PRINCIPLE COMPONENT ANALYSIS BASED APPROACH TO MODEL DEFORMATION OF WEB CORE BEAM

A Thesis

by

MATTHEW FISSELER

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Arun Srinivasa
Committee Members,	J.N. Reddy
	Darren Hartl
Head of Department,	Andreas A. Polycarpou

August 2021

Major Subject: Mechanical Engineering

ABSTRACT

The purpose of the present study was to develop a systematic way to determine the modes of deformation found in architected materials. Specifically, the mode of deformation for the web within a web core beam structure is studied. Localized deflection information was gathered and PCA was used to determine the common mode of deformation. This was used to develop a energy based model for a unit cell to determine the values of kinematic variables. The model was homogenized so a structure with many unit cells could be easily analysed by being discretized into elements much larger than the unit cell size. Using Castigliano's first theorem the values of the kinematic variables can be found. To minimize the objective function, MATLABs optimization toolbox was utilized.

For a web core beam it was found that the webs deform as a cubic function and 3 PCs were adequate to approximate the deformation. This information was used to develop the beam model and the accuracy of the center line deflection for a fixed-fixed beam was analyzed to validate the model.

While the work done does not provide a robust answer to what shape function to use in all uni-directional sandwich beam structures, it does provide a systematic method to determine the modes of deformation before the development of a analytical model is developed. This method can also be applied to other architected structures, and assist in future work related to determining deformation, failure, and instabilities within the micro-structure of architected materials.

The work done provides a systematic method to determine the shape function of uni-directional sandwich beam structures with any cross section. This method can also be applied to other types of architected structures. It can assist in future work related to determining deformation, failure, and instabilities within the micro-structure of architected structures. Furthermore, it shows that the implementation of an energy based method can be used rather than the typical system of equations. This allows for a more simple implementation when changing the structure's geometry, external loads, or boundary conditions.

DEDICATION

To my parents Patrick and Kristine and girlfriend Jennifer

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor Arun Srinivasa and J.N. Reddy of the Department of Mechanical Engineering and Professor Darren Hartl of the Department of Aerospace Engineering.

Administrative items were supported by Department of Mechanical Engineering Graduate Advising Office and the Office of Graduate and Professional Studies.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study was supported by a grant from the National Science Foundation for research on architected materials. Other funding was supported by a fellowship as well as a role as a Teaching Assistant from Texas A&M University Department of Mechanical Engineering.

NOMENCLATURE

GFRP	Glass Fiber Reinforced Polymer
PCA	Principal Component Analysis
SVD	Singular Value Decomposition
PC	Principle Component
FE	Finite Element
A	Top Face Plate
B	Bottom Face Plate
C	Web

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

TABLE Page

1. INTRODUCTION AND LITERATURE REVIEW

In any mechanical structure the material and geometry of the structure are major considerations because they control the strength, stiffness, stability, and durability of the structure. When selecting what material should be used, a large number of considerations must be accounted for such as, strength, stiffness, mass density, as well as its response to strain rate, temperature, cyclic loading, and corrosion. One method to find the relationship between these variables is to use charts developed by Ashby. These charts display the relationship between two factors and generally group materials into a region on the chart. Using these charts engineers are able to see the relationship between factors such as strength to density or stiffness to strength [10]. An example of one of these charts can be found in Figure 1.1.

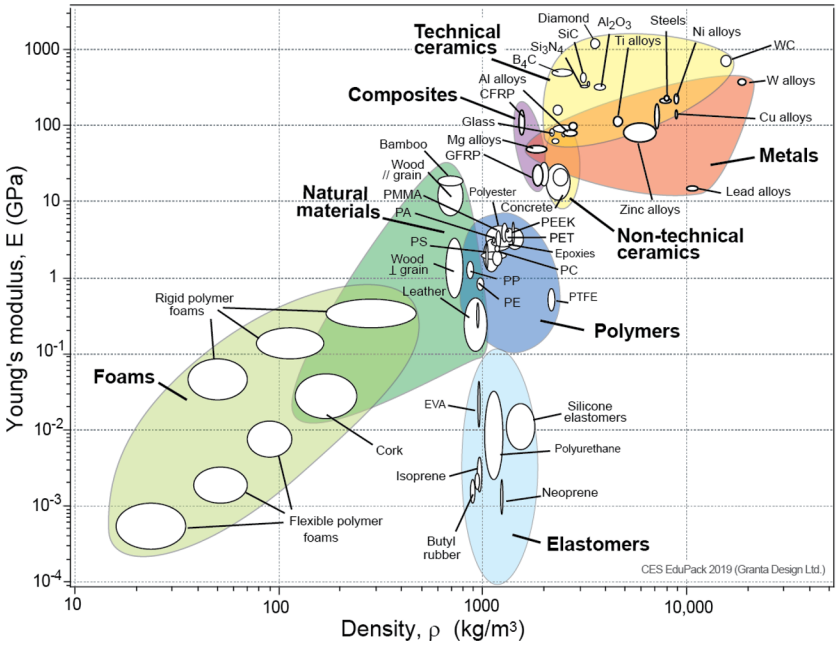


Figure 1.1: Young's Modulus vs Density. Level 2 Materials Chart [1]

Some locations on the chart are white because there is not currently a material that has that

combination of properties. An approach to filling in these spaces is to use chemistry to manipulate or develop new materials. For example, aluminum's properties can be changed by introducing small amounts of chromium, cobalt, copper, magnesium, iron, silicon, or zinc. Another approach is to changing the structure of a material using thermo-mechanical, thermo-chemical, or mechanical methods. For example, aluminum's properties can be changed using all three of these methods in processes such as cold rolling, heat treating, and anodizing. The third approach is to change the macro structure and use a combination of two or more different materials to obtain a material with desirable material properties on a macro scale. Examples of these types of are materials include: carbon-fiber reinforce composites, sandwich panels, and lattes structures [10, 11]. Figure 1.2 shows examples of sandwich structures with many different internal core structures. In general hybrid and composite materials have found applications in aerospace, automotive, construction, and marine industries [6, 10, 11, 12, 13, 14, 15, 16].

The phrase architected materials comes from a paper published by Ashby and Brechet [17] and was used to link the aspects of architecture and structural engineering to create reliable stiff structures. To further define architected materials, current literature defines them as a combination of multiple materials, or open space within the structure, configured to reach performance not typically available to a single material [10]. One example of this type of a material is extremely porous solids produced by additive manufacturing seen in Figure 1.3. In structures such as these, there is typically a single material with open spaces within the structure. This is done to reduce the material, cost, and time of manufacturing. Due to the geometry these structures have very high stiffness to weight ratios. The important variables to consider are the micro-architecture, length scale, and material properties of the material present [10]. An example of how these types of materials reach traditionally unreachable locations on the Ashby charts can be seen in Figure 1.4.

The length scale of the structure can range from tens of nano meters to centimeters. A diamond lattice as show in Figure 1.4 would be on the nano or micro meter scale, while a sandwich beam structures in Figure 1.2 would be larger and measure by centimeters [2]. As the length scale changes the challenges of designing and manufacturing the structure also change. Leading issues

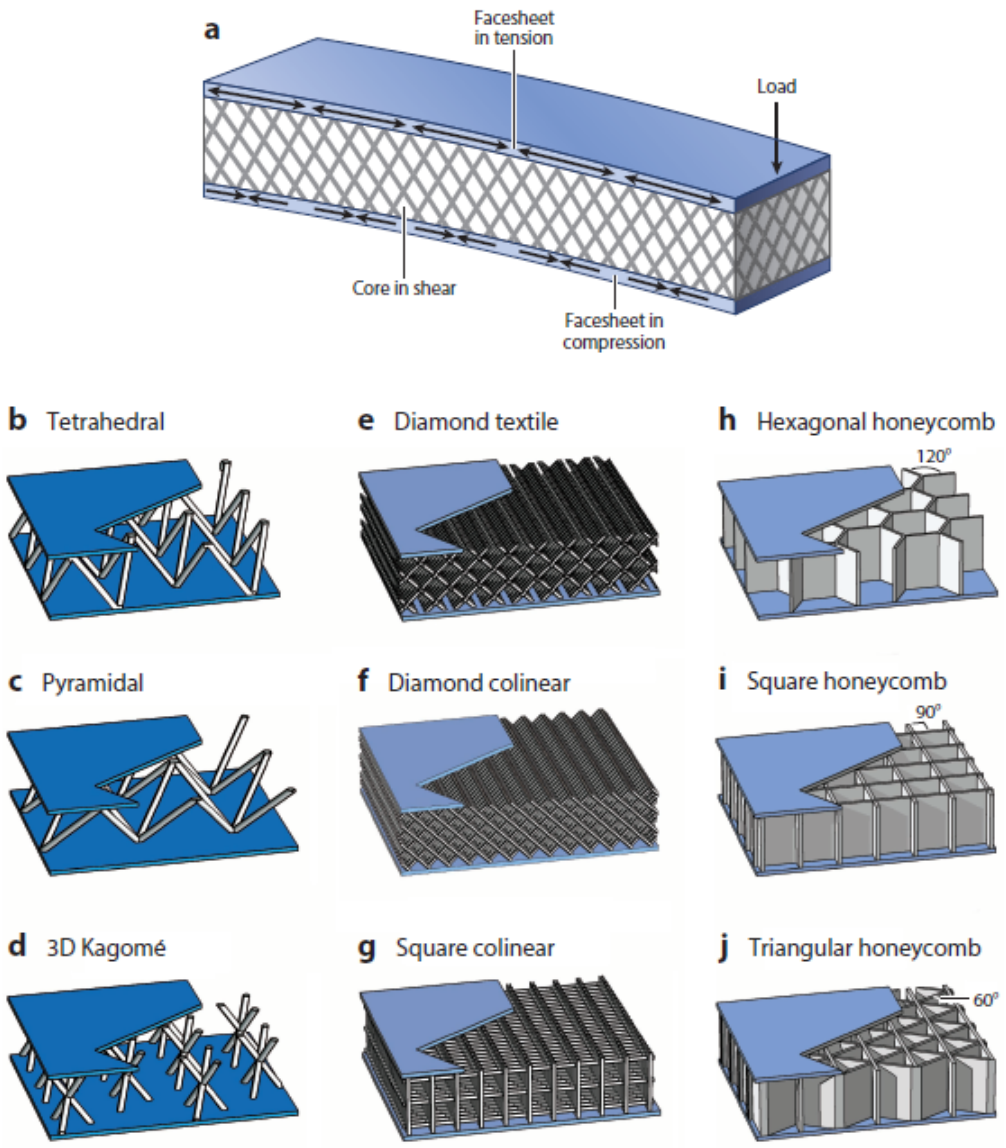


Figure 1.2: Examples of Sandwich Core Structures with Different Internal Structures [2]

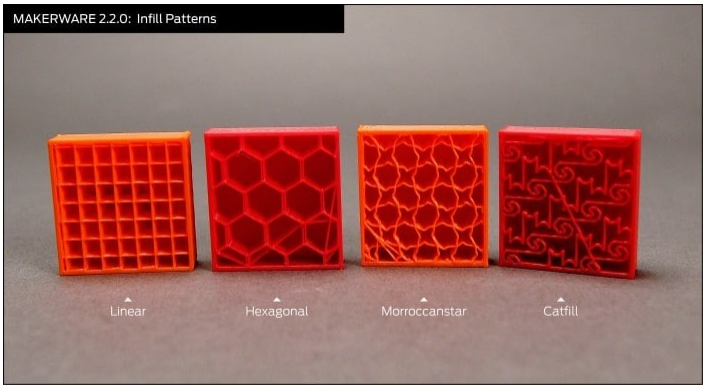


Figure 1.3: 3D Printing Infill Showing Different Micro Structures [3]

in the area of research today include consistent manufacturing and processing, finding optimal geometric designs, computational design tools, empirically determining specifics of localized material response, and application in electrochemical systems [10, 11]. One method used in both theoret-

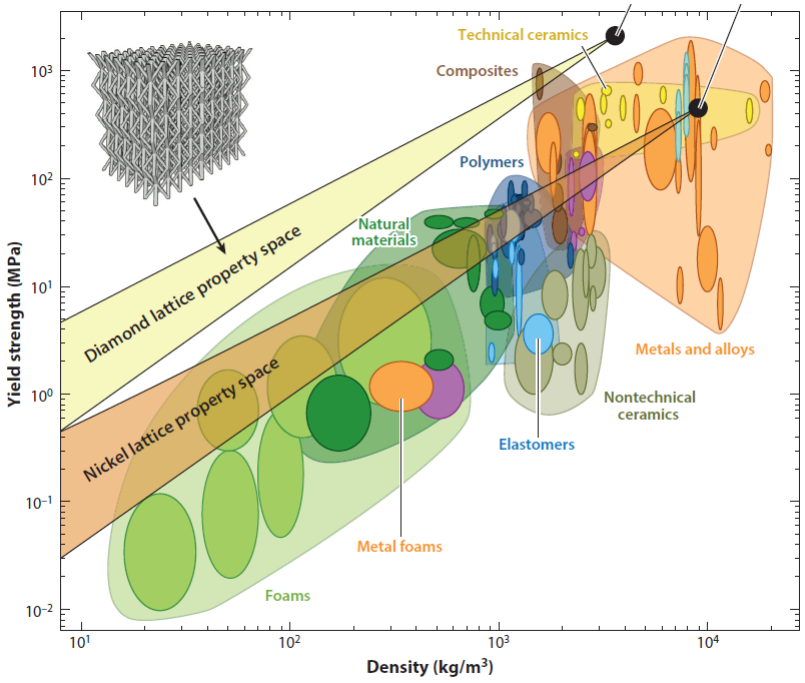


Figure 1.4: Young's Modulus vs Density with Architected Materials [2]

ical or empirical models to predict deformation and failure is to model the structure as a single continuum. For example, Schumacher et al. [4] studied optimized architected sheets and found the material properties of the sheet by using Kirchhoff rod simulation to find the deformation of a sample with in structure. Figure 1.5 shows a sample of the structure that was taken in order to run simulations. To simplify, the localized deformation of the sampled structure was used to generalize the material properties and approximate the structure as a single continuum in a sheet. This was applied to the remaining area of the sheet and the global deformation and strength could then be approximated. This method is called homogenization and relies on assuming that the detail of the local deformation and stress can lost while the global variables can still be accurately found.

This lowers the computational complexity and in large structures is found to be effective in this case. Figure 1.7 shows how the material properties can change a great deal by only changing the structure of the sheet. One of the issues with homogenizing the sheet is that because the localized deformation and stress are lost the global variables are not as accurate. Figure 1.6 shows the error between simulated and empirical values for tensile and bending stiffness of different patterns [4].

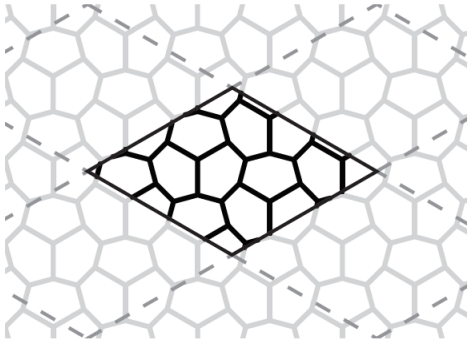
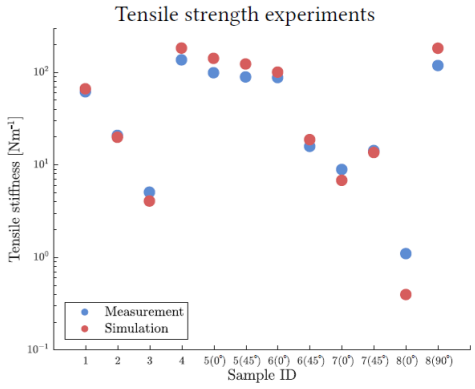
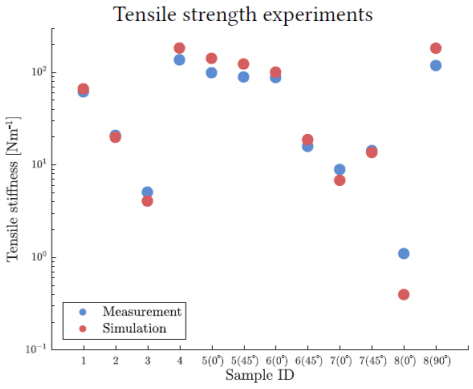


Figure 1.5: Sample of Architected Sheet [4]



(a) Comparing Empirical and Predicted Tensile Stiffness of Different Sheets [4]



(b) Comparison of Measured and Predicted Stiffness of Different Sheets [4]

Figure 1.6: Error in Simulation of Architected Sheets: ID numbers refer to sheet configurations in Figure 1.7, [4] refers to these graphs as tensile strength but shows measurements relate to stiffness

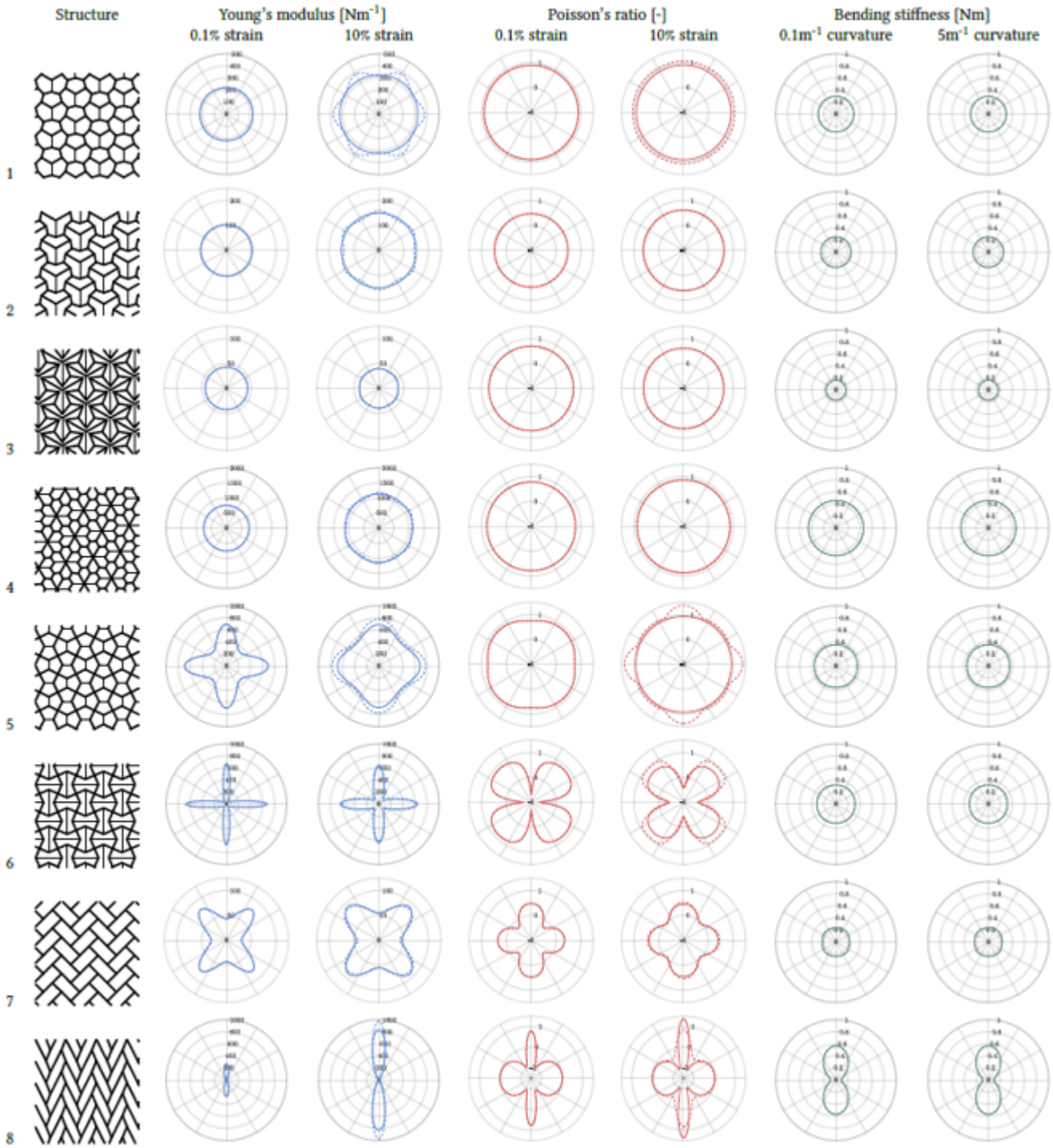


Figure 1.7: Materials Properties of Various Architected Sheets [4]

1.1 Introduction to Unidirectional Sandwich Panels

A specific group of sandwich beams shown in Figure 1.8 are structured such that the internal core is continuous in one direction while periodic in another. Practical examples of these sandwich structures can be found in ship design [10, 18, 19, 20, 21]. Sandwich panels with different interior structure have the ability to have a high stiffness while remaining light weight. Sandwich panels are comprised of face plates and an interior structure. The interior structure can be a variety of different geometries and provide the core to the structure as seen in Figure 1.8. This core supports the face plates continuously in the direction of the web plate and discretely in the transverse direction. Because the face plates are relativity far from the neutral axis of the beam, the structure is efficient in reducing deflection under bending loads [5]. Using the estimate from Winkle et al. [14] a ship deck constructed of steel sandwich panels can reduce the weight of the ship by 30 to 50 percent. While these structures are popular in navel construction they could also be used in the design of bridges or buildings. [6, 11, 12, 13, 14, 15, 16]. The structure can be made of a variety of material including, metal, wood, GFRP and are joined with adhesives, friction stir welding, or laser welding [13, 14, 18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29].

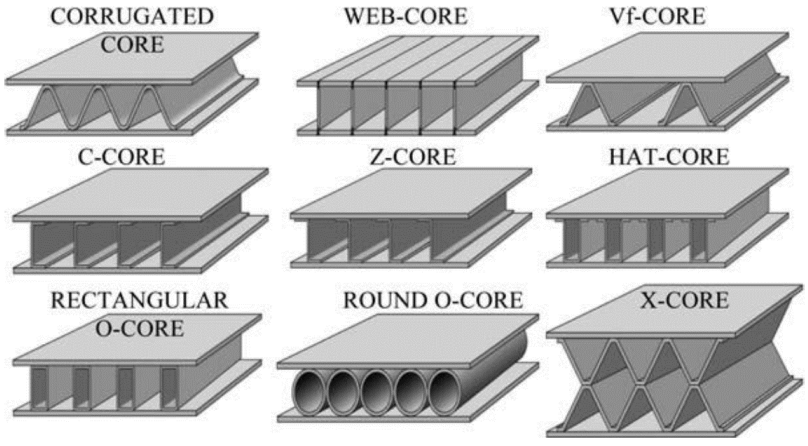


Figure 1.8: Various Types of Unidirectional Sandwich Panels [5]

A typical application of these structures uses steel plates joined by a laser welding technique

to form a web core sandwich panel shown in Figure 1.9. The plates on the top and bottom are called face plates and the interior plates that make up the core are called webs. The webs only run along one direction, this causes the structure to have different bending stiffness depending on the application of the load [5, 28].

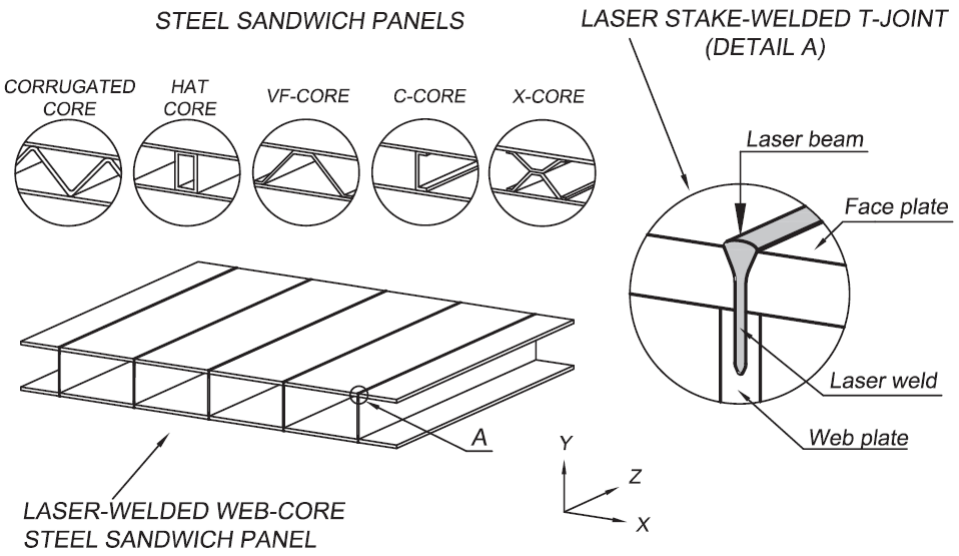


Figure 1.9: Steel Sandwich Panels Joined with Laser Stake-Welded T-Joint [6]

In the direction of the webs, the shear stiffness of the structure is relatively stronger. However perpendicular to this strong direction, the shear stiffness is far lower due to the lack of the webs continuously running along the direction, this creates a weak direction. The two directions can be evaluated theoretically or empirically to find an approximation for the bending stiffness and therefore have two different bending stiffness terms [5, 28].

The theoretical models that have been developed are generally suitable to analyze large structures so it can be applicable to analyzing ships or other similar scaled problems, see Table 1.1 for more information. Originally the model was built by homogenizing the core material so only the average response was considered with few degrees of freedom. Homogenized beam theory is cover in depth in sections 2.1 to 2.4. This can also be expanded on using frame analysis with beam

structures [5]. Separating the stress analysis into global and local components increases the accuracy of the models stress components. Beam elements are used because they are advantageous when combining global and local reactions. Modeling the structure in this way means the face plates carry the global bending moments and the core structure carries the shear forces [8]. The load becomes mixed because the core in sandwich panels is unidirectional. Figure 1.10 shows the internal forces acting at different point in the beam. In this figure, N is a normal force acting along the plate, M is a concentrated moment, V is a vertical force, and Q is a shear force. Super scripts n, t, and b refer to the web, top, and bottom plates, respectively [5].

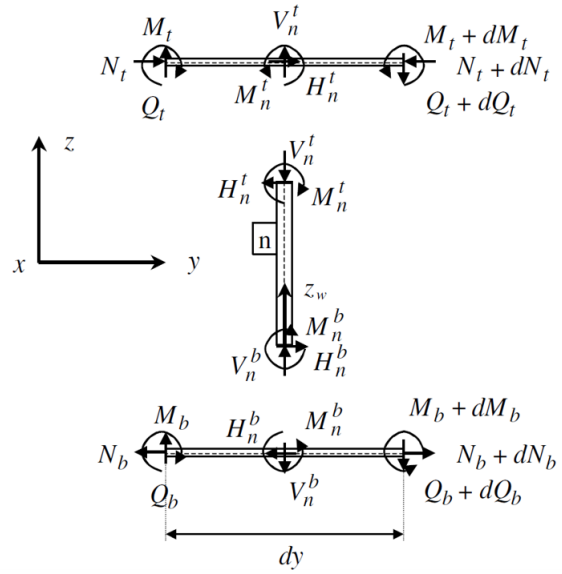


Figure 1.10: Internal Forces Between Face and Webs Plates [5]

When the load becomes mixed the unit cell response becomes complex and as a result high stresses are created [5, 30]. The deformation of each unit cell is complex due to the interactions between the web and face plates, see refs. [5, 8, 7, 23]. These interaction leads to bending loads in the web plates, which then creates concentrated bending stress on the face plate where it is connected to a web [5]. In order to find the stresses the deformation of each unit cell along with the global stiffness of the structure must be accounted for. Also it should be noted that both global

Table 1.1: Computation Methods to Find Response of Web Core Beam

Techniques/Theory	Structure	Reference
Commercial FE Method	Web Core Beam	[5, 8, 32, 33]
	Corrugated Core	[15, 16, 22, 24]
	X-Core	[22]
Plane Frame	Web Core	[5, 30]
Orthotropic plate theory	Web Core	[14, 25]
	Corrugated Core	[14, 15, 16]
Homogenized Classical Beam	Web Core	[5, 8, 30, 34]
	Corrugated Core	[22, 23]
	X-Core	[22, 23]
Homogenized Micropolar Beam	Web Core	[9, 30, 35]
	Lattice Core	[34, 36]
Empirical Studies or Fabrication	Any	[6, 8, 12, 14, 20, 23, 27, 28, 29, 18, 31, 32, 37]

failure and local failure at the joint should be checked, due to the high bending moment in the webs and face plates. The concentrated bending moment at the joints are locations of specific interest and research has been done specifically to analyze the joining method due to this interaction [31]. While these sandwich beams can be provide increased stiffness even in their weak direction the stress created by localized bending is an issue that must be accounted for when analyzing their design. This is explored in depth in sections 2.5 and 2.6 as well as chapter 2.

Table 1.1 summarizes some of the previous work done in this area. It categorizes papers by the technique or theory used to predict the deformation of the structures as well as what specific structures the paper explored.

1.1.1 Direct Solution of Microstructure

One approach that has been used in the modeling of these structures is the direct solution of the actual structure with full resolution of the microstructure using beam, 2-D, or 3-D elements using commercial software, see Table 1.1. They have the advantages of being able to fully resolve the deformations and requiring only basic material properties such as the elastic modulus, Poisson's ratio, and tensile strength. However, they suffer from a significant drawback, namely that the elements used need to be on the microstructural scale. This leads to a structure with millions of degrees of freedom even for a simple plate configuration due to the need for fine scale resolution.

Any change in geometry, material, loading, or fixtures requires a full reanalysis. This not only becomes prohibitively expensive in terms of computational power but also in terms of required time to solve. These limitations preclude such approaches from being used for iterative design and analysis of large scale structures using these architectured features.

1.1.2 Orthotropic Plate Theory

In the direction of the webs, the shear stiffness of the structure is relatively stronger. However perpendicular to this strong direction, the shear stiffness is far lower due to the lack of the webs continuously running along the direction, this creates a weak direction [5, 28]. Because the stiffness changes depending on the what plane the structure is bend it can be categorized as orthotropic. As a result, a ordinary plate theory, based on assuming the plate is isotropic cannot be used to accurately determine the stresses. [28]

Therefore, a small-deflection theory for flat orthotropic plates was developed in which deflections due to shear are accounted for [28, 38]. This theory is applicable to any type of orthotropic or isotropic sandwich that behaves similar to a plate, and when certain physical constants are known. The model can then describe the plates deformation for scenarios with simple loading conditions, similar to ref. [4]. These physical constants include a flexural stiffness, shear stiffness, and Poisson ratio for each bending direction, and share a torsional stiffness. For simple types of sandwich construction the physical constants can be found theoretically based on the material and geometry used [28]. For complicated types structures, the constants can be found by empirical tests on constructed samples of the structure, as described in ref. [38] and shown in ref. [20, 28, 37].

1.1.3 Homogenized Beam Theory

Because sandwich beams have regular repeating structure that allows for a repetitive unit cell to be homogenized into an equivalent material model to solve boundary value problems [8]. The equivalent material model for the unit cells can be found by experimental tests or by mathematical models. From a mathematical view, homogenization is a limit theory that uses asymptotic expansion and a known or assumed periodicity. An asymptotic expansion is a series expansion of

a function that's partial sums can be used to make an approximation to a given function. This is used to simplify differential equations with rapidly oscillating coefficients to a differential equation with constant or smoother variations [7, 8]. This is graphically shown in Figure 2.2 and 2.3. The new smooth differential equations can be more easily evaluated and can be used to predict both global and local response. However it should be noted due to the simplification much of the local response will be lost as the function is smoothed out. This includes the complex interaction between the webs and face plates, previously discussed. By dropping these terms the theory over predicts deflection near applied loads as shown in ref. [8].

Recently, to address issues with previous models and predict more accurate results accounting for this complex interaction, a coupled stress term was derived. This solution relies on Euler–Bernoulli or Timoshenko beam theories to account for micro-structural effects within the continuum. A coupled stress term is added to describe the effect created at the joints between the face plates and web plates. Shear stress is created at these joints because of the deflection of the face plate. The shear stress then creates a moment at the joint that bends the web plate [8]. This interaction is lost when the structure is homogenized with fewer degrees of freedom and with the introduction of the coupled stress term the model more closely predicts the center line deformation. [5, 8, 39, 40].

1.1.4 Micropolar beam models

The homogenized coupled stress approach is an improvement to the conventional homogenized method, but its full limitations are yet to be studied [9]. Coupled stress continuum theories [41, 42] are simplified versions of micropolar theories [43]. It is can also be called the Cosserat theory [44] or the theory of asymmetric elasticity [45]. A micropolar theory includes a local micro-rotation that is independent of the global macro-rotation [9], therefor increasing the degrees of freedom used in section 1.1.3.

The coupled stress term, discussed in section 1.1.3, is found through the assumption that the micro and macro rotations coincides with each other. In ref. [9] this assumption is relaxed and uses a micropolar thoery with Timoshenko beams to study the deformation of web core beams. It

also shows in some cases the homogenized coupled stress theory using Timoshenko beams may provide too stiff a response for web core beams due to the constraint on the independent rotations. It should be noted that micropolar theories using Timoshenko beams have been developed by several researchers, however were not applied to web core beam structures [46, 47, 48, 49, 50].

1.1.5 Empirical Studies

Studying the deflection and stress in a physical structure is important both to validate commercial FEA as well as theoretical models. It has also be used to create predictive models [8]. However with the wide verity of structures, materials, and boundary conditions it is difficult to crate a predictive models for every case. These tests are also difficult to conduct because of the time necessary to manufacture and testing a number of structures so statistical significance can be found. Examples of papers studying joining methods, failure modes, and response to loading can be found in Table 1.1.

1.2 Motivation and Plan

The micropolar and micromorphic approaches presented in the previous sections offer a way to model architected structures at macroscopic length scales while retaining the microscopic degrees of freedom. However two major items motivate additions and modifications of these methoids. First, the micropolar and micromorphic beams require intuition about the deformation of the fine scale cross sections in order to resolve the shape function of the webs deformation. In earlier work the deformed shape of the web was approximated using Taylor Series expansions [34] or by assuming the order of the function [5]. We seek to eliminate the need for these and possibly find a more accurate solution and propose an approach systematic to determining the modes of deformation using principle component analysis (PCA).

Furthermore, the conventional FE methods uses a force-balance based strategy to obtain governing equations. As the modes of deformation become more complex and the degrees of freedom become more abstract, these approaches become severely limited. Lagrangian mechanics offers a theoretically sound and computationally efficient means for obtaining the governing equations

without the need for postulating them a-priori. The statics counterpart, namely, the principle of minimum potential energy, Castigliano's first theorem, offers a versatile means for restating the problem as a minimization problem with great flexibility in choosing shape functions and solution methodologies. We seek to leverage the extensive open source software developed for optimization and machine-learning, such as conjugate gradient and Hessian based methods for solving the minimization problem with minimal effort.

A web core beam structure will be used because it has a simple geometry and as shown in Table 1.1 it is a well researched problem with both empirical and theoretical models available to compare with. However this approach is valid for other sandwich and architected materials in general.

The proposed strategy is as follows:

1. Use a fine scale FE model to obtain solutions for a relatively small structure under different loading conditions. Separate nodal displacements into two groups, a training group and a testing group.

2. Extract nodal deformation samples from the training group FEA.

3. Use Principle Component Analysis to extract the principal modes of the fine scale deformation.

4. Use these to develop the strain energy function for a coarse scale micromorphic beam model (so that we do not have to guess or loosely approximate the shape functions). We can also estimate the structural constants of the model.

5. Use energy minimization to find the solution to the coarse scale mode using the estimated structural constants.

6. Compare the deformation of the training and testing group FEA to improve the estimated structural constants.

The following sections cover the implementation of ANSYS to gather data, as well as principle component analysis to build an approximated model to find the shape function of the webs defor-

mation. The full development of the beam model for a single unit cell of a web core beam. And results comparing 2D FE results and theoretical beam model results for a simply supported beam, and a summary of the work done.

