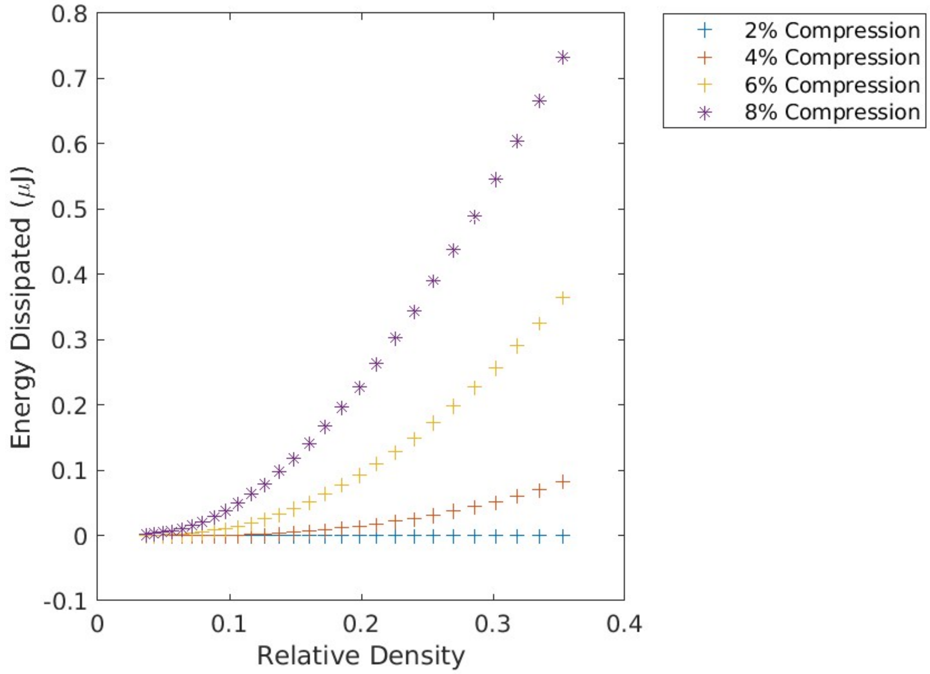

Figure 22: Energy dissipation vs relative density for loading in x-direction.

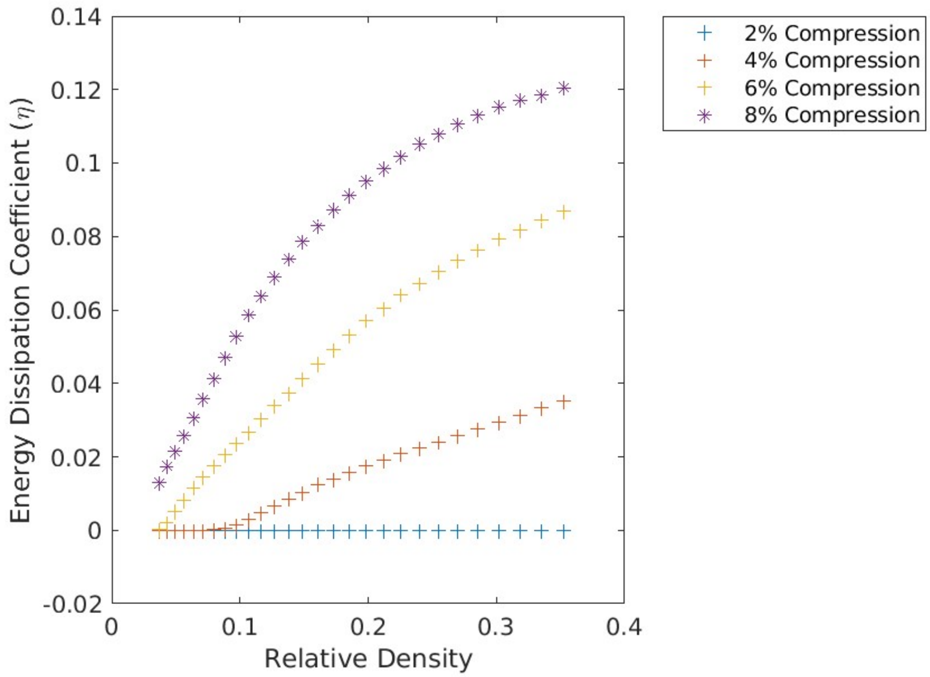

Figure 23: Energy dissipation coefficient vs relative density for loading in x-direction.

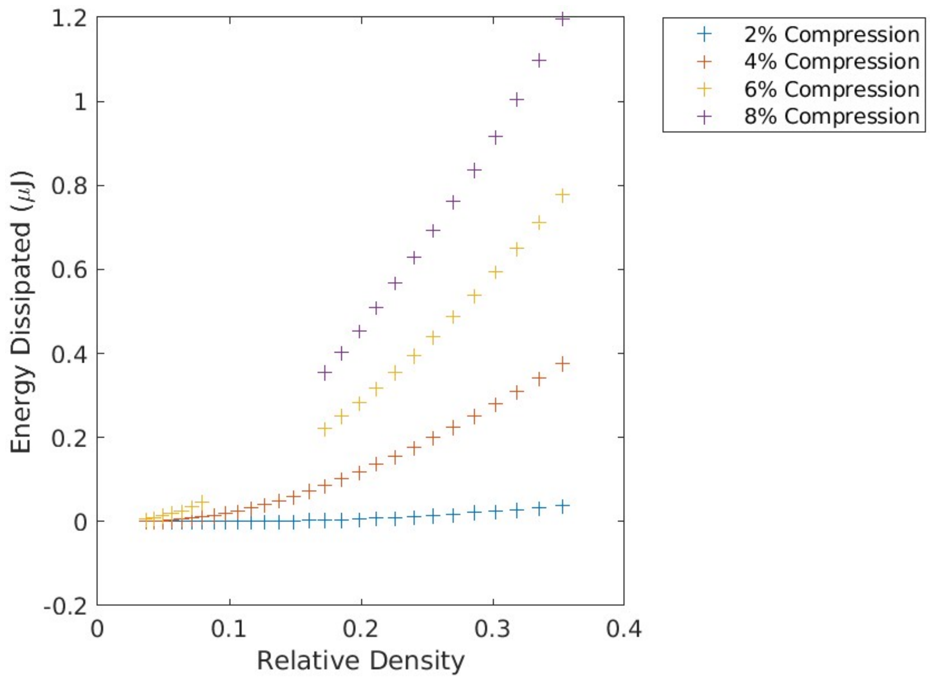

Figure 24: Energy dissipated vs relative density for loading in y-direction. Simulations that indicated buckling were not included in this plot.

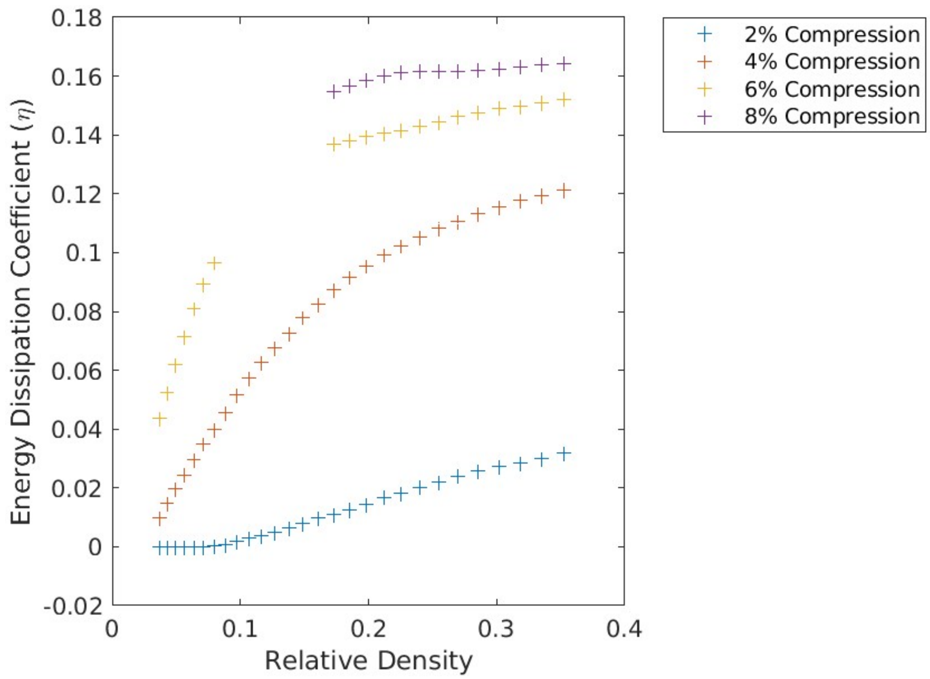

Figure 25: Energy dissipation coefficient vs relative density for loading in y-direction. Simulations that indicated buckling were not included in this plot.

V. Energy Dissipation Combined Study

Figure 26 and Figure 28 shows the energy dissipated plotted against the R_{var} and the relative density for a displacement of 4% of height in the x-direction and y-direction, respectively. Figure 27 and Figure 29 shows the Energy Dissipation Coefficient plotted against the R_{var} and the relative density for a displacement of 4% of height in the x-direction and y-direction, respectively. Only the 4% displacement data in this section as presenting the other plots does not provide additional meaning information these results Plots for other displacements, 1-8%, can be found in Appendix A. Peaks can be seen in the plots of Figure 26 and Figure 27. As the relative density increases, the Energy dissipation appears to increase exponentially, and the Energy Dissipation Coefficient appears to increase logarithmically and approach a plateau. As R_{var} increases, the energy dissipation and Energy Dissipation Coefficient decrease for the y-direction loading condition. These are the same trends for relative density and a change in the horizontal radius is found in the plots in the other energy dissipation studies. These plots show that the energy dissipation and the Energy Dissipation Coefficient can be controlled when manipulating the relative density and R_{var}. These results show that utilizing the Beam model is an efficient method of providing information on the energy dissipation properties of a superelastic NiTi rhombic dodecahedron LSM and its use in a design space. These results need to be verified with a DNS model. This model could be used with existing tabulated wire sizes to create a wire-woven design for a given relative density, which could be modeled using a Beam model and verified with a DNS model.

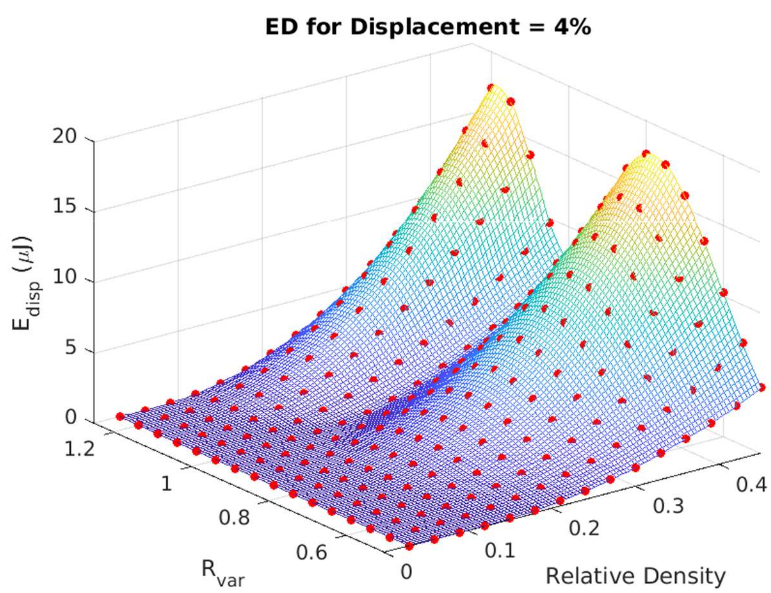

Figure 26: Energy Dissipated plotted against R_{var} and relative density for 4% displacement in x-direction

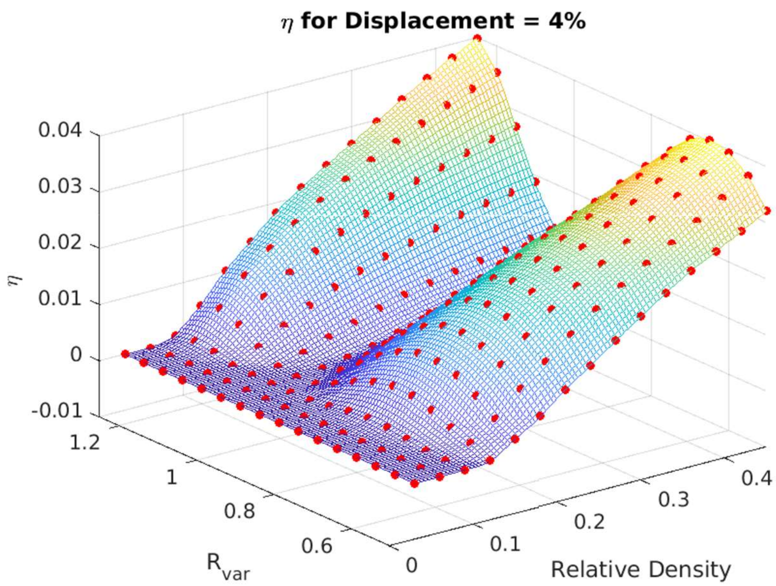

Figure 27: Energy Dissipation Coefficient, η, plotted against R_{var} and relative density for 4% displacement in x-direction

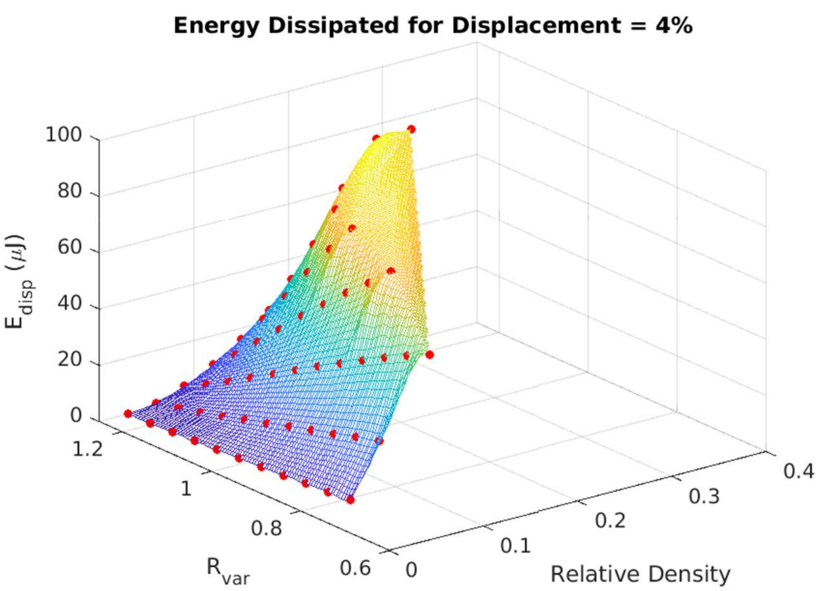

Figure 28: Energy Dissipated plotted against R_{var} and relative density for 4% displacement in y-direction. Simulations that exceed stress limit or that indicated buckling were not included in this plot.

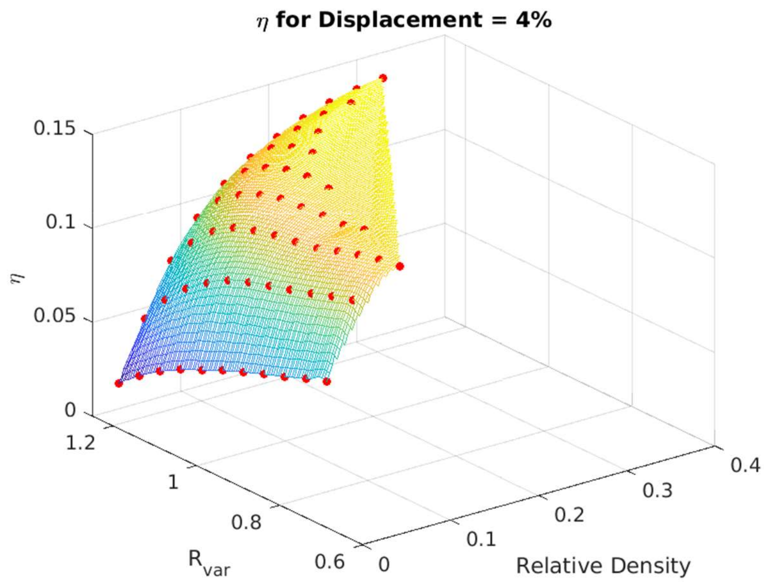

Figure 29: Energy Dissipation Coefficient, η, plotted against R_{var} and relative density for 4% displacement in y-direction. Simulations that exceed stress limit or that indicated buckling were not included in this plot.