2. TECHNICAL OVERVIEW OF HOMOGENIZED THEORIES

In this section there will be a short review of different models and approaches that have been researched to estimate the deformation for web core beam structures. The models that will be reviewed are homogenized beam theory, conventional Timoshenko, coupled stress Timoshenko, and micropolar approaches. All models will be compared using similar structures made up of vertical webs and horizontal plates in a two dimensional structure. A generic form of the structure can be seen in Figure 2.1. It should be noted that each of these theory's can be applied to various structures but for sake of comparison only web core beams will be used. A web core beam structure is commonly used in models because it is the simplest due to the webs being plates and being perpendicular to the face plates. This removes any of the design variables that would otherwise need to be considered such as the angle of the webs, or the length of the base of the square or c channel webs.

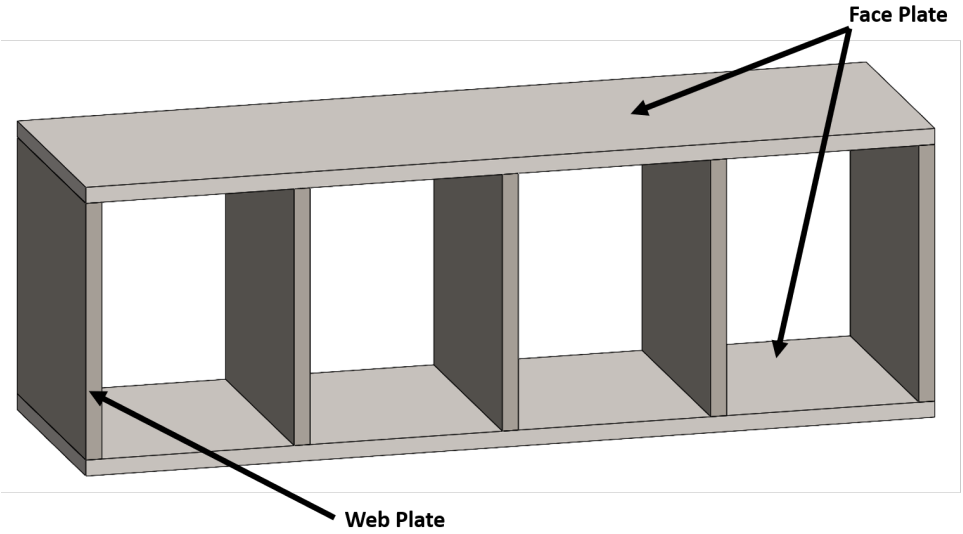


Figure 2.1: Naming Convention of Segments in Web Core Beam

2.1 Homogenized Beam Theory

Sandwich beams have regular repeating structure that allows for a repetitive unit cell to be homogenized into an equivalent material model to solve boundary value problems [8]. The equivalent material model for the unit cells can be found by experimental tests or by mathematical models. Typically, a mathematical model is used due to the large number of variations in material, manufacturing methods, and geometries that can be used. Homogenization of a structure applies to a wide variety of problems including solid mechanics, heat transfer, fluid flow in porous material, or electromagnetism in composites. From a mathematical view, homogenization is a limit theory that uses asymptotic expansion and a known or assumed periodicity. An asymptotic expansion is a series expansion of a function that's partial sums can be used to make an approximation to a given function. This is used to simplify differential equations with rapidly oscillating coefficients to a differential equation with constant or smoother variations as shown in Figure 2.2 and 2.3. [8, 7]. The new smooth differential equations can be more easily evaluated and can be used to predict both global and local response. However it should be noted due to the simplification much of the local response will be lost as the function is smoothed out.

Based on Hassani et al. [7] a heterogeneous medium can be stated to have regular periodicity if the physical properties or geometry of the structure can be described by functions of the form:

$$\mathcal{F}(x + NY) = \mathcal{F}x \tag{2.1}$$

where x is a position vector of a point, N is a square diagonal matrix with arbitrary values, and Y is a constant vector that determines the period of the structure. $\mathcal{F}$ can be a vector or a scalar.

2.2 Homogenized Unit Cell Response of Web Core Beam

When a web core beam structure is loaded in bending, the periodic deflection is complex function as seen in Figure 2.3 [8]. The function can be homogenized to smooth the function and give an approximated deflection described by global functions [7]. For example, if it is assumed that the face and interior structures act as Euler-Bernoulli beams, the elements neglects the shear de-

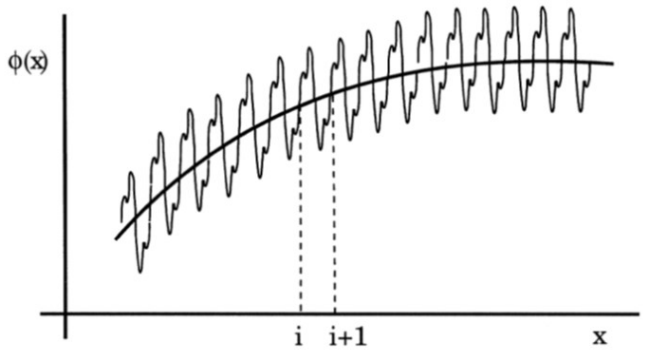


Figure 2.2: Homogenization of Periodic Function [7]

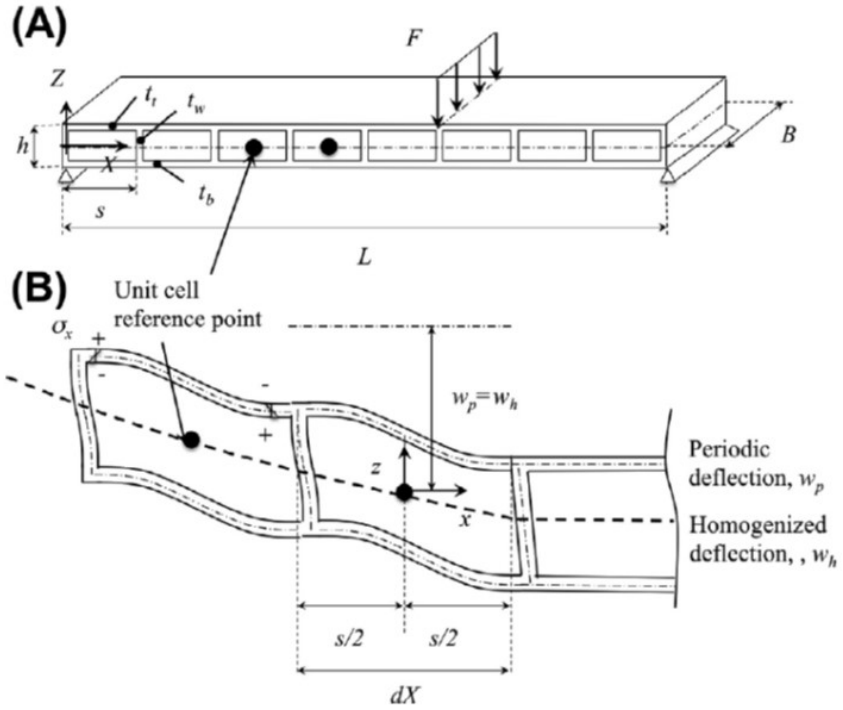


Figure 2.3: Homogenized Deflection Compared to Periodic Deflection [8]

formation on the face plates and assumes it to be negligible [8]. Based on work in shown in ref. [8] the following relations can be made assuming constant bending stiffness:

$$\begin{aligned} \theta_l &= w_l^1(x) \\ M_l(x) &= w_l^2(x) \\ Q_l(x) &= -Dw_l^3(x) \\ q_l(x) &= Dw_l^4(x) \end{aligned} \tag{2.2}$$

where w represents deflection, D represents the bending stiffness, and l represents that the deflection is in the local coordinate system. In general, the deflection at any point within the unit cell can be represented by functions with a combination of constant (w_0), even ($w_{even}(x)$), and odd ($w_{odd}(x)$) terms. Where (w_0) is a constant and the derivatives of ($w_{even}(x)$) and ($w_{odd}(x)$) we assumed to be non-zero. The total deflection can be defined as:

$$w_l(x) = w_0(x) + w_{l,odd}(x) + w_{l,even}(x) \tag{2.3}$$

The even term is associated with pure bending while the odd term is associated with out-of-plane shear deformation. The deflection terms were derived in reference [8]. The results of the derivation show that the odd terms contribute only to the slope and shear force, while the even terms only contribute to the deflection, moment, and distributed loading.

2.3 Assumptions for Loading on Homogenized Structure

According to Romanoff and Reddy [8] when the structure is homogenized the unit cell is combined into a infinitesimal small term. The loads can be dealt with using two assumptions:

- Assumption 1: An external load distributed between the web plates of a web core beam structure only contributes to local bending and is constant thought the unit cell [5].
- Assumption 2: An external load distributed between the web plates of a web core beam structure that contributes to global deflections can be moved as integrals to the web plates [5, 30].

Assuming the external loads effect local bending is constant in the unit cells means that it can be analyzed separate from the global analysis. If the external loads can be moved to the location of the web plates means that the shear force is constant [18].

2.4 Mathematical Model to Describe the Response of a Unit Cell

According to Romanoff and Reddy [8] when the structure is exposed to only odd-terms, out-of-plane shear deformation, the cell deforms as shown in Figure 2.4. The deflection can be described as such:

$$w_Q^i(x) = \frac{Q_Q s^2 d}{24 D_i} \frac{x}{s} [k_1^i \frac{s}{d}(-4\frac{x^2}{s^2} + 3) + 4(6\frac{D_i}{d k_\theta^i})] \tag{2.4}$$

Where $i = t, b$

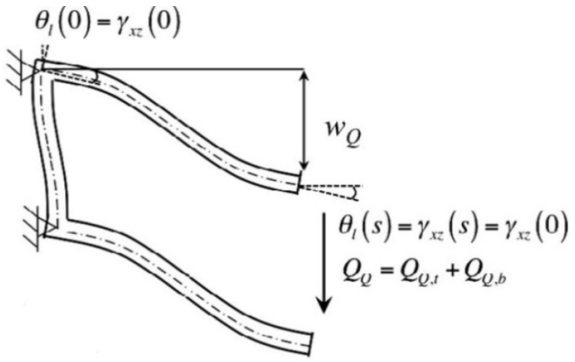


Figure 2.4: Unit Cell Deformation Due to Shear [7]

Q_Q the shear force from the face plate and involves only odd-terms up to the third degree with respect to the x axis. The variable s is the length of the unit cell in the y axis as shown in Figure 2.3 [5]. The stiffness values for the structure are as follows:

$$
\begin{aligned}
k_\theta^i &= Q_Q s/\theta_c^i \\
k_1^t &= 1 - k_Q \\
k_1^b &= k_Q \\
k_2^t &= 2 - 3k_Q \\
k_b^2 &= 3k_Q - 1
\end{aligned}
\tag{2.5}
$$

where

$$
k_Q = \frac{1 + 12\frac{D_t}{s}(\frac{1}{k_\theta^t} - \frac{1}{k_t heta^b}) + 6\frac{D_t}{D_w}\frac{d}{s}}{1 + 12\frac{D_t}{D_w}\frac{d}{s} + \frac{D_t}{D_b}} \tag{2.6}
$$

The variables t, b, and w represent the top, bottom, and web plate respectively. The variable k_θ represents the rotational stiffness of the joint connecting the web and face. The shear stiffness can be obtained by finding the deflection at the ends of the unit cell relative to the cells around it. Equation 2.4.4 represents the shear stiffness web and bottom plates:

$$
D_Q = \frac{Q_Q}{\Delta w_Q/s} = \frac{12 * D_w}{s^2(k_Q(\frac{D_w}{D_b} + 6\frac{d}{s})) + 12} \tag{2.7}
$$

Using assumption 2 the bending moment of the unit cell causes content elongation of the face plates. Therefore, the deflection is defined as:

$$
\begin{aligned}
\frac{d^2 w_B}{x d^2} &= \frac{M_{xx}}{D_{xx}} \\
w_B(0) &= 0 \\
w_B^1\frac{-s}{2} &= w_B^1\frac{-s}{2}
\end{aligned}
\tag{2.8}
$$

Therefore,

$$
w_B^1(x) = \frac{-M_{xx}}{2D_{xx}} \tag{2.9}
$$

The term D_{xx} can be described using equations 2.4.7 and 2.4.8

$$
D_{xx} = D_0 + D_t + D_b \tag{2.10}
$$

$$
D_t = \frac{E_t t_t E_b t_b d^2}{E_t t_t + E_b t_b}, D_i = \frac{E_i t_i^3}{12}, i = t, b \tag{2.11}
$$

The derivations and equations related to virtual work and the solution for the differential equation can be found in ref. [22]. By using the above terms and assembling the necessary matrices the deflection can be calculated for the structure.

2.5 Coupling Homogenized Beam Theory and the Unit Cell Response

When the sandwich beam deforms in shear and bending, the shear forces are transferred from the face plates to the web plates at the joints. As the shear forces are transferred, they induce a moment on web plates, as shown in Figure 2.1 [5, 39, 40]. It can be assumed that in all cases the sum of the moments on the web created at the top and bottom joints are zero. Therefore, this deformation and stress induced by the moments are lost when the structure is homogenized. In reality it is an important interaction that needs to be accounted for to accurately predict the global deformation of the beam [5]. Romanoff and Reddy [8] show by introducing a couple stress term it can account for the missing interaction. The couple stress stiffness as shown by:

$$S_{xy} = 4\frac{D_{xxD_f}}{D_0 + \frac{s^2}{2}D_Q}, D_f = D_t + D_b \tag{2.12}$$

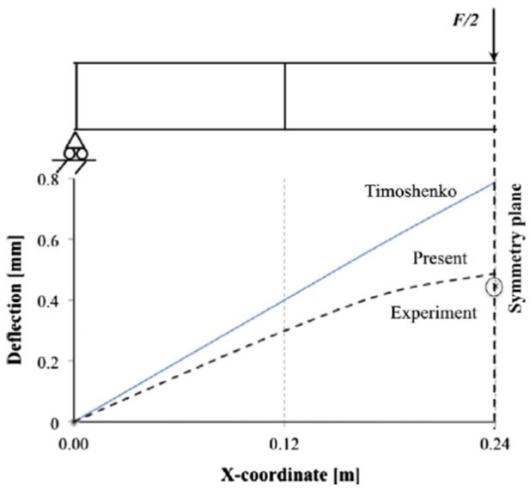
This term interacts with the solutions differential equation changing the values related to the stiffness of the structure. The presence of $D_f = D_t + D_b$ introduce the bending stiffness of the face and web plates as well as the rotational stiffness at the joints where the plate interacts. This allows for the moments induced by the shear stress to be accounted for in the solution [8].

2.6 Experimental Validation of Coupling Homogenized Beam Theory

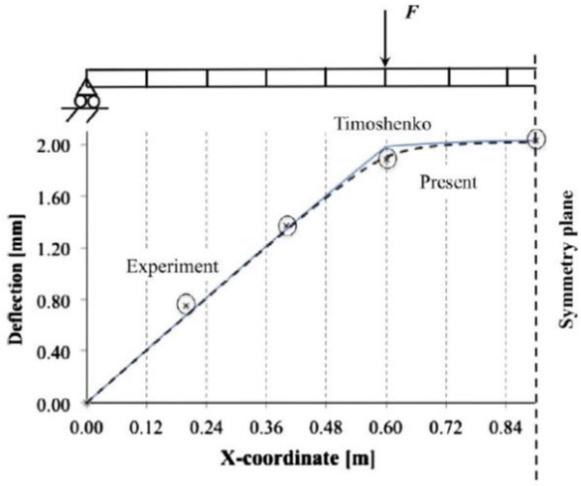
2.6.1 Experimental Setup

Experimental validation was done in ref. [8] and compares two different methods against experimental data. The details for the experimental procedures can be found in the following references [31, 37, 51]. In brief the experiment used steel with a Young's modulus of $E = 200 - -220$ GPa and Poisson's ratio of v = 0.3. In the experiment the deflection of the structure is measured in 3 and 4 point bending tests. Figure 2.5 shows the results of a three point bending test on a structure with 4 unit cells. The total length of the beam is 480 mm, the face plates have a thickness of 3.03 mm, and the web plate have a thickness of 4.03 mm. The force (F) is applied in the center of the beam with a magnitude of 1 N/m. Figure 2.6 shows the results of a 4 point bending test on a beam with 15 unit cells. The total length of the beam is 1.8 mm, the face plates

have a thickness of 2.86 mm, and the web plate have a thickness of 3.97 mm. The force (F) is applied to the beam is 1 N/m. In both these cases the unit's cells are all equal and spaced evenly across the length of the beam. It should also be noted that the beams are wide enough, so face and web plates are assumed to be in plane stress [8].



(a) Comparing Predicted Deflection of 4 Cell Web Core Beam in 3-point Bending. [8]



(b) Comparing Predicted Deflection of 15 Cell Web Core Beam in 4-point Bending [8]

Figure 2.5: Comparing Conventional and Modified Homogenized Timoshenko Beams Theories Predicted Deflection of Web Core Beam

2.6.2 Results Comparing Experimental Data with Conventional and Coupled Stress Homogenized Beam Theory

Figure 2.5 shows that the deflection predicted by the modified (Coupled Stress) conventional Timoshenko beam theory are fairly accurate. However, it overestimates the largest deflection. When looking at the present Timoshenko beam theory with the addition of the coupled stress term the accuracy is improved, and the prediction follows the experimental results more closely. As shown in equation (2.5.1) the term increases the local stiffness of the unit cell. The additional local stiffness affect the global stiffness of the beam and better approximates the maximum deflection.

This indicates that the addition of the couple stress term increases the accuracy near locations that a load is applied. This is because these locations have high shear force gradients were the interaction between the face and web plates is important when modeling the global response. It also indicates that the couple stress theory is more accurate for shorter beams when compared to the conventional Timoshenko beam theory. This is because a short structure will be dominated by shear rather than bending [8].

The approach of homogenizing web core beam structures provides a way to solve for the deflection and stress of the structure in bending. Previously face and web plates of the beam were modeled as conventional Euler-Bernoulli or Timoshenko beams. This was later changed to Timoshenko beams and a coupled stress term added to increase the accuracy of the mathematical model. This new theory accounts for the local effect of the web plates deforming under shear loads from the face plates. Accounting for the effect increases the global stiffness of the beam in both shear an bending loads. The modified theory was then compared to conventional Timoshenko beam theory and experimental results. Based on the results it was observed that the couple stress theory smooths the deflection at locations with higher shear force gradients and is more accurate when compared to experiment results [8].

2.7 Micropolar Theory

A micropolar theory includes a local microrotation that independent of the global macrorotation of the beam [9]. The Coupled stress term described in section 2.5 is found through the assumption that the micro and macro rotations coincides with each other. In [9] this assumption is relaxed and uses a micropolar theory with Timoshenko beams to study the deformation of web core beams. The derivation for the model can be found in ref. [9].

In short the unique features of the study are the derivation of an explicit general solution to the equilibrium equations of a micropolar Timoshenko beam. This is then used to develop a nodally-exact micropolar Timoshenko beam finite element (FE) for a web-core beam. Lastly, it applies the beam model and elements to a realistic problem with the micropolar equivalent single layer (ESL) stiffness parameters found through a unit cell analysis of a web-core sandwich beam. To evaluate

the accuracy of the model it is compared against the conventional homogenized beam and coupled stress homogenized beam theory.

2.7.1 Comparing Micropolar, Conventional Homogenized, and Coupled Stress Homogenized Timoshenko Beam Theories

In ref. [9] Three beam calculation examples were presented, however only one will be shown in this overview. The beams consist of rectangular web-core unit cells as shown in Figure 2.1 and are similar to those empirically tested in ref. [52]. The face and webs plates are made of steel ($E = 210$ GPa, ν = 0.3). All joints connecting faces and webs plates are modeled here as rigid connections. The beams overall width and height are 0.05m and 0.043m respectively. The web spacing between webs is 0.12 m and the face and web plate thicknesses are 3mm and 4 mm, respectively. Figure 2.6 shows the examples studied in [9]. Reference solutions were found using 2-D FE Euler–Bernoulli beam frames modeled in Abaqus.

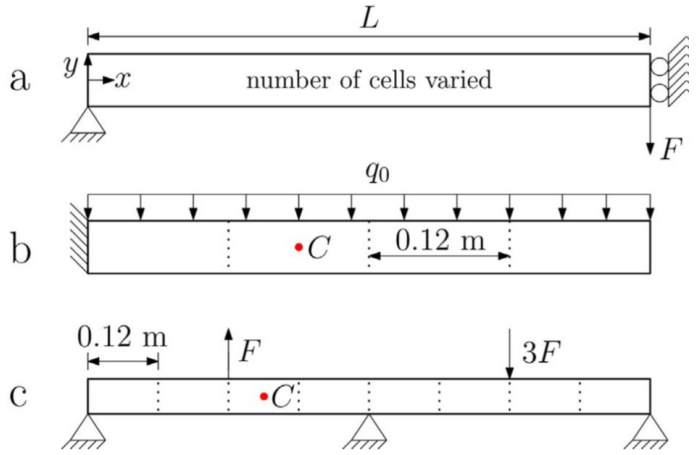


Figure 2.6: (a) Three-point bending of a web-core sandwich beam modeled by a symmetric half. (b) Web-core cantilever beam under a uniformly distributed load. (c) Web-core beam on three supports modeled by using four beam elements. [9]

Figure 2.7a shows the transverse deflections of the different ESL Timoshenko beam models based on figure 2.6a and Figure 2.7b shows the errors calculated from equation 2.13 in terms of

the maximum deflection.

$$\Delta u_y = 100 * \times \frac{(u_y^{1-DTimoshenko} - u_y^{2-DFE})}{u_y^{2-DFE}} \% \tag{2.13}$$

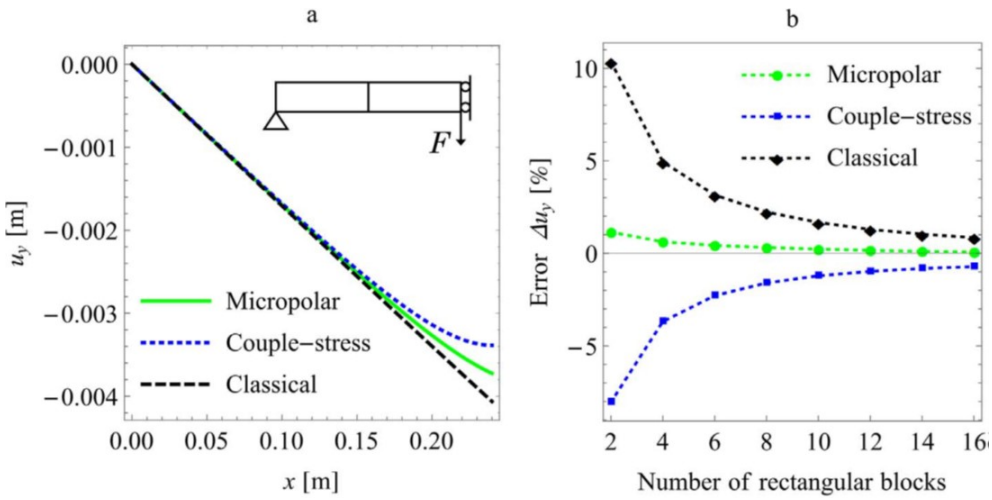


Figure 2.7: (a) Transverse deflections of 1-D conventional, coupled stress and micropolar ESL Timoshenko beams under three point bending. (b) Errors of the 1-D beam models in terms of maximum deflection in comparison to 2-D FE beam frame solution (face sheet deflection) calculated using Abaqus. [9]

The classical (conventional) and couple-stress models result in larger errors than the micropolar approach for short beams with only a few units cells. As the beams become longer, and more unit cells are used the errors between the models become smaller. To understand the reason for this, it can be shown that the difference between the rotation variables, macrorotation and microrotation, is non-zero only near the slider support as shown in figure 2.6a. The difference can be show to be directly proportional to the antisymmetric shear strain γ_a and shear force.

When evaluating points located at sufficient distance from the slider support, the unit cell only has symmetric shear strain, therefore ($\gamma_a = 0$) as shown by the 1-D micropolar model. The antisymmetric shear strain contributes to the approximation near the slider support which causes the 2-D unit cell attached to it to deform in such a way that can not be described with exclusively sym-

metric shear strain. Therefore, only micropolar approach can capture the antisymmetric behavior in this case. However, because this only occurs in the unit cells near the the slider support, the results given by the classical and couple-stress ESL beams are fairly accurate as the beam becomes longer and more unit cells are added as shown in Figure 2.7 [9].

2.8 Summary of Results

In the case of short beams with few unit cells, homogenized coupled stress theory using Timoshenko beams provides too stiff a response for structure because of the constraint on the independent rotations. All in all, the micropolar approach for modeling these types of structures has the greatest complexity but also is the most accurate currently developed [9]. It clearly shows the the local rotations and forces are important in accurately predicting the deformed shape of structure specifically near boundary conditions.

3. PRINCIPLE COMPONENT ANALYSIS

3.1 Justification and Mathematical Overview of PCA

To clearly determine the shape of the deformed webs, principle component analysis (PCA) was used. PCA allows a data set to be broken down into the most critical components that can be recombine to represent the data set. It does this by creating a group of vectors that align to minimize the distance between every point in the data set while also being orthogonal to the previous vector. These vectors are called principle components (PCs) and the number of components depends on the dimensions of the data set. These principle components represent the most meaningful trends, where the first PC is the most meaningful and the last PC the least meaningful. By finding all the PCs and removing the later PCs it allow the data to be reduced into the most meaning full trends while still retaining the general trends and meaning of the original data set. This can be used to reduce the size of the original data set while not losing very much information [53, 54]. This approach has applications in noise reduction [55] and machine learning [56].

To find the principle components in the data set a singular value decomposition (SVD) can be preformed on the data set. An SVD decomposes a matrix of any size into three separate matrices V, Σ, and U. Specifically, the original data would be held in an m column by n row matrix X and decomposed into U an $m \times m$ unitary matrix, Σ an $m \times n$ rectangular diagonal matrix with non negative real numbers, and V^T an $n \times n$ unitary matrix. Equation 3.1 shows the expression for the SVD. Any complex square matrix is unitary if its conjugate transpose is also its inverse. Therefore U and V it must meet the mathematical expression in equation 3.2.

$$X = U\Sigma V^T \tag{3.1}$$

$$I = U^*U = UU^* \tag{3.2}$$

The values in Σ can be found by finding the eigenvalues from X^TX or XX^T. The U matrix can be found by finding the eigenvectors of X^TX. Similarly V matrix can be found by finding the

eigenvectors of XX^T. If X is real then U and V^T will be real orthogonal matrices.

By using this technique the most prominent shape function can be found in a robust way. This could be done by visual inspection of the webs deflection, however it would be subjective to the viewer and in cases with many web it would be difficult to generalize. Utilizing this method this method it is mathematically sound to state the functions order for all the webs in the structure.

For example, an ellipse has a major and minor axis. These axis are orthogonal to each other and the major axis is larger than the minor axis. If PCA was applied to a data set with an elliptical shape, it would find 2 vectors. These vectors would match the major and minor axis. The major axis would be the first principle component because it represents the vector closest to all the points of the data set. The minor axis would be the second principal component because it is orthogonal to the first and represents the vector closest to all the points in that plane. The matrix X is the data set for an ellipse described by equation 3.3 sampled from -5 to 5 in steps of 1 along the x axis. U, Σ, and V^T are the matrices found by the SVD process. The values for these matrices can be found in the Appendix A. Figure 3.1 shows the data set along with the first and second principle components. Each principle component can be isolated by setting the eigenvalues for the PC to zero, as show in equation 3.4. The first eigenvalue belongs to the first PC and so on. In this example the eigenvalues for are each vector are 14.8 and 3.6 respectfully. There are only two eigenvalues because the data set is two dimensional. The U and V matrices are listed in Appendix A.

$$\frac{x^2}{25} + y^2 = 1 \quad or \quad y = \frac{(25 - x^2)^{1/2}}{5} \tag{3.3}$$

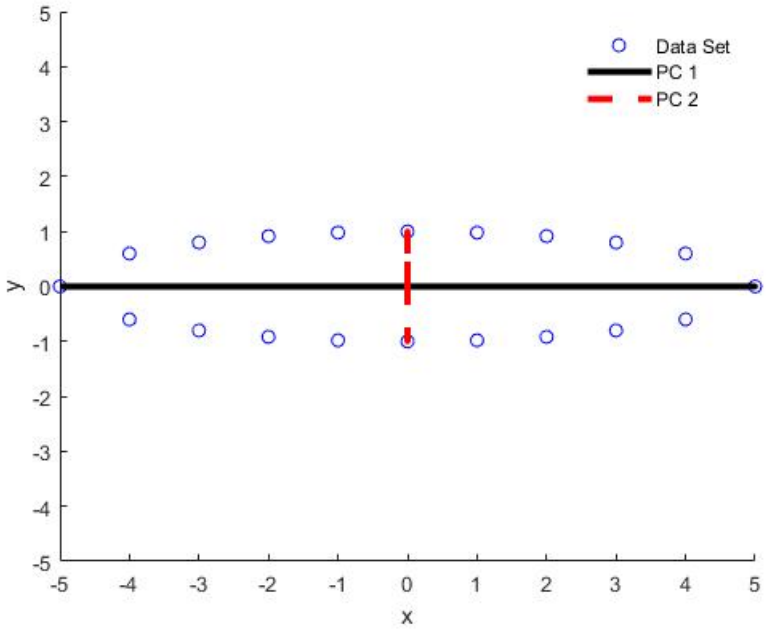


Figure 3.1: PC Vectors for Elliptical Shaped Data Set

$$X = U \begin{bmatrix} 14.8324 & 0 \\ 0 & 3.6332 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} V^T \tag{3.4}$$

Using different numbers of principle components changes the accuracy of the model produced

by the SVD. If only the first PC is used then in this case only the major axis is used. The introduction of the second PC exactly matches the data set because the example is 2D and the Σ matrix only has two values. It can be seen that the general information about the data, that it is extended mostly along the x-axis, can be seen using only one PC. This is the general shape and if the minor axis was much smaller then the major axis, this single PC would offer a good approximation. Figure 3.2 show the approximated points using different amounts of PCs.

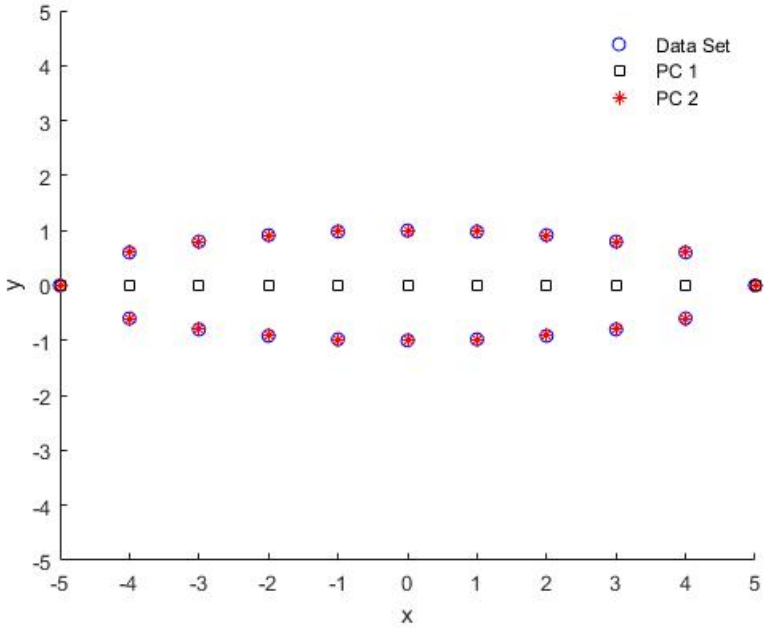


Figure 3.2: Model Approximations Using Different Numbers of PCs for Elliptical Shaped Data Set

With these two PCs the shape of the data set can be reformed. All matrices and code can be found in Appendix A. This technique commonly used in machine learning and intelligent systems to address the problem of using computer algorithms that can deal with huge amounts of data. By finding the PCs it is possible to find the most important trends in the data set to reduce the amount of data that needs to be transfer for communications and computation [53, 54, 56].

The importance of each PC can be shown in a scree plot. This is a diagnostic plot that shows

how much variation each PC has in the data set. Equation 3.4 shows how the variance for each PC is calculated were j can be any integer value between one and m and represents a given PC. Figure 3.3 shows the scree plot for the sampled data set. The figures shows that the first PC holds 80% of the variance in the data set. This means that depending on the accuracy and detail of the approximation needed, in this example it might be acceptable to only need the information about the first PC of this data set.

$$Variance(j) = \frac{\Sigma(j,j)}{\sum_{i=1}^{n} \Sigma(i,i)} \tag{3.5}$$

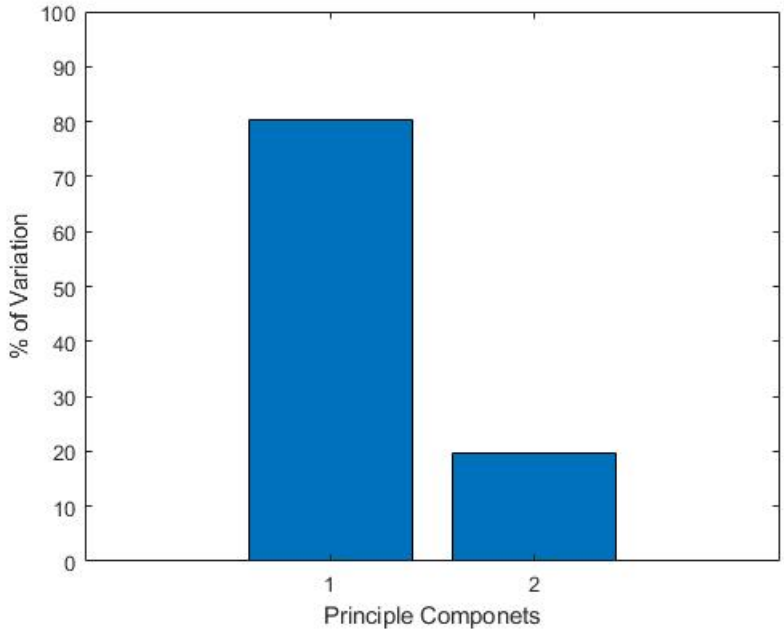
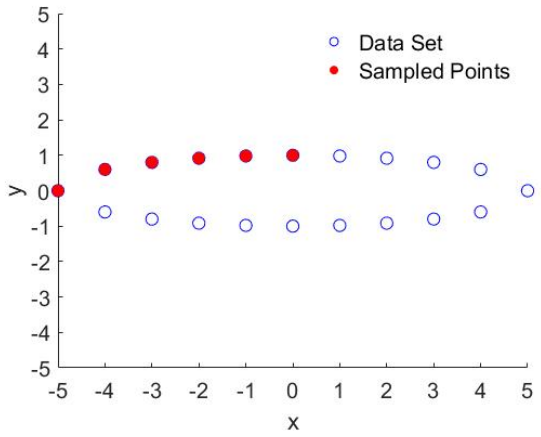
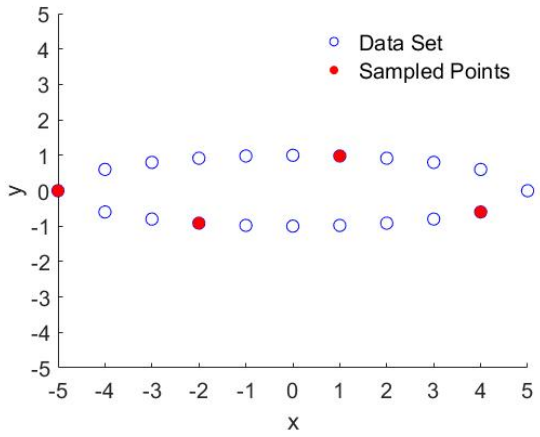


Figure 3.3: Comparing 'Influence' of PC for Elliptical Data Set

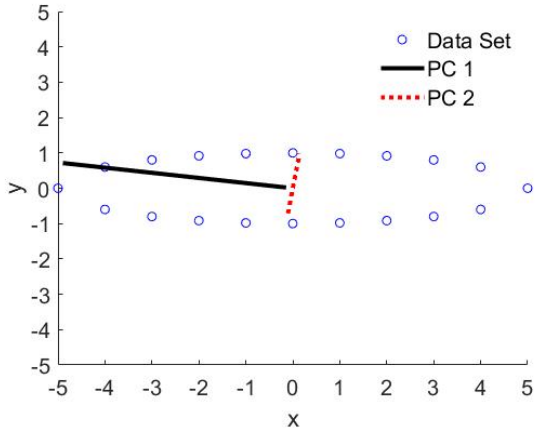
It should be noted in the above example all the data points were used to in the analysis however when the data set becomes large it might be unreasonable to use every data point in the analysis. In this case samples can be extracted from the data set and analysed to approximate the full data set. Using the same elliptical data set the points sampled can be show in Figures 3.5(a) and 3.5(b).