CONCLUSION

These studies show that an optimum for energy dissipation can be found by manipulating the geometry by changing the relative density and the horizontal radius of the superelastic NiTi rhombic dodecahedron LSM. The Beam model provides information about the mechanical behavior of the superelastic NiTi Rhombi-Dodecahedron LSM at a reduced computational expense compared to the DNS model. The Beam model was used to efficiently create the plots of the energy dissipation studies. The information provided by Figures 26-29 about the sensitivities of energy dissipation and the Energy Dissipation Coefficient to manipulation of the geometry of the LSM can be used as a tool to understand the design and implementation of these structures. It can also be used to weigh performance versus cost. The numerical studies shown in Figures 26-29 could be efficiently reproduced using other material parameters or other material models for NiTi without incurring the cost of using more computationally expensive DNS.

FUTURE WORK

This model could be expanded to other material parameters and superelastic materials. The studies in this thesis utilized the material parameters used to verify the original material model. Future simulations could utilize a wide scope of SMA material models of varying properties. These results should be validated with a DNS model before any experimentation. From there, these structures could be created via AM or a wire-woven process and validated experimentally. These concepts could be applied to other bending-dominated structures and other superelastic materials. This work only provided information on relatively small deformations of the structure as compared to full compaction of the LSM, however, DNS with contact and post-buckling analysis as well as experimental work would provide information for higher displacements. These results and methods could be utilized in a wide variety of applications such as noise suppression from the rapid expansion of gas, e.g., a firearm suppressor or a car muffler. They could also be used in the design space of mediating a sonic boom in aerospace applications.

BIBLIOGRAPHY

[1] J. Marx, M. Portanova, and A. Rabiei, "Performance of composite metal foam armors against various threat sizes," *J. Compos. Sci.*, vol. 4, no. 4, 2020.

[2] M. C. Messner, "Optimal lattice-structured materials," *J. Mech. Phys. Solids*, vol. 96, pp. 162–183, 2016.

[3] C. G. Ferro, S. Varetti, G. De Pasquale, and P. Maggiore, "Lattice structured impact absorber with embedded anti-icing system for aircraft wings fabricated with additive SLM process," *Mater. Today Commun.*, vol. 15, no. February, pp. 185–189, 2018.

[4] Z. Rao *et al.*, "Experimental and numerical studies on a novel shape-memory alloy wire–woven trusses capable of undergoing large deformation," *J. Intell. Mater. Syst. Struct.*, vol. 30, no. 15, pp. 2283–2298, 2019.

[5] M. Mazur, M. Leary, S. Sun, M. Vcelka, D. Shidid, and M. Brandt, "Deformation and failure behaviour of Ti-6Al-4V lattice structures manufactured by selective laser melting (SLM)," *Int. J. Adv. Manuf. Technol.*, vol. 84, no. 5–8, pp. 1391–1411, 2016.

[6] L. E. Murr *et al.*, "Metal Fabrication by Additive Manufacturing Using Laser and Electron Beam Melting Technologies," *J. Mater. Sci. Technol*, vol. 28, no. 1, pp. 1–14, 2012.

[7] V. S. Deshpande, M. F. Ashby, and N. A. Fleck, "Foam topology: Bending versus

stretching dominated architectures," *Acta Mater.*, vol. 49, no. 6, pp. 1035–1040, 2001.

[8] S. Babaee, B. H. Jahromi, A. Ajdari, H. Nayeb-Hashemi, and A. Vaziri, "Mechanical properties of open-cell rhombic dodecahedron cellular structures," *Acta Mater.*, vol. 60, no. 6–7, pp. 2873–2885, 2012.

[9] L. C. Brinson, "One-dimensional constitutive behavior of shape memory alloys: Thermomechanical derivation with non-constant material functions and redefined martensite internal variable," *J. Intell. Mater. Syst. Struct.*, vol. 4, no. 2, pp. 229–242, 1993.

[10] F. Auricchio and R. L. Taylor, "Shape-memory alloys: Modelling and numerical simulations of the finite-strain superelastic behavior," *Comput. Methods Appl. Mech. Eng.*, pp. 175–194, 1997.

[11] F. Auricchio, R. L. Taylor, and J. Lubliner, "SHAPE-MEMORY ALLOYS: macromodelling and numerical simulations of the superelastic behavior," *Comput. Methods Appl. Mech. Engrg*, vol. 146, pp. 281–312, 1997.

[12] H. U. Schuerch, "Certain Physical Properties and Applications of Nitinol," 1968.

[13] M. C. Piedboeuf and R. Guavin, "Damping Behaviour of Shape Memory Alloys: Strain Amplitude, Frequency and Temperature Effects," *J. Sound Vib.*, vol. 214, no. 5, pp. 885–901, 1998.

[14] Y. Gao, L. Casalena, M. L. Bowers, R. D. Noebe, M. J. Mills, and Y. Wang, "An origin of functional fatigue of shape memory alloys," *Acta Mater.*, vol. 126, pp.

389–400, Mar. 2017.

[15] Dassault-Systèmes, *ABAQUS Theory Manual: Superelasticity*. United States: Dassault Systèmes Simulia Corp, 2018.

[16] A. Ahmed, S. Arya, V. Gupta, H. Furukawa, and A. Khosla, “4D printing: Fundamentals, materials, applications and challenges,” *Polymer (Guildf).*, vol. 228, Jul. 2021.

[17] T. Belytschko, W. K. Kam, B. Moran, and K. I. Elkhodary, “Beams and Shells,” in *Nonlinear Finite Elements For Continua and Structures*, Second Edi., West Sussex, UK: John Wiley & Sons, Ltd., 2014, pp. 535–596.

[18] Dassault-Systèmes, *ABAQUS Theory Manual: Beam Elements*. United States: Dassault Systèmes Simulia Corp, 2018.

[19] Dassault-Systèmes, *ABAQUS Theory Manual: Three-dimensional solid element library*. United States: Dassault Systèmes Simulia Corp, 2018.

[20] Dassault-Systèmes, “ABAQUS Theory Manual: Finding the Maximum Value of Von Mises Stress,” 2018. [Online]. Available: https://help.3ds.com/2018/english/dssimulia_established/simacaecmdrefmap/simacmd-c-odbintroexamaxmisespyc.htm?contextscope=all.

APPENDIX A

This appendix shows all the plots from the third energy dissipation study for all displacements used. Omitted data points in plots are the simulations where buckling is indicated or the peak von-Mises Stress exceeds the limit set. All simulations for loading in the y-direction for displacements greater than 5% were not included as either the peak von-Mises stress was exceed or the simulation indicated buckling.

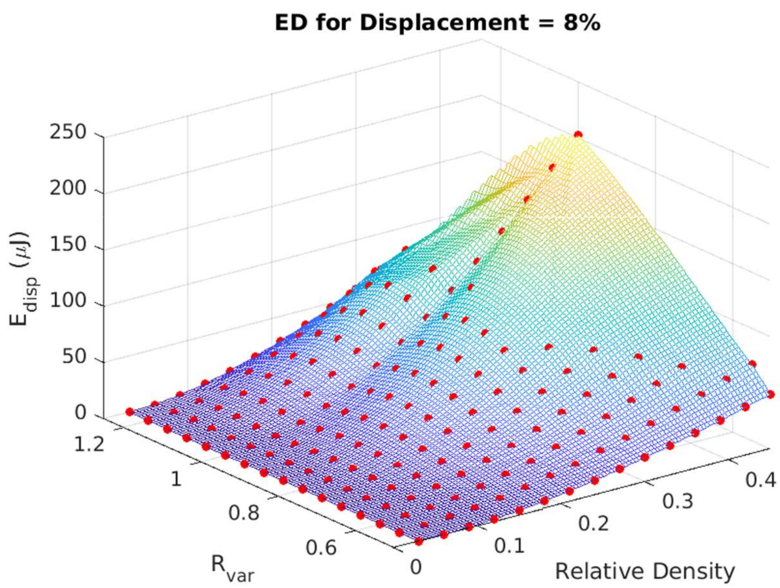

Figure 30: Energy Dissipated plotted against R_{var} and relative density for 8% displacement in x-direction. Simulations that exceeded stress limit or that indicated buckling were not included in this plot.

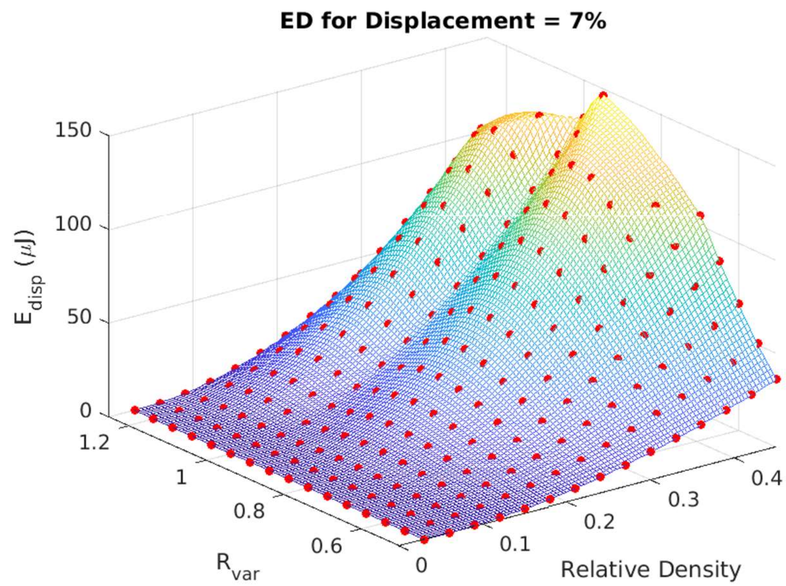

Figure 31: Energy Dissipated plotted against R_{var} and relative density for 7% displacement in x-direction. Simulations that exceeded stress limit or that indicated buckling were not included in this plot.

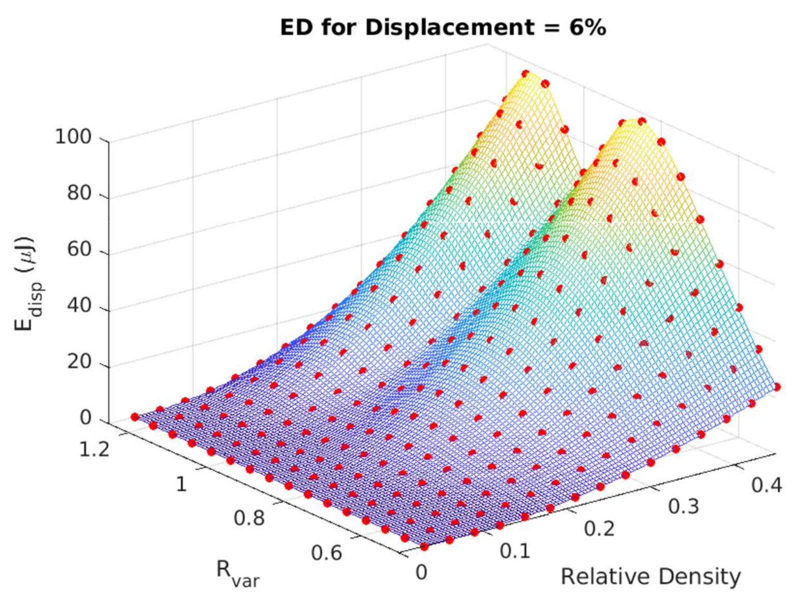

Figure 32: Energy Dissipated plotted against R_{var} and relative density for 6% displacement in x-direction. Simulations that exceeded stress limit or that indicated buckling were not included in this plot.

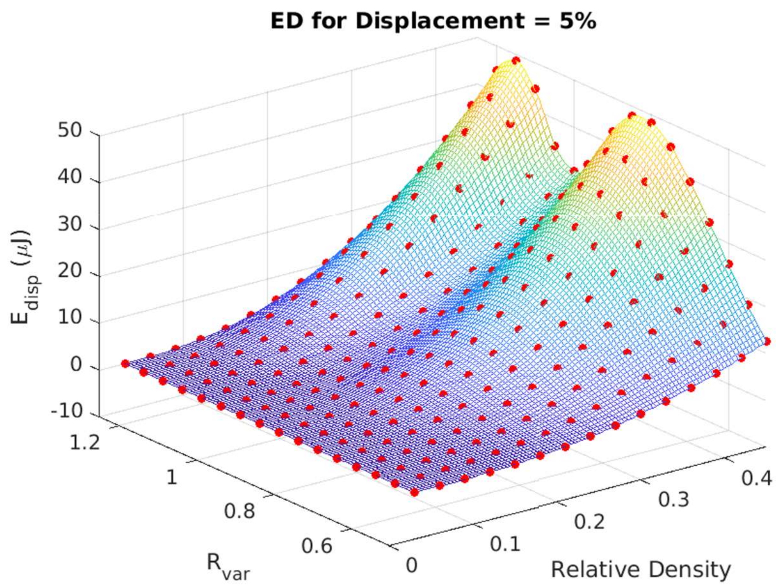

Figure 33: Energy Dissipated plotted against R_{var} and relative density for 5% displacement in x-direction. Simulations that exceeded stress limit or that indicated buckling were not included in this plot.

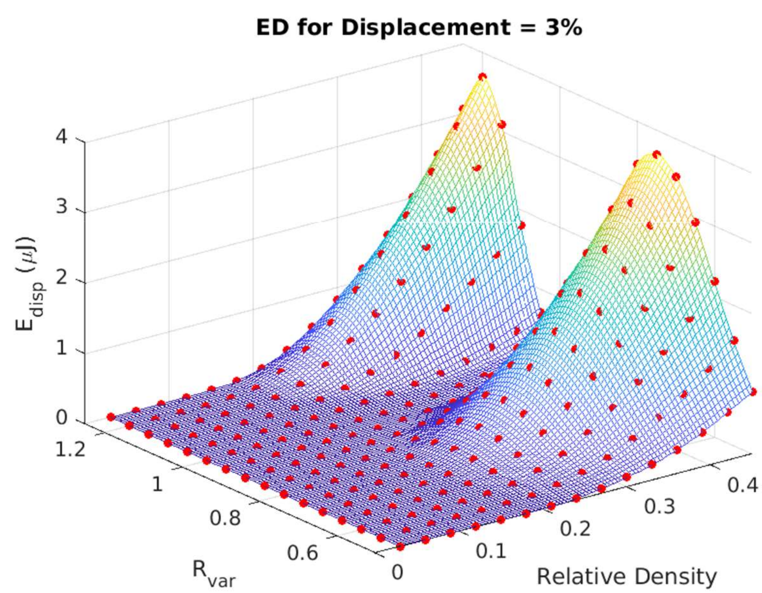

Figure 34: Energy Dissipated plotted against R_{var} and relative density for 3% displacement in x-direction.

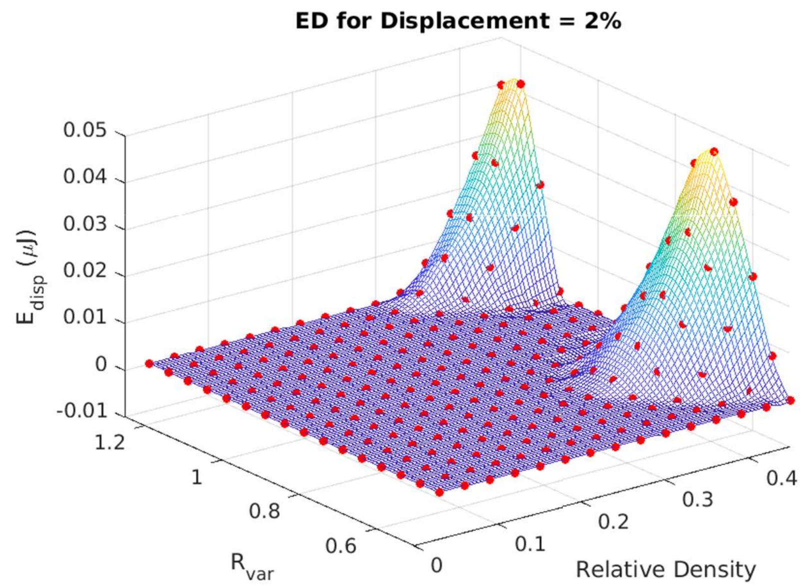

Figure 35: Energy Dissipated plotted against R_{var} and relative density for 2% displacement in x-direction.