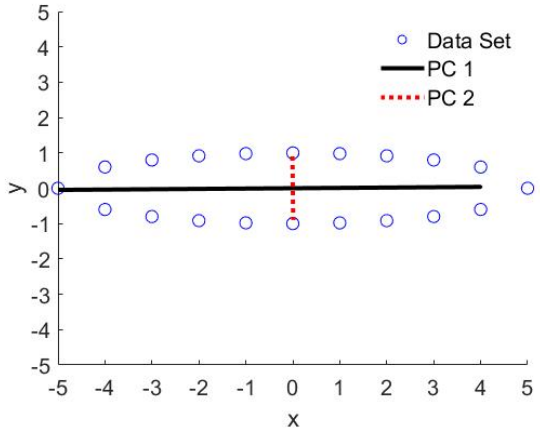
(a) Poorly Selected Samples, Elliptical Data Set with First Six Data Points Sampled for PCA



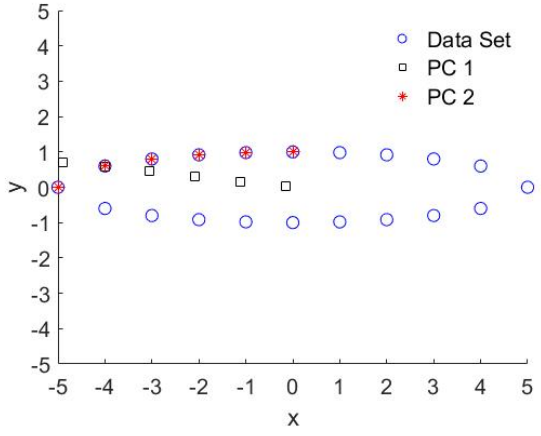
(b) Well Selected Samples, Elliptical Data Set with Four Data Points Sampled Every Sixth Point for PCA



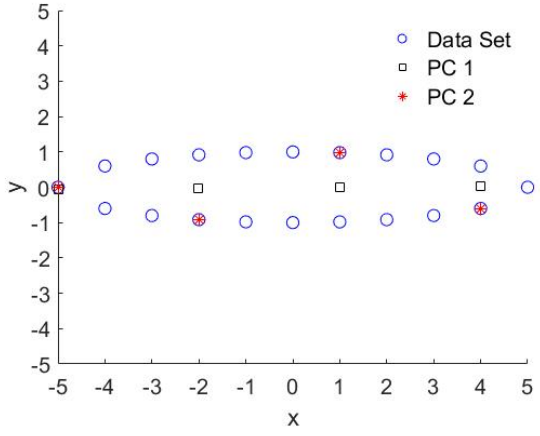
(c) Poorly Selected Samples, PC Vectors for Elliptical Shaped Data Set



(d) Well Selected Samples, PC Vectors for Elliptical Shaped Data Set



(e) Poorly Selected Samples, Model Approximations Using Different Numbers of PCs for Elliptical Shaped Data Set



(f) Well Selected Samples, Model Approximations Using Different Numbers of PCs for Elliptical Shaped Data Set

Figure 3.4: Sampled Points from Elliptical Data Set

Figures 3.4 a, c, and e show the results of poorly selecting points to base the reduced model on. The PC vectors shown in Figure 3.4 c clearly are not aligned with the major and minor axis of the data set, shown in Figure 3.1. By only selecting the first six data points in the set, the trends in the model do not approximate the overall data set well. Figure 3.4 b, d, and e show the results of well selected points to base the reduced model on. The PC vectors shown in Figure 3.4 d closely match with the major and minor axis of the data set, shown in Figure 3.1. This was accomplished by selecting points that reflect the overall shape of the data set rather than just a single part. It is important that when reducing the data set to capture points at extreme locations, so the full data set can be accurately approximated.

3.2 Application to Web Core Beam

The data set for the web core beam contains only the deformed shape of the webs as they deflect in the horizontal axis. The deformation in the y-axis would not affect the overall shape of the web only compress or extend the shape in the vertical axis. The data was gathered using ANSYS 2D static structural simulations using square elements and extracting the nodal displacements in the center of each web. The nodal displacements were found by applying a force in the center of a web core beam structure. The structure is constructed of six 2.5 inch tall webs spaced out by 7.25 inches and all members have a thickness of 0.25 inches and made of steel with an Elastic Modulus of 29 ksi and Poisson's Ratio of 0.3. Loads varied from 100 to 15,000 lb_f applied to the center, as shown in Figure 3.5. Figure 3.6 shows the results of deflection in the x-axis. The webs are numbered starting with web 1 on the far left side and ending with web 6 on the far right side. Figure 3.7 shows the mesh applied to the structure. 2D square shell elements were used because they capture the micro-deformation in the structure. Beam elements would simplify the connection between the webs and face plates and would not provide any resolution of points along the x-axis of the web if the information was deemed necessary. 3D elements would have not given any additional information because there is no deformation in the z-axis and would only increase the computational time need. More information about the simulation can be found in Appendix D.

The mesh produced a total of 52 data points along the center of the web but only 16 points were

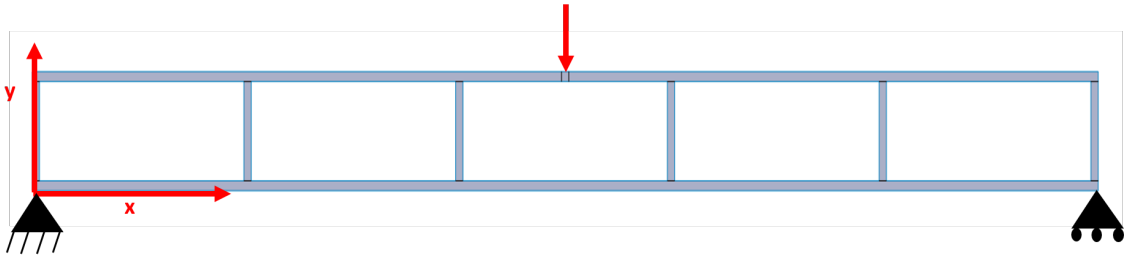


Figure 3.5: Location of Applied Load and Fixtures on Web Core Beam Structure

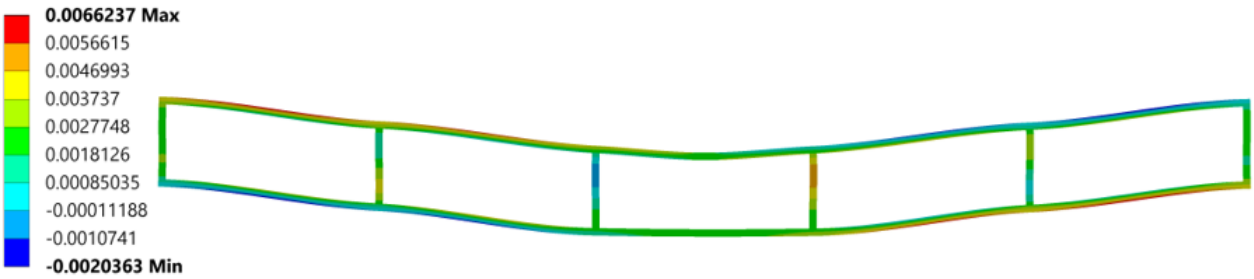


Figure 3.6: Resulting Deflection in the Horizontal Axis from Simply Supported Center Loaded Structure

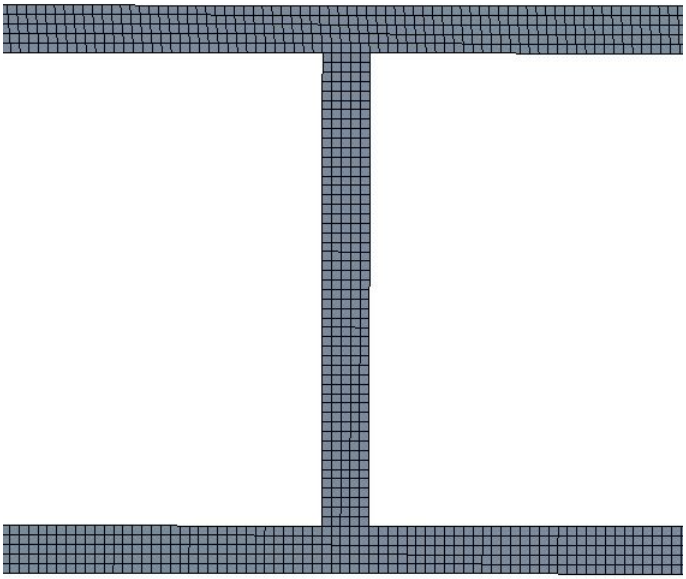


Figure 3.7: Fine Scale 2D Element Mesh at Web

sampled to be used in the mathematical analysis. This was done to lower the size and complexity of the mathematical analysis and provide evidence that the approach is suitable for similar situations. However as shown by the example using an elliptical data set it is important to capture the extreme point of the data set. In this case the points that need to be captured are at the locations where the slope of the web is zero with respect to the vertical axis. These locations are were the function will need to change direction and are important features. By subjectively observing were the change occurs along the y-axis for each web it can be noted that there is no common location along the y-axis where the webs slope is zero. Therefore more points need to be sampled to insure that all the points are accounted for in the simplified model. The sampled point formed an, m column by n row, matrix, X, were m is the number of webs and n is the number of samples. Using the SVD function built into MATLAB the matrices $U, \Sigma,$ and V^T were found then analyzed. Figure 3.8 shows the normalized deflection of each webs deflection in the horizontal axis, for the scenario shown in Figure 3.5 and 4.6 with a load of 10,000 lb_f. A single web, web 4, will be used to show how the analysis was done, before applying the same process to all the webs in the structure. The model used to approximate the deflection of web 4 is based on the PCs from the data set including all the webs in the structure, rather than just the single web.

Using the sampled points, the principle component vectors were found and used to approximate the deflection of the web. By only using the first few principle components and setting the others equal to zero, the most important shapes in the data can be seen. Later principle components while useful only help to reduce the approximations error by a small amount and do not contributed to the overall order of the function. Figure 3.9 shows how the approximation improves by adding each principle component, up to four. Using the approximated points an approximation function can be found by curve fitting to find what order fit is needed to approximate the shape of the deflected web, as shown in Figure 3.10.

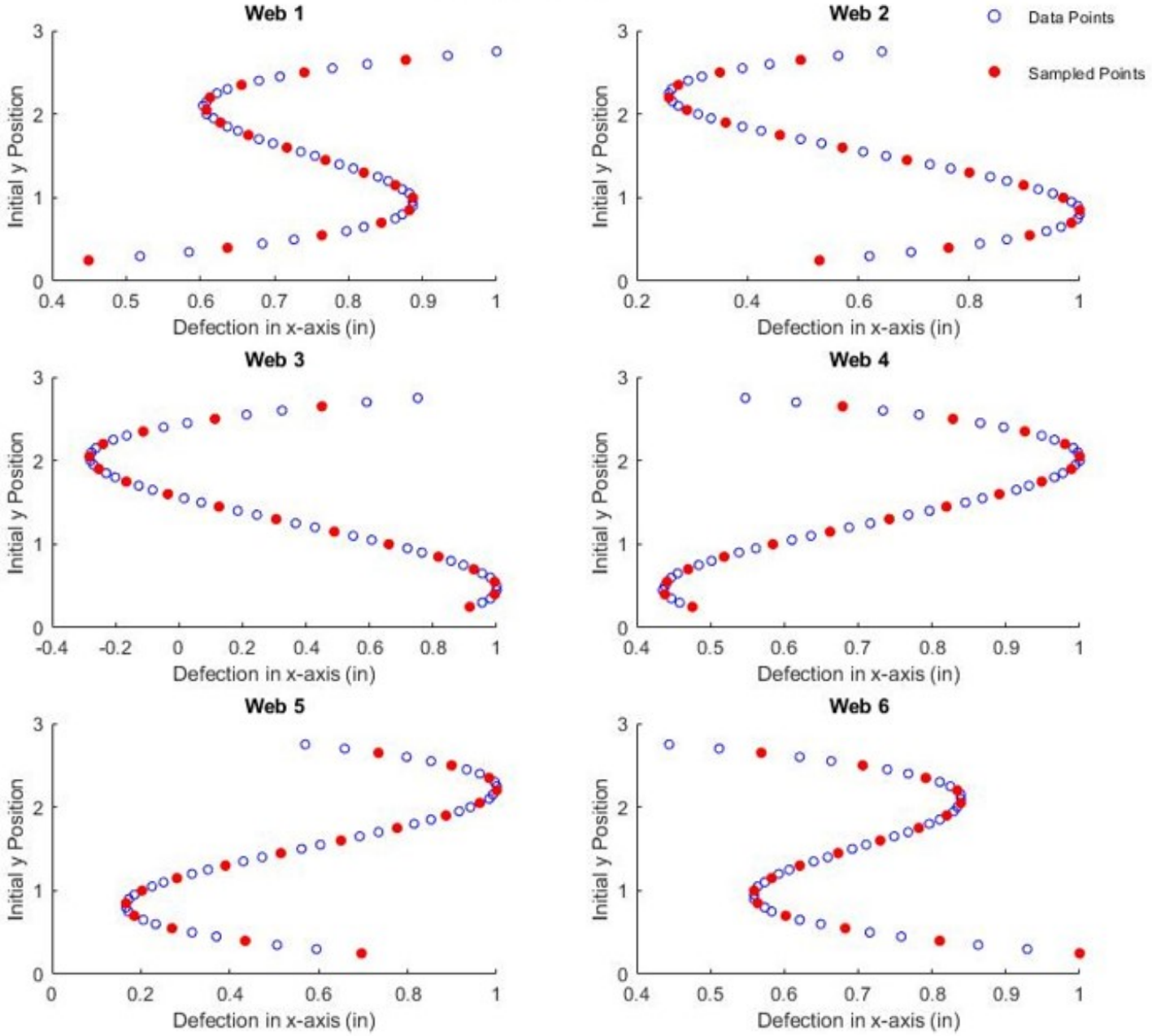


Figure 3.8: Normalized Nodal Deflection and Selected Nodes for all Webs, Load = 10,000 lb_f

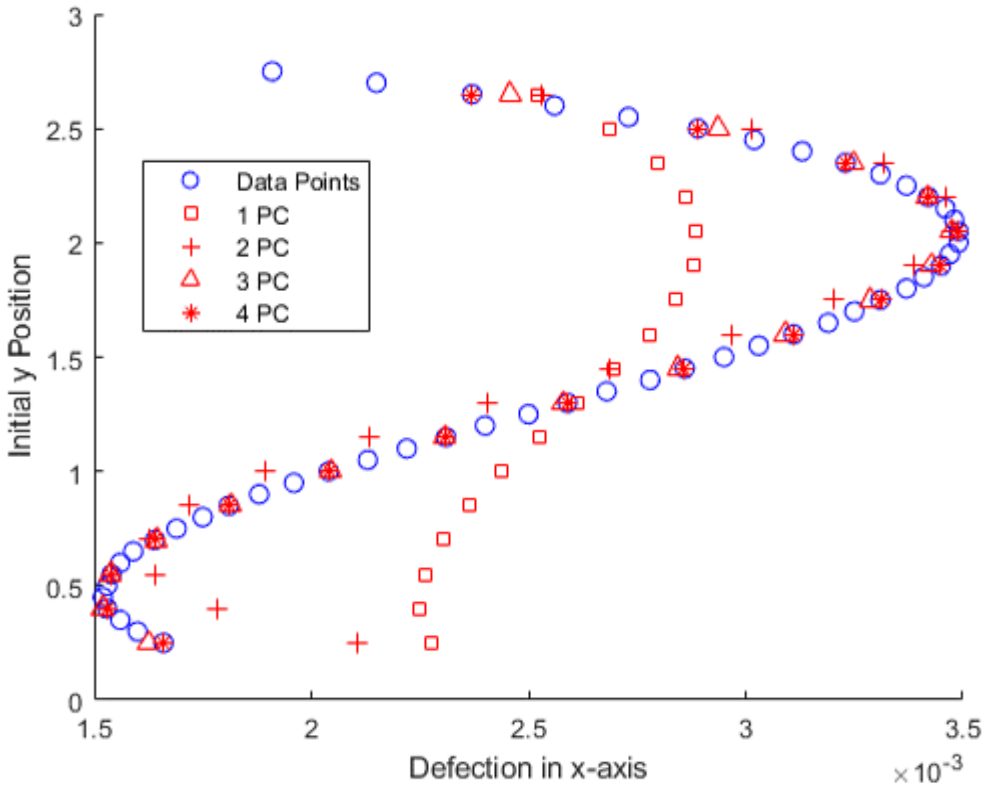
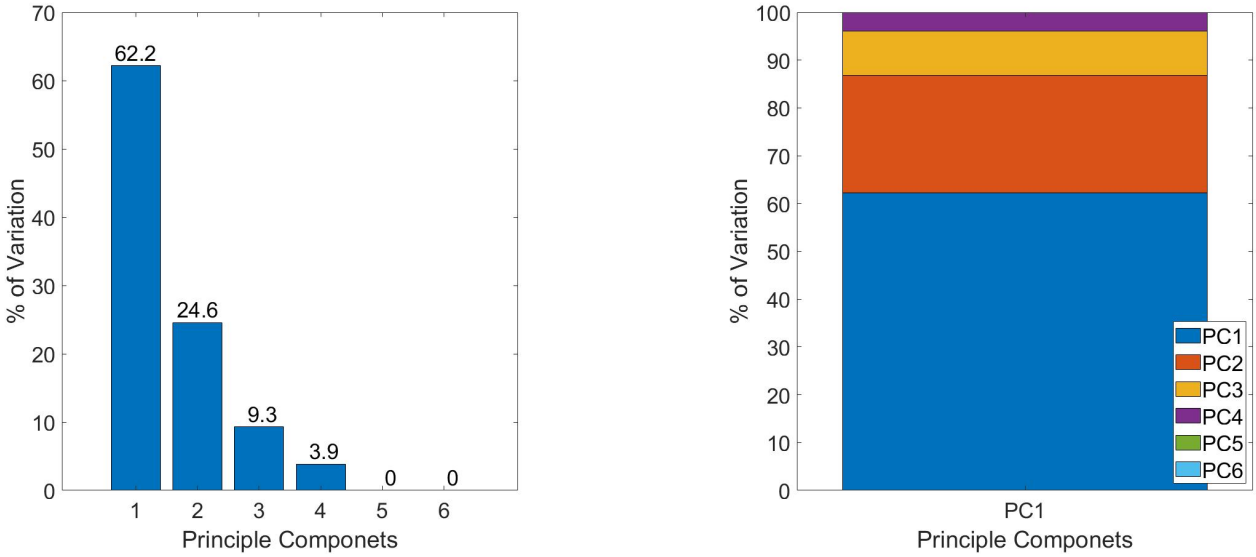


Figure 3.9: Modeled Deformation via Different Principle Components for Web Four

Using Figure 3.9 it can be seen that using just one or two principle components does not produce a close match to the nodal deformation along the web, however using three or four is much closer. The difference between three and four PCs is small and visually it is hard to tell a difference at some locations. This can also be observed in the scree plot, Figure 3.10, showing the variance in each PC for the model approximating all the web in the structure. It can be seen that the fifth and sixth PC do not add very much information to the model, less than 0.1% variance. PCs one and two account for around 85% while adding PC 3 raises the total variance accounts for around 95% of the total variance. This model needs a high number of PCs because all the webs are modeled together and to model the details of each web requires more PCs because each web deforms differently.

Using three principle components it can be shown that a cubic function best fits the data approximated points. Figure 3.11 shows different order fits over the approximated points found by



(a) Separated PCs

(b) Stacked PCs

Figure 3.10: Scree Plot for Sampled Points for Data Set from Simply Supported Center Loaded Structure

simplifying the full SVD down to three PCs. Table 3.1 shows the R^2 values for each of these fits.

This proves that the shape of the web can be most simply modeled as a third order function.

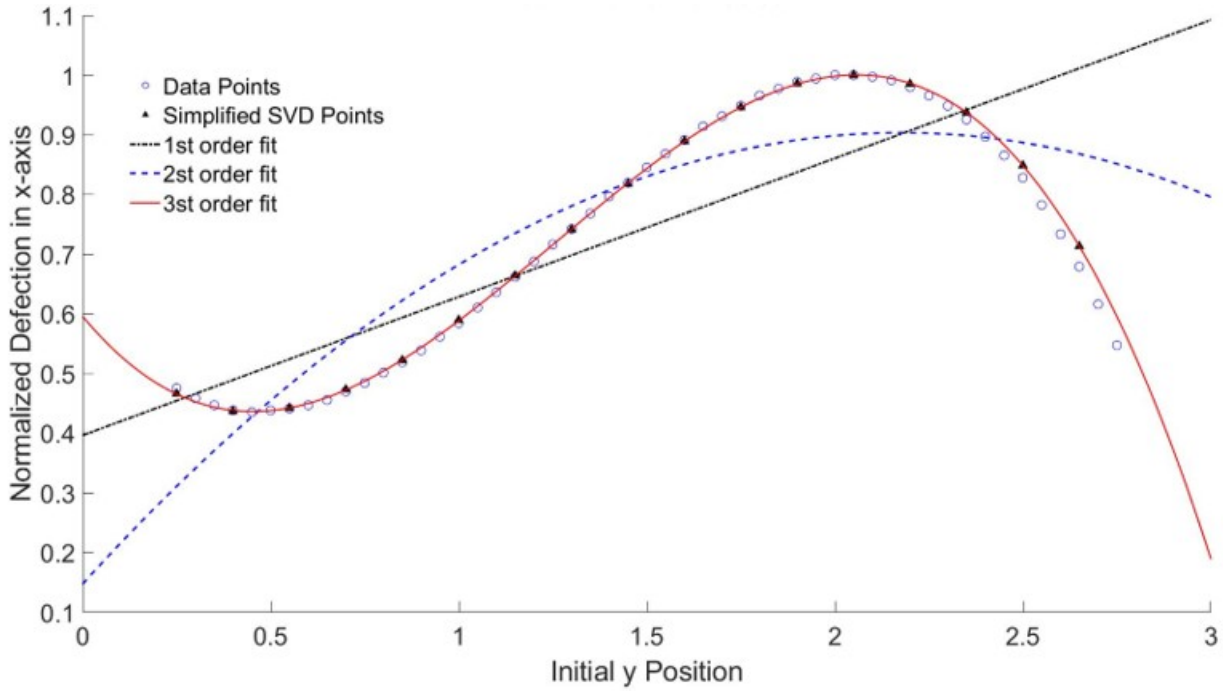


Figure 3.11: Comparing Different Order Fits for Web Four

Table 3.1: Error for Different Order Fit Functions

Order of Fit	R^2
1	.6941
2	0.8341
3	1.0000

Figure 3.12 shows the error between a cubic function fit over the approximation for different amounts of PCs and the original data set of nodal displacements. Equation 3.5 shows how the error was calculated. Based on the results the plot shows that using three PCs models the majority of the deformation well, but does decreases in accuracy at the ends of the web. Using four or more PCs results in close to no error between the approximation and original data set. Even when all six PCs are used in the model the error is still non zero when compared to the original data set. This is because by sampling the points some information about he overall data set is lost and leads to error between the sampled points.

$$\%Error = \frac{Approximated\ Displacement - Nodal\ Displacement\ Data}{Nodal\ Displacement\ Data} \cdot 100\% \tag{3.6}$$

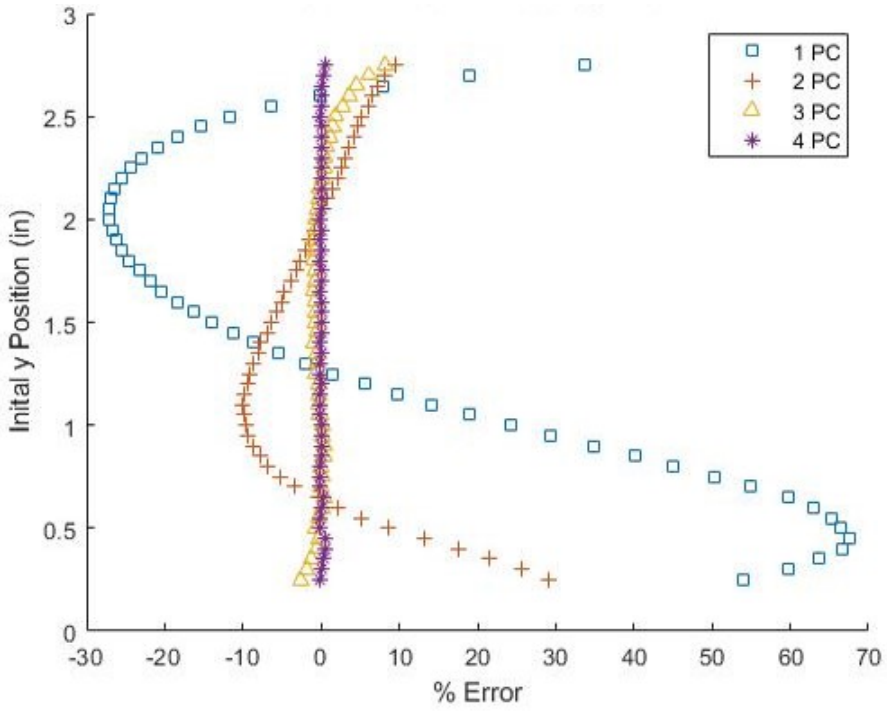


Figure 3.12: Error Between Modeled Deformation and Nodal Deformation via Different Principle Components for Web Four

In the above work only a single web was shown but as shown in Figure 3.8 each web deforms into a unique shape. To validate this conclusion all webs within the structure must be evaluated in a similar way. Figure 3.13 shows a cubic fit applied to an approximation using three PCs over the original data set. Figure 3.14 shows the error for approximations using different amounts of PC's similar to Figure 3.12, but for all the webs. The high error in the approximation of web three is due to the values of the deflection becoming near zero so even small errors differences can grow to a large percentage error due to the nature of Equation 3.6.

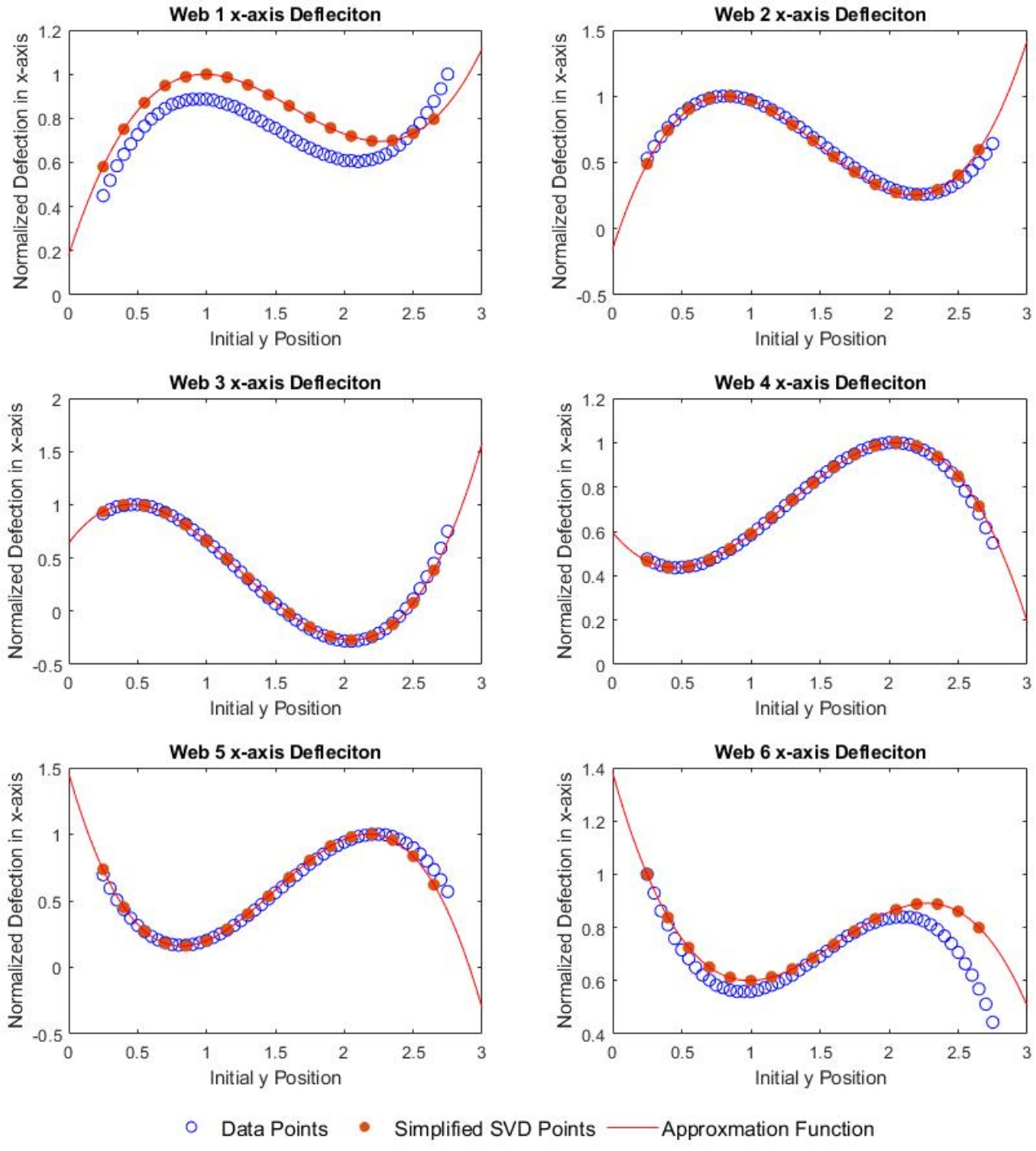


Figure 3.13: Comparing Cubic Fit on Simplified SVD Model with 3 PCs with Original Data Set

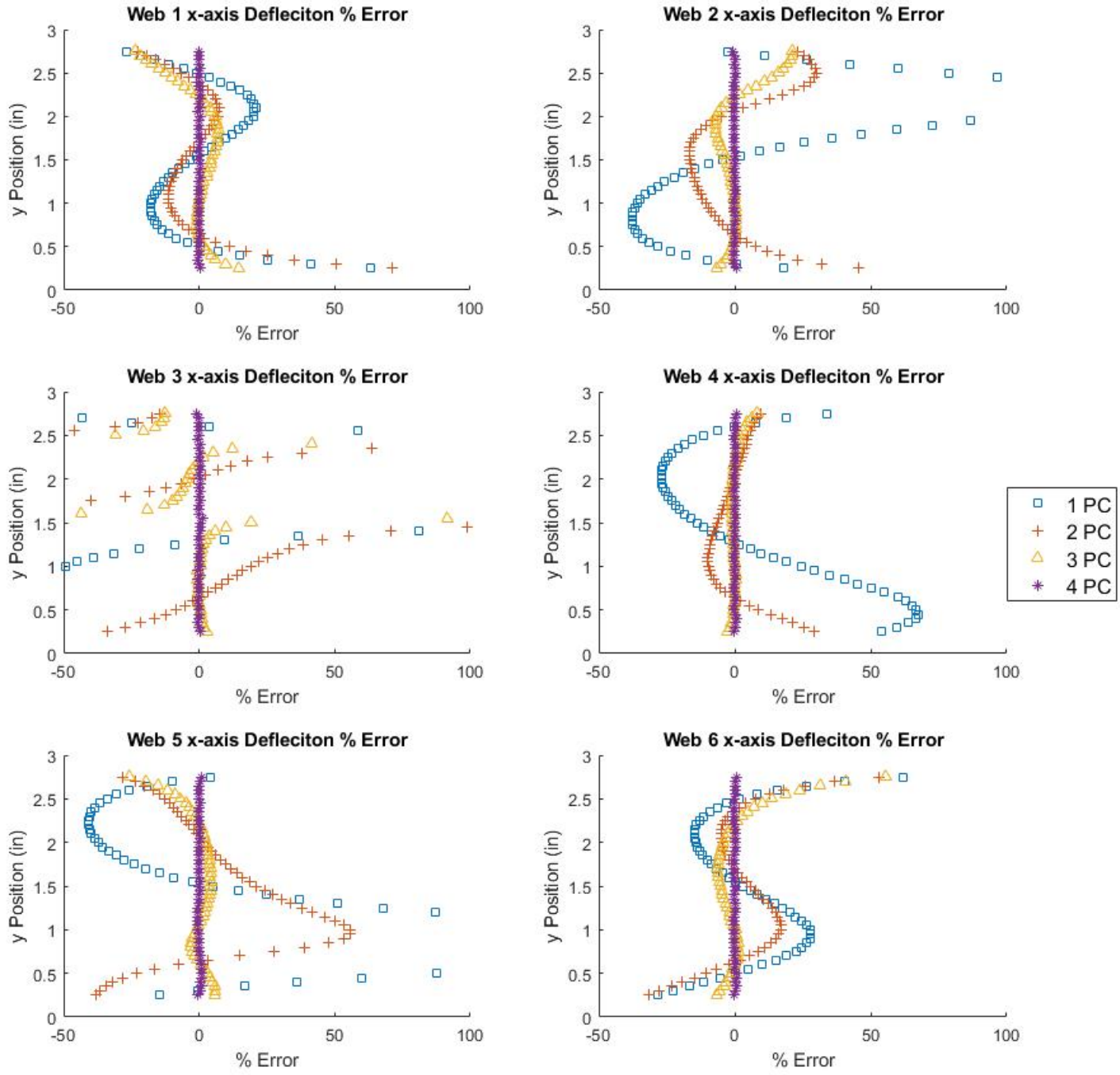


Figure 3.14: Comparing Nodal Displacements Approximated with Cubic Fit Error by Different Numbers of PC's for a Center Loaded Web Core Beam

Figures 3.13 and 3.14 shows that while the approximation is not perfect it does capture the general shape of each web. The webs in the center of the structure are more closely approximated while the webs near the boundary conditions, web 1 and 6, do not match as closely. Based on previous findings this is because the specific shape created near the boundary conditions is unique compared to that of the webs away from the boundary conditions. This difference is because anti-symmetric shear strain contributes to the deformation near the boundary conditions, however the stain is more symmetric away from the boundary conditions. Karttunen et al., show this is the case in ref [9], and is the reason way the micro-polar approach is able to more accurately approximate the global deformation of short beams.