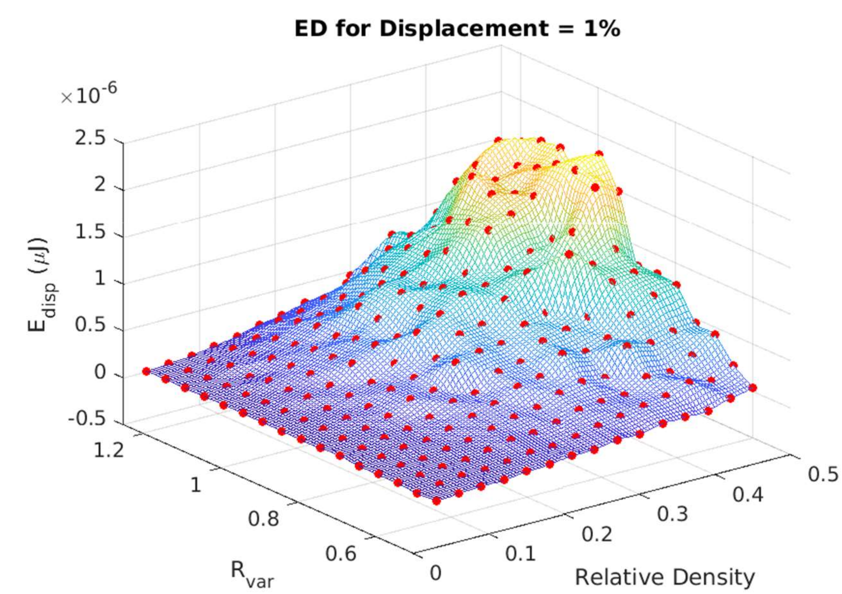

Figure 36: Energy Dissipated plotted against R_{var} and relative density for 2% displacement in x-direction.

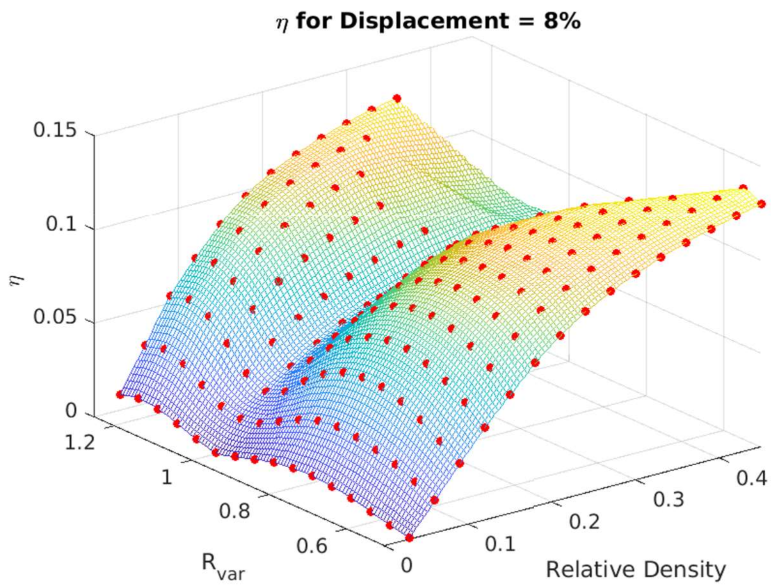

Figure 37: Energy Dissipation Coefficient plotted against R_{var} and relative density for 8% displacement in x-direction. Simulations that exceeded stress limit or that indicated buckling were not included in this plot.

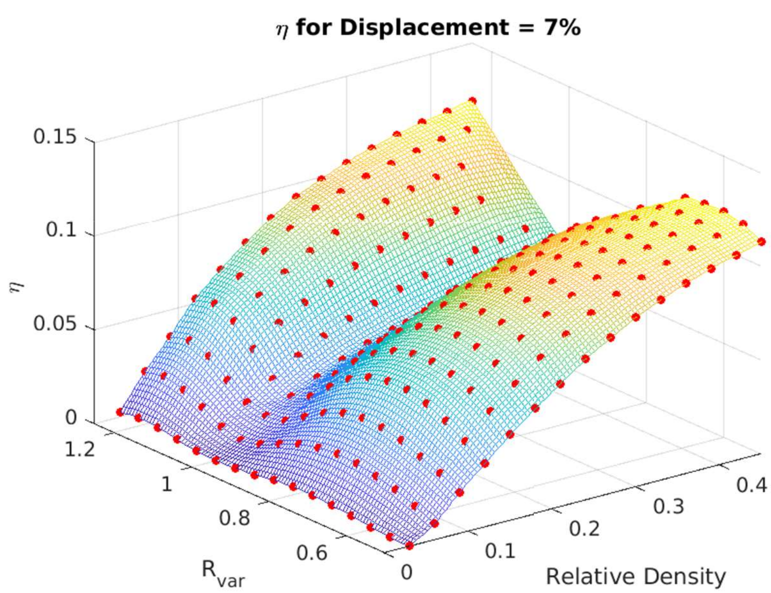

Figure 38: Energy Dissipation Coefficient plotted against R_{var} and relative density for 7% displacement in x-direction

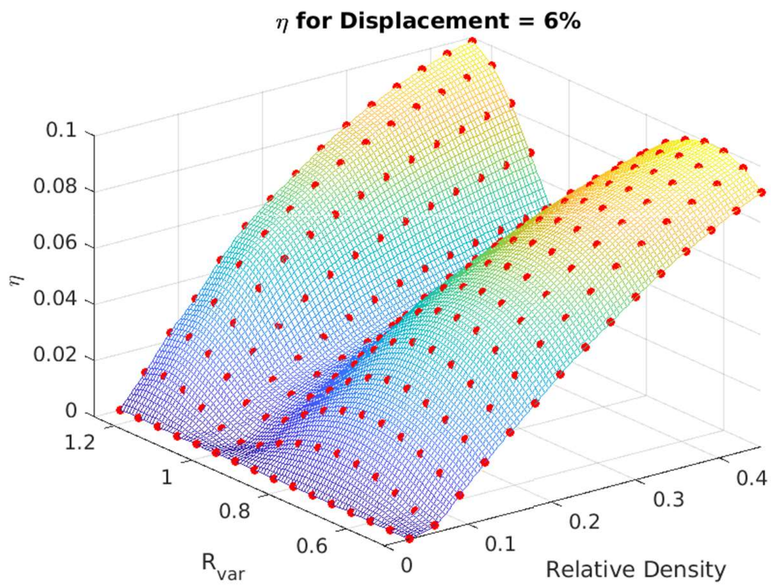

Figure 39: Energy Dissipation Coefficient plotted against R_{var} and relative density for 6% displacement in x-direction

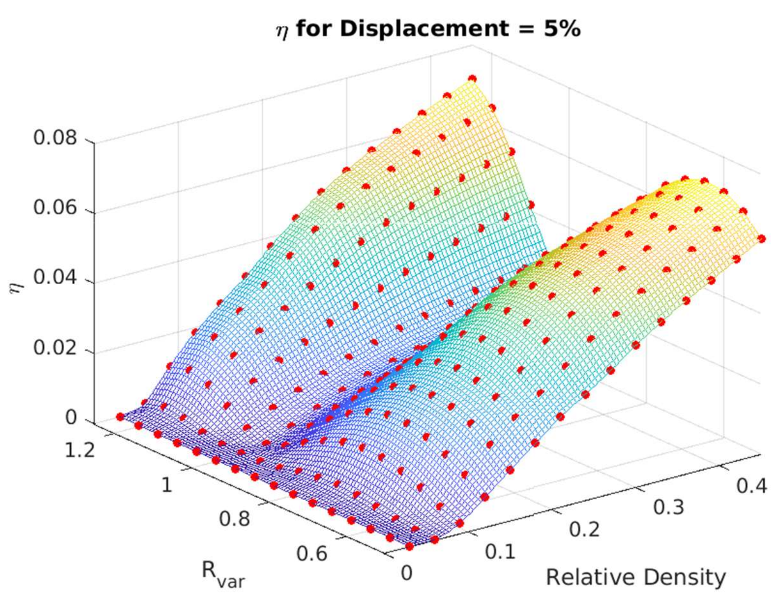

Figure 40: Energy Dissipation Coefficient plotted against R_{var} and relative density for 5% displacement in x-direction

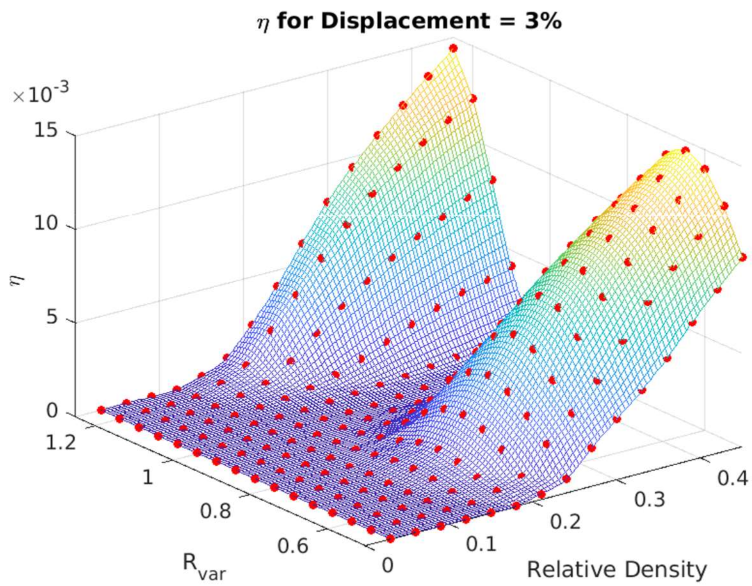

Figure 41: Energy Dissipation Coefficient plotted against R_{var} and relative density for 3% displacement in x-direction

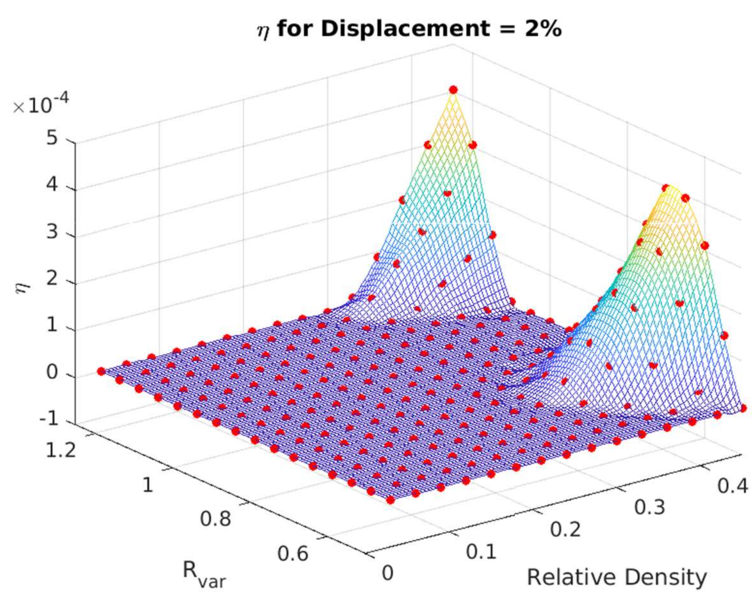

Figure 42: Energy Dissipation Coefficient plotted against R_{var} and relative density for 2% displacement in x-direction

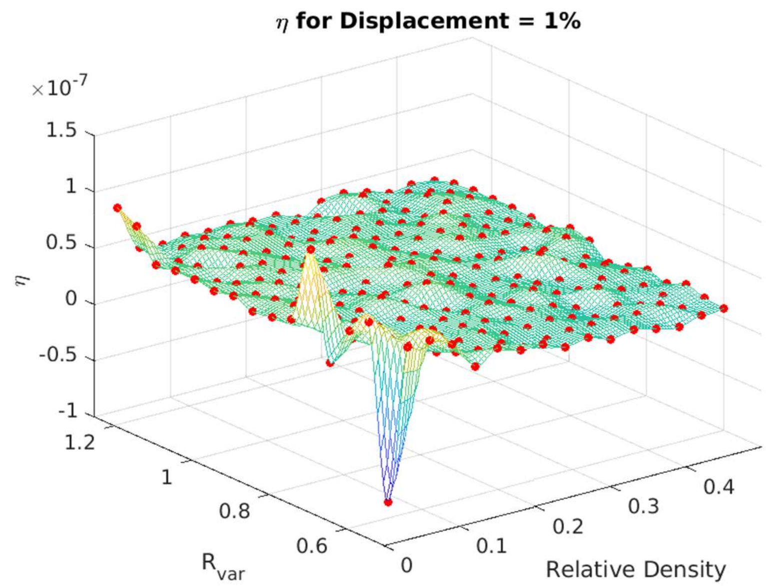

Figure 43: Energy Dissipation Coefficient plotted against R_{var} and relative density for 1% displacement in x-direction. All values of η are effectively zero.

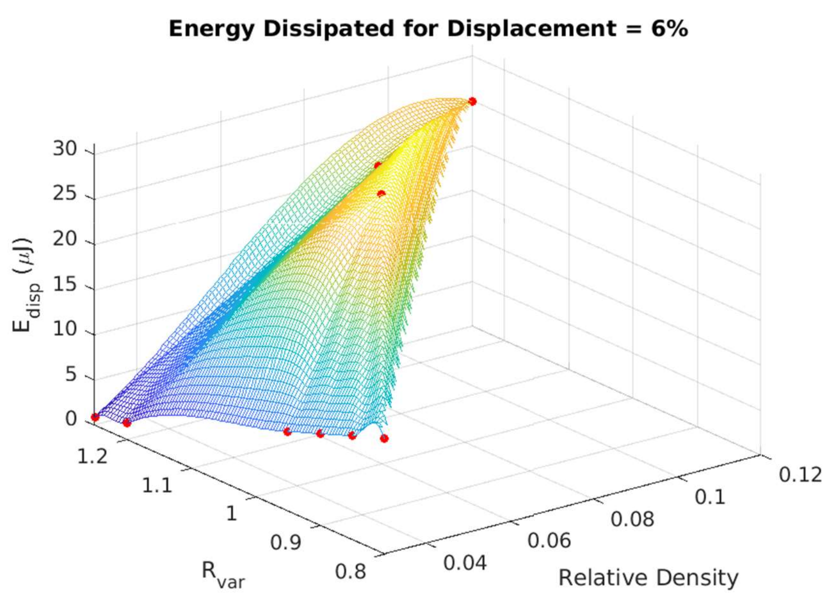

Figure 44: Energy Dissipated plotted against R_{var} and the relative density for 6% displacement in y-direction

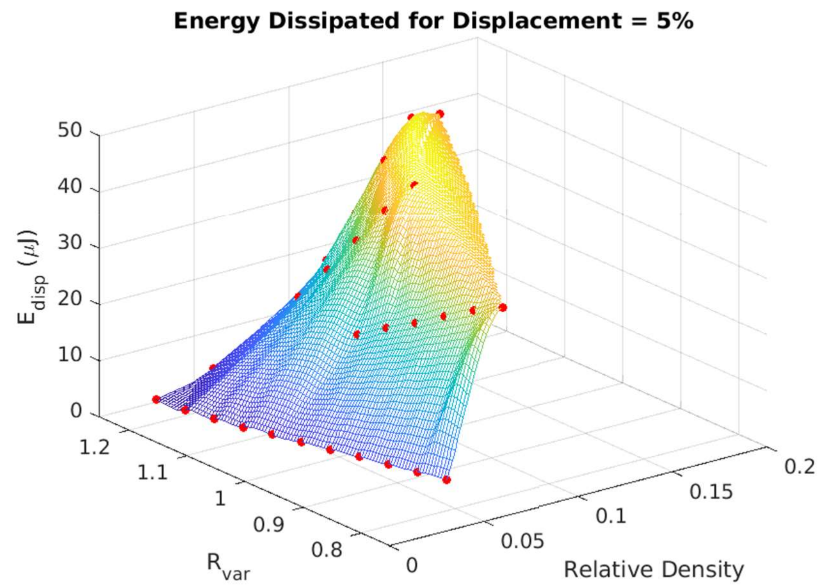

Figure 45: Energy Dissipated plotted against R_{var} and relative density for 5% displacement in y-direction