The above process was also applied to a web core beam loaded in the center as shown in Figure 3.1 with different magnitude loads applied. Figure 3.15 shows the nodal deflection for a variety of loads. It is shown the general shape of the deformation is the same for a given web despite the load changing. As the load increases the deformation increases and the web also translates along the x-axis due to deformation in the face plates, this is why the webs do not stay in a consistent location. Figure 3.16 shows the same scenario and loading but with normalized deformation for each web. This shows the the shape of the deformation is the same and only the magnitude of the deformation changes with changes in load. The result makes sense because the deflection occurs in the linear elastic region of the material so deformation is correlated with the force applied and the elastic modulus of the material.

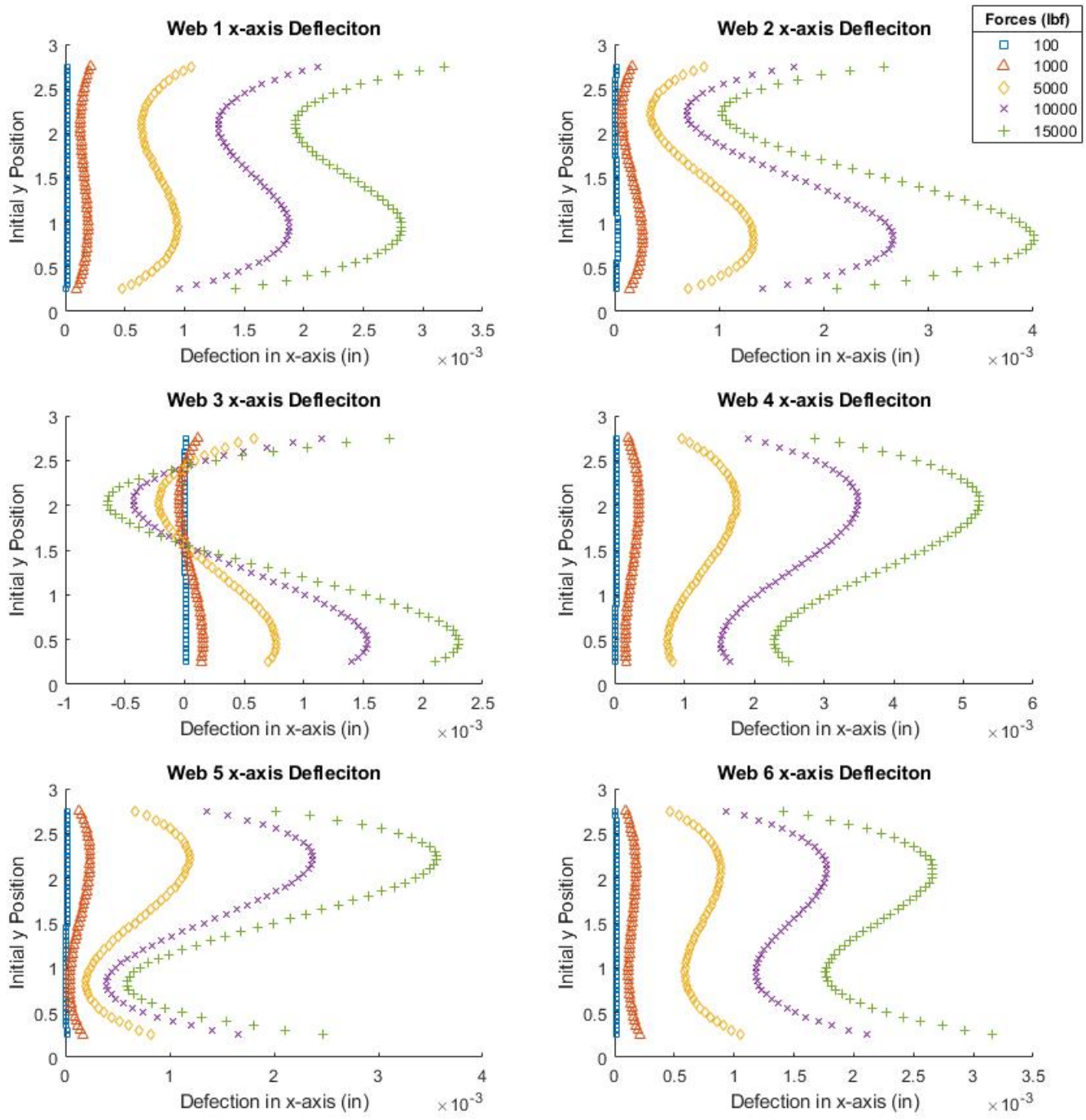


Figure 3.15: Comparing Deflection Caused by Different Magnitude Loads for Center Loaded Web Core Beam as shown in Figure 3.5

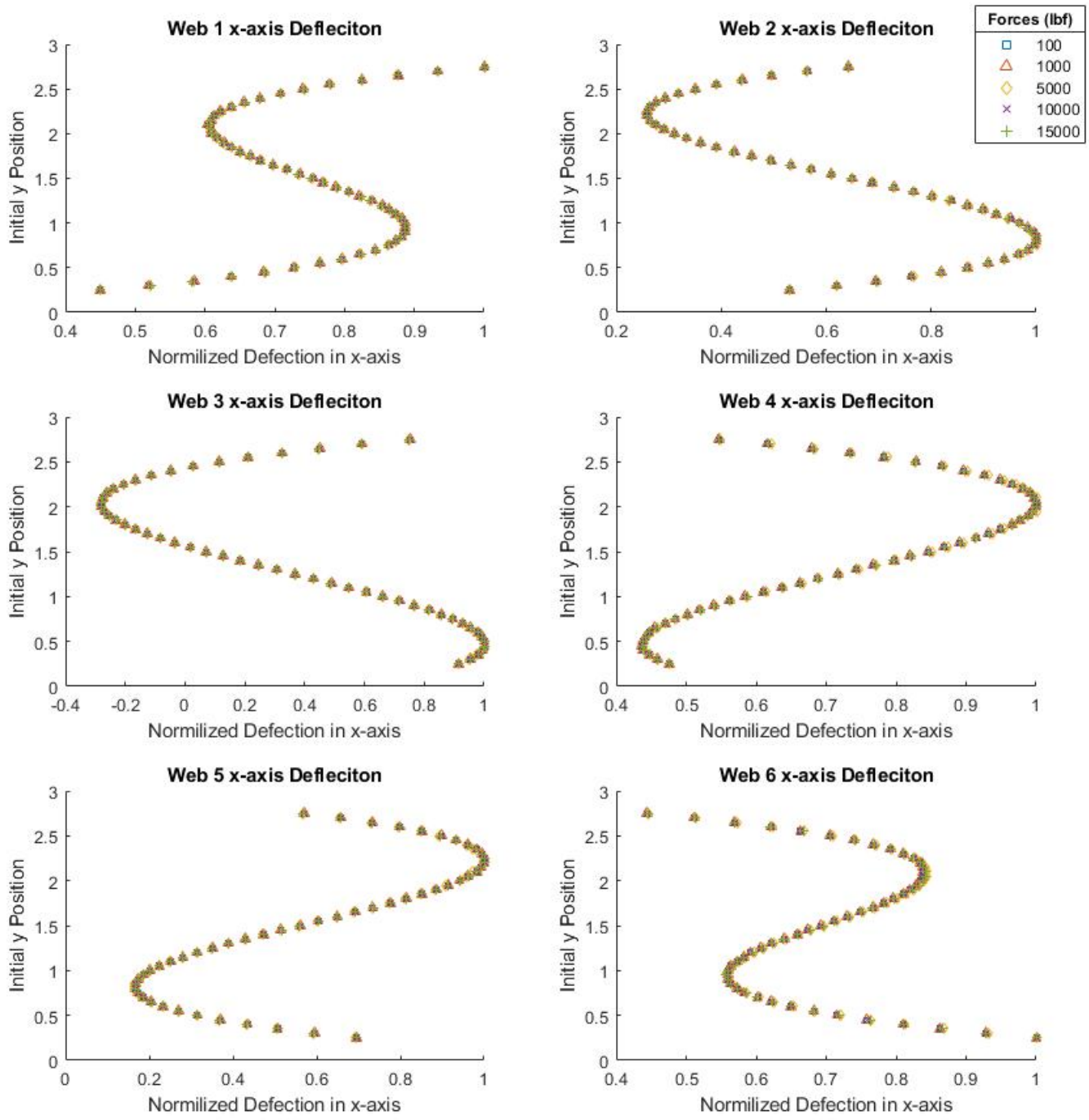


Figure 3.16: Comparing Normalized Deflection Caused by Different Magnitude Loads for Center Loaded Web Core Beam as shown in Figure 3.5

However when the load is moved to different locations along the beam the shape of deformation changes for each web. Figure 3.17 shows where different loads were applied along the beam and Figure 3.18 shows the shape of the deformation changes as the load is applied in different locations. The force was kept constant at $10,000\ lb_f$ for each location.

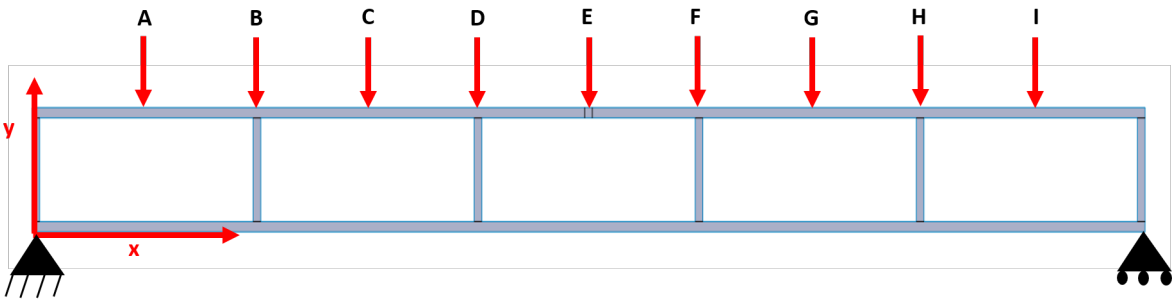


Figure 3.17: Locations of Applied Loads on Simply Supported Web Core Beam

As the location of loading changes the webs magnitude and shape of deformation changes. This is due to the direction of the stress acting on each web changing direction. The direction depends on if the applied force is moved between the web and the nearest fixture or outside of this space.

Similar to the center loaded web core beam, when the location of the load is changed the approximate shape of deformation can be modeled as a cubic function using only three PC's. Figure 3.19 shows the error for approximations using different amount of PC's for a $10,000\ lb_f$ load applied at location C.

Similar plots for each force location can be found in the appendix as well as all code involved in generating the results. Based on the results it can be concluded that a third order polynomial should be used to approximate the webs deformation.

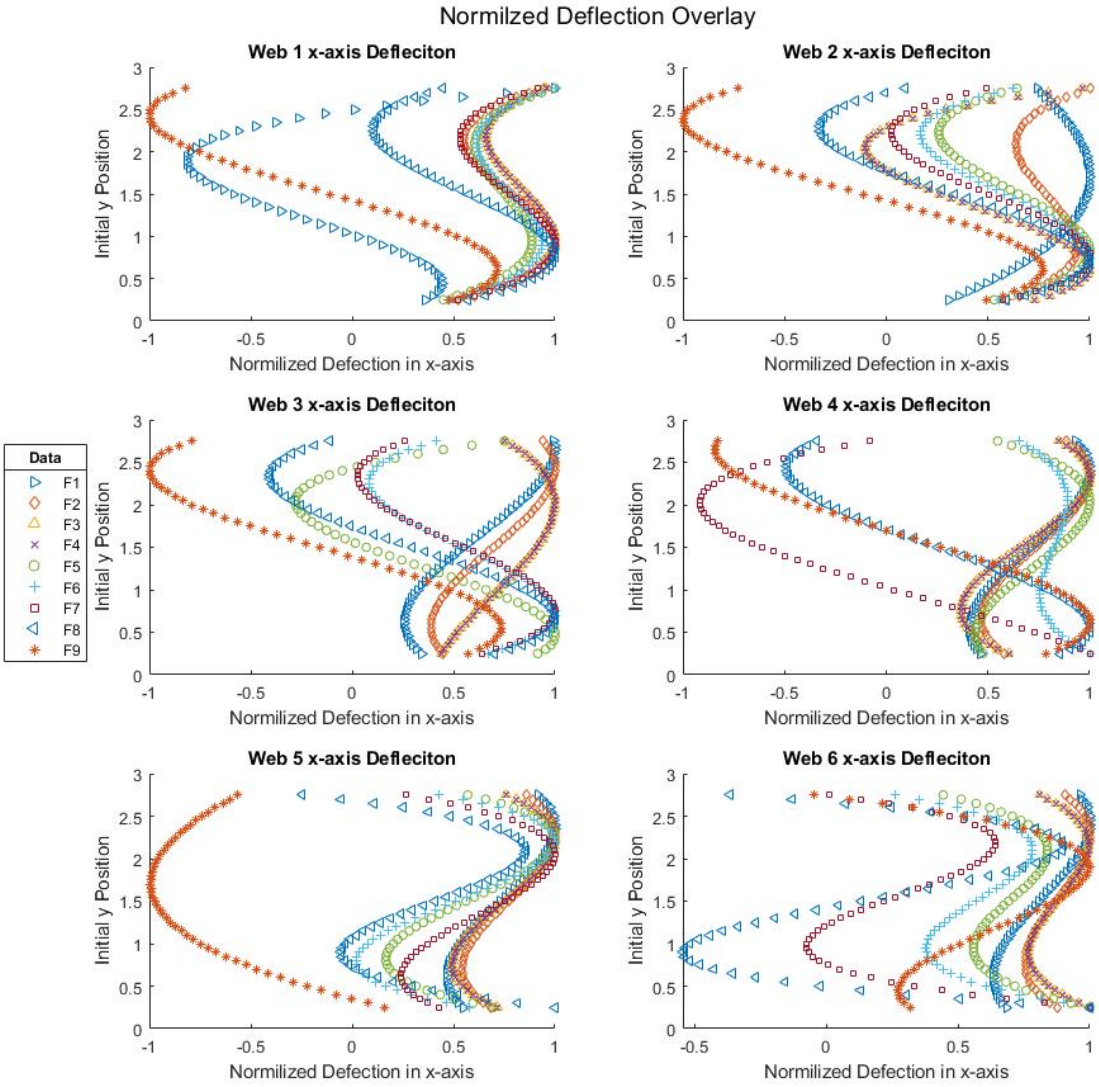


Figure 3.18: Comparing Normalized Deflection Caused by Different Load Locations for Simply Supported Web Core Beam as shown in Figure 3.11

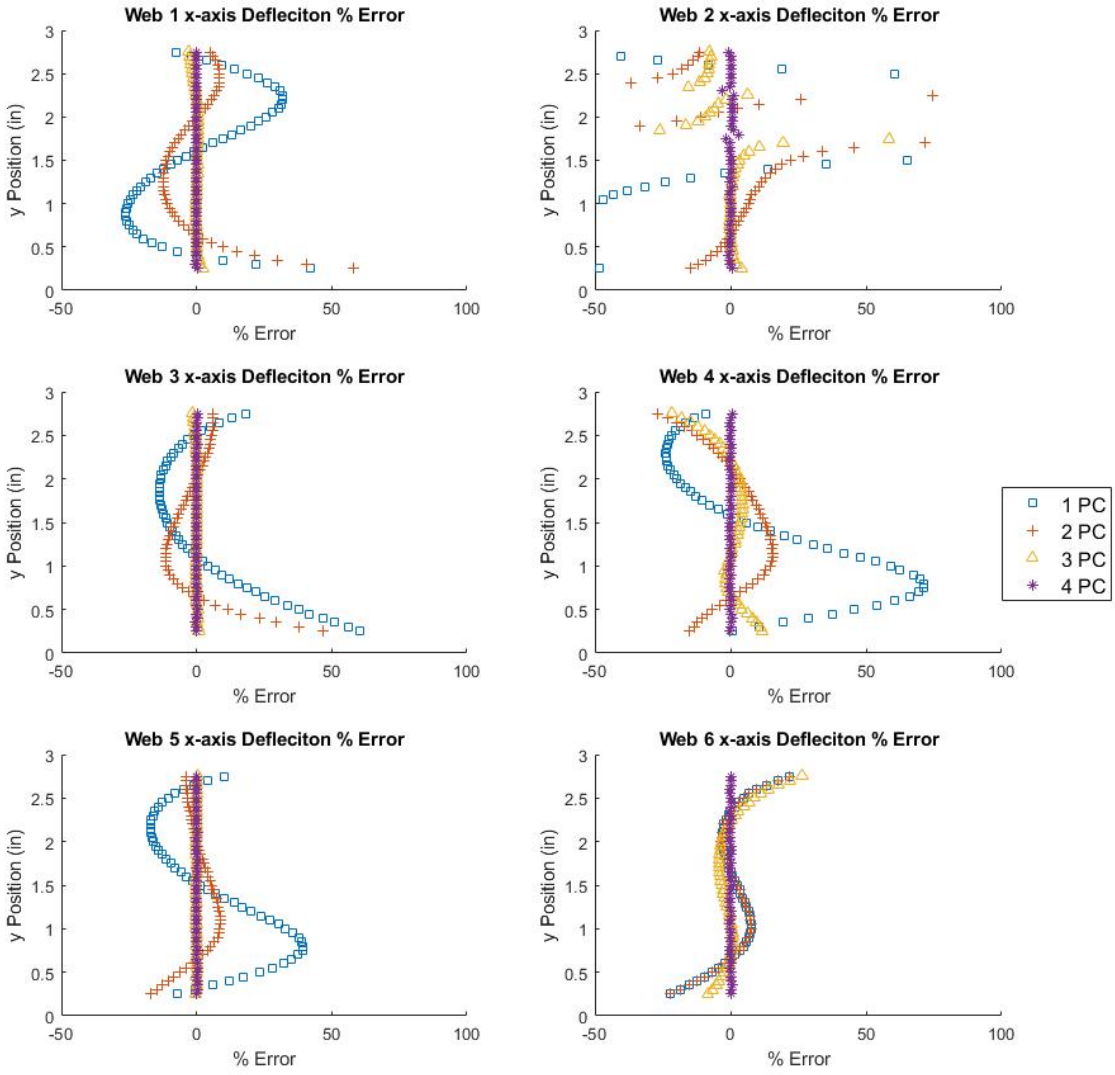


Figure 3.19: Comparing Normalized Deflection Caused by Cubic Fit Error with Nodal Displacements Approximated by Different Numbers of PC's for a Web Core Beam Loaded at Location C as shown in Figure 3.13

4. BEAM MODEL

Based on the results in chapter 3, the beam model is started with a cubic Hermite polynomial, shown in equation 4.1. A Hermite polynomial is selected because it only requires the slope and location at the two end points to define a cubic function, and therefore mid-side nodes are not required. Figure 4.1 shows a geometric representation. Where v is the vertical displacement, m is the slope, and u is the horizontal displacement, all with respect to the end points of a beam. l is the length of the beam along the x direction. This model for a single bar will later be applied to a unit cell to later approximate the energy of a full structure.

$$v = (3t^2 - 2t^3)v_1 + (t^3 - t^2)m_1 l - 2(1-t)^3 v_0 - ((1-t)^3 - (1-t)^2)m_0 l \tag{4.1}$$

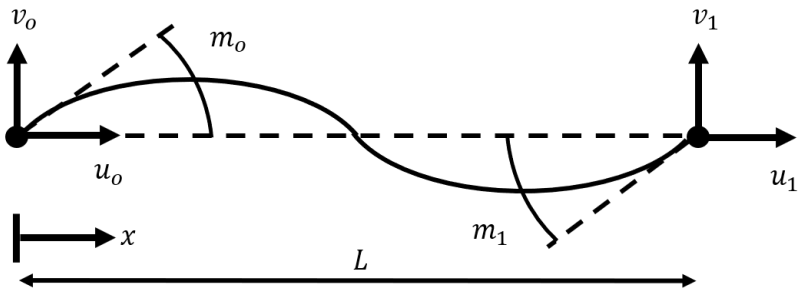


Figure 4.1: Single Beam Deformed

The variable t in equation 4.1 is defined as $t = x/l$. The strain energy of a single bar can be found by integrating the square of the curvature (approximated as the second derivative of the vertical displacement). This results in the equation 5.2 below where E is the elastic modules and

I is the second moment of area.

$$\Pi = \frac{EI}{2}\int_0^l v''^2\,dx = \frac{EIl}{2}[(\frac{m_1 - m_0}{l})^2 + \frac{12}{l^2}(\frac{(m_1 + m_0)}{2} + \frac{(v_0 - v_1)}{l})^2] \tag{4.2}$$

Equation 4.2 accounts for the energy of the beam as it undergoes bending, however the axial deformation must be accounted for because the axial extension and compression of the face places are critical to the response of the beam. Introducing the axial displacements of the two ends of the beam as u_0 and u_1, the resulting strain energy of the beam is given by equation 4.3.

$$\Pi = \frac{EIl}{2}[(\frac{m_1 - m_0}{l})^2 + \frac{12}{l^2}(\frac{(m_1 + m_0)}{2} + \frac{(v_0 - v_1)}{l})^2] + \frac{EAl}{2}(\frac{u_1 - u_0}{l})^2 \tag{4.3}$$

4.1 Energy of Unit Cell

Now that the strain energy has been found for a single beam, all the members of a unit cell can be treated as similar beam elements and combined to then approximate the deformation of the web core beam. A single unit cell consists of two face plates and a single web that is located at the right end of the face plates. Figure 4.2 shows the unit cell and variables at each end of each beam. Equation 4.4 shows the total strain energy Π for a unit cell show, based on the results in equation 4.3. It should be noted a new variable, n, is introduced to account for the slope between the deformed web and horizontal axis.

$$\begin{aligned}\Pi = {} & \frac{EI^Al}{2}[(\frac{m_1^A - m_0^A}{l})^2 + \frac{12}{l^2}(\frac{(m_1^A + m_0^A)}{2} + \frac{(v_0^A - v_1^A)}{l})^2] + \frac{EA^Al}{2}(\frac{u_1^A - u_0^A}{l})^2 \\ & + \frac{EI^Bl}{2}[(\frac{m_1^B - m_0^B}{l})^2 + \frac{12}{l^2}(\frac{(m_1^B + m_0^B)}{2} + \frac{(v_0^B - v_1^B)}{l})^2] + \frac{EA^Bl}{2}(\frac{u_1^B - u_0^B}{l})^2 \\ & + \frac{EI^Ch}{2}[(\frac{n_0^B - n_0^A}{h})^2 + \frac{12}{h^2}(\frac{(n_0^B + n_0^A)}{2} - \frac{(u_0^B - u_0^A)}{h})^2] + \frac{EA^Ch}{2}(\frac{v_1^A - v_1^B}{h})^2\end{aligned} \tag{4.4}$$

In equation (4.4), l is the length of the element and h is the height of the web; A, B, and C indicate variables for the top face plate, bottom face plate, and web respectfully. l is the same for both face plates insuring a square or rectangular unit cell. Note the web does not have a Δ because it is not

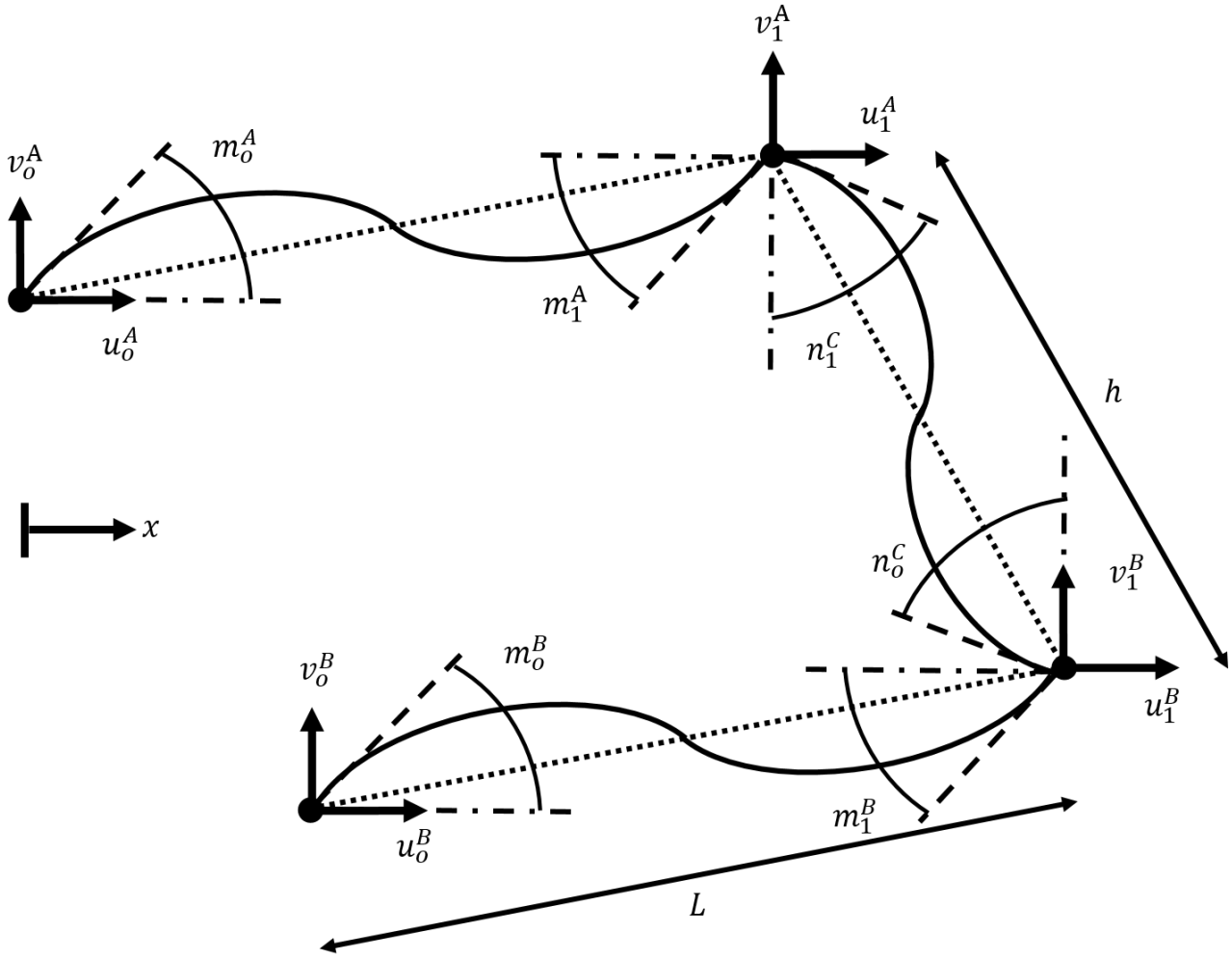


Figure 4.2: Deformed Unit Cell

changing over the x-axis due to the web being vertical.

4.1.1 Rescaling

The central idea of the beam formulation for architected structure is that the individual cells are much smaller than the size of the structure. We do this by replacing the original box structure by an "infinitesimal" box (i.e., the limit as the box size l vanishes) and using the tools of calculus. Thus the energy per unit length of a beam composed of such infinitesimal boxes is given by equation 4.5.

$$\lim_{l\to 0}(\frac{\Pi}{l}) = \frac{EI^A}{2}[(m^{A'})^2 + \frac{12}{l^2}(m^A - v^{A'})^2] + \frac{EA^A}{2}(u^{A'})^2 + \frac{EI^B}{2}[(m^{B'})^2 + \frac{12}{l^2}(m^B - v^{A'})^2] + \frac{EA^B}{2}(u^{B'})^2 + \frac{EI^C h}{2l}[(\frac{m^B - m^A}{h})^2 + \frac{12}{h^2}(\frac{(m^B + m^A)}{2} - \frac{(u^B - u^A)}{h})^2] \quad (4.5)$$

Note that in obtaining the above equations, we have replaced differences of kinematical variables between the two ends of the box divided by the box length by derivatives of the kinematical variables and mean values of a variable by the variable itself. Also note that the box length scale is retained in the constitutive parameters. This is how the model retains an additional length scale in its formulation. The result is a strain energy function for a beam that has a larger number of degrees of freedom in addition to the usual ones.

$$\lim_{l\to 0}(\frac{\Pi}{l}) = \frac{EI^A}{2}[(m^{A'})^2 + \frac{12}{l^2}(m^A - v^{A'})^2] + \frac{EA^A}{2}(u^{A'})^2 + \frac{EI^B}{2}[(m^{B'})^2 + \frac{12}{l^2}(m^B - v^{A'})^2] + \frac{EA^B}{2}(u^{B'})^2 + \frac{EI^C h}{2l}[(\frac{m^B - m^A}{h})^2 + \frac{12}{h^2}(\frac{(m^B + m^A)}{2} - \frac{(u^B - u^A)}{h})^2] \quad (4.6)$$

To further reduce the degrees of freedom in the element the following assumption are made:

- The transverse displacements of the two face plates are the same, that is, in Figure 4.2, $v^A = v^B$. This implies that the there is no axial strain energy for the web (member C) for the beam and the vertical deflection can be treated as the center line deflection of the beam.
- The center line axial deflection of the beam is zero. Therefore, in Figure 4.2, we assume that $(u^A + u^b)/2 = 0$
- The connection between the web and face plate is rigid therefore the rotation of the face plate is the same as the rotational of the web at the connection. Therefore based on Figure

4.2, $n_1^C = m_1^A$ and $n_0^C = m_1^B$.

We now introduce the variable χ to describe the rotation angle between end points of the web with respect to the vertical axis, as shown in Figure 4.3. When the system undergoes bending one side of the beam will be in tension and the other in compression. This combined with the shearing of the top and bottom plates will cause the web to tilt, this angle is defined as χ. Note that the if the top plate purely shears with respect to the bottom plate, the χ is now the shear strain. A triangle is formed between the face plate, vertical axis, and line between the upper and lower points of the web. This triangle has a hypotenuse of $h/2$ because it was assumed the web does not deform axially. Also assuming that the angle χ is small the approximations in equations 4.6, 4.7, and 4.8 can be made:

$$u^B - u^A = \chi \cdot h \quad or \quad \chi = \frac{u^B - u^A}{h} \tag{4.7}$$

This, when combined with the requirement that $(u^A + u^B)/2 = 0$ implies that the axial deformation of the face plates can be redefined in terms of χ.

$$u^B = \chi \frac{h}{2} \tag{4.8}$$

$$u^A = -\chi \frac{h}{2} \tag{4.9}$$

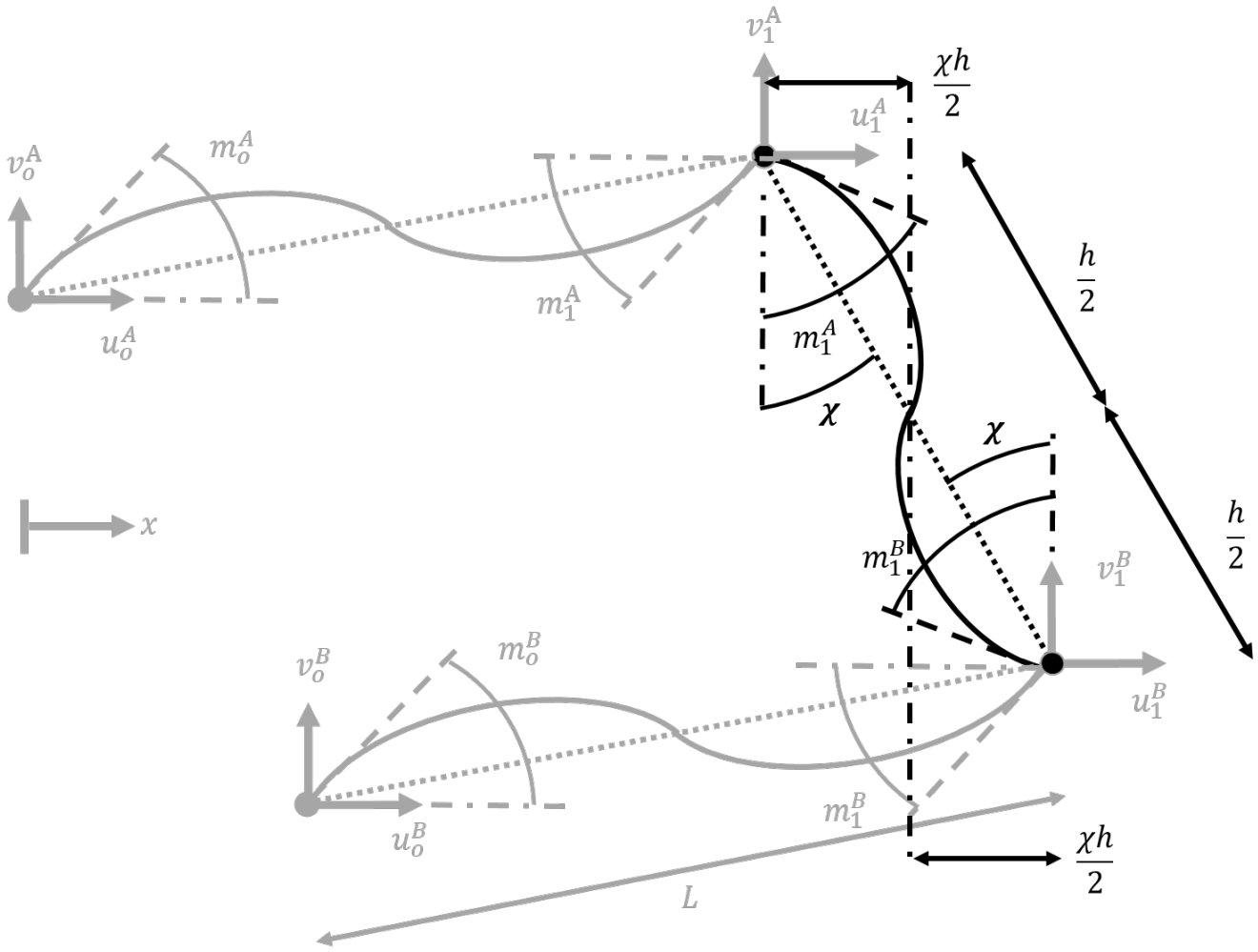


Figure 4.3: Geometric Representation of χ on Unit Cell

This changes equation 4.5 and becomes what is shown in equations 4.9 after some algebraic simplifications.

$$\lim_{l\to 0}(\frac{\Pi}{l}) = \frac{EI^A}{2}[(m^{A'})^2 + \frac{12}{l^2}(m^A - v^{A'})^2] + \frac{EA^A h^2}{8}\chi'^2 + \frac{EI^B}{2}[(m^{B'})^2 + \frac{12}{l^2}(m^B - v^{A'})^2] + \frac{EA^B h^2}{8}\chi'^2 + \frac{EI^C h}{2l}[(\frac{m^B - m^A}{h})^2 + \frac{12}{h^2}(\frac{(m^B + m^A)}{2} - \chi)^2] \quad (4.10)$$

Initially, it is not evident that the strain energy density function listed above is that of a beam,

since no curvature terms appear. Furthermore, we wish to separate the global macroscopic response, from the microscopic or local degrees of freedom. For this purpose,we decompose the rotations m^A and $m^B$and the rotation χ of the web of the beam into that which define the macroscopic rotation and the additional local rotation of the beam. Since the macroscopic rotation is given, in the small deformation assumption, by v' we now define the variables ϕ and ψ. Where ϕ is related to the rotation of a face plate and ψ is related to the rotation of the web.