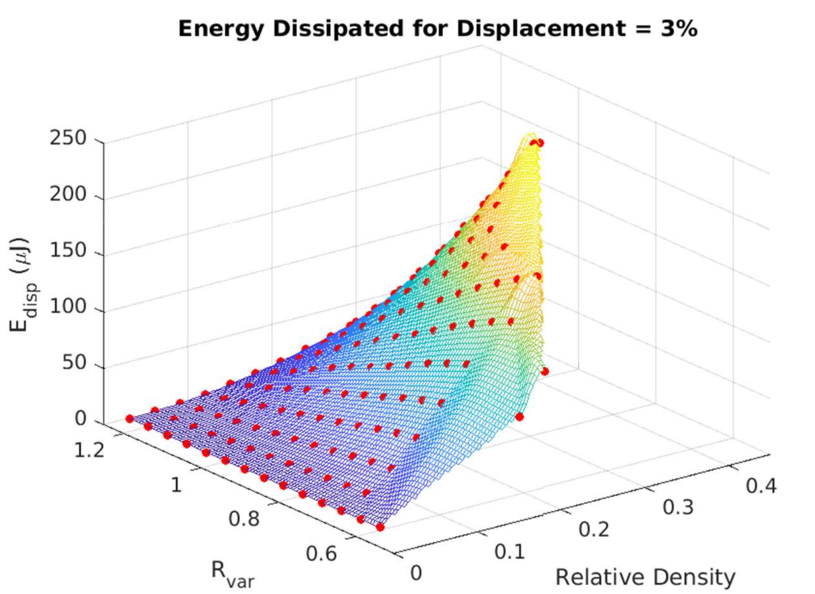

Figure 46: Energy Dissipated plotted against R_{var} and relative density for 3% displacement in y-direction

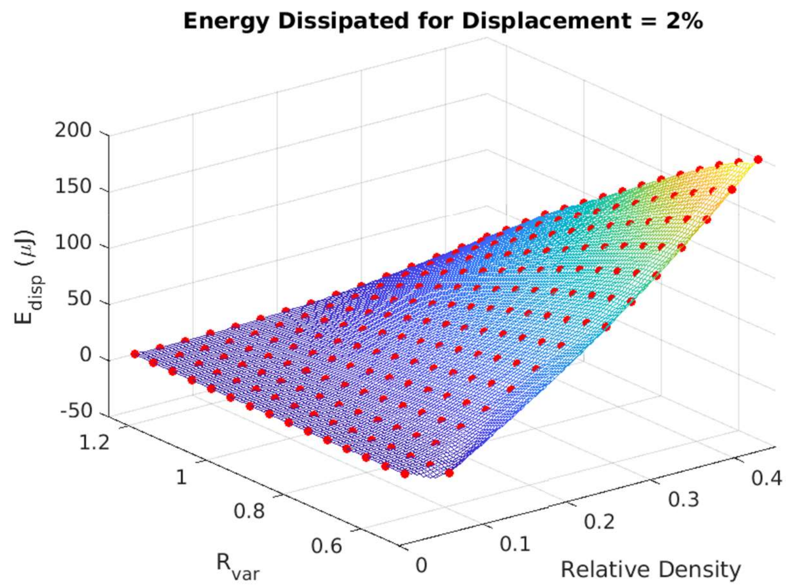

Figure 47: Energy Dissipated plotted against R_{var} and relative density for 2% displacement in y-direction

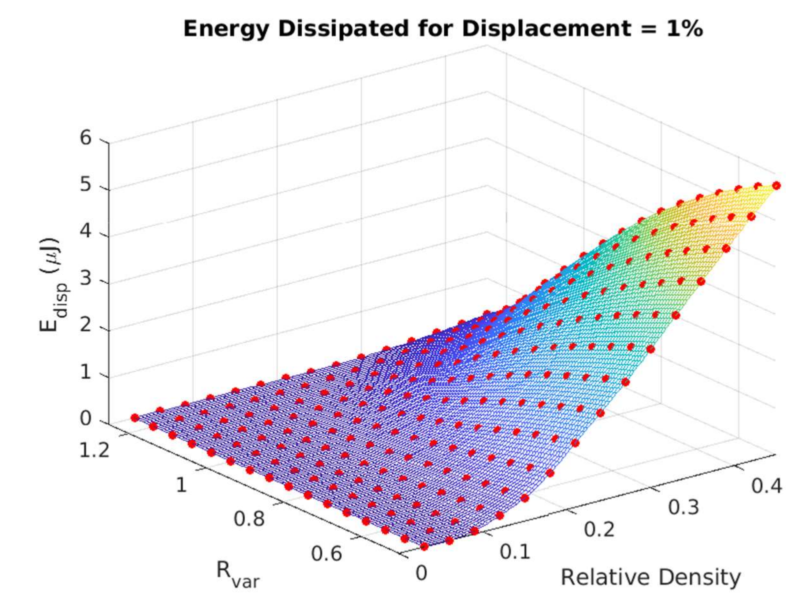

Figure 48: Energy Dissipated plotted against R_{var} and relative density for 1% displacement in y-direction

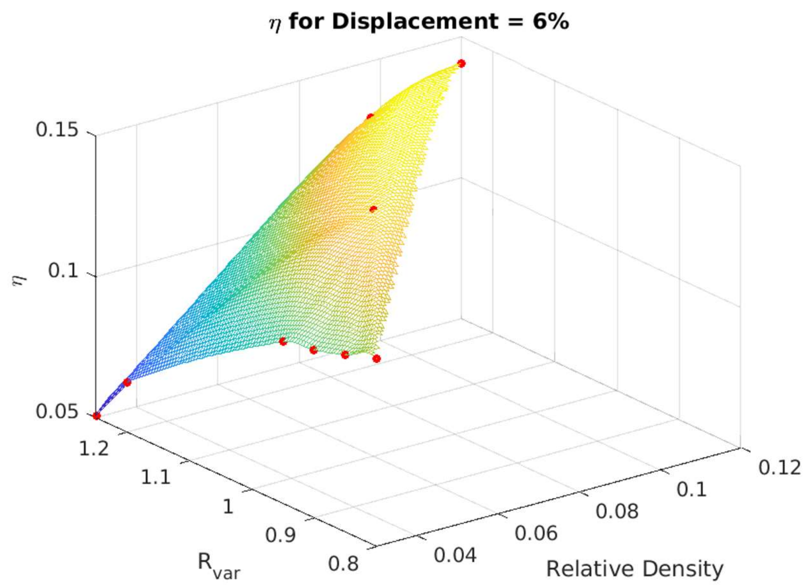

Figure 49: Energy Dissipation Coefficient plotted against R_{var} and relative density for 6% displacement in y-direction

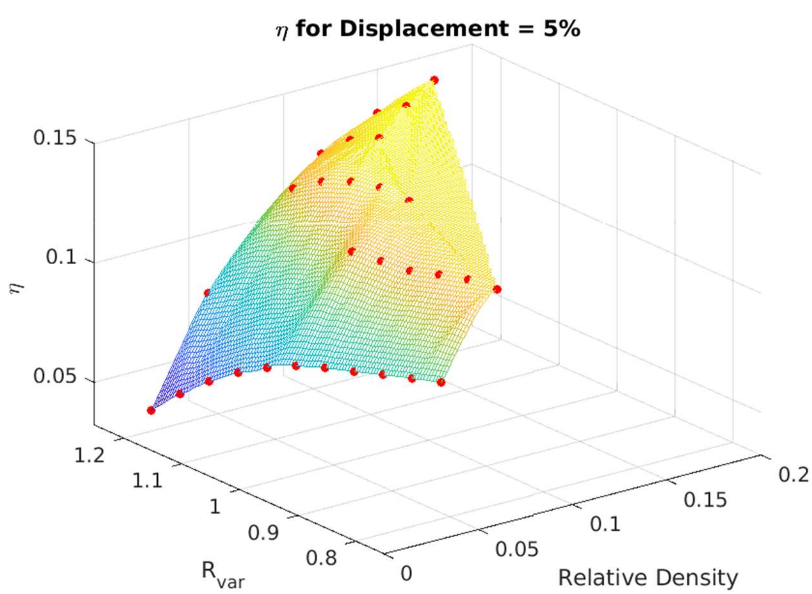

Figure 50: Energy Dissipation Coefficient plotted against R_{var} and relative density for 5% displacement in y-direction

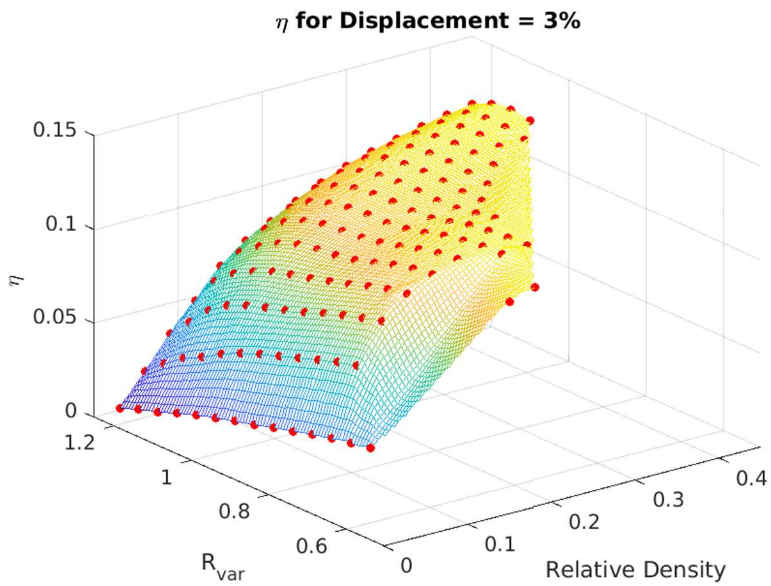

Figure 51: Energy Dissipation Coefficient plotted against R_{var} and relative density for 3% displacement in y-direction

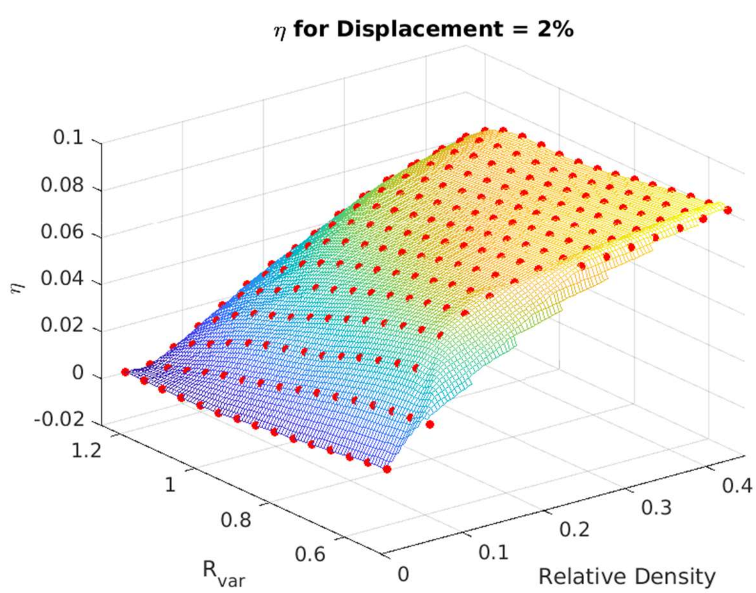

Figure 52: Energy Dissipation Coefficient plotted against R_{var} and relative density for 2% displacement in y-direction

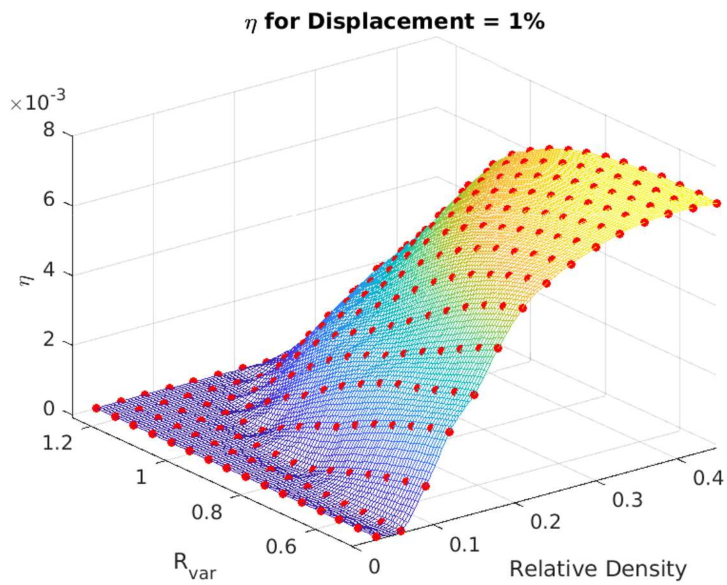

Figure 53: Energy Dissipation Coefficient plotted against R_{var} and relative density for 1% displacement in y-direction

APPENDIX B

This appendix shows examples of the python scripts that were used to write and submit input decks (.inp files), vary parameters, and post-process from outputs (.odb and .msg files) with ABAQUS.

I. writeinput.py

```
# Author: Ian P. Morrissey
# Description: writes ABAQUS .inp files for a range of displacements, relative density, and radius variation

import math
import numpy
# displacement magnitude
d1L=5.7735 # direction 1 or x-direction
dispMag=d1L*numpy.linspace(-.08,-.01,8)

# relative density
rd=numpy.linspace(0.03,0.45,15)
print('rd='+str(rd))

# calculate nominal radius based on r
r=numpy.sqrt((rd*2*pow(5,2))/(3*numpy.pi*numpy.sqrt(3)))
print('r='+str(r))

# calculate total volume for unit cell
totVol=32/9*numpy.sqrt(3)*pow(5,3)
print('totVol='+str(totVol))

# calculate volume of occupied
volume=24*pow(r,2)*numpy.pi*5
print('volume='+str(volume))

# radius variation coefficient
rvar=numpy.linspace(0.5,1.25,16)
print('rvar='+str(rvar))

# number of loads to index by
numLoads=len(dispMag)
print('numLoads='+str(numLoads))

# number of relative desnities to index by
numrd=len(rd)
print('numrd='+str(numrd))