$$\phi = m - v', \psi = \chi - v' \tag{4.11}$$

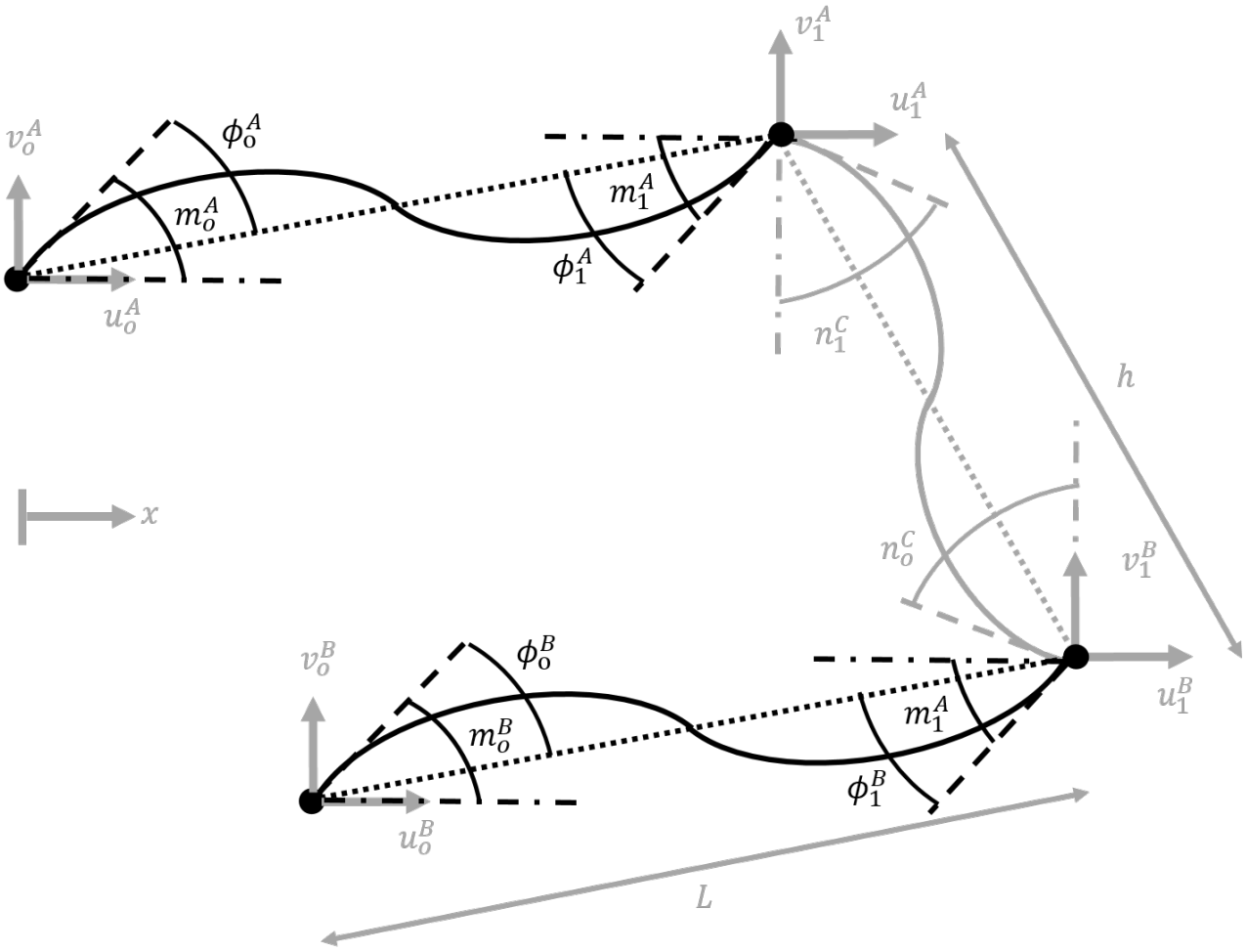


Figure 4.4: Geometric Representation of ϕ on Unit Cell

Now substituting the above expression into the strain energy density function and after some algebraic simplifications the final equation for the strain energy per length of the beam is

$$\begin{aligned}\lim_{l\to 0}(\frac{\Pi}{l}) = \Psi = \frac{EI^A}{2}[(v''+\phi^{A'})^2 + \frac{12}{l^2}(\phi^A)^2] + \frac{EA^Ah^2}{8}(v''+\psi')^2 \\ + \frac{EI^B}{2}[(v''+\phi^{B'})^2 + \frac{12}{l^2}(\phi^B)^2] + \frac{EA^Bh^2}{8}(v''+\psi')^2 \\ + \frac{EI^Ch}{2l}[(\frac{\phi^B-\phi^A}{h}))^2 + \frac{12}{h^2}(\frac{(\phi^B+\phi^A)}{2}-\psi)^2] \quad (4.12)\end{aligned}$$

We now see the familiar second derivative terms in the strain energy function. As a verification of the above model, if we set $\phi^A = \phi^B = \psi = 0$, that is, there are no micro-rotation terms, we obtain equation 4.13 which is the strain energy function for a classical beam.

$$\Psi|_{\text{no micro-rotation}} = \frac{EI^A + EI^B + EA^Ah^2/4 + EA^Bh^2/4}{2}(v'')^2 \quad (4.13)$$

4.2 Full Structure

With the developed unit cell model, the total energy of the entire structure is given by the functional,

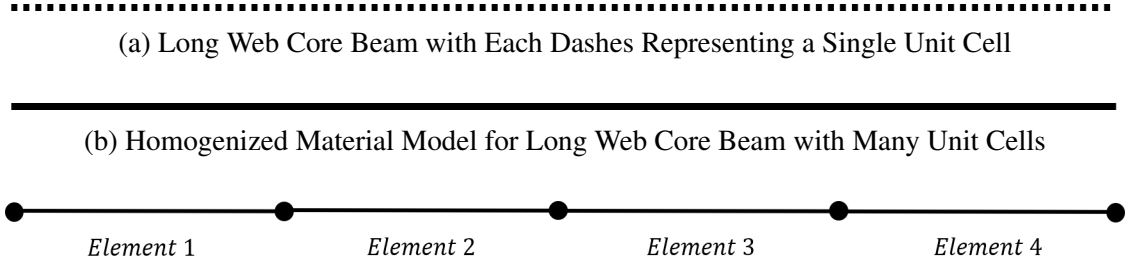
$$U(v,\phi^A,\phi^b,\psi) = \int_0^L \Psi dx \quad (4.14)$$

We can now frame the equilibrium configuration of the web core beam subject to distributed loads as the following minimization problem:

> Find $v(x), \phi^A(x), \phi^B(x), \psi(x)$ such that the functional $U - \int_0^L W(x)v(x)\,dx$ is minimized subjected to essential boundary conditions.

We can solve for each degrees of freedom by using a FE approach and discretizing the beam, regardless of the cell sizes. Figure 4.5 shows a visual representation of an FE model applied to a long beam with many unit cells.

The interpolation functions must be chosen carefully to prevent shear locking. Since we have second derivative terms of v, we will use a cubic Hermite polynomial for the center line deflection



(a) Long Web Core Beam with Each Dashes Representing a Single Unit Cell

(b) Homogenized Material Model for Long Web Core Beam with Many Unit Cells

(c) FE for Homogenized Material Model for Web Core Beam

Figure 4.5: Representation of FE model Applied to Long Web Core Beam with Many Unit Cells

v, by carefully observing the terms in equation 4.27 we note that it is necessary to assume that ψ^A, ψ^B, and ϕ should be linear. The interpolation functions for this element are listed below from equation 4.15 to 4.18. Thus for each element, where $t = x/l$, we will assume the following interpolation functions.

$$\begin{aligned} v &= (3t^2 - 2t^3)\alpha_1 + (t^3 - t^2)\beta_1 l + (3(1-t)^2 - 2(1-t)^3)\alpha_0 - ((1-t)^3 - (1-t)^2)\beta_0 l \\ v' &= \frac{1}{l}[(6t - 6t^2)\alpha_1 + (3t^2 - 2t)\beta_1 l + (6t^2 - 6t)\alpha_0 - (-3t^2 + 4t - 1)\beta_0 l] \\ v'' &= \frac{1}{l^2}[(6 - 12t)\alpha_1 + (6t - 2)\beta_1 l + (12t - 6)\alpha_0 - (-6t + 4)\beta_0 l] \end{aligned} \tag{4.15}$$

$$\phi^A = \phi_1^A t + \phi_0^A, \implies \phi^{A'} = \frac{\phi_1^A}{l} \tag{4.16}$$

$$\phi^B = \phi_1^B t + \phi_0^B, \implies \phi^{B'} = \frac{\phi_1^B}{l} \tag{4.17}$$

$$\psi = \psi_1 t + \psi_0, \implies \psi' = \frac{\psi_1}{l} \tag{4.18}$$

To assist the minimization solver and insure than a minimum point has been reached the gradient will also be provided along with the energy. Knowing that Ψ must be integrated, a general pattern can be utilized. Groups of kinematic variables can be be separated and because of the order of the interpolation functions selected each group is at most linear. This is because of the cubic interpolation for the deflection, leading to a linear variation of the curvature, coupled with a linear interpolations for the rotations of the top and bottom plates and the web.

The relationship shown in equation (4.19) can be obtained for any linear function.

$$\int_{-1}^{1}(at+b)^2\,dt = \int_{-1}^{1}(a^2t^2+b^2+2abt)\,dt = 2(a^2+b^2/3) \tag{4.19}$$

Observing that $\Delta y = 2a$ and $y_{avg} = b$, we obtain the following

$$2(a^2+b^2/3) = (y_{avg}^2 + \Delta y^2/12) \tag{4.20}$$

This result will be repeatedly used in the developments below for finding the energy functional of the beam.

To find the total energy from equation (4.14), we notice from equation (4.12) that the energy density is the sum of squares of linearly varying functions that are grouped together. In order to integrate each group, we will utilize the step shown in equation 4.20 repeatedly, specifically noting that, for each group the average and difference in the variables are needed. We thus list the averages and differences in each group of variable below:

$$v'' + \phi' \implies G^i_{avg} = (dv'' + d\phi^{i'})/l,\;\; dG^4 = 12((v_{avg}{}'' - dv^{i'}/l)/l) \tag{4.21}$$

$$v'' + \psi' \implies G^3_{avg} = (dv'' + d\psi^{i'})/l,\;\; dG^4 = 12((v_{avg}{}'' - dv^{i'}/l)/l) \tag{4.22}$$

$$\phi^B - \phi^A \implies G^5_{avg} = (\phi^B - \phi^A)_{avg},\;\; dG^5 = d\phi^B - d\phi^A \tag{4.23}$$

$$\frac{\phi^B - \phi^A}{2} - \phi \implies G^6_{avg} = \frac{(\phi^B - \phi^A)_{avg}}{2} - \psi_{avg},\;\; dG^6 = \frac{d\phi^B - d\phi^A}{2} - \psi_{avg} \tag{4.24}$$

$$\phi^A \implies G^7_{avg} = \phi^A_{avg},\;\; dG^7 = d\phi^A \tag{4.25}$$

$$\phi^B \implies G^8_{avg} = \phi^B_{avg},\;\; dG^8 = d\phi^B \tag{4.26}$$

Note that in each case, the averages and differences are written entirely in terms of the nodal degrees of freedom. Using these newly defined variables, the total energy functional of the discretized beam can we rewritten in terms of the groups.

$$U(v,\phi^A,\phi^b,\psi) = \int_0^L \Psi dx = \frac{EI^A}{2}[(G^1_{avg}+\frac{dG^4}{12})^2+\frac{12}{l^2}(G^7_{avg}+\frac{dG^7}{12})^2]+\frac{EA^Ah^2}{8}(G^3_{avg}+\frac{dG^4}{12})^2 + \frac{EI^B}{2}[(G^2_{avg}+\frac{dG^4}{12})^2+\frac{12}{l^2}(G^8_{avg}+\frac{dG^8}{12})^2]+\frac{EA^Bh^2}{8}(G^3_{avg}+\frac{dG^4}{12})^2 + \frac{EI^Ch}{2l}[(\frac{G^5_{avg}+\frac{dG^5}{12}}{h}))^2+\frac{12}{h^2}(G^6_{avg}+\frac{(dG^6)}{12})^2] \quad (4.27)$$

The gradient can also be defined using these groups, specifically by the product of a vector holding the material properties and kinematic variables and a matrix of constants. Because the energy function is quadratic the gradient will be linear. This means that rather than the average and difference for each group being sum and squared they are only summed together in the gradient. In particular, the vector $\vec{k}$ of stiffnesses can be written as

$$\vec{k} = \left[EI^A/2, \quad EI^B/2, \quad E(A^A+A^B)h^2/4, \quad EI^C/2lh, \quad 12EI^A/2l, \quad 12EI^B/2l\right] \quad (4.28)$$

We next define the vector $\vec{G}$ by

$$\vec{G} = [G^2_{avg}, G^1_{avg}, G^3_{avg}, dG^4, G^5_{avg}, G^6_{avg}, dG^5, dG^6, G^7_{avg}, G^8_{avg}, dG^7, dG^8] \quad (4.29)$$

Then, the derivative of the energy in each element with respect to the vector $\vec{G}$ is given as

$$\vec{g} = \frac{\partial \Psi_{ele}}{\partial \vec{G}} = \begin{bmatrix} G^1_{avg} k^1 \\ G^2_{avg} k^2 \\ G^3_{avg} k^3 \\ \frac{1}{12}(dG^4(k^1 + k^2 + k^3)) \\ G^5_{avg} k^4 \\ G^6_{avg} 12 k^4 \\ \frac{dG^5}{12} k^4 \\ \frac{dG^6}{12} k^4 \\ G^7_{avg} k^5 \\ G^8_{avg} k^6 \\ \frac{dG^7}{12} k^5 \\ \frac{dG^8}{12} k^6 \\ -Force \end{bmatrix} \tag{4.30}$$

We next define the degrees of freedom in each element $\vec{q}$ as

$$\vec{q} = [\alpha_0, \beta_0, \phi_0^A, \phi_0^B, \psi_0, \alpha_1, \beta_1, \phi_1^A, \phi_1^B, \psi_1] \tag{4.31}$$

where the subscript 0 is for the left end of the element and 1 is for the right end of the element. With this notation, the derivative of $\vec{G}$ with respect of $\vec{q}$ is given by

$$M := \frac{\partial [G]}{\partial [q]} = \begin{array}{c} \\ G^1_{avg} \\ G^2_{avg} \\ G^3_{avg} \\ dG^1 \\ G^5_{avg} \\ G^6_{avg} \\ dG^5 \\ dG^6 \\ G^7_{avg} \\ G^8_{avg} \\ dG^7 \\ dG^8 \\ Force \end{array} \begin{array}{c} \begin{array}{cccccccccc} \alpha_0 & \beta_0 & \phi_0^A & \phi_0^B & \psi_0 & \alpha_1 & \beta_1 & \phi_1^A & \phi_1^B & \psi_1 \end{array} \\ \begin{bmatrix} 0 & -z & -z & 0 & 0 & 0 & z & z & 0 & 0 \\ 0 & -z & 0 & -z & 0 & 0 & z & 0 & z & 0 \\ 0 & -z & 0 & 0 & -z & 0 & z & 0 & 0 & z \\ 12z^2 & 6z & 0 & 0 & 0 & -12z^2 & 6z & 0 & 0 & 0 \\ 0 & 0 & -.5 & .5 & 0 & 0 & 0 & -.5 & .5 & 0 \\ 0 & 0 & .25 & .25 & -.5 & 0 & 0 & .25 & .25 & -.5 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -.5 & -.5 & 1 & 0 & 0 & .5 & .5 & -1 \\ 0 & 0 & .5 & 0 & 0 & 0 & 0 & .5 & 0 & 0 \\ 0 & 0 & 0 & .5 & 0 & 0 & 0 & 0 & .5 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ .5 & 0 & 0 & 0 & 0 & .5 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \tag{4.32}$$

where $z = 1/l$

Thus, over each element, we can write

$$\nabla(U - \int_0^L W(x)v(x)\,dx) = \vec{g}^T \cdot M \tag{4.33}$$

With the energy and gradient expressions above, we can model the response of a web core beam under different loading and boundary conditions. The validation and results for a single case can be found in Chapter 5 and code found in Appendix C. Possible boundary conditions for each kinematic variable are listed in Table 4.1. In the cases were there is no value the variable is known and free to move under that fixture.

A list of the constraints on the system for different kinds of support are provided in the table below.

Table 4.1: Constraints on Kinematic Variables for Different Simple Supports

Fixture Type	v	m	ϕ^A	ϕ^B	ψ
Pin on Bottom Face Plate	0				
Pin on Both Face Plates	0				0
Vertical Roller on Bottom Face Plate				0	0
Vertical Roller on Both Face Plates		0	0	0	0
Horizontal Roller on Bottom Face Plate	0	0		0	
Horizontal Roller on Both Face Plates	0	0	0	0	
Fixed	0	0	0	0	0

5. RESULTS, VALIDATION, AND DISCUSSION

To determine the deformation of the structure the total energy must be minimized. When the energy is minimized the structure is static and the values for each degree of freedom can be defined, as stated in Castigliano's first theorem. To minimize the total energy MATLABs optimization toolbox was used, specifically the function $fmincon$, which can find the minimum of a supplied constrained nonlinear multi-variable function. Depending upon the settings used, the function can use many algorithms including Interior Point Algorithm [57, 58], sequential quadratic programming (SQP) [59, 60, 61, 62], and trust-region-reflective that involves preconditioned conjugate gradients (PCG) [63, 64]. We choose the conjugate gradient based solver since it is easily implemented with the energy and gradient values that we have obtained.

In general, because $fmincon$ is a gradient based local optimizer it does not insure that the global minimum within the constrained design space is found. However because the energy function used is quadratic in the nodal variables, and the constraints are linear, there is at most one minimum point. As long as a reasonable initial point supplied then the solver will converge to the minimum point quickly. The halting criteria and other options were changed in order to tune the solver, these can be found in Appendix C.

5.1 Verification

To validate the the beam model and code are set up correctly a verification step can be done by eliminating the all rotations as show in equation (4.13), and checking if the model behaves like a classical beam. This can be done by applying equality constraints to set each rotation to zero or by increasing the stiffness terms to lead groups with only rotations. By increasing the stiffness terms by orders of magnitude then the rotations are reduced greatly and do no play a factor in the final solution when the energy is minimized.

The structure used for this verification step is a web core beam with nine unit cells and is fixed on each end. A vertical load is applied to the middle two webs. Figure 5.1 shows the free body

diagram of a steel structure ($E = 29007547\ psi$). Figure 5.2 shows the dimensions of unit cell of a unit cell. Based on *Roark's Formulas for Stress and Strain* [65] the maximum deflection for a classical beam fixed on each end and loaded in the center can be described by equation 5.1.

$$\delta_{max} = \frac{FL^3}{192EI} \tag{5.1}$$

Because the load is not applied to the web core beam is not directly in the center the relationship will only be approximate but should have a similar relationship to the equation. By normalizing the stiffness terms and length of the beam the relationship between the maximum deflection δ_{max} and the force F is a factor of 192. In order to normalize the stiffness terms all terms not associated with rotations that will be forced to zero and summed and individually divided by the sum. The length of the beam is normalized in the same way so that the length of the beam now spans from zero to one.

$$k_{norm} = k \cdot (\frac{EI^A + EI^B + EA^A h^2/4 + EA^B h^2/4}{2})^{-1} \tag{5.2}$$

$$L_{norm} = L/L \tag{5.3}$$

Figure 5.8 shows the deflection results from the model with normalized stiffness and length. A load of 0.5 was applied at each location shown in Figure 5.2. In this case the structure was broken up into nine elements however more or less could be used depending on the situation and information required.

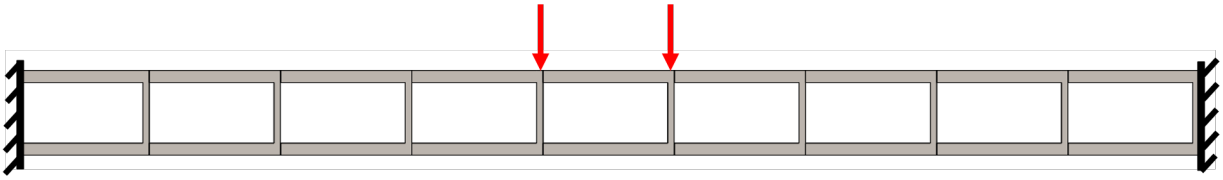


Figure 5.1: FBD of 9 Cell Web Core Beam Structure

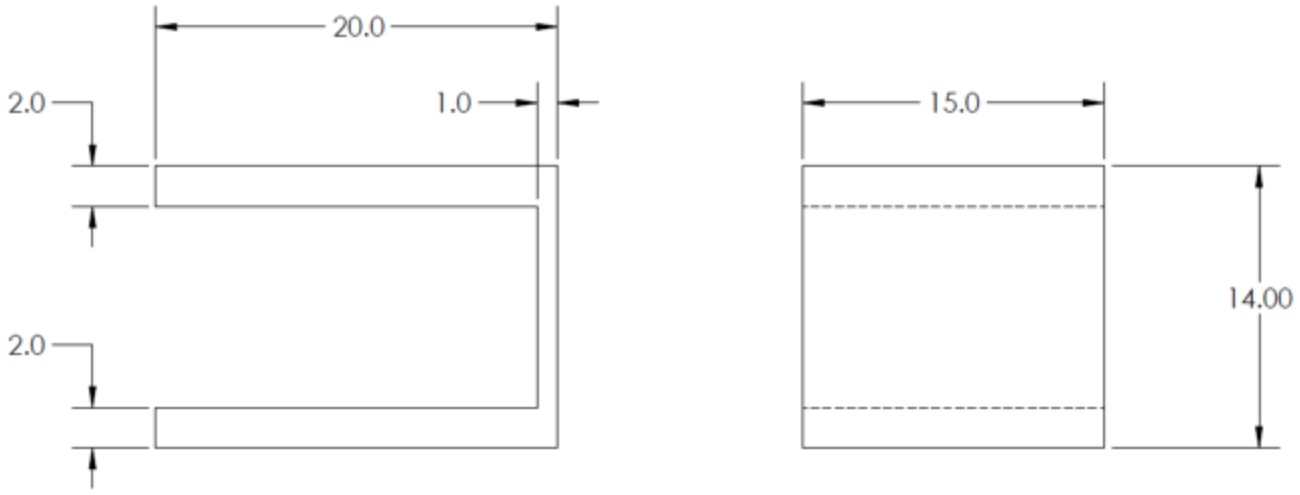


Figure 5.2: Dimensions of Unit Cell (Units: inches)

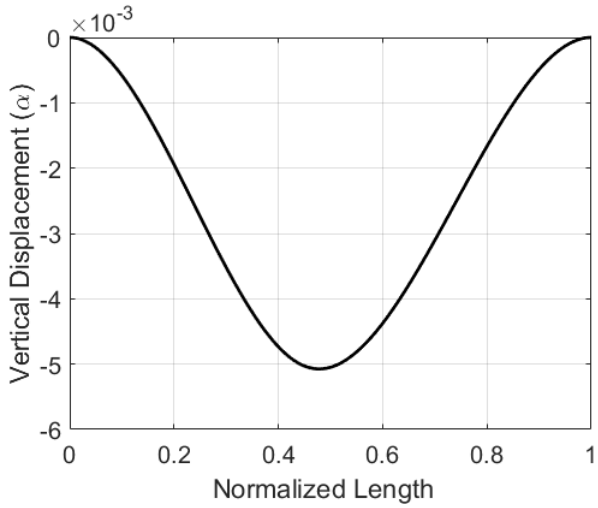


Figure 5.3: Verification of Normalized Web Core Beam Model to find Deflection Response by Solving as Classical Beam

The results show that the maximum deflection when a total force value of 1 is applied the maximum deflection is ≈ 0.005. By using equation 5.1 the relationship between the force and deflection can be found, as shown in equation 5.2.

$$\frac{F}{\delta_{max}} = \frac{1}{.00507} \approx 197 \approx 192 \tag{5.4}$$

The result indicates the model and code are set up correctly and now allows for results for a web core beam to be found and compared to results form commercial FEA. While the results do not directly match with those of a classical beam this was expected because the forces were applied at two separate locations near the center rather than directly at he center of the structure.

5.2 Results

To determine the accuracy of the developed beam model the center line deflection results were found and compared to Ansys FE solutions. The Ansys simulation used a 3D structure as described in Figures 5.1 and 5.2, as well as Figures in Appendix D. 3D cube elements were used with a fine mesh, spacing of 0.25 inches. More information about the structure, mesh, and results can be found in Appendix D. A vertical load of $2,500\ lb_f$ was applied at each location pointed in the negative y-axis. The deflection, von Mises stress, and maximum shear stress results are shown in Figure 5.5 and 5.6. Figure 5.4 shows the mesh applied to a section of the structure in Ansys. A total of 777600 element 3526011 nodes were used. All elements were square 3D elements with a size of 0.25 inches and were uniform thought out. Figure 5.7 shows the value of the maximum shear stress at each web. The results comparing the values between the solver and FE results shown in Figure 5.8.

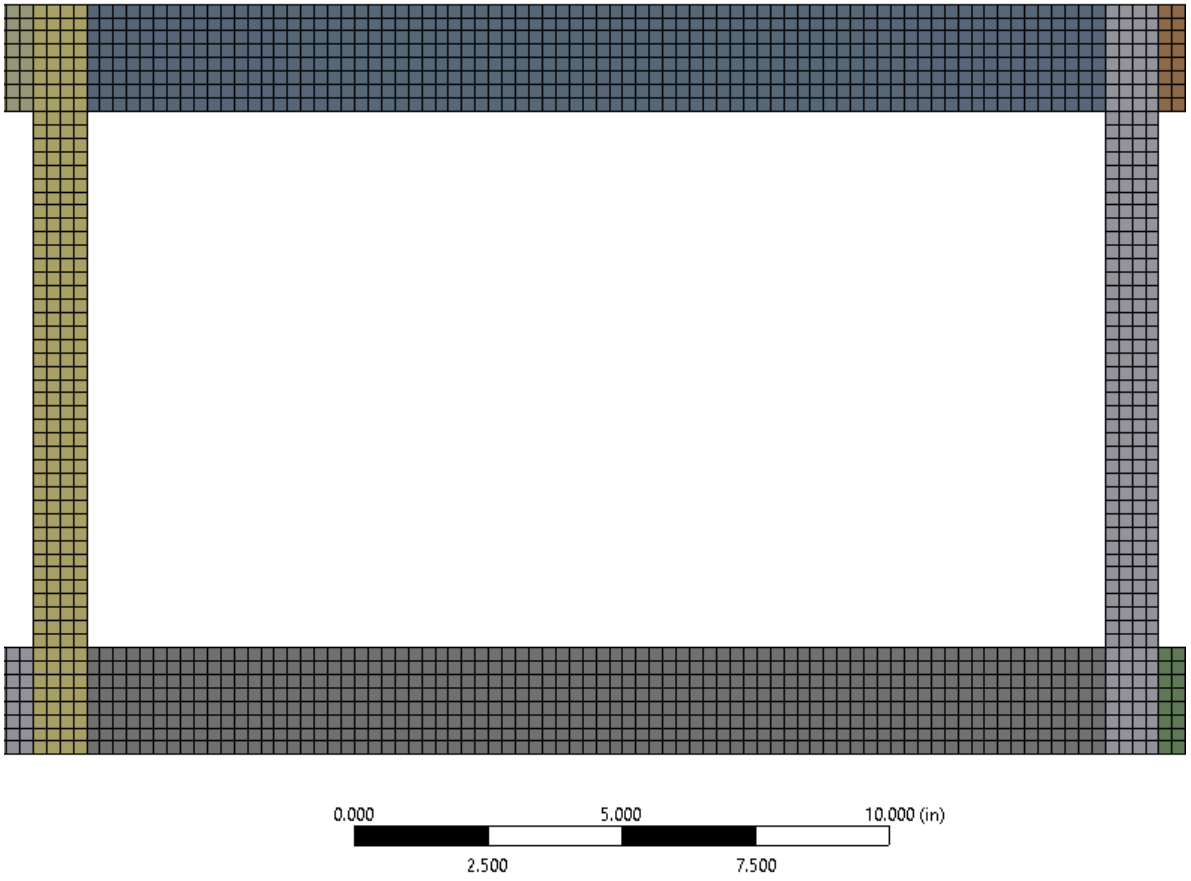
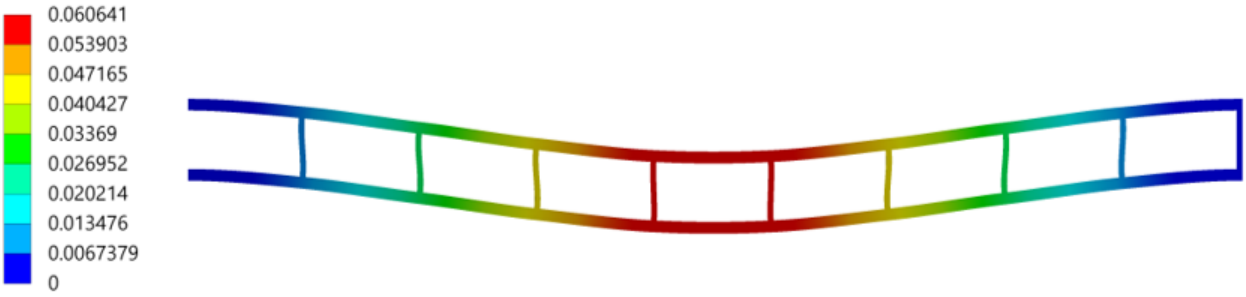
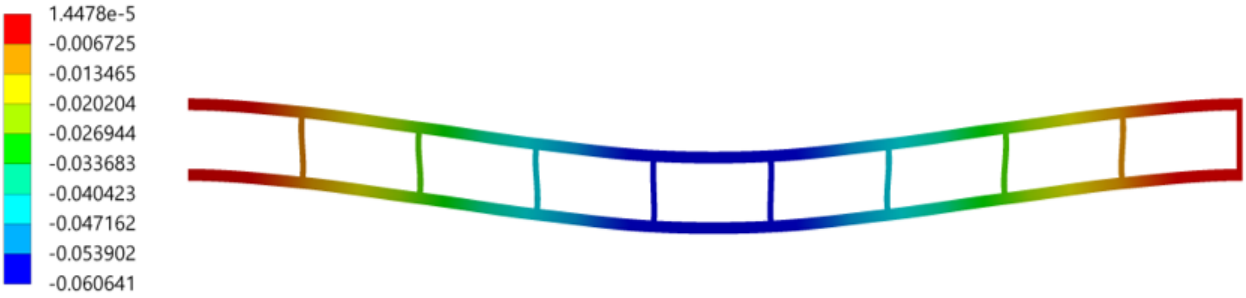


Figure 5.4: Front View of Mesh Size for Ansys Simulation to Compare Results of Beam Model

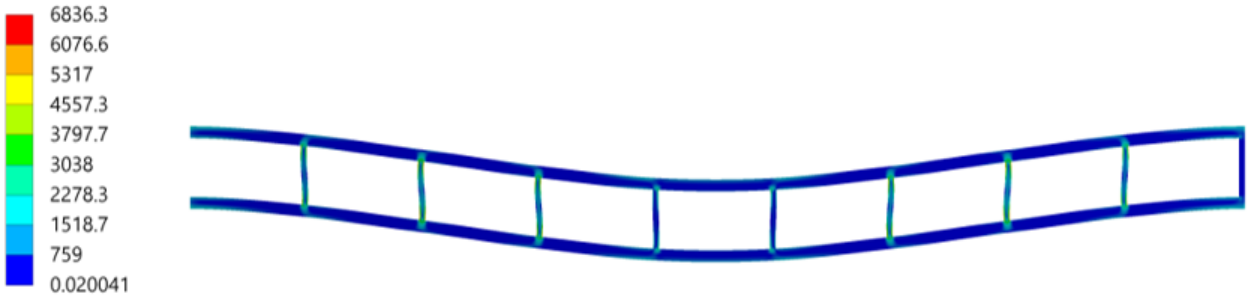


(a) Vertical Deflection (Unit: in)

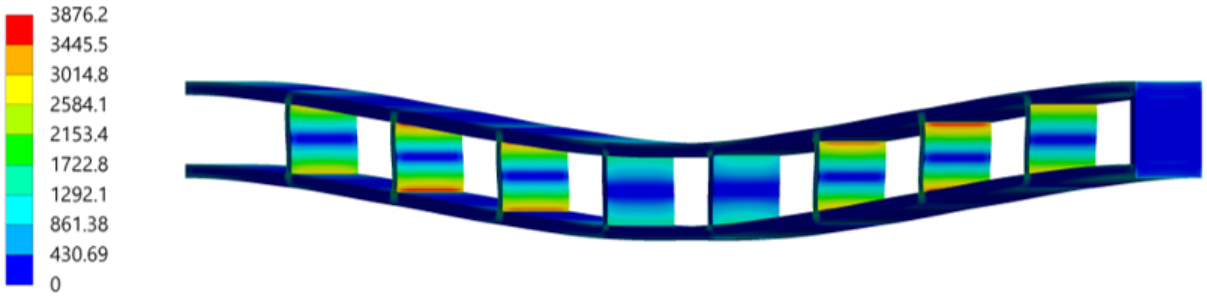


(b) Von Mises Stress (Unit: psi)

Figure 5.5: Ansys Simulation Results for Deflection of Web Core Beam Fixed on Each End and Loaded in the Center, As Shown in Figure 5.1



(a) Shear Stress distribution in the beam (Unit: psi)



(b) Skewed View of Shear Stress Results Highlighting the Distribution in the Web (Unit: psi)

Figure 5.6: Ansys Simulation Results for Stress in Web Core Beam Fixed on Each End and Loaded in the Center, As Shown in Figure 5.1

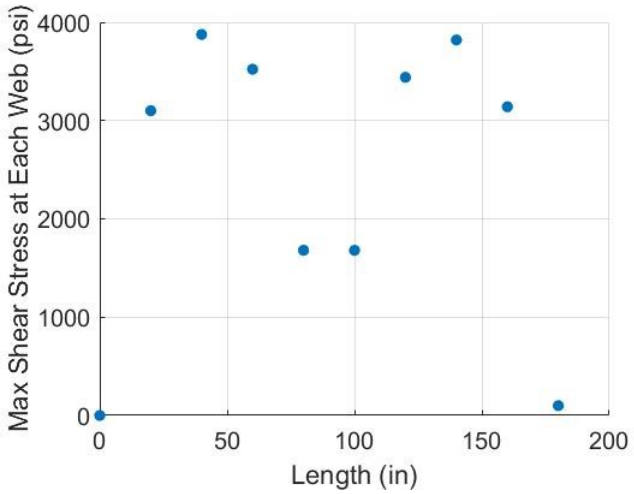
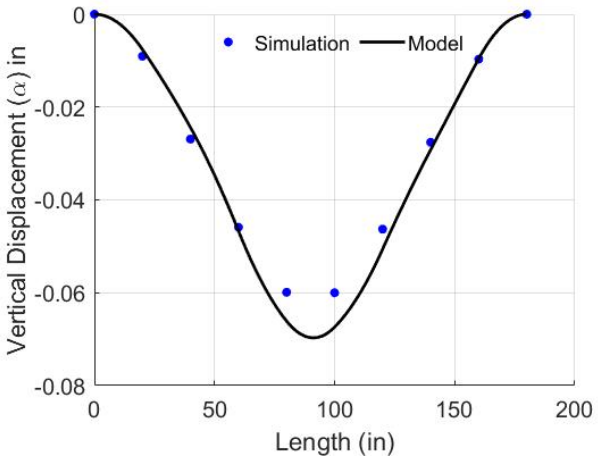
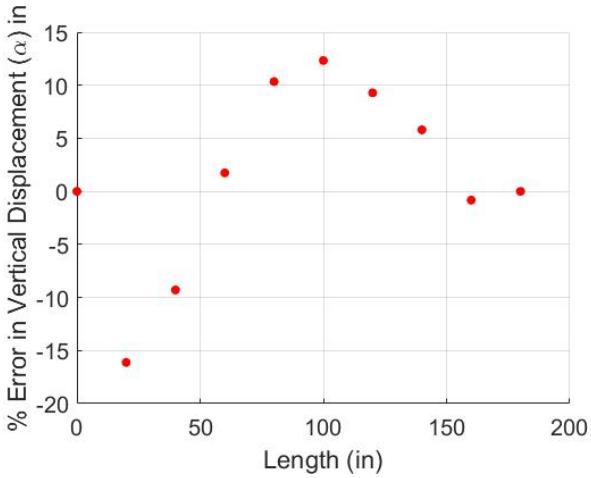


Figure 5.7: Maximum Value for Shear Stress from Ansys Simulation at Each Web Along the Web Core Beam



(a) Web Core Beam Center Line Deflection Compared to Ansys Simulation Nodal Deflection



(b) Difference Between Center Line Deflection Compared to Simulation Nodal Deflection

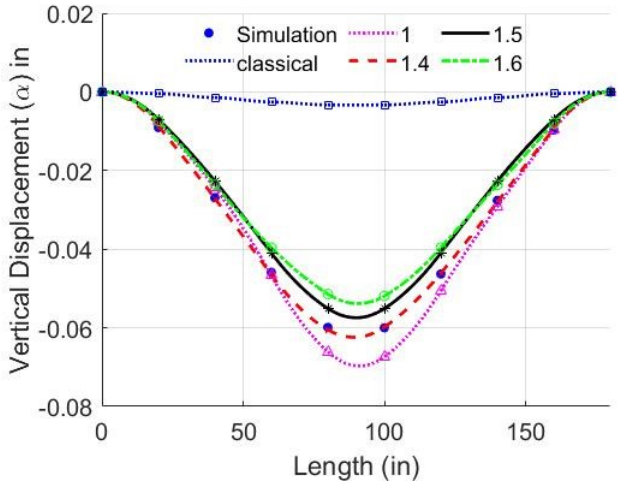
Figure 5.8: Comparing Web Core Beam Model to Ansys FE Simulations Results

5.3 Discussion

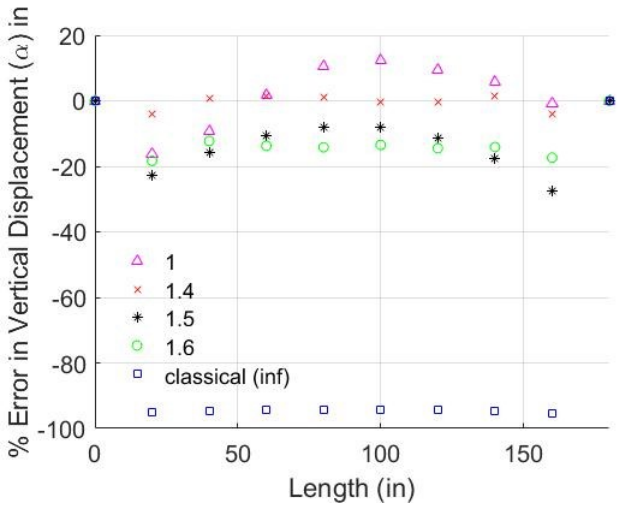
As shown in Figure 5.8, the maximum center line deflection is slightly over predicted near the center of the beam. We hypothesize that the errors in the beam deflection are due to the neglect of the fact that the web core portion of the beam is very short and is stiffer than the assumed bending mode. Error might be caused by assuming that the connection between web and face plate is rigid and treating the web as a beam rather than a short plate. Because the length to width ratio of the web is small the web acts differently than the longer thinner face plates. This has also been discussed by Karttunen and Reddy [66]. They address the issue by applying a correction factor to the term associated with the webs stiffness, see page 178 [66]. By the introduction of a correction factor they are able to more closely predicted the deformation using the micropolar theory discussed in Chapters 2 and 3. Application of correction factors for beams has also been discussed in ref. [67] and in sandwich beams in [68, 69].