```

```
# number of radius variation coefficient to vary by
numrvar=len(rvar)
print('numrvar='+str(numrvar))

# index over number of loads/displacements, relative densities, and radius variations
for j in range(numLoads):
   for i in range(numrd):
      for k in range(numrvar):
       rh=r[i]*rvar[k] #calculate horizontal radius
       rv=numpy.sqrt((volume[i]-12*5*numpy.pi*pow(rh,2))/(12*5*numpy.pi)) #calculate vertical radius

#inp file name
       filename='D1B-'+str(j+1)+'-'+str(i+1)+'-'+str(k+1)
#write inp file
       fileOutput = open(filename + '.inp','w')
       fileOutput.write("""*Heading
** Job name: BD1ex Model name: BD1ex
** Generated by: Abaqus/CAE 2019
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=PART-1
*Node
      1, 2.887000145, 4.08250004, 0
      2, 5.7735002, 4.08250004, 4.08250004
      3, 2.887000145, 0, 4.08250004
      4, 0, 0, 8.1650001
      5, 2.887000145, 4.08250004, 8.1650001
      6, 2.887000145, 8.1650001, 4.08250004
      7, 0, 8.1650001, 8.1650001
      8, 0, 0, 0
      9, 0, 8.1650001, 0
      10, -2.887000145, 4.08250004, 0
      11, -2.887000145, 0, 4.08250004
      12, -5.7735002, 4.08250004, 4.08250004
      13, -2.887000145, 4.08250004, 8.1650001
      14, -2.887000145, 8.1650001, 4.08250004
      15, 3.03132504, 4.08250004, 0.204124991
      16, 3.17564994, 4.08250004, 0.4082499815
      17, 3.31997514, 4.08250004, 0.61237499
      18, 3.464300035, 4.08250004, 0.816499965
      19, 3.608624935, 4.08250004, 1.02062501
      20, 3.75295013, 4.08250004, 1.22474998
      21, 3.89727503, 4.08250004, 1.42887503
      22, 4.04159993, 4.08250004, 1.632999925
      23, 4.185925125, 4.08250004, 1.837124975
      24, 4.330250025, 4.08250004, 2.04125002
      25, 4.474574925, 4.08250004, 2.245375065
      26, 4.61890012, 4.08250004, 2.449499965
      27, 4.76322502, 4.08250004, 2.65362501
      28, 4.90754992, 4.08250004, 2.85775006
      29, 5.0518751, 4.08250004, 3.061875105
      30, 5.1962, 4.08250004, 3.265999855
```

```
    31, 5.3405249, 4.08250004, 3.4701249
    32, 5.4848498, 4.08250004, 3.674249945
    33, 5.6291747, 4.08250004, 3.878374995
    34, 5.6291747, 3.878374995, 4.08250004
    35, 5.4848498, 3.674249945, 4.08250004
    36, 5.3405249, 3.4701249, 4.08250004
    37, 5.1962, 3.265999855, 4.08250004
    38, 5.0518751, 3.061875105, 4.08250004
    39, 4.90754992, 2.85775006, 4.08250004
    40, 4.76322502, 2.65362501, 4.08250004
    41, 4.61890012, 2.449499965, 4.08250004
    42, 4.474574925, 2.245375065, 4.08250004
    43, 4.330250025, 2.04125002, 4.08250004
    44, 4.185925125, 1.837124975, 4.08250004
    45, 4.04159993, 1.632999925, 4.08250004
    46, 3.89727503, 1.42887503, 4.08250004
    47, 3.75295013, 1.22474998, 4.08250004
    48, 3.608624935, 1.02062501, 4.08250004
    49, 3.464300035, 0.816499965, 4.08250004
    50, 3.31997514, 0.61237499, 4.08250004
    51, 3.17564994, 0.4082499815, 4.08250004
    52, 3.03132504, 0.204124991, 4.08250004
    53, 2.742649915, 0, 4.286625085
    54, 2.59829998, 0, 4.490750135
    55, 2.453950045, 0, 4.694874885
    56, 2.309599965, 0, 4.89899993
    57, 2.165250035, 0, 5.103125
    58, 2.02089995, 0, 5.30725
    59, 1.87655002, 0, 5.51137505
    60, 1.732199935, 0, 5.7155001
    61, 1.587850005, 0, 5.91962515
    62, 1.44350007, 0, 6.1237502
    63, 1.29914999, 0, 6.32787525
    64, 1.154799985, 0, 6.5319997
    65, 1.010449975, 0, 6.73612475
    66, 0.86609997, 0, 6.9402498
    67, 0.721750035, 0, 7.14437485
    68, 0.57739999, 0, 7.3484999
    69, 0.4330499845, 0, 7.55262495
    70, 0.2886999955, 0, 7.75675
    71, 0.144349998, 0, 7.96087505
    72, 2.742649915, 3.878374995, 8.1650001
    73, 2.59829998, 3.674249945, 8.1650001
    74, 2.453950045, 3.4701249, 8.1650001
    75, 2.309599965, 3.265999855, 8.1650001
    76, 2.165250035, 3.061875105, 8.1650001
    77, 2.02089995, 2.85775006, 8.1650001
    78, 1.87655002, 2.65362501, 8.1650001
    79, 1.732199935, 2.449499965, 8.1650001
    80, 1.587850005, 2.245375065, 8.1650001
    81, 1.44350007, 2.04125002, 8.1650001
    82, 1.29914999, 1.837124975, 8.1650001
    83, 1.154799985, 1.632999925, 8.1650001
    84, 1.010449975, 1.42887503, 8.1650001
    85, 0.86609997, 1.22474998, 8.1650001
    86, 0.721750035, 1.02062501, 8.1650001
    87, 0.57739999, 0.816499965, 8.1650001
```

```
        88, 0.4330499845, 0.61237499, 8.1650001
        89, 0.2886999955, 0.4082499815, 8.1650001
        90, 0.144349998, 0.204124991, 8.1650001
        91, 3.03132504, 4.08250004, 7.96087505
        92, 3.17564994, 4.08250004, 7.75675
        93, 3.31997514, 4.08250004, 7.55262495
        94, 3.464300035, 4.08250004, 7.3484999
        95, 3.608624935, 4.08250004, 7.14437485
        96, 3.75295013, 4.08250004, 6.9402498
        97, 3.89727503, 4.08250004, 6.73612475
        98, 4.04159993, 4.08250004, 6.5319997
        99, 4.185925125, 4.08250004, 6.32787525
        100, 4.330250025, 4.08250004, 6.1237502
        101, 4.474574925, 4.08250004, 5.91962515
        102, 4.61890012, 4.08250004, 5.7155001
        103, 4.76322502, 4.08250004, 5.51137505
        104, 4.90754992, 4.08250004, 5.30725
        105, 5.0518751, 4.08250004, 5.103125
        106, 5.1962, 4.08250004, 4.89899993
        107, 5.3405249, 4.08250004, 4.694874885
        108, 5.4848498, 4.08250004, 4.490750135
        109, 5.6291747, 4.08250004, 4.286625085
        110, 3.03132504, 7.96087505, 4.08250004
        111, 3.17564994, 7.75675, 4.08250004
        112, 3.31997514, 7.55262495, 4.08250004
        113, 3.464300035, 7.3484999, 4.08250004
        114, 3.608624935, 7.14437485, 4.08250004
        115, 3.75295013, 6.9402498, 4.08250004
        116, 3.89727503, 6.73612475, 4.08250004
        117, 4.04159993, 6.5319997, 4.08250004
        118, 4.185925125, 6.32787525, 4.08250004
        119, 4.330250025, 6.1237502, 4.08250004
        120, 4.474574925, 5.91962515, 4.08250004
        121, 4.61890012, 5.7155001, 4.08250004
        122, 4.76322502, 5.51137505, 4.08250004
        123, 4.90754992, 5.30725, 4.08250004
        124, 5.0518751, 5.103125, 4.08250004
        125, 5.1962, 4.89899993, 4.08250004
        126, 5.3405249, 4.694874885, 4.08250004
        127, 5.4848498, 4.490750135, 4.08250004
        128, 5.6291747, 4.286625085, 4.08250004
        129, 2.742649915, 8.1650001, 4.286625085
        130, 2.59829998, 8.1650001, 4.490750135
        131, 2.453950045, 8.1650001, 4.694874885
        132, 2.309599965, 8.1650001, 4.89899993
        133, 2.165250035, 8.1650001, 5.103125
        134, 2.02089995, 8.1650001, 5.30725
        135, 1.87655002, 8.1650001, 5.51137505
        136, 1.732199935, 8.1650001, 5.7155001
        137, 1.587850005, 8.1650001, 5.91962515
        138, 1.44350007, 8.1650001, 6.1237502
        139, 1.29914999, 8.1650001, 6.32787525
        140, 1.154799985, 8.1650001, 6.5319997
        141, 1.010449975, 8.1650001, 6.73612475
        142, 0.86609997, 8.1650001, 6.9402498
        143, 0.721750035, 8.1650001, 7.14437485
        144, 0.57739999, 8.1650001, 7.3484999
```

```
        145, 0.4330499845, 8.1650001, 7.55262495
        146, 0.2886999955, 8.1650001, 7.75675
        147, 0.144349998, 8.1650001, 7.96087505
        148, 2.742649915, 4.286625085, 8.1650001
        149, 2.59829998, 4.490750135, 8.1650001
        150, 2.453950045, 4.694874885, 8.1650001
        151, 2.309599965, 4.89899993, 8.1650001
        152, 2.165250035, 5.103125, 8.1650001
        153, 2.02089995, 5.30725, 8.1650001
        154, 1.87655002, 5.51137505, 8.1650001
        155, 1.732199935, 5.7155001, 8.1650001
        156, 1.587850005, 5.91962515, 8.1650001
        157, 1.44350007, 6.1237502, 8.1650001
        158, 1.29914999, 6.32787525, 8.1650001
        159, 1.154799985, 6.5319997, 8.1650001
        160, 1.010449975, 6.73612475, 8.1650001
        161, 0.86609997, 6.9402498, 8.1650001
        162, 0.721750035, 7.14437485, 8.1650001
        163, 0.57739999, 7.3484999, 8.1650001
        164, 0.4330499845, 7.55262495, 8.1650001
        165, 0.2886999955, 7.75675, 8.1650001
        166, 0.144349998, 7.96087505, 8.1650001
        167, 0.144349998, 0.204124991, 0
        168, 0.2886999955, 0.4082499815, 0
        169, 0.4330499845, 0.61237499, 0
        170, 0.57739999, 0.816499965, 0
        171, 0.721750035, 1.02062501, 0
        172, 0.86609997, 1.22474998, 0
        173, 1.010449975, 1.42887503, 0
        174, 1.154799985, 1.632999925, 0
        175, 1.29914999, 1.837124975, 0
        176, 1.44350007, 2.04125002, 0
        177, 1.587850005, 2.245375065, 0
        178, 1.732199935, 2.449499965, 0
        179, 1.87655002, 2.65362501, 0
        180, 2.02089995, 2.85775006, 0
        181, 2.165250035, 3.061875105, 0
        182, 2.309599965, 3.265999855, 0
        183, 2.453950045, 3.4701249, 0
        184, 2.59829998, 3.674249945, 0
        185, 2.742649915, 3.878374995, 0
        186, 0.144349998, 7.96087505, 0
        187, 0.2886999955, 7.75675, 0
        188, 0.4330499845, 7.55262495, 0
        189, 0.57739999, 7.3484999, 0
        190, 0.721750035, 7.14437485, 0
        191, 0.86609997, 6.9402498, 0
        192, 1.010449975, 6.73612475, 0
        193, 1.154799985, 6.5319997, 0
        194, 1.29914999, 6.32787525, 0
        195, 1.44350007, 6.1237502, 0
        196, 1.587850005, 5.91962515, 0
        197, 1.732199935, 5.7155001, 0
        198, 1.87655002, 5.51137505, 0
        199, 2.02089995, 5.30725, 0
        200, 2.165250035, 5.103125, 0
        201, 2.309599965, 4.89899993, 0
```

```
    202, 2.453950045, 4.694874885, 0
    203, 2.59829998, 4.490750135, 0
    204, 2.742649915, 4.286625085, 0
    205, -2.742649915, 4.286625085, 0
    206, -2.59829998, 4.490750135, 0
    207, -2.453950045, 4.694874885, 0
    208, -2.309599965, 4.89899993, 0
    209, -2.165250035, 5.103125, 0
    210, -2.02089995, 5.30725, 0
    211, -1.87655002, 5.51137505, 0
    212, -1.732199935, 5.7155001, 0
    213, -1.587850005, 5.91962515, 0
    214, -1.44350007, 6.1237502, 0
    215, -1.29914999, 6.32787525, 0
    216, -1.154799985, 6.5319997, 0
    217, -1.010449975, 6.73612475, 0
    218, -0.86609997, 6.9402498, 0
    219, -0.721750035, 7.14437485, 0
    220, -0.57739999, 7.3484999, 0
    221, -0.4330499845, 7.55262495, 0
    222, -0.2886999955, 7.75675, 0
    223, -0.144349998, 7.96087505, 0
    224, -0.144349998, 0.204124991, 0
    225, -0.2886999955, 0.4082499815, 0
    226, -0.4330499845, 0.61237499, 0
    227, -0.57739999, 0.816499965, 0
    228, -0.721750035, 1.02062501, 0
    229, -0.86609997, 1.22474998, 0
    230, -1.010449975, 1.42887503, 0
    231, -1.154799985, 1.632999925, 0
    232, -1.29914999, 1.837124975, 0
    233, -1.44350007, 2.04125002, 0
    234, -1.587850005, 2.245375065, 0
    235, -1.732199935, 2.449499965, 0
    236, -1.87655002, 2.65362501, 0
    237, -2.02089995, 2.85775006, 0
    238, -2.165250035, 3.061875105, 0
    239, -2.309599965, 3.265999855, 0
    240, -2.453950045, 3.4701249, 0
    241, -2.59829998, 3.674249945, 0
    242, -2.742649915, 3.878374995, 0
    243, -0.144349998, 0, 0.204124991
    244, -0.2886999955, 0, 0.4082499815
    245, -0.4330499845, 0, 0.61237499
    246, -0.57739999, 0, 0.816499965
    247, -0.721750035, 0, 1.02062501
    248, -0.86609997, 0, 1.22474998
    249, -1.010449975, 0, 1.42887503
    250, -1.154799985, 0, 1.632999925
    251, -1.29914999, 0, 1.837124975
    252, -1.44350007, 0, 2.04125002
    253, -1.587850005, 0, 2.245375065
    254, -1.732199935, 0, 2.449499965
    255, -1.87655002, 0, 2.65362501
    256, -2.02089995, 0, 2.85775006
    257, -2.165250035, 0, 3.061875105
    258, -2.309599965, 0, 3.265999855
```

```
        259, -2.453950045, 0, 3.4701249
        260, -2.59829998, 0, 3.674249945
        261, -2.742649915, 0, 3.878374995
        262, -3.03132504, 0.204124991, 4.08250004
        263, -3.17564994, 0.4082499815, 4.08250004
        264, -3.31997514, 0.61237499, 4.08250004
        265, -3.464300035, 0.816499965, 4.08250004
        266, -3.608624935, 1.02062501, 4.08250004
        267, -3.75295013, 1.22474998, 4.08250004
        268, -3.89727503, 1.42887503, 4.08250004
        269, -4.04159993, 1.632999925, 4.08250004
        270, -4.185925125, 1.837124975, 4.08250004
        271, -4.330250025, 2.04125002, 4.08250004
        272, -4.474574925, 2.245375065, 4.08250004
        273, -4.61890012, 2.449499965, 4.08250004
        274, -4.76322502, 2.65362501, 4.08250004
        275, -4.90754992, 2.85775006, 4.08250004
        276, -5.0518751, 3.061875105, 4.08250004
        277, -5.1962, 3.265999855, 4.08250004
        278, -5.3405249, 3.4701249, 4.08250004
        279, -5.4848498, 3.674249945, 4.08250004
        280, -5.6291747, 3.878374995, 4.08250004
        281, -5.6291747, 4.08250004, 3.878374995
        282, -5.4848498, 4.08250004, 3.674249945
        283, -5.3405249, 4.08250004, 3.4701249
        284, -5.1962, 4.08250004, 3.265999855
        285, -5.0518751, 4.08250004, 3.061875105
        286, -4.90754992, 4.08250004, 2.85775006
        287, -4.76322502, 4.08250004, 2.65362501
        288, -4.61890012, 4.08250004, 2.449499965
        289, -4.474574925, 4.08250004, 2.245375065
        290, -4.330250025, 4.08250004, 2.04125002
        291, -4.185925125, 4.08250004, 1.837124975
        292, -4.04159993, 4.08250004, 1.632999925
        293, -3.89727503, 4.08250004, 1.42887503
        294, -3.75295013, 4.08250004, 1.22474998
        295, -3.608624935, 4.08250004, 1.02062501
        296, -3.464300035, 4.08250004, 0.816499965
        297, -3.31997514, 4.08250004, 0.61237499
        298, -3.17564994, 4.08250004, 0.4082499815
        299, -3.03132504, 4.08250004, 0.204124991
        300, 0.144349998, 0, 0.204124991
        301, 0.2886999955, 0, 0.4082499815
        302, 0.4330499845, 0, 0.61237499
        303, 0.57739999, 0, 0.816499965
        304, 0.721750035, 0, 1.02062501
        305, 0.86609997, 0, 1.22474998
        306, 1.010449975, 0, 1.42887503
        307, 1.154799985, 0, 1.632999925
        308, 1.29914999, 0, 1.837124975
        309, 1.44350007, 0, 2.04125002
        310, 1.587850005, 0, 2.245375065
        311, 1.732199935, 0, 2.449499965
        312, 1.87655002, 0, 2.65362501
        313, 2.02089995, 0, 2.85775006
        314, 2.165250035, 0, 3.061875105
        315, 2.309599965, 0, 3.265999855
```

```
316, 2.453950045, 0, 3.4701249
317, 2.59829998, 0, 3.674249945
318, 2.742649915, 0, 3.878374995
319, -2.742649915, 3.878374995, 8.1650001
320, -2.59829998, 3.674249945, 8.1650001
321, -2.453950045, 3.4701249, 8.1650001
322, -2.309599965, 3.265999855, 8.1650001
323, -2.165250035, 3.061875105, 8.1650001
324, -2.02089995, 2.85775006, 8.1650001
325, -1.87655002, 2.65362501, 8.1650001
326, -1.732199935, 2.449499965, 8.1650001
327, -1.587850005, 2.245375065, 8.1650001
328, -1.44350007, 2.04125002, 8.1650001
329, -1.29914999, 1.837124975, 8.1650001
330, -1.154799985, 1.632999925, 8.1650001
331, -1.010449975, 1.42887503, 8.1650001
332, -0.86609997, 1.22474998, 8.1650001
333, -0.721750035, 1.02062501, 8.1650001
334, -0.57739999, 0.816499965, 8.1650001
335, -0.4330499845, 0.61237499, 8.1650001
336, -0.2886999955, 0.4082499815, 8.1650001
337, -0.144349998, 0.204124991, 8.1650001
338, -0.144349998, 7.96087505, 8.1650001
339, -0.2886999955, 7.75675, 8.1650001
340, -0.4330499845, 7.55262495, 8.1650001
341, -0.57739999, 7.3484999, 8.1650001
342, -0.721750035, 7.14437485, 8.1650001
343, -0.86609997, 6.9402498, 8.1650001
344, -1.010449975, 6.73612475, 8.1650001
345, -1.154799985, 6.5319997, 8.1650001
346, -1.29914999, 6.32787525, 8.1650001
347, -1.44350007, 6.1237502, 8.1650001
348, -1.587850005, 5.91962515, 8.1650001
349, -1.732199935, 5.7155001, 8.1650001
350, -1.87655002, 5.51137505, 8.1650001
351, -2.02089995, 5.30725, 8.1650001
352, -2.165250035, 5.103125, 8.1650001
353, -2.309599965, 4.89899993, 8.1650001
354, -2.453950045, 4.694874885, 8.1650001
355, -2.59829998, 4.490750135, 8.1650001
356, -2.742649915, 4.286625085, 8.1650001
357, 0.144349998, 8.1650001, 0.204124991
358, 0.2886999955, 8.1650001, 0.4082499815
359, 0.4330499845, 8.1650001, 0.61237499
360, 0.57739999, 8.1650001, 0.816499965
361, 0.721750035, 8.1650001, 1.02062501
362, 0.86609997, 8.1650001, 1.22474998
363, 1.010449975, 8.1650001, 1.42887503
364, 1.154799985, 8.1650001, 1.632999925
365, 1.29914999, 8.1650001, 1.837124975
366, 1.44350007, 8.1650001, 2.04125002
367, 1.587850005, 8.1650001, 2.245375065
368, 1.732199935, 8.1650001, 2.449499965
369, 1.87655002, 8.1650001, 2.65362501
370, 2.02089995, 8.1650001, 2.85775006
371, 2.165250035, 8.1650001, 3.061875105
372, 2.309599965, 8.1650001, 3.265999855
```

```
        373, 2.453950045, 8.1650001, 3.4701249
        374, 2.59829998, 8.1650001, 3.674249945
        375, 2.742649915, 8.1650001, 3.878374995
        376, -0.144349998, 8.1650001, 0.204124991
        377, -0.2886999955, 8.1650001, 0.4082499815
        378, -0.4330499845, 8.1650001, 0.61237499
        379, -0.57739999, 8.1650001, 0.816499965
        380, -0.721750035, 8.1650001, 1.02062501
        381, -0.86609997, 8.1650001, 1.22474998
        382, -1.010449975, 8.1650001, 1.42887503
        383, -1.154799985, 8.1650001, 1.632999925
        384, -1.29914999, 8.1650001, 1.837124975
        385, -1.44350007, 8.1650001, 2.04125002
        386, -1.587850005, 8.1650001, 2.245375065
        387, -1.732199935, 8.1650001, 2.449499965
        388, -1.87655002, 8.1650001, 2.65362501
        389, -2.02089995, 8.1650001, 2.85775006
        390, -2.165250035, 8.1650001, 3.061875105
        391, -2.309599965, 8.1650001, 3.265999855
        392, -2.453950045, 8.1650001, 3.4701249
        393, -2.59829998, 8.1650001, 3.674249945
        394, -2.742649915, 8.1650001, 3.878374995
        395, -5.6291747, 4.286625085, 4.08250004
        396, -5.4848498, 4.490750135, 4.08250004
        397, -5.3405249, 4.694874885, 4.08250004
        398, -5.1962, 4.89899993, 4.08250004
        399, -5.0518751, 5.103125, 4.08250004
        400, -4.90754992, 5.30725, 4.08250004
        401, -4.76322502, 5.51137505, 4.08250004
        402, -4.61890012, 5.7155001, 4.08250004
        403, -4.474574925, 5.91962515, 4.08250004
        404, -4.330250025, 6.1237502, 4.08250004
        405, -4.185925125, 6.32787525, 4.08250004
        406, -4.04159993, 6.5319997, 4.08250004
        407, -3.89727503, 6.73612475, 4.08250004
        408, -3.75295013, 6.9402498, 4.08250004
        409, -3.608624935, 7.14437485, 4.08250004
        410, -3.464300035, 7.3484999, 4.08250004
        411, -3.31997514, 7.55262495, 4.08250004
        412, -3.17564994, 7.75675, 4.08250004
        413, -3.03132504, 7.96087505, 4.08250004
        414, -3.03132504, 4.08250004, 7.96087505
        415, -3.17564994, 4.08250004, 7.75675
        416, -3.31997514, 4.08250004, 7.55262495
        417, -3.464300035, 4.08250004, 7.3484999
        418, -3.608624935, 4.08250004, 7.14437485
        419, -3.75295013, 4.08250004, 6.9402498
        420, -3.89727503, 4.08250004, 6.73612475
        421, -4.04159993, 4.08250004, 6.5319997
        422, -4.185925125, 4.08250004, 6.32787525
        423, -4.330250025, 4.08250004, 6.1237502
        424, -4.474574925, 4.08250004, 5.91962515
        425, -4.61890012, 4.08250004, 5.7155001
        426, -4.76322502, 4.08250004, 5.51137505
        427, -4.90754992, 4.08250004, 5.30725
        428, -5.0518751, 4.08250004, 5.103125
        429, -5.1962, 4.08250004, 4.89899993
```

```
      430, -5.3405249, 4.08250004, 4.694874885
      431, -5.4848498, 4.08250004, 4.490750135
      432, -5.6291747, 4.08250004, 4.286625085
      433, -2.742649915, 8.1650001, 4.286625085
      434, -2.59829998, 8.1650001, 4.490750135
      435, -2.453950045, 8.1650001, 4.694874885
      436, -2.309599965, 8.1650001, 4.89899993
      437, -2.165250035, 8.1650001, 5.103125
      438, -2.02089995, 8.1650001, 5.30725
      439, -1.87655002, 8.1650001, 5.51137505
      440, -1.732199935, 8.1650001, 5.7155001
      441, -1.587850005, 8.1650001, 5.91962515
      442, -1.44350007, 8.1650001, 6.1237502
      443, -1.29914999, 8.1650001, 6.32787525
      444, -1.154799985, 8.1650001, 6.5319997
      445, -1.010449975, 8.1650001, 6.73612475
      446, -0.86609997, 8.1650001, 6.9402498
      447, -0.721750035, 8.1650001, 7.14437485
      448, -0.57739999, 8.1650001, 7.3484999
      449, -0.4330499845, 8.1650001, 7.55262495
      450, -0.2886999955, 8.1650001, 7.75675
      451, -0.144349998, 8.1650001, 7.96087505
      452, -2.742649915, 0, 4.286625085
      453, -2.59829998, 0, 4.490750135
      454, -2.453950045, 0, 4.694874885
      455, -2.309599965, 0, 4.89899993
      456, -2.165250035, 0, 5.103125
      457, -2.02089995, 0, 5.30725
      458, -1.87655002, 0, 5.51137505
      459, -1.732199935, 0, 5.7155001
      460, -1.587850005, 0, 5.91962515
      461, -1.44350007, 0, 6.1237502
      462, -1.29914999, 0, 6.32787525
      463, -1.154799985, 0, 6.5319997
      464, -1.010449975, 0, 6.73612475
      465, -0.86609997, 0, 6.9402498
      466, -0.721750035, 0, 7.14437485
      467, -0.57739999, 0, 7.3484999
      468, -0.4330499845, 0, 7.55262495
      469, -0.2886999955, 0, 7.75675
      470, -0.144349998, 0, 7.96087505
*Element, type=B31
  1,   1,  15
  2,  15,  16
  3,  16,  17
  4,  17,  18
  5,  18,  19
  6,  19,  20
  7,  20,  21
  8,  21,  22
  9,  22,  23
 10,  23,  24
 11,  24,  25
 12,  25,  26
 13,  26,  27
 14,  27,  28
 15,  28,  29
```

```
16,  29,  30
17,  30,  31
18,  31,  32
19,  32,  33
20,  33,   2
21,   2,  34
22,  34,  35
23,  35,  36
24,  36,  37
25,  37,  38
26,  38,  39
27,  39,  40
28,  40,  41
29,  41,  42
30,  42,  43
31,  43,  44
32,  44,  45
33,  45,  46
34,  46,  47
35,  47,  48
36,  48,  49
37,  49,  50
38,  50,  51
39,  51,  52
40,  52,   3
41,   3,  53
42,  53,  54
43,  54,  55
44,  55,  56
45,  56,  57
46,  57,  58
47,  58,  59
48,  59,  60
49,  60,  61
50,  61,  62
51,  62,  63
52,  63,  64
53,  64,  65
54,  65,  66
55,  66,  67
56,  67,  68
57,  68,  69
58,  69,  70
59,  70,  71
60,  71,   4
61,   5,  72
62,  72,  73
63,  73,  74
64,  74,  75
65,  75,  76
66,  76,  77
67,  77,  78
68,  78,  79
69,  79,  80
70,  80,  81
71,  81,  82
72,  82,  83
```

```
 73,  83,  84
 74,  84,  85
 75,  85,  86
 76,  86,  87
 77,  87,  88
 78,  88,  89
 79,  89,  90
 80,  90,   4
 81,   5,  91
 82,  91,  92
 83,  92,  93
 84,  93,  94
 85,  94,  95
 86,  95,  96
 87,  96,  97
 88,  97,  98
 89,  98,  99
 90,  99, 100
 91, 100, 101
 92, 101, 102
 93, 102, 103
 94, 103, 104
 95, 104, 105
 96, 105, 106
 97, 106, 107
 98, 107, 108
 99, 108, 109
100, 109,   2
101,   6, 110
102, 110, 111
103, 111, 112
104, 112, 113
105, 113, 114
106, 114, 115
107, 115, 116
108, 116, 117
109, 117, 118
110, 118, 119
111, 119, 120
112, 120, 121
113, 121, 122
114, 122, 123
115, 123, 124
116, 124, 125
117, 125, 126
118, 126, 127
119, 127, 128
120, 128,   2
121,   6, 129
122, 129, 130
123, 130, 131
124, 131, 132
125, 132, 133
126, 133, 134
127, 134, 135
128, 135, 136
129, 136, 137
```

```
130, 137, 138
131, 138, 139
132, 139, 140
133, 140, 141
134, 141, 142
135, 142, 143
136, 143, 144
137, 144, 145
138, 145, 146
139, 146, 147
140, 147,   7
141,   5, 148
142, 148, 149
143, 149, 150
144, 150, 151
145, 151, 152
146, 152, 153
147, 153, 154
148, 154, 155
149, 155, 156
150, 156, 157
151, 157, 158
152, 158, 159
153, 159, 160
154, 160, 161
155, 161, 162
156, 162, 163
157, 163, 164
158, 164, 165
159, 165, 166
160, 166,   7
161,   8, 167
162, 167, 168
163, 168, 169
164, 169, 170
165, 170, 171
166, 171, 172
167, 172, 173
168, 173, 174
169, 174, 175
170, 175, 176
171, 176, 177
172, 177, 178
173, 178, 179
174, 179, 180
175, 180, 181
176, 181, 182
177, 182, 183
178, 183, 184
179, 184, 185
180, 185,   1
181,   9, 186
182, 186, 187
183, 187, 188
184, 188, 189
185, 189, 190
186, 190, 191
```

```
187, 191, 192
188, 192, 193
189, 193, 194
190, 194, 195
191, 195, 196
192, 196, 197
193, 197, 198
194, 198, 199
195, 199, 200
196, 200, 201
197, 201, 202
198, 202, 203
199, 203, 204
200, 204,   1
201,  10, 205
202, 205, 206
203, 206, 207
204, 207, 208
205, 208, 209
206, 209, 210
207, 210, 211
208, 211, 212
209, 212, 213
210, 213, 214
211, 214, 215
212, 215, 216
213, 216, 217
214, 217, 218
215, 218, 219
216, 219, 220
217, 220, 221
218, 221, 222
219, 222, 223
220, 223,   9
221,   8, 224
222, 224, 225
223, 225, 226
224, 226, 227
225, 227, 228
226, 228, 229
227, 229, 230
228, 230, 231
229, 231, 232
230, 232, 233
231, 233, 234
232, 234, 235
233, 235, 236
234, 236, 237
235, 237, 238
236, 238, 239
237, 239, 240
238, 240, 241
239, 241, 242
240, 242,  10
241,   8, 243
242, 243, 244
243, 244, 245
```

```
244, 245, 246
245, 246, 247
246, 247, 248
247, 248, 249
248, 249, 250
249, 250, 251
250, 251, 252
251, 252, 253
252, 253, 254
253, 254, 255
254, 255, 256
255, 256, 257
256, 257, 258
257, 258, 259
258, 259, 260
259, 260, 261
260, 261,  11
261,  11, 262
262, 262, 263
263, 263, 264
264, 264, 265
265, 265, 266
266, 266, 267
267, 267, 268
268, 268, 269
269, 269, 270
270, 270, 271
271, 271, 272
272, 272, 273
273, 273, 274
274, 274, 275
275, 275, 276
276, 276, 277
277, 277, 278
278, 278, 279
279, 279, 280
280, 280,  12
281,  12, 281
282, 281, 282
283, 282, 283
284, 283, 284
285, 284, 285
286, 285, 286
287, 286, 287
288, 287, 288
289, 288, 289
290, 289, 290
291, 290, 291
292, 291, 292
293, 292, 293
294, 293, 294
295, 294, 295
296, 295, 296
297, 296, 297
298, 297, 298
299, 298, 299
300, 299,  10
```

```
301,   8, 300
302, 300, 301
303, 301, 302
304, 302, 303
305, 303, 304
306, 304, 305
307, 305, 306
308, 306, 307
309, 307, 308
310, 308, 309
311, 309, 310
312, 310, 311
313, 311, 312
314, 312, 313
315, 313, 314
316, 314, 315
317, 315, 316
318, 316, 317
319, 317, 318
320, 318,   3
321,  13, 319
322, 319, 320
323, 320, 321
324, 321, 322
325, 322, 323
326, 323, 324
327, 324, 325
328, 325, 326
329, 326, 327
330, 327, 328
331, 328, 329
332, 329, 330
333, 330, 331
334, 331, 332
335, 332, 333
336, 333, 334
337, 334, 335
338, 335, 336
339, 336, 337
340, 337,   4
341,   7, 338
342, 338, 339
343, 339, 340
344, 340, 341
345, 341, 342
346, 342, 343
347, 343, 344
348, 344, 345
349, 345, 346
350, 346, 347
351, 347, 348
352, 348, 349
353, 349, 350
354, 350, 351
355, 351, 352
356, 352, 353
357, 353, 354
```

```
358, 354, 355
359, 355, 356
360, 356,  13
361,   9, 357
362, 357, 358
363, 358, 359
364, 359, 360
365, 360, 361
366, 361, 362
367, 362, 363
368, 363, 364
369, 364, 365
370, 365, 366
371, 366, 367
372, 367, 368
373, 368, 369
374, 369, 370
375, 370, 371
376, 371, 372
377, 372, 373
378, 373, 374
379, 374, 375
380, 375,   6
381,   9, 376
382, 376, 377
383, 377, 378
384, 378, 379
385, 379, 380
386, 380, 381
387, 381, 382
388, 382, 383
389, 383, 384
390, 384, 385
391, 385, 386
392, 386, 387
393, 387, 388
394, 388, 389
395, 389, 390
396, 390, 391
397, 391, 392
398, 392, 393
399, 393, 394
400, 394,  14
401,  12, 395
402, 395, 396
403, 396, 397
404, 397, 398
405, 398, 399
406, 399, 400
407, 400, 401
408, 401, 402
409, 402, 403
410, 403, 404
411, 404, 405
412, 405, 406
413, 406, 407
414, 407, 408
```

```
415, 408, 409
416, 409, 410
417, 410, 411
418, 411, 412
419, 412, 413
420, 413,  14
421,  13, 414
422, 414, 415
423, 415, 416
424, 416, 417
425, 417, 418
426, 418, 419
427, 419, 420
428, 420, 421
429, 421, 422
430, 422, 423
431, 423, 424
432, 424, 425
433, 425, 426
434, 426, 427
435, 427, 428
436, 428, 429
437, 429, 430
438, 430, 431
439, 431, 432
440, 432,  12
441,  14, 433
442, 433, 434
443, 434, 435
444, 435, 436
445, 436, 437
446, 437, 438
447, 438, 439
448, 439, 440
449, 440, 441
450, 441, 442
451, 442, 443
452, 443, 444
453, 444, 445
454, 445, 446
455, 446, 447
456, 447, 448
457, 448, 449
458, 449, 450
459, 450, 451
460, 451,   7
461,  11, 452
462, 452, 453
463, 453, 454
464, 454, 455
465, 455, 456
466, 456, 457
467, 457, 458
468, 458, 459
469, 459, 460
470, 460, 461
471, 461, 462
```

```
472, 462, 463
473, 463, 464
474, 464, 465
475, 465, 466
476, 466, 467
477, 467, 468
478, 468, 469
479, 469, 470
480, 470,   4
*Nset, nset=WIRE-2-SET-1
   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,  13,  14,  15,  16
  17,  18,  19,  20,  21,  22,  23,  24,  25,  26,  27,  28,  29,  30,  31,  32
  33,  34,  35,  36,  37,  38,  39,  40,  41,  42,  43,  44,  45,  46,  47,  48
  49,  50,  51,  52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64
  65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,  78,  79,  80
  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,  91,  92,  93,  94,  95,  96
  97,  98,  99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112
 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128
 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144
 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160
 161, 162, 163, 164, 165, 166, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252
 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268
 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284
 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300
 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316
 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332
 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348
 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364
 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380
 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396
 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412
 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428
 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444
```

```
      445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460
      461, 462, 463, 464, 465, 466, 467, 468, 469, 470
     *Elset, elset=WIRE-2-SET-1
        1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,  13,  14,  15,  16
       17,  18,  19,  20,  21,  22,  23,  24,  25,  26,  27,  28,  29,  30,  31,  32
       33,  34,  35,  36,  37,  38,  39,  40,  41,  42,  43,  44,  45,  46,  47,  48
       49,  50,  51,  52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64
       65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,  78,  79,  80
       81,  82,  83,  84,  85,  86,  87,  88,  89,  90,  91,  92,  93,  94,  95,  96
       97,  98,  99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112
      113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128
      129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144
      145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160
      241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256
      257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272
      273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288
      289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304
      305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320
      321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336
      337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352
      353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368
      369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384
      385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400
      401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416
      417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432
      433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448
      449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464
      465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480
     *Nset, nset=SET-2, generate
        1,  470,    1
     *Elset, elset=SET-2, generate
```

```
      1,  480,    1
*Nset, nset=SET-4, generate
      1,  470,    1
*Elset, elset=SET-4, generate
      1,  480,    1
*Nset, nset=SET-7
      1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,  13,  14,  15,  16
     17,  18,  19,  20,  21,  22,  23,  24,  25,  26,  27,  28,  29,  30,  31,  32
     33,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,  65,  66,  67
     68,  69,  70,  71,  91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102
    103, 104, 105, 106, 107, 108, 109, 129, 130, 131, 132, 133, 134, 135, 136, 137
    138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 243, 244, 245, 246, 247, 248
    249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 281, 282, 283
    284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299
    300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315
    316, 317, 318, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369
    370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385
    386, 387, 388, 389, 390, 391, 392, 393, 394, 414, 415, 416, 417, 418, 419, 420
    421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436
    437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452
    453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468
    469, 470
*Elset, elset=SET-7
      1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,  13,  14,  15,  16
     17,  18,  19,  20,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,  52
     53,  54,  55,  56,  57,  58,  59,  60,  81,  82,  83,  84,  85,  86,  87,  88
     89,  90,  91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 121, 122, 123, 124
    125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140
    241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256
    257, 258, 259, 260, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292
    293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308
    309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 361, 362, 363, 364
```

```
     365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380
     381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396
     397, 398, 399, 400, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432
     433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448
     449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464
     465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480
    *Nset, nset=AVERTICAL
       1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,  13,  14,  34,  35
      36,  37,  38,  39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51
      52,  72,  73,  74,  75,  76,  77,  78,  79,  80,  81,  82,  83,  84,  85,  86
      87,  88,  89,  90, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121
     122, 123, 124, 125, 126, 127, 128, 148, 149, 150, 151, 152, 153, 154, 155, 156
     157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172
     173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188
     189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204
     205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220
     221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236
     237, 238, 239, 240, 241, 242, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271
     272, 273, 274, 275, 276, 277, 278, 279, 280, 319, 320, 321, 322, 323, 324, 325
     326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341
     342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 395
     396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411
     412, 413
    *Elset, elset=AVERTICAL
      21,  22,  23,  24,  25,  26,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36
      37,  38,  39,  40,  61,  62,  63,  64,  65,  66,  67,  68,  69,  70,  71,  72
      73,  74,  75,  76,  77,  78,  79,  80, 101, 102, 103, 104, 105, 106, 107, 108
     109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 141, 142, 143, 144
     145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160
     161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176
```

```
     177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192
     193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208
     209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224
     225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240
     261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276
     277, 278, 279, 280, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332
     333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348
     349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 401, 402, 403, 404
     405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420
    *Nset, nset=SET-9, generate
       1,  470,    1
    *Elset, elset=SET-9, generate
       1,  480,    1
    *Nset, nset=AHORIZONTAL
       1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,  13,  14,  15,  16
      17,  18,  19,  20,  21,  22,  23,  24,  25,  26,  27,  28,  29,  30,  31,  32
      33,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,  65,  66,  67
      68,  69,  70,  71,  91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102
     103, 104, 105, 106, 107, 108, 109, 129, 130, 131, 132, 133, 134, 135, 136, 137
     138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 243, 244, 245, 246, 247, 248
     249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 281, 282, 283
     284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299
     300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315
     316, 317, 318, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369
     370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385
     386, 387, 388, 389, 390, 391, 392, 393, 394, 414, 415, 416, 417, 418, 419, 420
     421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436
     437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452
     453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468
     469, 470
    *Elset, elset=AHORIZONTAL
       1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,  13,  14,  15,  16
```

```
    17,  18,  19,  20,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,  52
    53,  54,  55,  56,  57,  58,  59,  60,  81,  82,  83,  84,  85,  86,  87,  88
    89,  90,  91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 121, 122, 123, 124
   125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140
   241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256
   257, 258, 259, 260, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292
   293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308
   309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 361, 362, 363, 364
   365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380
   381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396
   397, 398, 399, 400, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432
   433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448
   449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464
   465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480
*Nset, nset=SET-10, generate
   1,  470,    1,
** Section: Section-1-AHORIZONTAL  Profile: Profile-1
*Beam Section, elset=AHORIZONTAL, material=NITI_SE_AURRICHIO, poisson = 0.3, temperature=GRADIENTS, section=CIRC
""")
       fileOutput.write(str(rh)) # write horizontal radius
       fileOutput.write("""
0.,0.,-1.
** Section: Section-2-AVERTICAL  Profile: Profile-2
*Beam Section, elset=AVERTICAL, material=NITI_SE_AURRICHIO, poisson = 0.3, temperature=GRADIENTS, section=CIRC
""")
       fileOutput.write(str(rv)) # write vertical radius
       fileOutput.write("""
0.,0.,-1.
*End Part
**
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=PART-1-1, part=PART-1
*End Instance
**
*Nset, nset=BOTTOM, instance=PART-1-1
```

```
  3,  4,  8, 11
*Nset, nset=TOP, instance=PART-1-1
  6,  7,  9, 14
*Nset, nset=XYPLANE, instance=PART-1-1
  1,  8,  9, 10
*Nset, nset=YZPLANE, instance=PART-1-1
 12,
 *Nset, nset=ZXBOTTOM, instance=PART-1-1
   3,   4,   8,  11,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64
  65,  66,  67,  68,  69,  70,  71, 243, 244, 245, 246, 247, 248, 249, 250, 251
 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 300, 301, 302, 303, 304, 305
 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 452, 453, 454
 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470
*Nset, nset=ZXTOP, instance=PART-1-1
   6,   7,   9,  14, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140
 141, 142, 143, 144, 145, 146, 147, 357, 358, 359, 360, 361, 362, 363, 364, 365
 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381
 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 433, 434, 435
 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451
*Nset, nset=XYBOTTOM, instance=PART-1-1
   1,   8,   9,  10, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178
 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194
 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210
 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226
 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242
*Nset, nset=XYTOP, instance=PART-1-1
   4,   5,   7,  13,  72,  73,  74,  75,  76,  77,  78,  79,  80,  81,  82,  83
  84,  85,  86,  87,  88,  89,  90, 148, 149, 150, 151, 152, 153, 154, 155, 156
 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 319, 320, 321, 322, 323, 324
 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340
 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356
*Nset, nset=ZYBOTTOM, instance=PART-1-1
 12,
*Nset, nset=ZYTOP, instance=PART-1-1
 2,
*End Assembly
**
```

```
** MATERIALS
**
*Material, name=NITI_SE_AURRICHIO
*SUPERELASTIC
60000.,   0.3, 0.075,   520.,   600.,   300.,   200.,   520.
    0.,     0.,     0.,     0.
*Density
 6.5e-06,
*Elastic
60000., 0.3
**
** BOUNDARY CONDITIONS
**
** Name: Disp-BC-1 Type: Displacement/Rotation
*Boundary
ZXBOTTOM, 2, 2
** Name: Disp-BC-2 Type: Displacement/Rotation
*Boundary
XYBOTTOM, 3, 3
** Name: Disp-BC-3 Type: Displacement/Rotation
*Boundary
YZPLANE, 1, 1
** ----------------------------------------------------------------
**
** STEP: LOAD
**
*Step, name=LOAD, nlgeom=YES
*Static
0.01, 1., 1e-05, 0.1
**
** BOUNDARY CONDITIONS
**
** Name: BC-4 Type: Displacement/Rotation
*Boundary
""")
       fileOutput.write('ZYTOP, 1, 1, '+str(dispMag[j])) # write displacement magnitude
       fileOutput.write("""
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT
*End Step
** ----------------------------------------------------------------
**
** STEP: UNLOAD
**
```

```
*Step, name=UNLOAD, nlgeom=YES
*Static
0.01, 1., 1e-05, 0.1
**
** BOUNDARY CONDITIONS
**
** Name: BC-4 Type: Displacement/Rotation
*Boundary, op=NEW
** Name: BC-5 Type: Displacement/Rotation
*Boundary, op=NEW
ZYTOP, 1, 1
** Name: Disp-BC-1 Type: Displacement/Rotation
*Boundary, op=NEW
ZXBOTTOM, 2, 2
** Name: Disp-BC-2 Type: Displacement/Rotation
*Boundary, op=NEW
XYBOTTOM, 3, 3
** Name: Disp-BC-3 Type: Displacement/Rotation
*Boundary, op=NEW
YZPLANE, 1, 1
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-4
**
*Output, field
*Contact Output
CDISP, CSTRESS
**
** FIELD OUTPUT: F-Output-2
**
*Node Output
CF, RF, U
**
** FIELD OUTPUT: F-Output-3
**
*Element Output, directions=YES
LE, MVF, PE, PEEQ, PEMAG, S
**
** HISTORY OUTPUT: H-Output-2
**
*Output, history, variable=PRESELECT
*End Step

 """)
        fileOutput.close() #close inp file
```