Therefore to address this issue we also applied a shear correction factor applied to the stiffness of the web ($EI^C/2lh$). A parametric study was conducted to elucidate the effect of the shear

correction factor on the beam deflection. Figure 5.9 shows different correction factors effect the predicted deflection.



(a) Web Core Beam Modeled Center Line Deflection Compared to Ansys Simulation Nodal Deflection for Different Shear Correction Factors



(b) Difference Between Web Core Beam Modeled Center Line Deflection Compared to Ansys Simulation Nodal Deflection for Different Shear Correction Factors

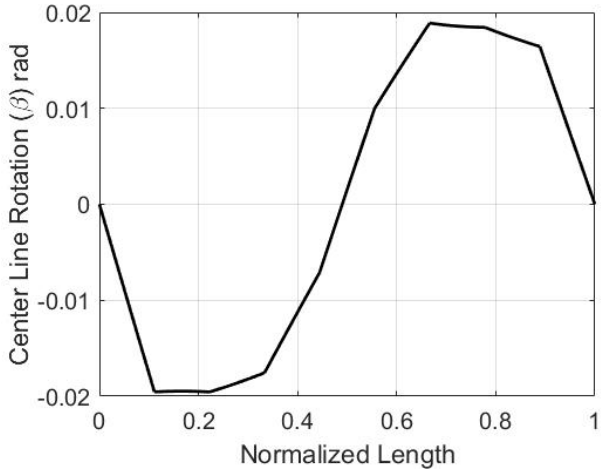
Figure 5.9: Comparing Web Core Beam Model with Different Shear Correction Factors to Ansys FE Simulations Results. Note that the a simple Bernoulli-Euler Beam Model (classical) severely under-predicts the deflections.

When a shear correction factor of 1.4 is used, the web stiffness is increased and the results match that from the 3D FEA quite closely as shown in Figure 5.9. It is important to note that a Bernoulli-Euler beam model servery under predicts the deformation giving a maximum deflection of around .003 inches or around an error of 95% as show in Figure 5.9. Therefore this factor is appropriate for the structure used and shows the beam model is able to model the center line deflection accurately.

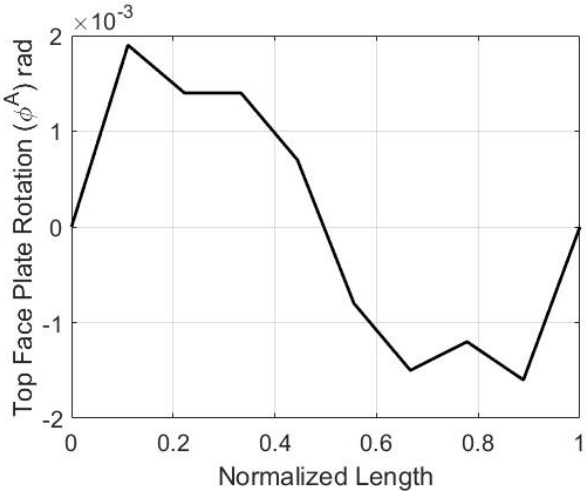
Figure 5.9 shows the results for the kinematic variables related to rotations. As the center line rotation of the structure increases the rotations related to the face plate ϕ and psi also increase. Because the structure is fixed on both ends and loaded in the center the natural and essential

boundary conditions are the rotation at the center and at the fixed ends is zero. This is shown in the results for each rotation. It can be subjectively observed that the results for the rotation of the cross section (β) matches with the results shown in chapter 3, specifically Figure 3.8, as well as Figure 5.5. Rotation of the webs psi on the left side rotate in a clockwise direction while while webs on the right side rotate in a counter clockwise direction. An increase in cross sectional rotation can also be verified by the results of the shear stress shown in Figure 5.6 and 5.7. The direction of rotation is due to the fact that the webs rotation is coupled with center line rotation, see Figures 4.3 and 4.4. Another point of verification is seen by noting the shear stress is highest in the same area the rotation of the cross section is also highest. Rotations related to the top and bottom face plates ϕ are not the same however are very similar. These results also subjectively match the results show in Figure 5.5 and the center line rotation. Again the maximum rotation coincides with the locations of maximum shear stress. Lastly because the scenario analyzed is symmetric the rotations are symmetric for each variable.

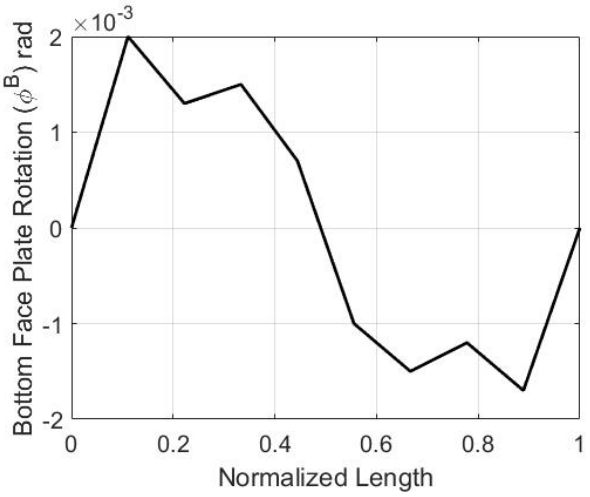
Work related to evaluating the accuracy of this model related to predicted rotations was only subjectively evaluated in this paper, however can be done in more detail in future work.



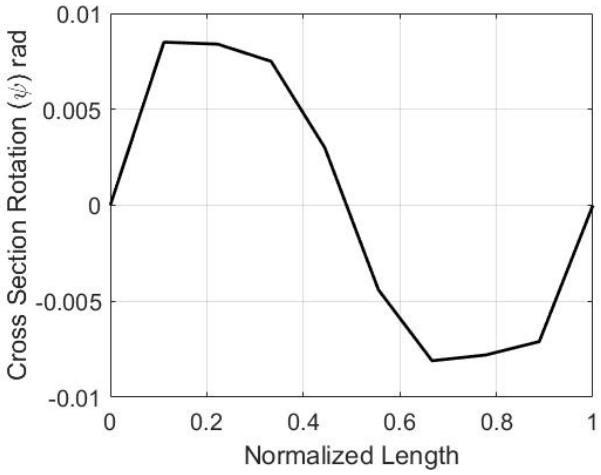
(a) Center Line Rotation β



(b) Top Face Plate Rotation ϕ^A



(c) Bottom Face Plate Rotation ϕ^B



(d) Cross Section Rotation ψ

Figure 5.10: Web Core Beam Model Rotation Results for Fixed-Fixed Web Core Beam Loaded in the Center

6. SUMMARY AND CONCLUSION

In this thesis, the mode of deformation was analytically found and used in order to develop a reduced order finite element model for a web core beam structure. In order to do this the localized deflection information of a web core beams webs was gathered and sampled. Then using the information PCA was used to find the a common order mode of deformation between all the webs in the structure. Unknowing what the minimum order mode of deformation is required to capture the micro deformation a reduced FE model was developed using an energy based approach. Using the energy function for the structure developed minimization techniques the results were compared with commercial FE simulations.

For a web core beam it was found that the webs deform as a cubic function and 3 PCs were adequate to approximate the deformation of a structure with five units cells. Knowing that a cubic function was needed, the strain energy for a single bar was developed using a cubic Hermite polynomial. This was then applied to a unit cell before being homogenized and simplified by different assumptions about the relationship between kinematic variables. The modified unit cell energy was then used to find the strain energy destiny of a full structure, before boundary conditions were applied and the total energy minimized in order to determine values for the kinematic variables.

The center line deflection for a fixed-fixed beam were analyzed and a shear correction factor was applied to the increase the stiffness of the web and therefore obtain more accurate results. This model could also be used to find local degrees of freedom and results related to macro and micro rotations. It can also be used to find the stress in the structure, however this has not been evaluated nor its accuracy determined.

The work done provides a systematic method to determine the shape function of uni-directional sandwich beam structures with any cross section. This method can also be applied to other architected structures, especially those were intuition is used rather than a data driven approach. It can also assist in future work related to determining failure and instabilities within the micro-structure of architected structures. Furthermore it shows that the implementation of an energy based method

can be used rather than the typical system of equations. This allows for a more simple implementation when changing the structures geometry, external loads, or boundary conditions. However it sacrifices information about internal forces in the structure as shown by the homogenized micropolar beam theory approach [9, 30, 35]. With the homogenized micropolar beam theory approach a free body diagram between each element can be formed and the information found using the formed system of equations.

REFERENCES

[1] Granta-Design, "Material property charts," *https://www.grantadesign.com/education/students/charts/*, 2021.

[2] T. A. Schaedler and W. B. Carter, "Architected cellular materials," *Annu. Rev. Mater. Res. 2016.*, pp. 46–187,210., 2016.

[3] MakerBot, "Makerware 2.2.0 | more features."

[4] C. Schumacher, S. Marschner, M. Gross, and B. Thomaszewski, "Mechanical characterization of structured sheet materials," *ACM Trans. Graph.*, vol. 37, July 2018.

[5] J. Romanoff and P. Varsta, "Bending response of web-core sandwich beams," *Composite Structures*, vol. 73, no. 4, pp. 478–487, 2006.

[6] S. J. Bright SR, "A new design for steel bridge decks using laser fabrication," *Proceedings of the 2nd International Conference on Sandwich Construction. Gainesville, Florida*, 2004.

[7] B. Hassani and E. Hinton, "A review of homogenization and topology optimization i—homogenization theory for media with periodic structure," *Computers & Structures*, vol. 69, no. 6, pp. 707–717, 1998.

[8] J. Romanoff and J. Reddy, "Experimental validation of the modified couple stress timoshenko beam theory for web-core sandwich panels," *Composite Structures*, vol. 111, pp. 130–137, 2014.

[9] A. T. Karttunen, J. Reddy, and J. Romanoff, "Micropolar modeling approach for periodic sandwich beams," *Composite Structures*, vol. 185, pp. 656–664, 2018.

[10] J. R. Greer and V. S. Deshpande, "Three-dimensional architected materials and structures: Design, fabrication, and mechanical behavior.," *MRS Bulletin*, vol. 44, pp. 750–757, 2019.

[11] D. Pasini and J. K. Guest, "Imperfect architected materials: Mechanics and topology optimization," *MRS BULLETIN*, vol. 44, pp. 766 – 772, 2019.

[12] S. J. Bright SR, "Fatigue performance of laser-welded steel bridge decks," *Proceedings of the 2nd International Conference on Sandwich Construction. Gainesville, Florida*, 2004.

[13] C. Briscoe, S. Mantell, J. Davidson, and T. Okazaki, "Design procedure for web core sandwich panels for residential roofs," *Journal of Sandwich Structures & Materials - J SANDW STRUCT MATER*, vol. 12, 01 2010.

[14] W. I. Smith EM, Cowling MJ, "Adhesively bonded sandwich structures in marine technology," *Proceedings of the 2nd international conference on sandwich construction. Gainesville, Florida*, pp. 347—-62, 1992.

[15] A. S. Nilsson P, Al-Emrani M, "Transverse shear stiffness of corrugated core steel sandwich panels with dual weld lines," *Proceedings of the 2nd International Conference on Sandwich Construction. Gainesville, Florida*, pp. 117:98—-112, 2017.

[16] K. A. Kujala P, "A steel sandwich panels in marine applications," *Proceedings of the 2nd International Conference on Sandwich Construction. Gainesville, Florida*, 2005.

[17] M. Ashby and Y. Bréchet, "Designing hybrid materials," *Acta Materialia*, vol. 51, no. 19, pp. 5801–5821, 2003. The Golden Jubilee Issue. Selected topics in Materials Science and Engineering: Past, Present and Future.

[18] R. T. Roland F, "Laser welded sandwich panels for the shipbuilding industry," *Lightweight construction latest developments*, pp. 1—-12, 2000.

[19] T. JA, "The application of sandwich structures to ship design: phase four summary report," *Department of Naval Architecture and Shipbuilding,University of Newcastle-upon-Tyne*, 1988.

[20] L. S. Seo SI, Lee CS, "Basic studies on the production technology for application of large extrusion profiles of aluminum a6005a alloy to ship structures," *J Ship Product*, vol. 15, no. 3, pp. 156—-163, 1999.

[21] L. S. Seo SI, Lee CS, "A study on the application of large aluminum hollow extrusion profiles to ship structures," *J Ship Product*, vol. 15, no. 3, pp. 179—-190, 1999.

[22] E. A. Valdevit L, Hutchinson JW, "Structurally optimized sandwich panels with prismatic cores," *Int J Solids Struct*, 2004.

[23] L. Valdevit, Z. Wei, C. Mercer, F. Zok, and A. Evans, "Structural performance of near-optimal sandwich panels with corrugated cores," *International Journal of Solids and Structures*, vol. 43, no. 16, pp. 4888–4905, 2006.

[24] J. D. Poirier, S. S. Vel, and V. Caccese, "Multi-objective optimization of laser-welded steel sandwich panels for static loads using a genetic algorithm," *Engineering Structures*, vol. 49, pp. 508–524, 2013.

[25] H. Kolsters and P. Wennhage, "Optimisation of laser-welded sandwich panels with multiple design constraints," *Marine Structures*, vol. 22, no. 2, pp. 154–171, 2009.

[26] C. JD, "Predicting the properties of adhesively bonded corrugated core sandwich panels.," *Second international conference adhesion '87*, pp. 1—-6, 1987.

[27] T. Marsico, P. Denney, and A. Furió, "Laser welding of lightweight structural steel panels," 1993.

[28] E. Knox, M. Cowling, and I. Winkle, "Adhesively bonded steel corrugated core sandwich construction for marine applications," *Marine Structures*, vol. 11, no. 4, pp. 185–204, 1998.

[29] K. Minamida, M. Oikawa, A. Sugihashi, and M. Kido, "Laser welding of metal honeycomb panel with multiple reflecting effects of high power laser beams," *International Congress on Applications of Lasers & Electro-Optics*, vol. 1992, no. 1, pp. 557–564, 1992.

[30] Z. Bažant and M. Christensen, "Analogy between micropolar continuum and grid frameworks under initial stress," *International Journal of Solids and Structures*, vol. 8, no. 3, pp. 327–346, 1972.

[31] J. Romanoff, H. Remes, G. Socha, M. Jutila, and P. Varsta, "The stiffness of laser stake welded t-joints in web-core sandwich structures," *Thin-Walled Structures*, vol. 45, no. 4, pp. 453–462, 2007.

[32] H. R. D. Frank, J. Romanoff, "Fatigue strength assessment of laser stake-welded web-core steel sandwich panels," *Fatigue and Fracture of Engineering Materials and Structures*, vol. 36, no. 8, pp. 724–737, 2013.

[33] Y. Jie, "Deflection of a web-core sandwich beam in weak direction," *The Open Ocean Engineering Journal*, vol. 6, pp. 82–87, 2013.

[34] P. Nampally, A. T. Karttunen, and J. Reddy, "Nonlinear finite element analysis of lattice core sandwich beams," *European Journal of Mechanics - A/Solids*, vol. 74, pp. 431–439, 2019.

[35] A. T. Karttunen, J. Reddy, and J. Romanoff, "Two-scale constitutive modeling of a lattice core sandwich beam," *Composites Part B: Engineering*, vol. 160, pp. 66–75, 2019.

[36] S. Roy Chowdhury and J. Reddy, "Geometrically exact micropolar timoshenko beam and its application in modelling sandwich beams made of architected lattice core," *Composite Structures*, vol. 226, p. 111228, 2019.

[37] N. K. Romli, J. Xiaoxia, S. M. Sofie, and M. R. M. Rejab, "Three-point bending response of laser-welded sandwich structure with varying number of core and span length," *IOP Conference Series: Materials Science and Engineering*, vol. 788, p. 012011, jun 2020.

[38] S. B. Libove, Charles & Batdorf, "A general small-deflection theory for flat sandwich plates," *University of North Texas Libraries, UNT Digital Library*, September 30, 1947.

[39] J. Romanoff, P. Varsta, and A. Klanac, "Stress analysis of homogenized web-core sandwich beams," *Composite Structures*, vol. 79, no. 3, pp. 411–422, 2007.

[40] J. Romanoff and P. Varsta, "Bending response of web-core sandwich plates," *Composite Structures*, vol. 81, no. 2, pp. 292–302, 2007.

[41] H. Tiersten and J. Bleustein, "Generalized elastic continua," in *RD Mindlin and applied mechanics*, pp. 67–103, Elsevier, 1974.

[42] F. Yang, A. Chong, D. Lam, and P. Tong, "Couple stress based strain gradient theory for elasticity," *International Journal of Solids and Structures*, vol. 39, no. 10, pp. 2731–2743, 2002.

[43] A. C. Eringen, *Microcontinuum field theories: I. Foundations and solids*. Springer Science & Business Media, 2012.

[44] C. F. Cosserat E, "Théorie des corps déformables.," *Nature*, vol. 81, p. 67, 1909.

[45] W. Nowacki, "Theory of asymmetric elasticity," *Pergamon Press, Headington Hill Hall, Oxford OX 3 0 BW, UK, 1986.*, 1986.

[46] S. Ramezani, R. Naghdabadi, and S. Sohrabpour, "Analysis of micropolar elastic beams," *European Journal of Mechanics - A/Solids*, vol. 28, no. 2, pp. 202–208, 2009.

[47] A. Nobili, "On the generalization of the timoshenko beam model based on the micropolar linear theory: static case," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[48] R. A. Regueiro and Z. Duan, "Static and dynamic micropolar linear elastic beam finite element formulation, implementation, and analysis," *Journal of Engineering Mechanics*, vol. 141, no. 8, p. 04015026, 2015.

[49] S. Shaw, "High frequency vibration of a rectangular micropolar beam: A dynamical analysis," *International Journal of Mechanical Sciences*, vol. 108-109, pp. 83–89, 2016.

[50] V. Zozulya, "Micropolar curved rods. 2-d, high order, timoshenko's and euler-bernoulli models," *Curved and Layered Structures*, vol. 4, no. 1, pp. 104–118, 2017.

[51] J. Romanoff, A. Laakso, and P. Varsta, "Improving the shear properties of web-core sandwich structures using filling material," 2009.

[52] A. T. Karttunen, M. Kanerva, D. Frank, J. Romanoff, H. Remes, J. Jelovica, S. Bossuyt, and E. Sarlin, "Fatigue strength of laser-welded foam-filled steel sandwich beams," *Materials & Design*, vol. 115, pp. 64–72, 2017.

[53] N. Salem and S. Hussein, "Data dimensional reduction and principal components analysis," *Procedia Computer Science*, vol. 163, pp. 292–299, 2019. 16th Learning and Technology Conference 2019Artificial Intelligence and Machine Learning: Embedding the Intelligence.

[54] J. Feng, L. T. Yang, G. Dai, W. Wang, and D. Zou, "A secure high-order lanczos-based orthogonal tensor svd for big data reduction in cloud environment," *IEEE Transactions on Big Data*, vol. 5, no. 3, pp. 355–367, 2019.

[55] P. Sadasivan and D. Dutt, "Svd based technique for noise reduction in electroencephalographic signals," *Signal Processing*, vol. 55, no. 2, pp. 179–189, 1996.

[56] YongchangWang and L. Zhu, "Research and implementation of svd in machine learning," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pp. 471–475, 2017.

[57] R. Byrd, J. Gilbert, and J. Nocedal, "A trust region method based on interior point techniques for nonlinear programming," *Mathematical Programming*, vol. 89, no. 1, pp. 149–185, 2000.

[58] R. Waltz, J. Morales, J. Nocedal, and D. Orban, "An interior algorithm for nonlinear optimization that combines line search and trust region steps," *Mathematical Programming*, vol. 107, pp. 391–408, July 2006.

[59] P. M.J.D., *The convergence of variable metric matrices in unconstrained optimization*. Berlin, Heidelberg: Springer, 1978.

[60] P. M. Ren-pu G., "The convergence of variable metric matrices in unconstrained optimization," *Mathematical Programming*, vol. 27, p. 123, 1983.

[61] S. P. Han, "A globally convergent method for nonlinear programming," *Optimization Theory and Applications*, vol. 22, p. 297, 1997.

[62] G. P. E., W. Murray, and M. H. Wright, *Practical Optimization*. London: Academic Press, 1981.

[63] T. F. Coleman and Y. Li, "An interior, trust region approach for nonlinear minimization subject to bound," *SIAM Journal on Optimization*, vol. 6, no. 2, pp. 418–445, 1996.

[64] T. F. Coleman and Y. Li, "On the convergence of reflective newton methods for large-scale nonlinear minimization subject to bounds," *Mathematical Programming*, vol. 67, no. 2, pp. 189–224, 1994.

[65] W. Young, R. Budynas, and A. Sadegh, *Roark's Formulas for Stress and Strain, Eighth Edition*. New York: McGraw-Hill Education, 2012.

[66] A. T. Karttunen and J. Reddy, "Hierarchy of beam models for lattice core sandwich structures," *International Journal of Solids and Structures*, vol. 204-205, pp. 172–186, 2020.

[67] C. Wang, J. Reddy, and K. Lee, *Shear Deformable Beams and Plates*. The Boulevard, Langford Lane Kidlington, Oxford, UK: Elsevier Ltd, 2000.

[68] H. G. Allen, *Analysis and Design of Structural Sandwich Panels*. Pergamon Press, 1969.

[69] D. Zenkert, *The handbook of sandwich construction*. Engineering Materials Advisory Services, 1997.

APPENDIX A

SVD MATRICES AND CODE FOR ELLIPTICAL DATA EXAMPLE

A.1 Code for Elliptical Data Example

Listing A.1: For educational purposes

```
clear; clc; close all
a = 1;
b = 5;
x = -5:1:5;
y = (a*sqrt(b^2 - x.^2))./b;
s = length(x);
data = [x',y'];
data(s+1:s*2,:) = [-x',-y'];
scatter(data(:,1),data(:,2),'b')
hold on
%% PC Modifications
%PC 1
[U,S,V] = svd(data);
S_mod = S;
S_mod(2,2) = 0;
model_PC1 = U*S_mod*V';
plot(model_PC1(:,1),model_PC1(:,2),'k-','LineWidth',3)
% PC 2
S_mod = S;
S_mod(1,1) = 0;
model_PC2 = U*S_mod*V';
plot(model_PC2(:,1),model_PC2(:,2),'r--','LineWidth',3)
% Figure 1 plot config
ylim([-b,b])
xlim([-b,b])
h = legend('Data Set','PC 1','PC 2');
legend('boxoff')
%title('PC Vectors for Elliptical Shaped Data Set')
xlabel('x')
ylabel('y')
%% Full Model
figure
scatter(data(:,1),data(:,2),'b')
hold on
%1 PC
[U,S,V] = svd(data);
S_mod = S;
```

```
S_mod(2,2) = 0;
model_PC1 = U*S_mod*V';
scatter(model_PC1(:,1),model_PC1(:,2),'sk')
% 2 PCs
model_PC2 = U*S*V';
scatter(model_PC2(:,1),model_PC2(:,2),'*r')
% Figure 2 config
ylim([-b,b])
xlim([-b,b])
h = legend('Data Set','PC 1','PC 2');
legend('boxoff')
%title({'Model Approximations Using Diffrent Numbers of',' PCs for
   Elliptical Shaped Data Set'})
xlabel('x')
ylabel('y')
%% Screen Plot
[m,n] = size(data);
f = n;
var = S;
sum_var = sum(sum(var));
influence = sum(sort((var/sum_var)*100));
figure
x_names= 1:1:2;
bar(x_names,influence(1,:))
xlabel('Principle Componets');
ylabel('% of Variation');
ylim([0 100]);
```

A.2 Matrix Values for Elliptical Data Example

$$S = \begin{bmatrix} 14.8324 & 0 \\ 0 & 3.6332 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{A.1}$$

$$V^T = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{A.2}$$

$$U = \begin{vmatrix}
-0.34 & 0.00 & -0.15 & -0.08 & -0.01 & 0.06 & 0.13 & 0.19 & 0.25 & 0.31 & 0.34 & 0.34 & 0.23 & 0.15 & 0.08 & 0.01 & -0.06 & -0.13 & -0.19 & -0.25 & -0.31 & -0.34 \\
-0.27 & -0.17 & -0.25 & -0.27 & -0.27 & -0.26 & -0.24 & -0.21 & -0.17 & -0.10 & 0.07 & 0.07 & 0.21 & 0.25 & 0.27 & 0.27 & 0.26 & 0.24 & 0.21 & 0.17 & 0.10 & -0.07 \\
-0.20 & -0.22 & 0.94 & -0.06 & -0.05 & -0.04 & -0.03 & -0.02 & 0.00 & 0.02 & 0.05 & 0.05 & 0.07 & 0.06 & 0.06 & 0.05 & 0.04 & 0.03 & 0.02 & 0.00 & -0.02 & -0.05 \\
-0.13 & -0.25 & -0.06 & 0.94 & -0.06 & -0.05 & -0.05 & -0.04 & -0.02 & 0.00 & 0.03 & 0.03 & 0.06 & 0.06 & 0.06 & 0.06 & 0.05 & 0.05 & 0.04 & 0.02 & 0.00 & -0.03 \\
-0.07 & -0.27 & -0.06 & -0.06 & 0.94 & -0.06 & -0.06 & -0.05 & -0.04 & -0.02 & 0.02 & 0.02 & 0.05 & 0.06 & 0.06 & 0.06 & 0.06 & 0.06 & 0.05 & 0.04 & 0.02 & -0.02 \\
0.00 & -0.28 & -0.05 & -0.06 & -0.06 & 0.93 & -0.06 & -0.06 & -0.05 & -0.04 & 0.00 & 0.00 & 0.04 & 0.05 & 0.06 & 0.06 & 0.07 & 0.06 & 0.06 & 0.05 & 0.04 & 0.00 \\
0.07 & -0.27 & -0.04 & -0.05 & -0.06 & -0.07 & 0.93 & -0.07 & -0.06 & -0.05 & -0.02 & -0.02 & 0.03 & 0.04 & 0.05 & 0.06 & 0.07 & 0.07 & 0.07 & 0.06 & 0.05 & 0.02 \\
0.13 & -0.25 & -0.03 & -0.05 & -0.06 & -0.07 & -0.07 & 0.93 & -0.07 & -0.07 & -0.03 & -0.03 & 0.01 & 0.03 & 0.05 & 0.06 & 0.07 & 0.07 & 0.07 & 0.07 & 0.07 & 0.03 \\
0.20 & -0.22 & -0.02 & -0.04 & -0.05 & -0.06 & -0.07 & -0.08 & 0.92 & -0.08 & -0.05 & -0.05 & 0.00 & 0.02 & 0.04 & 0.05 & 0.06 & 0.07 & 0.08 & 0.08 & 0.08 & 0.05 \\
0.27 & -0.17 & 0.00 & -0.02 & -0.04 & -0.05 & -0.06 & -0.07 & -0.08 & 0.91 & -0.07 & -0.07 & -0.02 & 0.00 & 0.02 & 0.04 & 0.05 & 0.06 & 0.07 & 0.08 & 0.09 & 0.07 \\
0.34 & 0.00 & 0.04 & 0.02 & 0.00 & -0.02 & -0.03 & -0.05 & -0.06 & -0.08 & 0.92 & -0.08 & -0.06 & -0.04 & -0.02 & 0.00 & 0.02 & 0.03 & 0.05 & 0.06 & 0.08 & 0.08 \\
0.34 & 0.00 & 0.04 & 0.02 & 0.00 & -0.02 & -0.03 & -0.05 & -0.06 & -0.08 & -0.08 & 0.92 & -0.06 & -0.04 & -0.02 & 0.00 & 0.02 & 0.03 & 0.05 & 0.06 & 0.08 & 0.08 \\
0.27 & 0.17 & 0.06 & 0.05 & 0.04 & 0.03 & 0.01 & 0.00 & -0.02 & -0.04 & -0.07 & -0.07 & 0.93 & -0.06 & -0.05 & -0.04 & -0.03 & -0.01 & 0.00 & 0.02 & 0.04 & 0.07 \\
0.20 & 0.22 & 0.06 & 0.06 & 0.05 & 0.04 & 0.03 & 0.02 & 0.00 & -0.02 & -0.05 & -0.05 & -0.07 & 0.94 & -0.06 & -0.05 & -0.04 & -0.03 & -0.02 & 0.00 & 0.02 & 0.05 \\
0.13 & 0.25 & 0.06 & 0.06 & 0.06 & 0.05 & 0.05 & 0.04 & 0.02 & 0.00 & -0.03 & -0.03 & -0.06 & -0.06 & 0.94 & -0.06 & -0.05 & -0.05 & -0.04 & -0.02 & 0.00 & 0.03 \\
0.07 & 0.27 & 0.06 & 0.06 & 0.06 & 0.06 & 0.06 & 0.05 & 0.04 & 0.02 & -0.02 & -0.02 & -0.05 & -0.06 & -0.06 & 0.94 & -0.06 & -0.06 & -0.05 & -0.04 & -0.02 & 0.02 \\
0.00 & 0.28 & 0.05 & 0.06 & 0.06 & 0.07 & 0.06 & 0.06 & 0.05 & 0.04 & 0.00 & 0.00 & -0.04 & -0.05 & -0.06 & -0.06 & 0.93 & -0.06 & -0.06 & -0.05 & -0.04 & 0.00 \\
-0.07 & 0.27 & 0.04 & 0.05 & 0.06 & 0.07 & 0.07 & 0.07 & 0.06 & 0.05 & 0.02 & 0.02 & -0.03 & -0.04 & -0.05 & -0.06 & -0.07 & 0.93 & -0.07 & -0.06 & -0.05 & -0.02 \\
-0.13 & 0.25 & 0.03 & 0.05 & 0.06 & 0.07 & 0.07 & 0.07 & 0.07 & 0.07 & 0.03 & 0.03 & -0.01 & -0.03 & -0.05 & -0.06 & -0.07 & -0.07 & 0.93 & -0.07 & -0.07 & -0.03 \\
-0.20 & 0.22 & 0.02 & 0.04 & 0.05 & 0.06 & 0.07 & 0.08 & 0.08 & 0.08 & 0.05 & 0.05 & 0.00 & -0.02 & -0.04 & -0.05 & -0.06 & -0.07 & -0.08 & 0.92 & -0.08 & -0.05 \\
-0.27 & 0.17 & 0.00 & 0.02 & 0.04 & 0.05 & 0.06 & 0.07 & 0.08 & 0.09 & 0.07 & 0.07 & 0.02 & 0.00 & -0.02 & -0.04 & -0.05 & -0.06 & -0.07 & -0.08 & 0.91 & -0.07 \\
-0.34 & 0.00 & -0.04 & -0.02 & 0.00 & 0.02 & 0.03 & 0.05 & 0.06 & 0.08 & 0.08 & 0.08 & 0.06 & 0.04 & 0.02 & 0.00 & -0.02 & -0.03 & -0.05 & -0.06 & -0.08 & 0.92
\end{vmatrix} \tag{A.3}$$

APPENDIX B

SIMPLIFIED SVD MODEL FIGURES FOR FORCES APPLIED AT DIFFERENT LOCATIONS

In all cases see Figure 4.13 for fixtures and load location. Also note that load is kept constant at $10,000 lb_f$