II. inpAbaqusRunAll.py

```
#Author: Ian P. Morrissey
#Description: Submits every input deck in current directory
# and provides the job status for each submission to end each job before submitting the next job

import os

#create array textstring of .inp file names with file extensions
path=os.curdir
inpNames = [f for f in os.listdir(path) if f.endswith('.inp')]

#number of input files or decks
numinps=len(inpNames)

#remove .inp extension from text string
inpNamesNoExt =[]
inpNamesNoExt =[0 for i in range(numinps)]
for i in range(1,numinps+1):
    inpNamesNoExt[i-1] = inpNames[i-1].replace('.inp', '')

#submit all input files and display job status for each submission to end each job before submitting the next job
for i in range(1,numinps+1):
    os.system('abaqus job='+inpNamesNoExt[i-1]+' int')
print('Done')
```

III. post_tensor.py

```
# Author: Dr. John A. Moore
# Edited by: Ian P. Morrissey
# Changes: changed node sets, input name
# Description: Extracts reaction Force values in 1, 2, & 3 directions from node set at every frame in each step and sums
# Extracts displacment values in 1, 2, & 3 directions from node set at every frame in each step and sums and averages


import math
import datetime
from odbAccess import *
from abaqusConstants import *
# number of displacements
numLoads=8
# number of relative densities
numruns=15
# number of radii variation
numrvar=16
for j in range(1,numLoads+1):
```

```
    for i in range(1,numruns+1):
        for k in range(1,numrvar+1):
            runnum = str(i)
            loadnum = str(j)
            rvarnum = str(k)

            # Input and ouput file names
            InputName='D1B-'+loadnum+'-'+runnum+'-'+rvarnum

            # Sum of Reaction Forces
            OutName1='RF'
            # top displacement
            OutName2='U'

            print 'ODB = ' + InputName

            print 'Output Variable = ' + OutName1
            print 'Output Variable = ' + OutName2

            #vector component

            vecComp = 2


            outputfilename1=InputName+'-'+str(OutName1)+'-'+str(1)+'.txt' #output file name of reaction force in 1-direction
            outputfilename2=InputName+'-'+str(OutName1)+'-'+str(2)+'.txt' #ofn rf in 2
            outputfilename3=InputName+'-'+str(OutName1)+'-'+str(3)+'.txt' #ofn rf in 3
            outputfilename4=InputName+'-'+str(OutName2)+'-'+str(1)+'.txt' #ofn displacement in 1-direction
            outputfilename5=InputName+'-'+str(OutName2)+'-'+str(2)+'.txt' #ofn d in 2
            outputfilename6=InputName+'-'+str(OutName2)+'-'+str(3)+'.txt' #ofn d in 3


            ## open txt file to write to
            out1 = open(outputfilename1,'w') #w is mode used to open a file to write or to create one if it does not exist
            out2 = open(outputfilename2,'w')
            out3 = open(outputfilename3,'w')
            out4 = open(outputfilename4,'w')
            out5 = open(outputfilename5,'w')
            out6 = open(outputfilename6,'w')

            # open .odb file to read from
            odb = openOdb(InputName+'.odb')

            # part
            partInstance = odb.rootAssembly.instances['PART-1-1']

            # assembly
            assembly = odb.rootAssembly
```

```

            #n node sets

            nsetName = 'ZYTOP' #node set name of center top nodes
            nsetTop = assembly.nodeSets[nsetName]


            steps = ['LOAD','UNLOAD'] #load steps
            #steps = ['LOAD']
            for s in steps:
                print s

                # number of frames in step
                numberFrame = len(odb.steps[s].frames)

                # extract frame 1 to determine how many elements it has
                frame = odb.steps[s].frames[1]
                outvar1=frame.fieldOutputs[str(OutName1)]
                outvar2=frame.fieldOutputs[str(OutName1)]
                outvar3=frame.fieldOutputs[str(OutName1)]
                outvar4=frame.fieldOutputs[str(OutName2)]
                outvar5=frame.fieldOutputs[str(OutName2)]
                outvar6=frame.fieldOutputs[str(OutName2)]

                numelem = len(outvar2.values)
                print 'Number of Elements ' + str(numelem)


                # loop over all times (ie frames)
                for ns in range (1,numberFrame):
                        # these just let you know its running and how long it takes
                        print ns
                        print str(datetime.datetime.now())

                        # extract varibles desired
                        frame = odb.steps[s].frames[ns]

                        outvar1 = frame.fieldOutputs['RF'].getSubset(region=nsetTop)
                        outvar2 = frame.fieldOutputs['U'].getSubset(region=nsetTop)

                        # number of nodes in sets
                        nNsetTop   =  len(outvar1.values)

                        # average over all nodes in set
                        # this is adding all variable values together, starting the variable at 0 value
                        sumvar1=0.
                        sumvar2=0.
                        sumvar3=0.
                        sumvar4=0.
                        sumvar5=0.
                        sumvar6=0.
```

```


                        for n in range (0,nNsetTop):
                                # sum of reaction for all nodes
                                var1 =outvar1.values[n].data[0]
                                sumvar1 += var1
                                var2 =outvar1.values[n].data[1]
                                sumvar2 += var2
                                var3 =outvar1.values[n].data[2]
                                sumvar3 += var3
                                # sum of displacment for all nodes
                                var4 =outvar2.values[n].data[0]
                                sumvar4 += var4
                                var5 =outvar2.values[n].data[1]
                                sumvar5 += var5
                                var6 =outvar2.values[n].data[2]
                                sumvar6 += var6

                        # average of varibale for all displacment (forces don't need averaged)
                        avevar4 = sumvar4/nNsetTop
                        avevar5 = sumvar5/nNsetTop
                        avevar6 = sumvar6/nNsetTop

                        # write data
                        out1.write(str(sumvar1)+'\n')
                        out2.write(str(sumvar2)+'\n')
                        out3.write(str(sumvar3)+'\n')
                        out4.write(str(avevar4)+'\n')
                        out5.write(str(avevar5)+'\n')
                        out6.write(str(avevar6)+'\n')


            # close input and output files
            out1.close()
            out2.close()
            out3.close()
            out4.close()
            out5.close()
            out6.close()
```