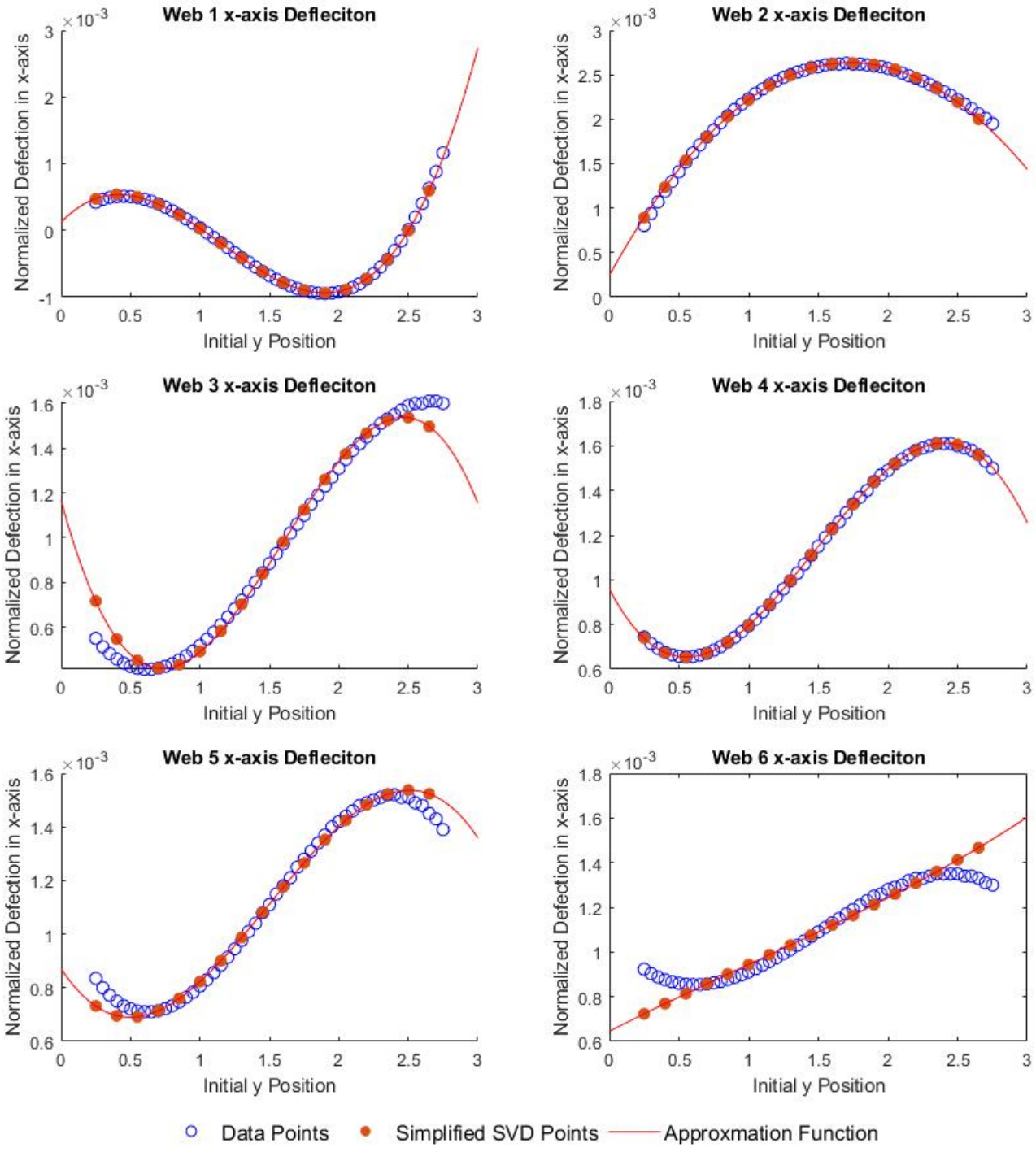


Figure B.1: Comparing Cubic Fit on Simplified SVD Model with 3 PCs to Original Points When Load Applied at Location A

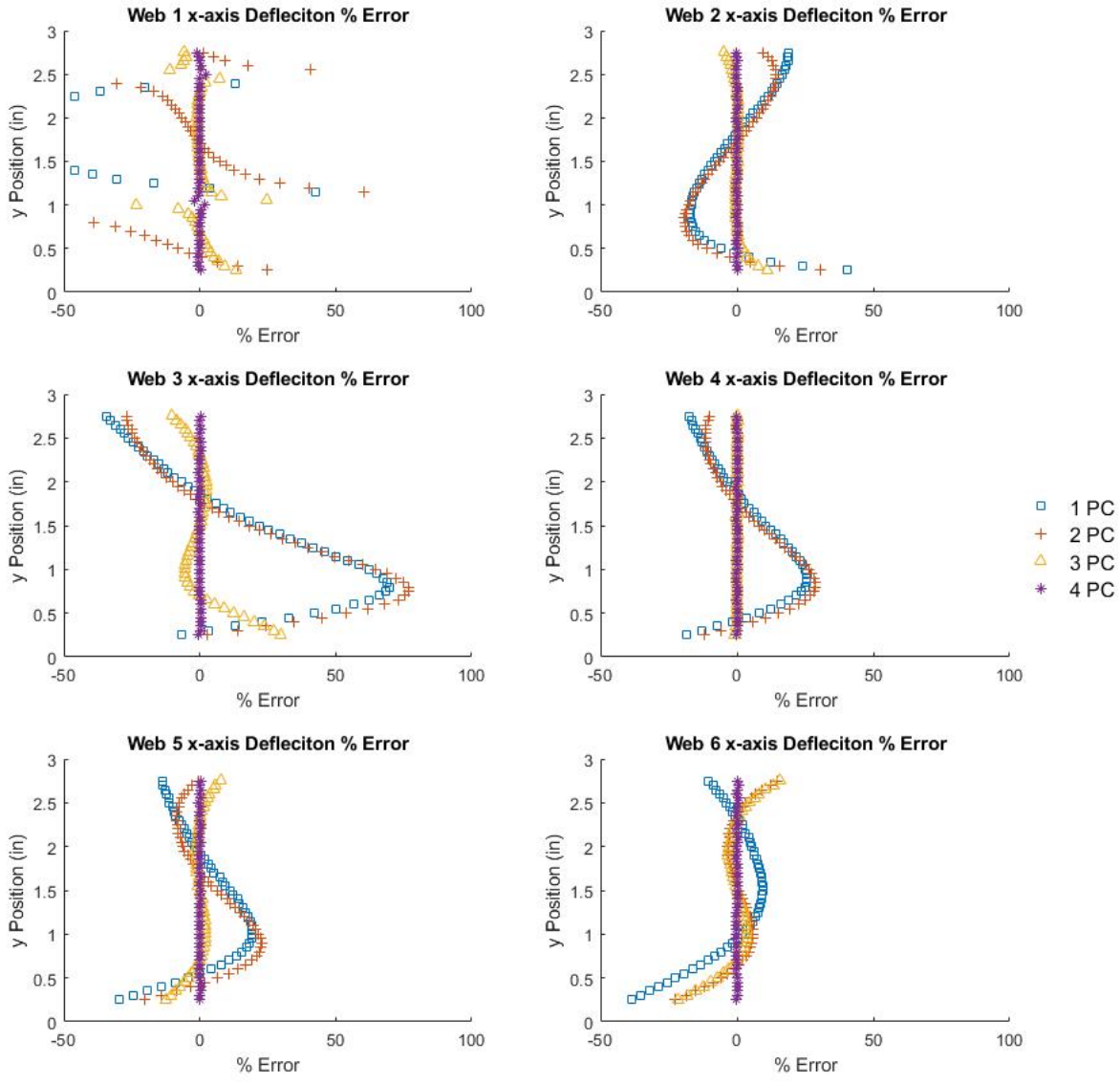


Figure B.2: Comparing Cubic Fit Error with Nodal Displacements Approximated by Different Numbers of PC's When Load Applied at Location A

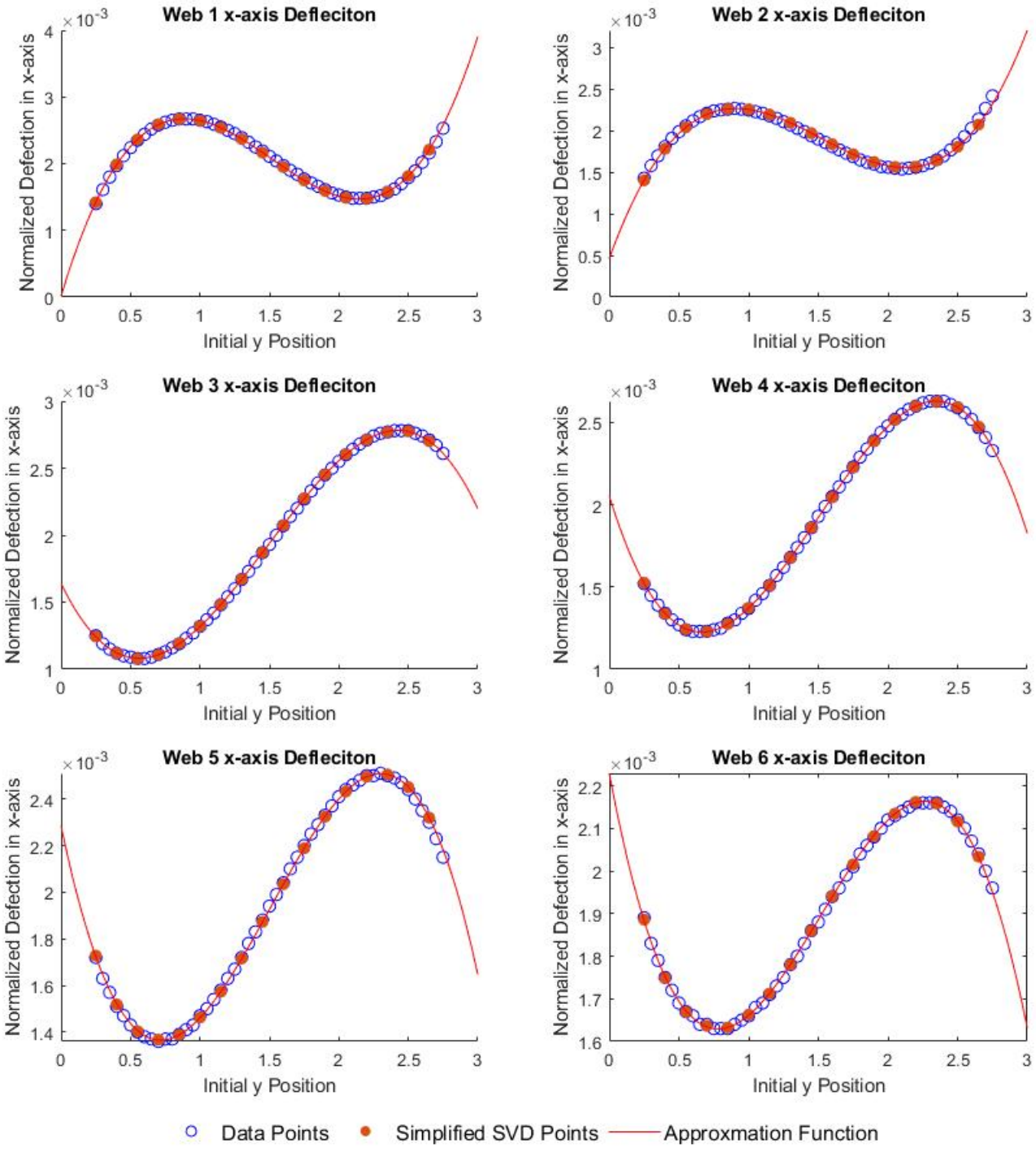


Figure B.3: Comparing Cubic Fit on Simplified SVD Model with 3 PCs to Original Points When Load Applied at Location B

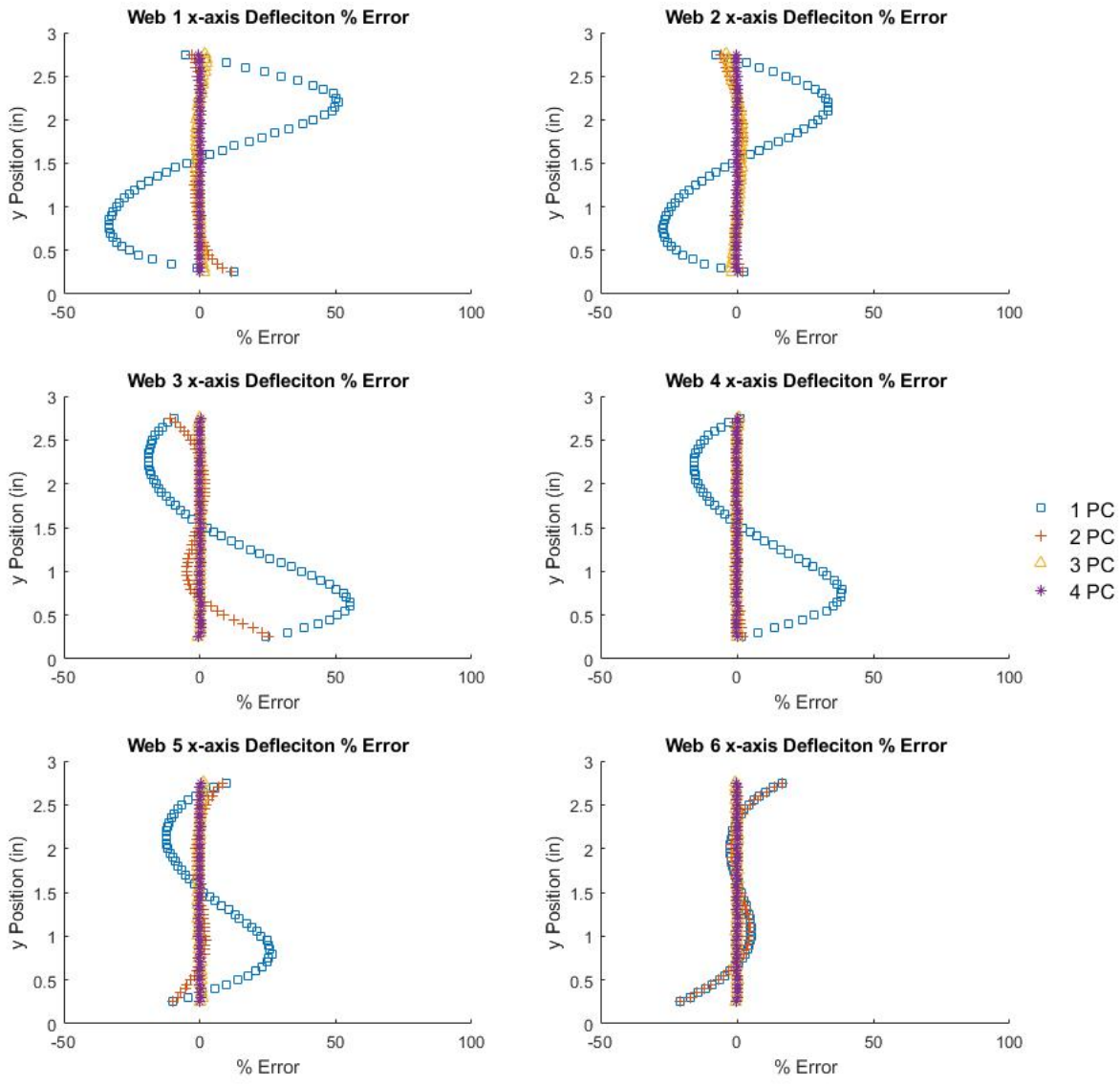


Figure B.4: Comparing Cubic Fit Error with Nodal Displacements Approximated by Different Numbers of PC's When Load Applied at Location B

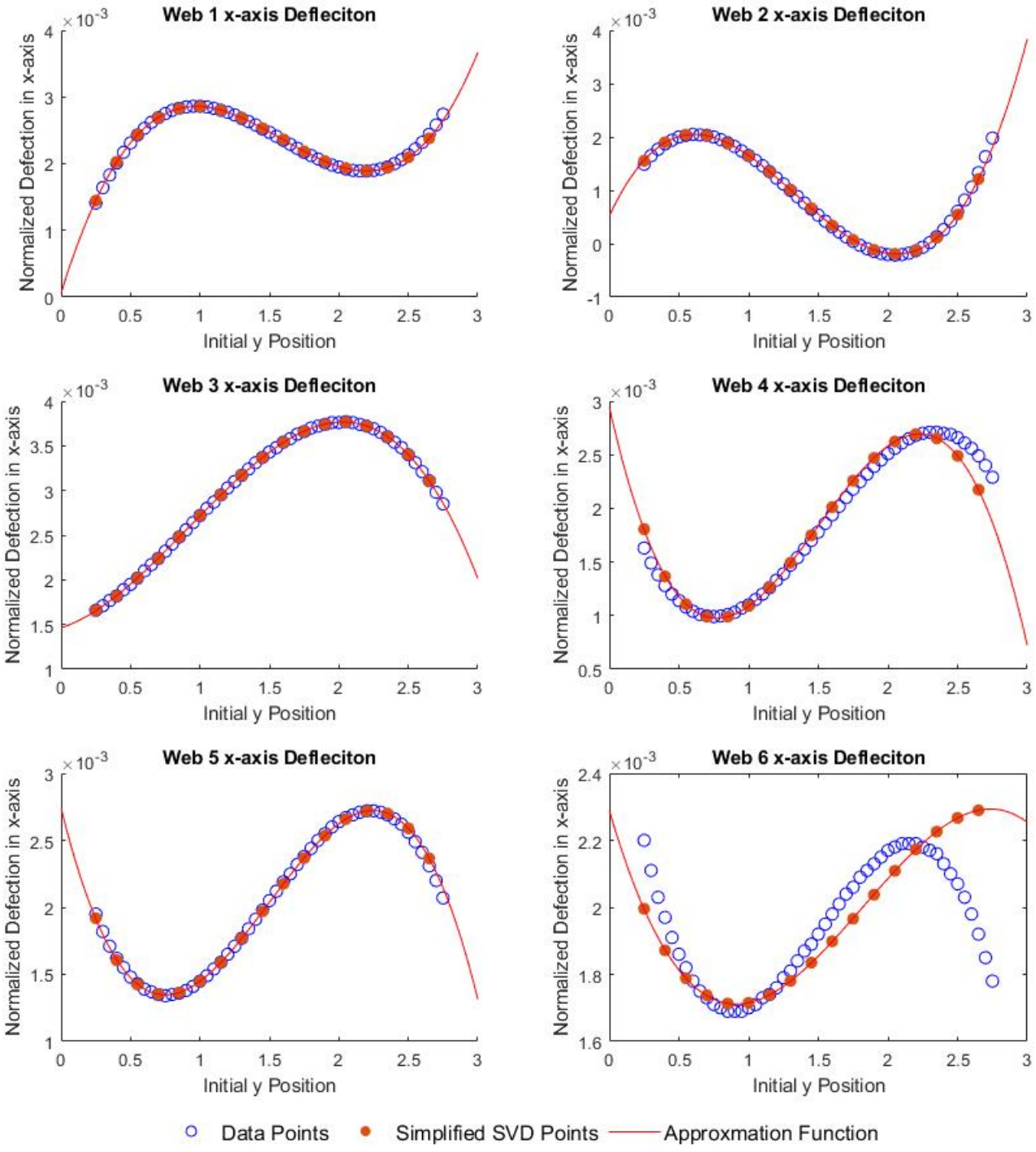


Figure B.5: Comparing Cubic Fit on Simplified SVD Model with 3 PCs to Original Points When Load Applied at Location C

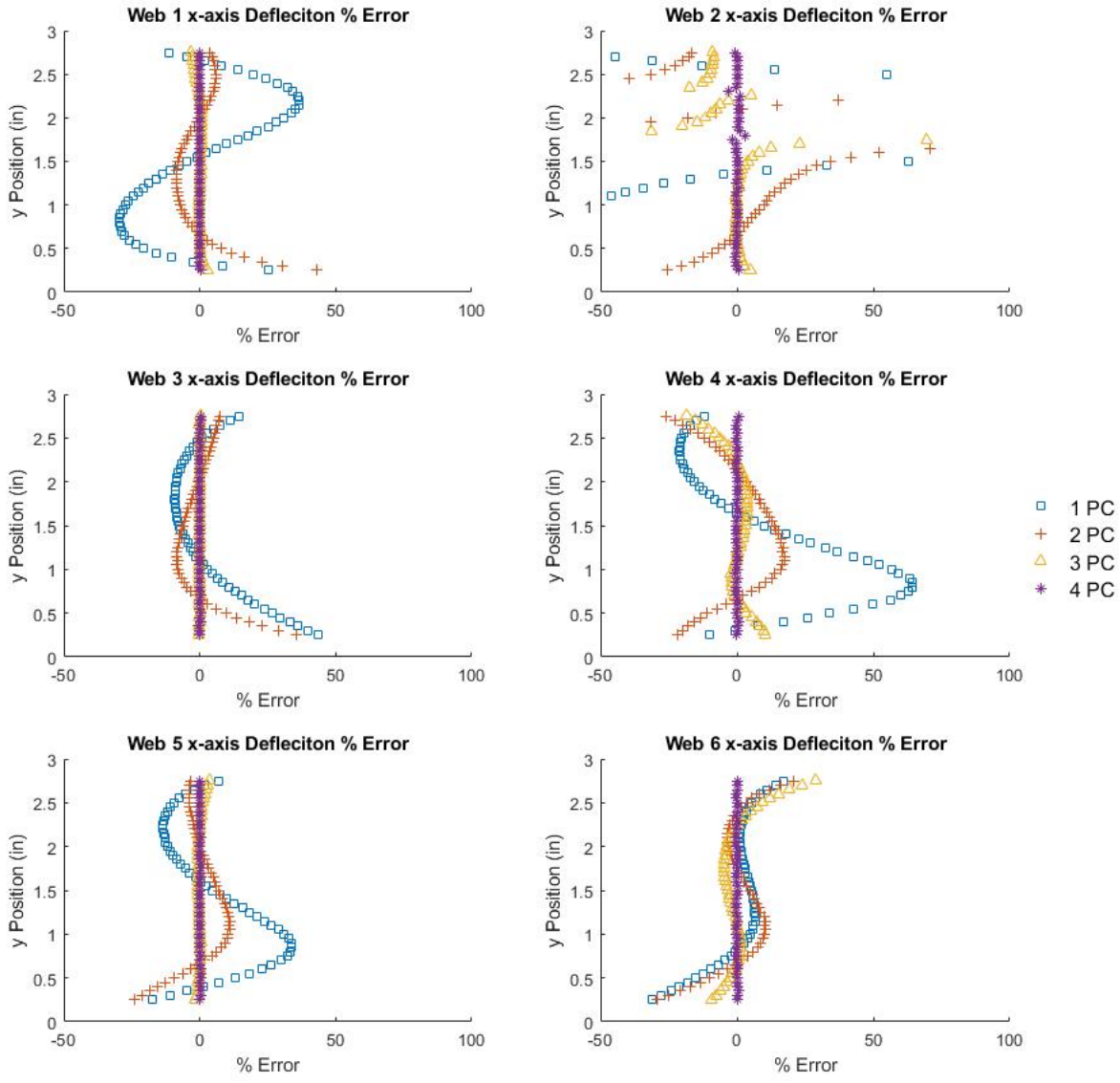


Figure B.6: Comparing Cubic Fit Error with Nodal Displacements Approximated by Different Numbers of PC's When Load Applied at Location C

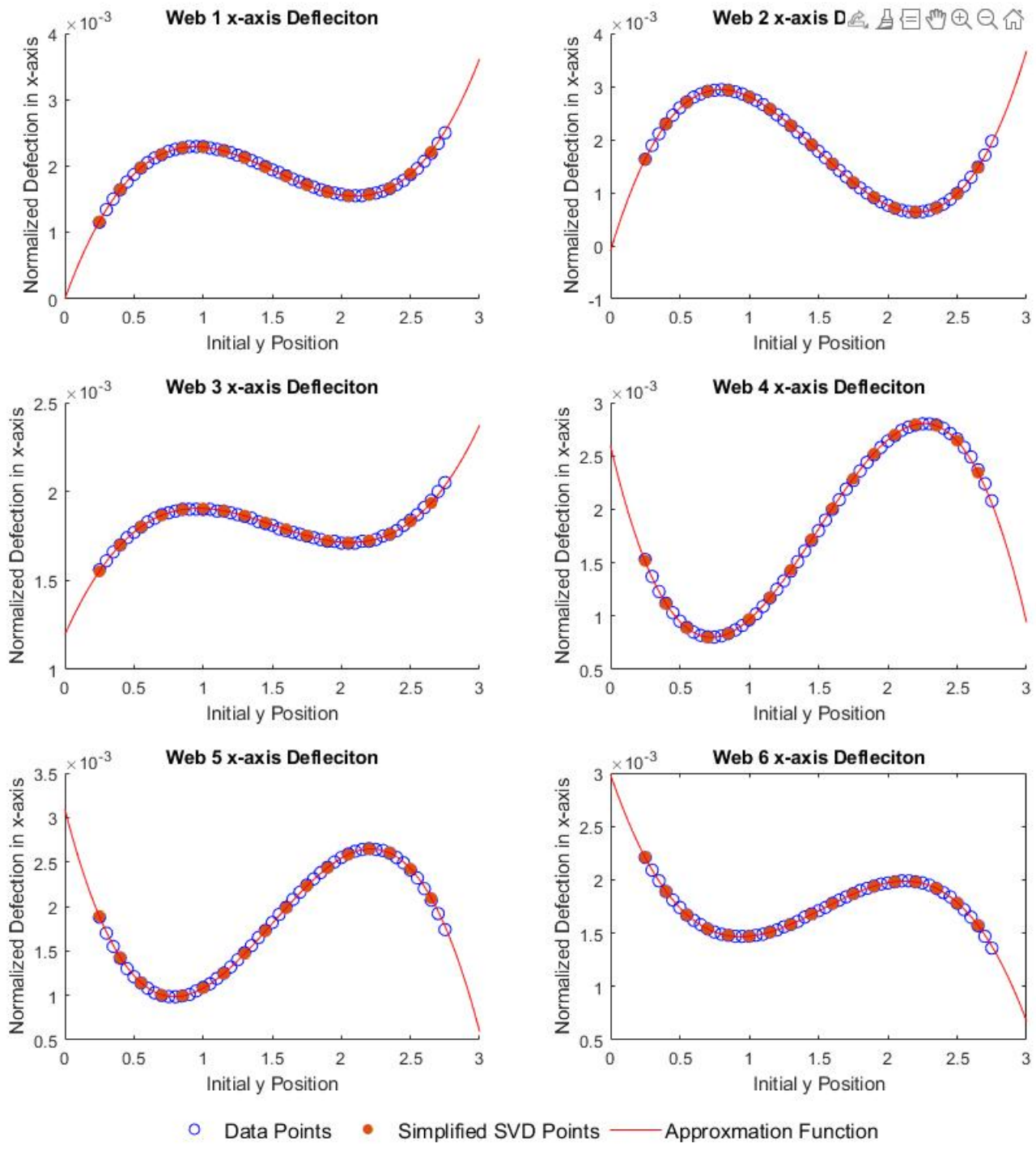


Figure B.7: Comparing Cubic Fit on Simplified SVD Model with 3 PCs to Original Points When Load Applied at Location D

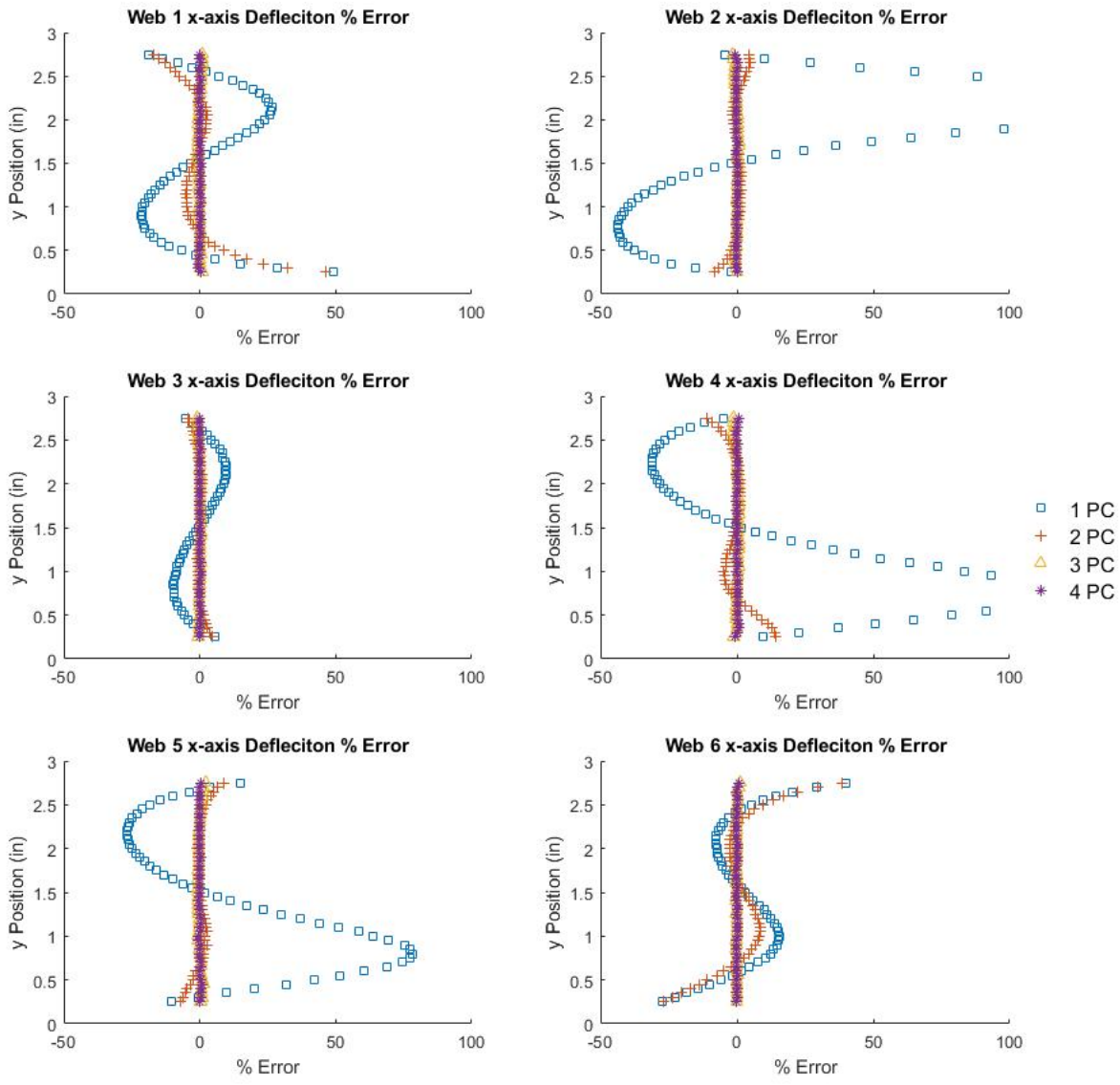


Figure B.8: Comparing Cubic Fit Error with Nodal Displacements Approximated by Different Numbers of PC's When Load Applied at Location D

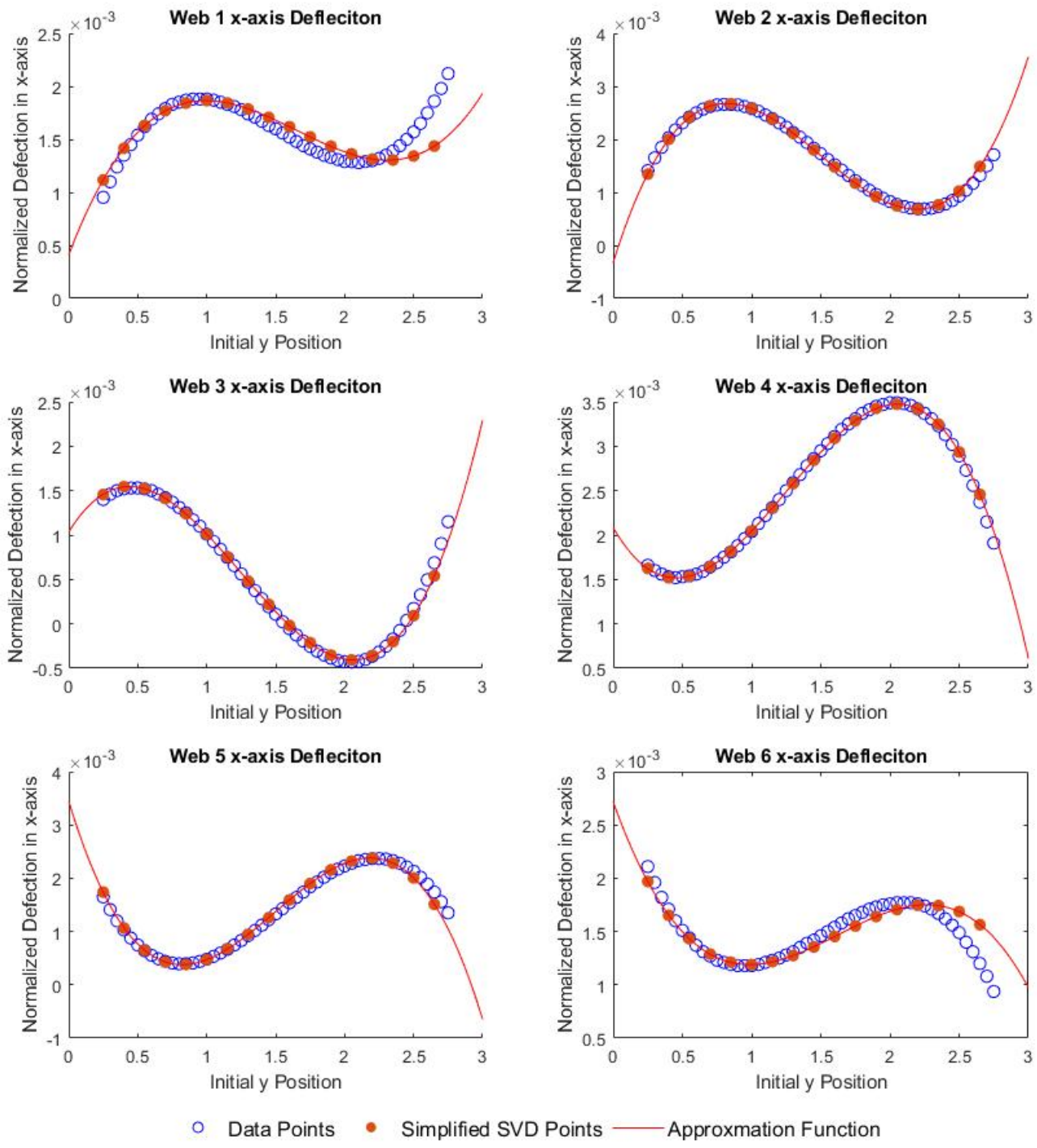


Figure B.9: Comparing Cubic Fit on Simplified SVD Model with 3 PCs to Original Points When Load Applied at Location E