IV. odbMaxMises.py [20]

```
#code is from
#Title:odbMaxMises.py
#Author: D'assault Systemes
#Date:2018
#Code Version: N/A
```

```
#Location: https://help.3ds.com/2018/english/dssimulia_established/simacaecmdrefmap/simacmd-c-odbintroexamaxmisespyc.htm?contextscope=all
#
#
#Changes: Peak stress is written to text file based off of odb file name
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
from odbAccess import *
from sys import argv,exit
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

def rightTrim(input,suffix):
    if (input.find(suffix) == -1):
        input = input + suffix
    return input
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

def getMaxMises(odbName,elsetName):
    """ Print max mises location and value given odbName
        and elset(optional)
    """
    elset = elemset = None
    region = "over the entire model"
    """ Open the output database """
    opname=str(odbName)
    opname=opname.replace('.odb','')
    odb = openOdb(odbName)
    assembly = odb.rootAssembly

    """ Check to see if the element set exists
        in the assembly
    """
    if elsetName:
        try:
            elemset = assembly.elementSets[elsetName]
            region = " in the element set : " + elsetName;
        except KeyError:
            print 'An assembly level elset named %s does' \
                   'not exist in the output database %s' \
                   % (elsetName, odbName)
            odb.close()
            exit(0)

    """ Initialize maximum values """
    maxMises = -0.1
    maxElem = 0
    maxStep = "_None_"
    maxFrame = -1
    Stress = 'S'
    isStressPresent = 0
    for step in odb.steps.values():
        print 'Processing Step:', step.name
        for frame in step.frames:
            allFields = frame.fieldOutputs
            if (allFields.has_key(Stress)):
                isStressPresent = 1
```

```
                stressSet = allFields[Stress]
                if elemset:
                    stressSet = stressSet.getSubset(
                        region=elemset)
                for stressValue in stressSet.values:
                    if (stressValue.mises > maxMises):
                        maxMises = stressValue.mises
                        maxElem = stressValue.elementLabel
                        maxStep = step.name
                        maxFrame = frame.incrementNumber
    if(isStressPresent):
        print 'Maximum von Mises stress %s is %f in element %d'%(
            region, maxMises, maxElem)
        outt=open(opname+'-PVM.txt','w') # open a text file named after the odb
        outt.write(str(maxMises)) #write peak vm stress
        outt.close() #close text file
        print 'Location: frame # %d  step:  %s '%(maxFrame,maxStep)
    else:
        print 'Stress output is not available in' \
              'the output database : %s\n' %(odb.name)

    """ Close the output database before exiting the program """
    odb.close()

#===================================================================
# S T A R T
#
if __name__ == '__main__':

    odbName = None
    elsetName = None
    argList = argv
    argc = len(argList)
    i=0
    while (i < argc):
        if (argList[i][:2] == "-o"):
            i += 1
            name = argList[i]
            odbName = rightTrim(name,".odb")
        elif (argList[i][:2] == "-e"):
            i += 1
            elsetName = argList[i]
        elif (argList[i][:2] == "-h"):
            print __doc__
            exit(0)
        i += 1
    if not (odbName):
        print ' **ERROR** output database name is not provided'
        print __doc__
        exit(1)
    getMaxMises(odbName,elsetName)
```

V. PVM.py

```
# Author: Ian Morrissey
# Description: Runs odbMaxMisesText.py for all .odb files in the current directory
import os

# create array textstring of .odb file names with file extensions in current directory
path = os.curdir
odbNames = [f for f in os.listdir(path) if f.endswith('.odb')]
numodbs=len(odbNames)
#submit odb for peak von mises
for i in range(1,numodbs+1):
    os.system('abaqus python odbMaxMisesText.py -odb '+odbNames[i-1]+' -elset " ALL ELEMENTS"')
    print('**'+odbNames[i-1]+' executed**')
print('Done')
```

VI. indBuck.py

```
# Author: Ian P. Morrissey
# Description: Reads ABAQUS .msg file for buckling warning, writes 1 if no buckling warning
# writes 0 if negative eignenvalue warning is present


import os

path=os.curdir
#create array of textstring of .msg file names with file extensions within current directory
msgNames = [f for f in os.listdir(path) if f.endswith('.msg')]
numMsg=len(msgNames)
#remove file extension
outNames=[]
outNames=[0 for i in range(numMsg)]
for i in range(1,numMsg+1):
    outNames[i-1]=msgNames[i-1].replace('.msg','')

# text string that appears in .msg file when buckling does not occur
eigString= '0  ANALYSIS WARNINGS ARE NEGATIVE EIGENVALUE MESSAGES'

#output files for each simulation

for i in range(1,numMsg+1):
```

```
    currMsg = open(msgNames[i-1],'r') #current .msg file to open
    readMsg = currMsg.read() #read message file
    buckMsg=open(outNames[i-1]+'-Buck.txt','w') #open output file to write
    if eigString in readMsg:
        buckMsg.write('1') #write No Buckling warning
    else:
        buckMsg.write('0') # write Negative Eigen-value warning
    currMsg.close #close current msg
buckMsg.close #close current buck msg

print('Done')
```

VII. writeRunPost.py

```
#Author: Ian P. Morrissey
#Description: Runs all scripts for writing input decks, submitting input decks, extracting values from odb and msg files

import os

#write .inp files
os.system('python writeinput.py')
print('                **.inp files written**')

#submit .inp files
os.system('python inpAbaqusRunAll.py')
print('                **jobs completed**')

#extract force and displacement from .odb files
os.system('abaqus python post_tensor.py')
print('                **force and displacement processed**')

#extract peak von mises stresses from .odb files
os.system('python PVM.py')
print('                **peak von mises processed**')

#indicate whether or not buckling occured
os.system('python indBuck.py')
print('                **buckling indication processed**')

#done
print('                **All Scripts Complete**')
```

APPENDIX C

This appendix shows samples of the MATLAB scripts that were used to process and create plots from the extracted values from ABAQUS.

I. PlotresultsD1DNSBeam.m

```
% Author: Ian Morrissey
% description import force and displacment of DNS and Beam models and plot
clc; clear all; close all



%import force f and displacement u DNS models
% 2 compression
uc2=load(['D1C2-U-1.txt'])
Fc2=load(['D1C2-RF-1.txt'])
% 4 compression
uc4=load(['D1C4-U-1.txt'])
Fc4=load(['D1C4-RF-1.txt'])
% 6 compression
uc6=load(['D1C6-U-1.txt'])
Fc6=load(['D1C6-RF-1.txt'])
% 8 compression
uc8=load(['D1C8-U-1.txt'])
Fc8=load(['D1C8-RF-1.txt'])
% 2 tensile
ut2=load(['D1T2-U-1.txt'])
Ft2=load(['D1T2-RF-1.txt'])
% 4 tensile
ut4=load(['D1T4-U-1.txt'])
Ft4=load(['D1T4-RF-1.txt'])
% 6 tensile
ut6=load(['D1T6-U-1.txt'])
Ft6=load(['D1T6-RF-1.txt'])
% 8 tensile
ut8=load(['D1T8-U-1.txt'])
Ft8=load(['D1T8-RF-1.txt'])




%import force f and displacement u beam models at nominal
% 2 compression
Buc2=load(['D1B-1-1-U-1.txt'])
BFc2=load(['D1B-1-1-RF-1.txt'])
% 4 compression
```

```
Buc4=load(['D1B-2-1-U-1.txt'])
BFc4=load(['D1B-2-1-RF-1.txt'])
% 6 compression
Buc6=load(['D1B-3-1-U-1.txt'])
BFc6=load(['D1B-3-1-RF-1.txt'])
% 8 compression
Buc8=load(['D1B-4-1-U-1.txt'])
BFc8=load(['D1B-4-1-RF-1.txt'])
% 2 tensile
But2=load(['D1B-5-1-U-1.txt'])
BFt2=load(['D1B-5-1-RF-1.txt'])
% 4 tensile
But4=load(['D1B-6-1-U-1.txt'])
BFt4=load(['D1B-6-1-RF-1.txt'])
% 6 tensile
But6=load(['D1B-7-1-U-1.txt'])
BFt6=load(['D1B-7-1-RF-1.txt'])
% 8 tensile
But8=load(['D1B-8-1-U-1.txt'])
BFt8=load(['D1B-8-1-RF-1.txt'])



%plot dns and beam force-displacement
figure(1)
p1=plot(uc2,Fc2,'r')
p1.LineWidth = 1.5
hold on
p2=plot(ut2,Ft2,'r')
p2.LineWidth = 1.5
p3=plot(Buc2,BFc2,'r--')
p3.LineWidth = 1.5
p4=plot(But2,BFt2,'r--')
p4.LineWidth = 1.5
p5=plot(uc4,Fc4,'b')
p6=plot(ut4,Ft4,'b')
p7=plot(Buc4,BFc4,'b--')
p8=plot(But4,BFt4,'b--')
p9=plot(uc6,Fc6,'g')
p10=plot(ut6,Ft6,'g')
p11=plot(Buc6,BFc6,'g--')
p12=plot(But6,BFt6,'g--')
p13=plot(uc8,Fc8,'k')
p14=plot(ut8,Ft8,'k')
p15=plot(Buc8,BFc8,'k--')
p16=plot(But8,BFt8,'k--')
hold off
xlabel('Displacement (mm)');ylabel('Force (N)');grid
axis([-0.2 0.2 -15 15]);
legend([p1 p5 p9 p13 p3 p7 p11 p15],{'DNS 2%','DNS 4%','DNS 6%','DNS 8%','Beam 2%','Beam 4%','Beam 6%','Beam 8%'});
legend('Location','northeastoutside')
saveas(gcf,'BeamDNSFDDirection1.svg')
```

II. plotresults1cont.m

```
% Author: Ian P. Morrissey

% Description: Import displacement and force values
% Imports peak Von Mises stress values and Buckling indication
% Calculates energy dissipation and energy dissipation coefficient
% Plots Energy Dissipation and Energy Dissipation Coefficient for set-displacements vs Rvar and Relative Density

clc; clear all; close all;
disp=linspace(-0.08,-0.01,8); % displacment values
rvar=linspace(0.5,1.25,16); % radius variance coefficient values
rd=linspace(0.03,0.45,15); % relative density values
k=0; %initialize k index
for u=1:8; %index by displacment
    for ii=1:15; % index by reltive desnity
        for j=1:16; %index by rvar
            clear d F di Fi F1 F2 E1 E2 h; % clear variables from last loop
            vm=load(['D1B-' num2str(u) '-' num2str(ii) '-' num2str(j) '-PVM.txt']); %load peak von mises stress
            buck=load(['D1B-' num2str(u) '-' num2str(ii) '-' num2str(j) '-Buck.txt']); %load buckling indication
            if buck==1 && vm<=700   % only proceed if buckling did not occur and the stress limit was not reached
                di=load(['D1B-' num2str(u) '-' num2str(ii) '-' num2str(j) '-U-1.txt']); %import displacement
                Fi=load(['D1B-' num2str(u) '-' num2str(ii) '-' num2str(j) '-RF-1.txt']); %import force
                k=k+1; %index k
                vab(k,1)=rd(ii); % create rd that is to be used
                vab(k,2)=rvar(j); %create rvar to be used
                d(1)=0; %initialize displacment
                F(1)=0; %initialize force
                for i= 1:length(di); %convert
                    d(i+1)=-di(i);
                    F(i+1)=-Fi(i);
                end
                %separate curves for calculating area under loading and
                %unloading for energy calculations and calculate energy
                %dissipated and ED coefficeint
                [maxval,maxind]=max(d);
                d1=d(1:maxind);
                d2=d(maxind:length(d));
                F1=F(1:maxind);
                F2=F(maxind:length(d));
                E1=trapz(d1,F1);
                E2=-trapz(d2,F2);
                vab(k,3)=E1-E2; %energy calc;
                vab(k,4)=(E1-E2)./((E1-(E1-E2)./2)*pi); % energy dissipation coefficient
```

```
            else % continue to next if vm limit exceed
                %or if buckling occured
            end
        end
    end
    clear k %clear and reinitialize k
    k=0;

    % put mesh over points so they can be seen on plot clearly
    x1=linspace(min(vab(:,1)),max(vab(:,1)),100)
    y1=linspace(min(vab(:,2)),max(vab(:,2)),100)
    [X,Y]=meshgrid(x1,y1)
    Z1=griddata(vab(:,1),vab(:,2),vab(:,3),X,Y,'cubic');

    %plot ED and save svg of plot
    figure(2*u-1)
    mesh(X,Y,Z1),grid;title(['ED for Displacement = ' num2str(-100*disp(u)) '%']);xlabel('Relative Density');ylabel('R_{var}');zlabel('E_{disp} ({\mu}J)');
    hold on
    plot3(vab(:,1),vab(:,2),vab(:,3),'r.','MarkerSize',15),grid
    saveas(gcf,['D1ED-' num2str(u) '.svg'])

    %plot EDC and save svg of plot
    figure(2*u)
    Z2=griddata(vab(:,1),vab(:,2),vab(:,4),X,Y,'cubic');
    mesh(X,Y,Z2),grid;title(['\eta for Displacement = ' num2str(-100*disp(u)) '%']);xlabel('Relative Density');ylabel('R_{var}');zlabel('\eta ');
    hold on
    plot3(vab(:,1),vab(:,2),vab(:,4),'r.','MarkerSize',15),grid
    saveas(gcf,['D1EDC-' num2str(u) '.svg'])
end
```