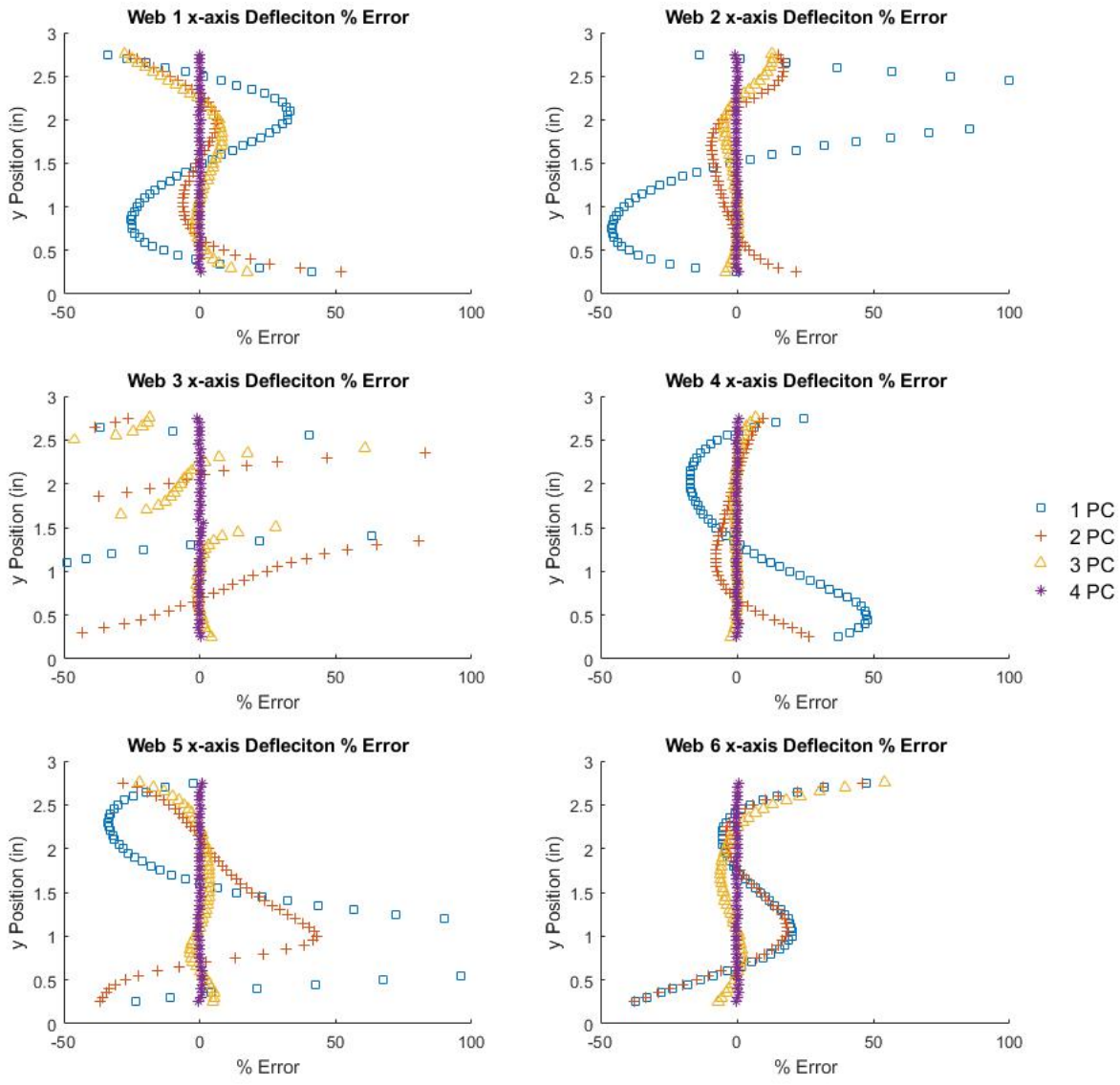


Figure B.10: Comparing Cubic Fit Error with Nodal Displacements Approximated by Different Numbers of PC's When Load Applied at Location E

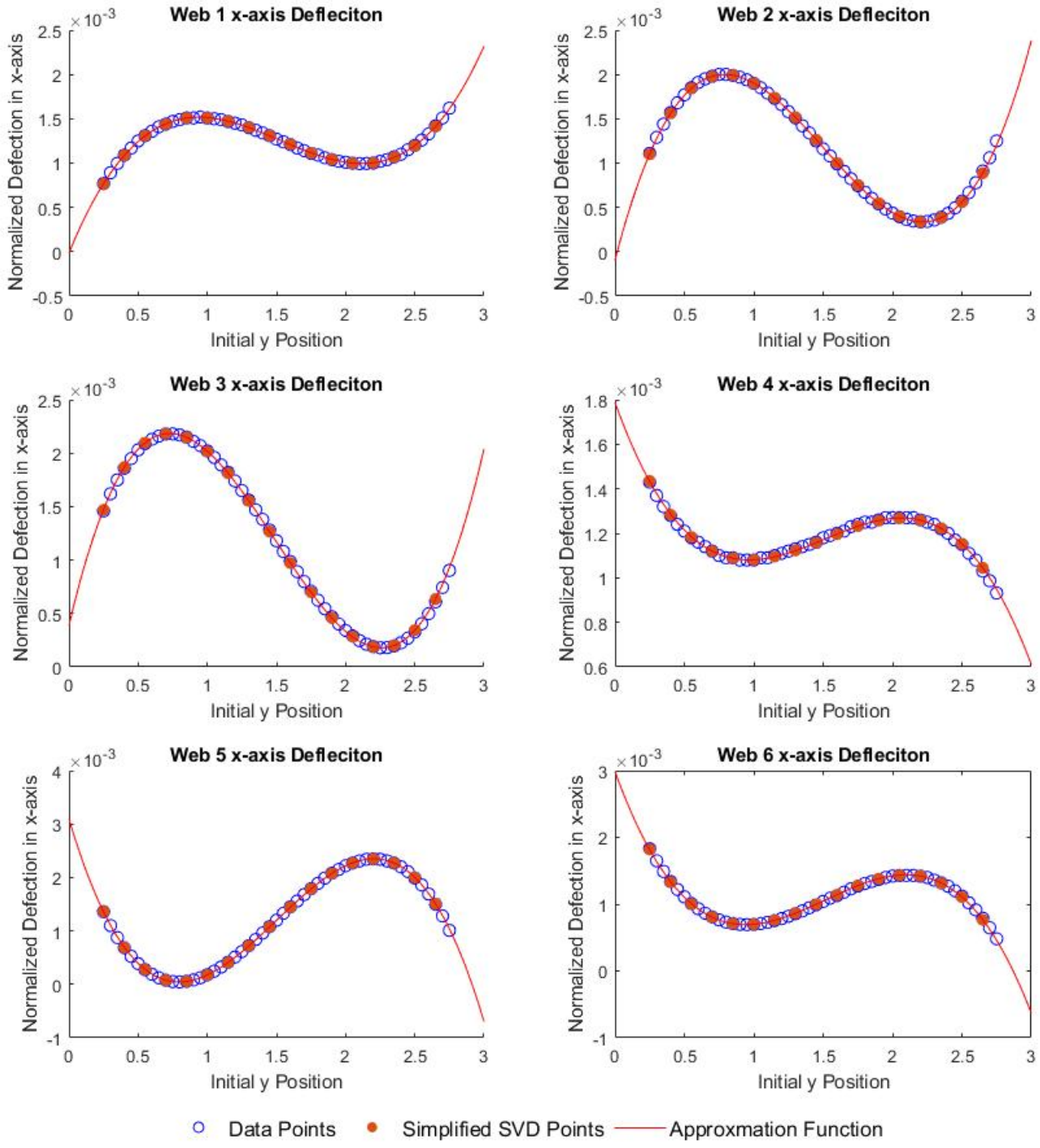


Figure B.11: Comparing Cubic Fit on Simplified SVD Model with 3 PCs to Original Points When Load Applied at Location F

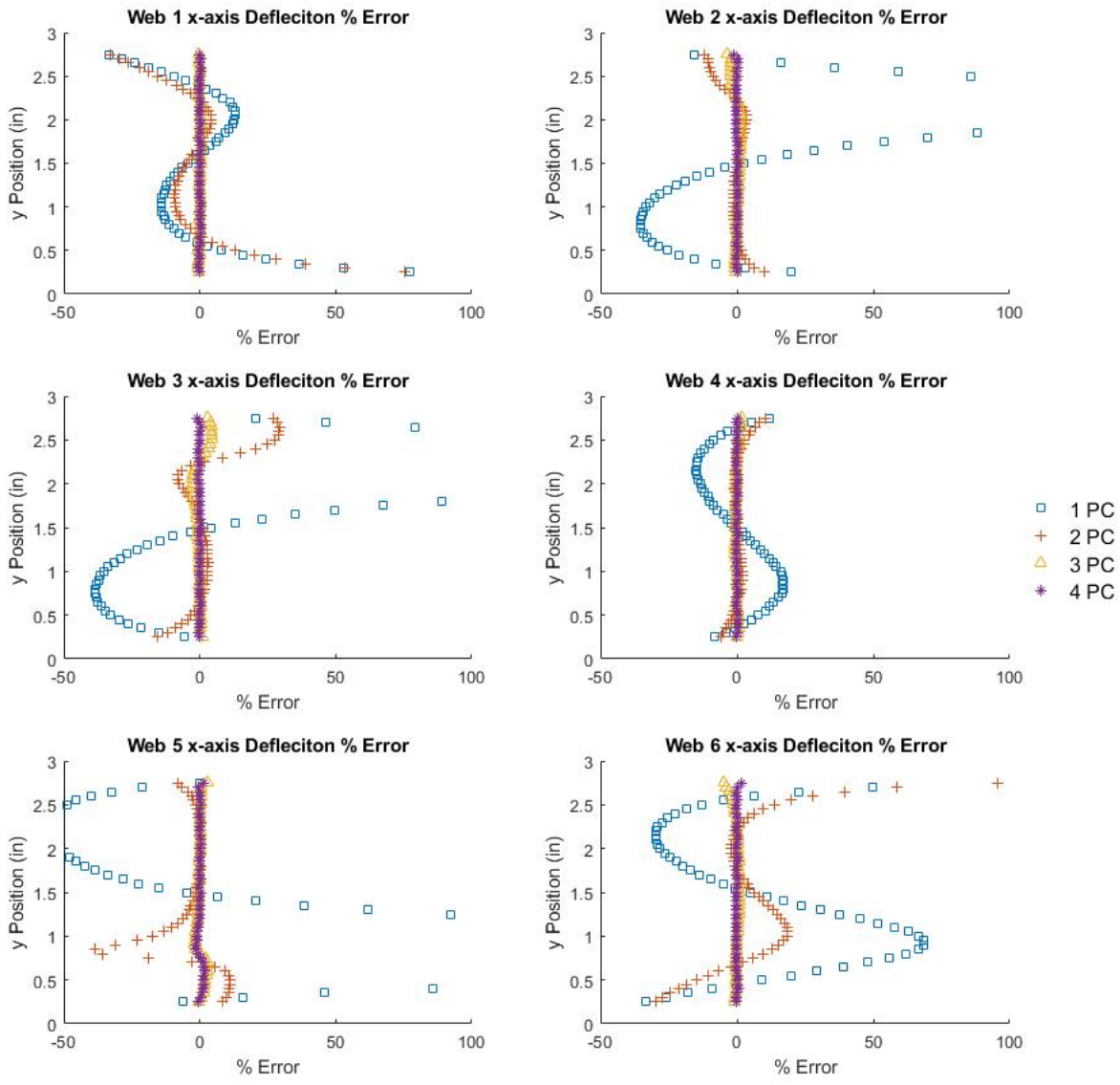


Figure B.12: Comparing Cubic Fit Error with Nodal Displacements Approximated by Different Numbers of PC's When Load Applied at Location F

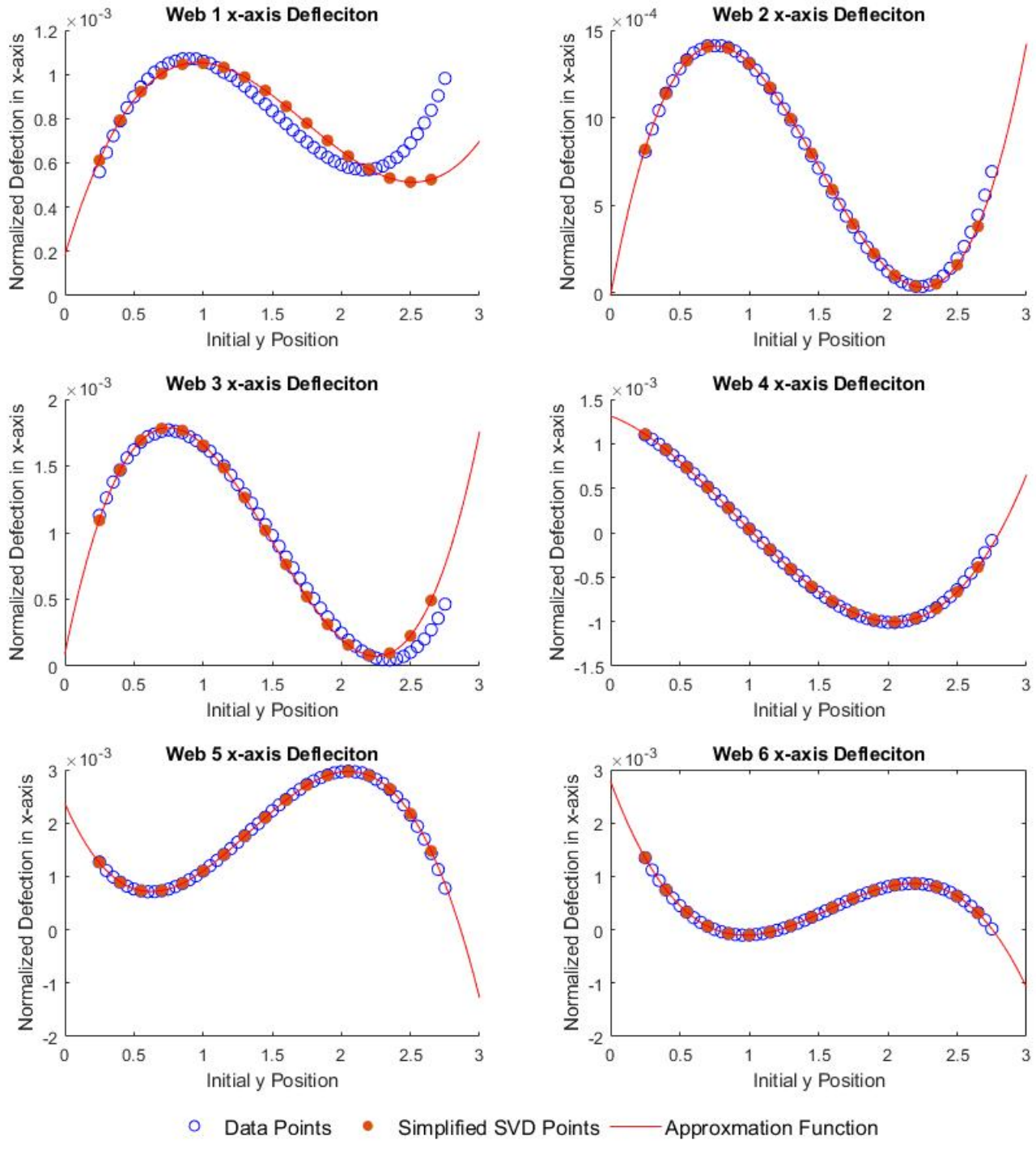


Figure B.13: Comparing Cubic Fit on Simplified SVD Model with 3 PCs to Original Points When Load Applied at Location G

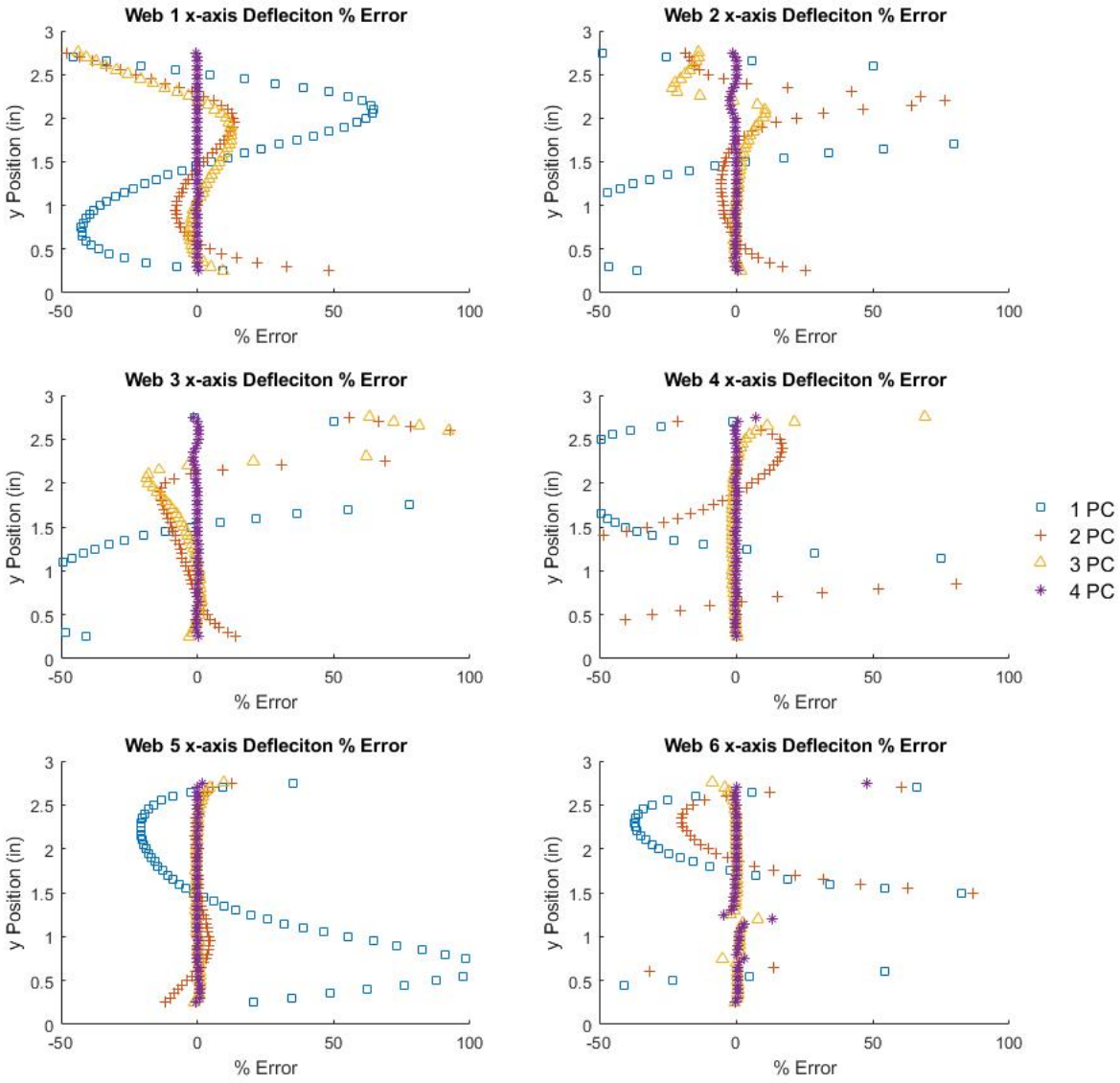


Figure B.14: Comparing Cubic Fit Error with Nodal Displacements Approximated by Different Numbers of PC's When Load Applied at Location G

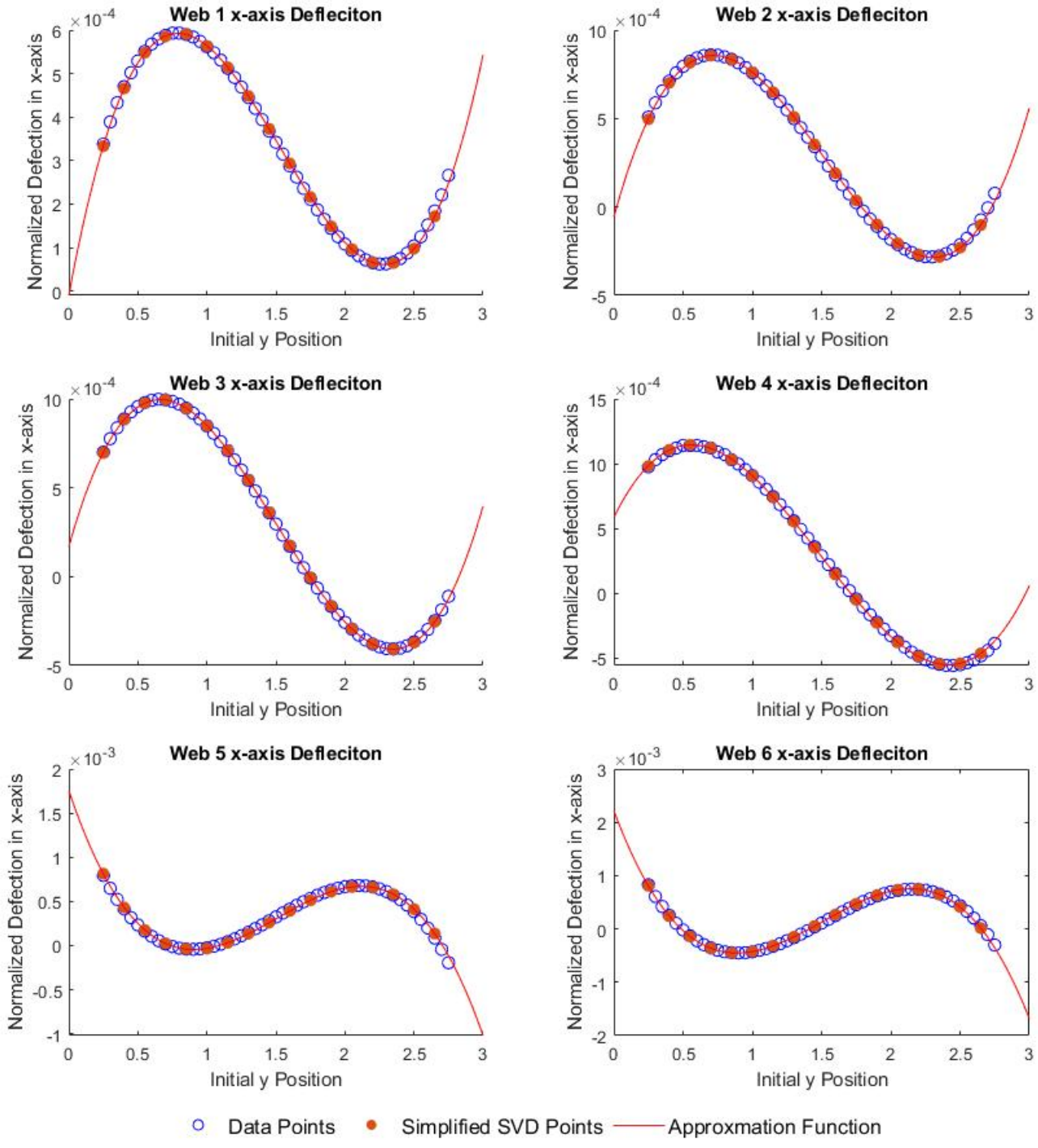


Figure B.15: Comparing Cubic Fit on Simplified SVD Model with 3 PCs to Original Points When Load Applied at Location H

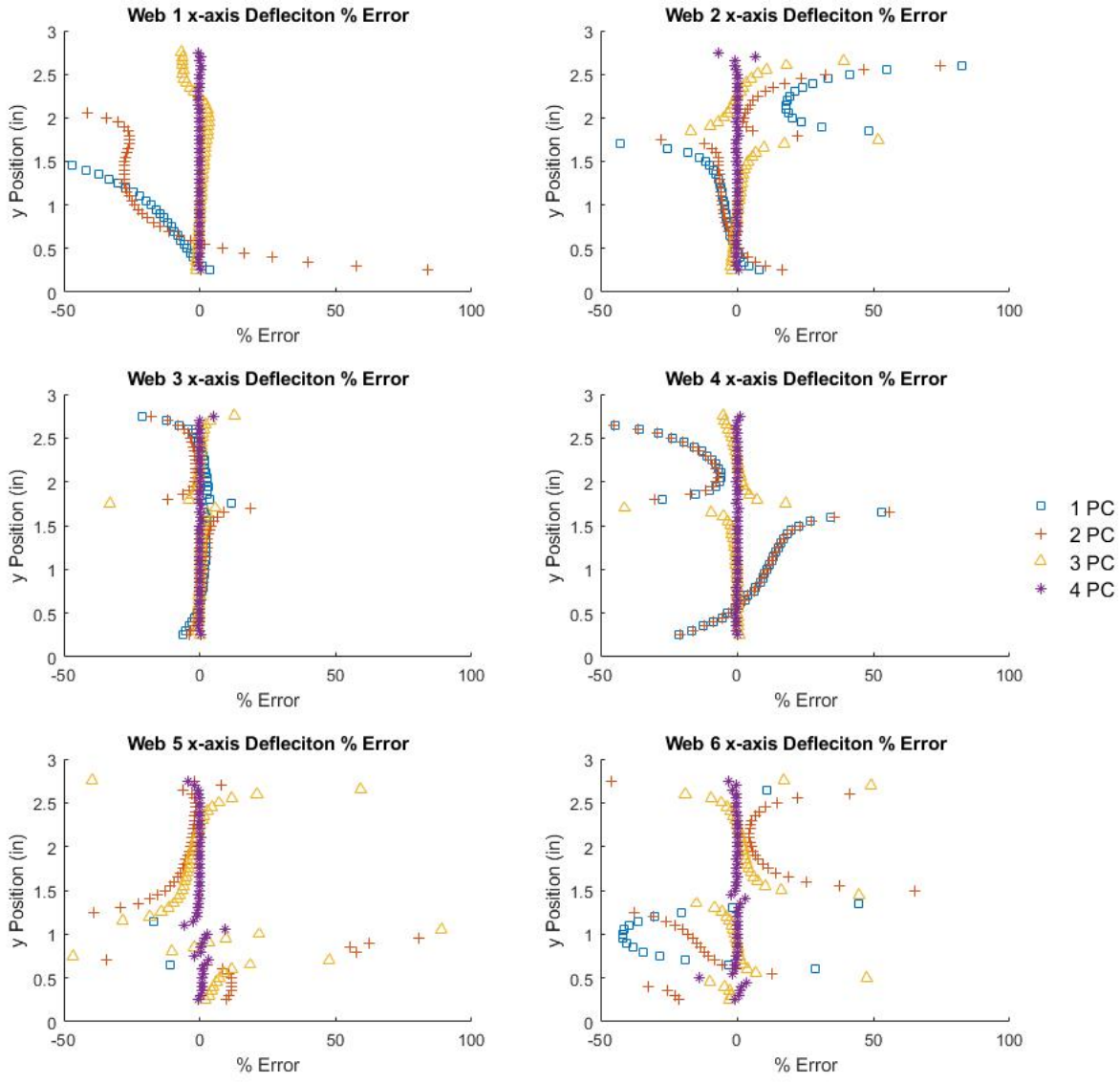


Figure B.16: Comparing Cubic Fit Error with Nodal Displacements Approximated by Different Numbers of PC's When Load Applied at Location H

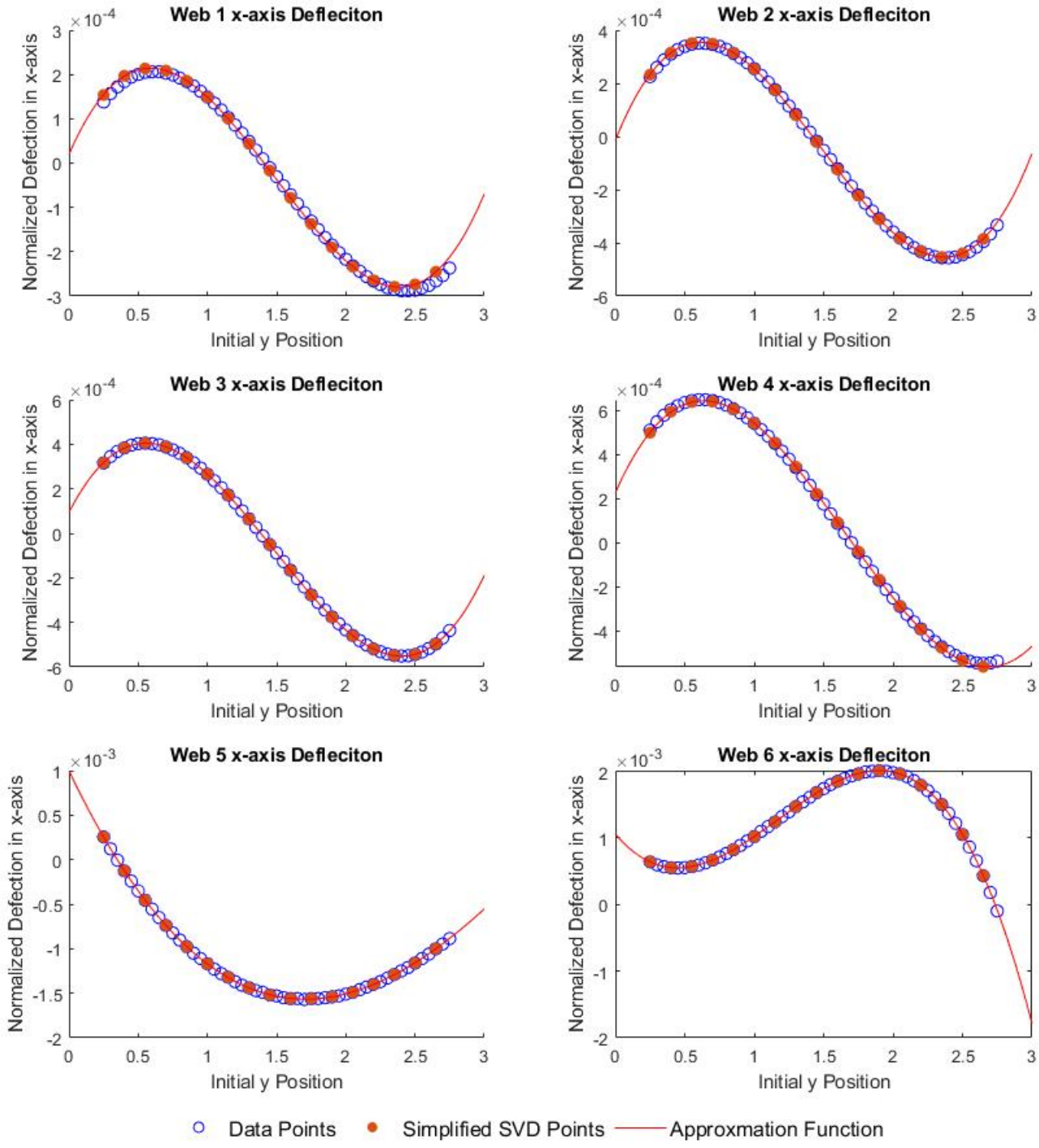


Figure B.17: Comparing Cubic Fit on Simplified SVD Model with 3 PCs to Original Points When Load Applied at Location I

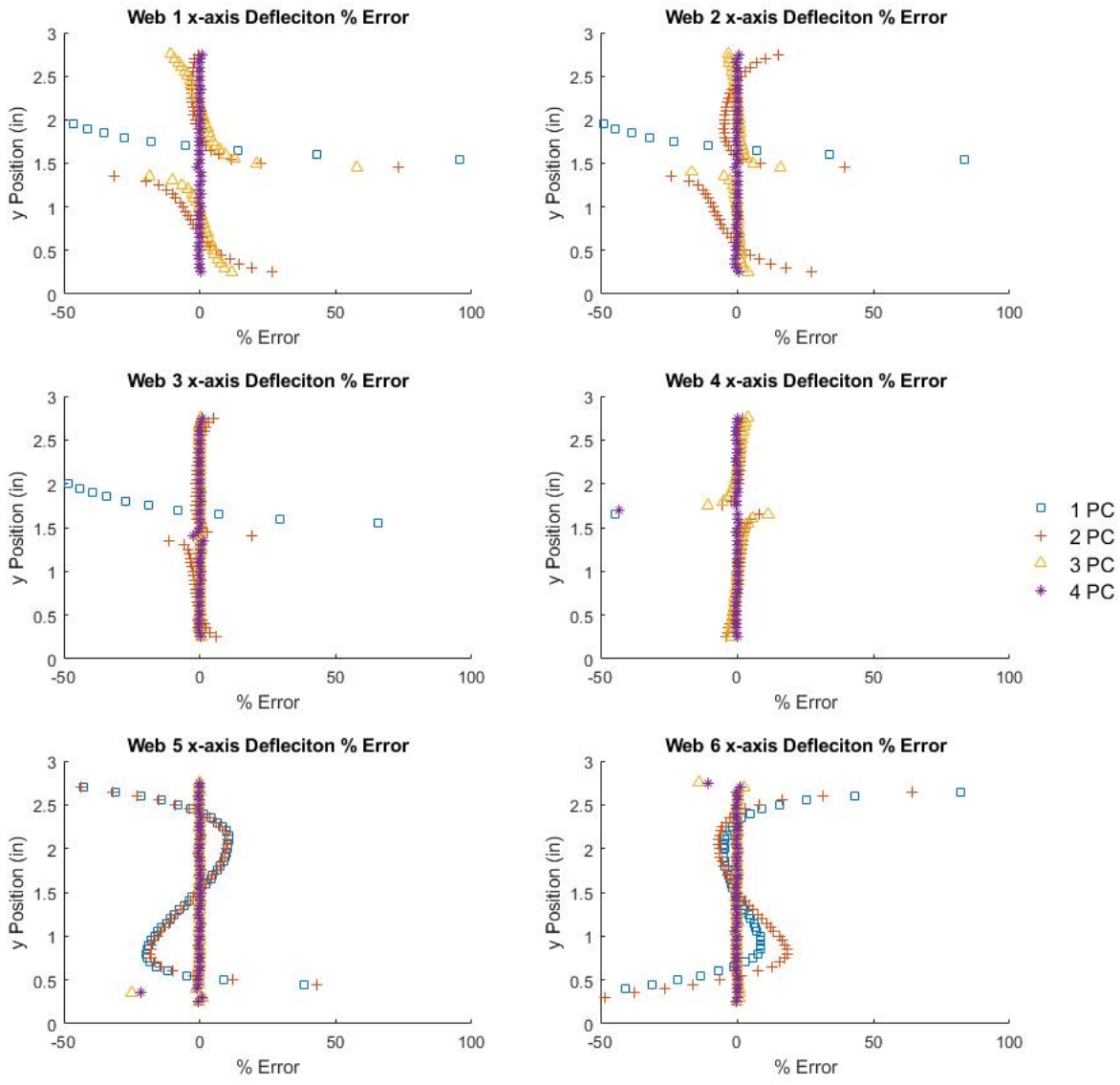


Figure B.18: Comparing Cubic Fit Error with Nodal Displacements Approximated by Different Numbers of PC's When Load Applied at Location I

APPENDIX C

CODE FOR MINIMIZING ENERGY AND PLOTTING RESULTS

C.1 Code for Elliptical Data Example

wcb-solver.m:

```
clear; close all
tic
%% BC's
material_properties
x0 = (-1 + (1+1)*rand(1,10 + 5*(e_num - 1))); % initial guess
A = [];
b = [];
lb = [];
ub = [];
nonlcon = [];
Aeq = zeros(1,10 + 5*(e_num - 1));
Aeq(1,1) = 1; Aeq(2,2) = 1;Aeq(3,3) = 1;Aeq(4,4) = 1;Aeq(5,5) = 1;    %
   Boundary Conditions
Aeq(6,46) = 1; Aeq(7,47) = 1;Aeq(8,48) = 1;Aeq(9,49) = 1;Aeq(10,50) = 1;

beq = zeros(10,1);                 % fixed at all 0's
%% options
options = optimoptions('fmincon',...
    'SpecifyObjectiveGradient', true,...
    'StepTolerance',1*10^(-30),...
    'MaxIterations', 1*10^3,...
    'MaxFunctionEvaluations', 1*10^6);
    %'Algorithm','trust-region-reflective',...
    %'FunctionTolerance',1*10^-30);
    %'HessianFcn',@hessinterior);
    %'OptimalityTolerance',1*10^-10, ...
%% Solve
fun = @wcb;
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(fun,x0,A,b,Aeq,beq,lb
   ,ub,[],options);

%% Print
x_list = ['   alpha      ','beta       ','phiA      ','phiB      ','psi
        '];
st = zeros(1,length(x)/5);
alpha = st;
beta = st;
```

```
phiA = st;
phiB = st;
psi = st;
j = 1;
for i = 1:5:(length(x)-1)
alpha(1,j) = x(1,i);
beta(1,j) = x(1,i+1);
phiA(1,j) = x(1,i+2);
phiB(1,j) = x(1,i+3);
psi(1,j) = x(1,i+4);
j = j+1;
end
M = [alpha;beta;phiA;phiB;psi];
M = M'
disp(x_list)
toc

%% Plot
x_plot = 0:L:L*e_num;
plot(x_plot,M(:,1))
Lseg = L0;
for i = 1:1:e_num
j = 0;
for t = 0:.01:1
    displacment(1+j,1) = ...
        ((3*t^2-2*t^3)*M(i+1,1)...
        +(t^3-t^2)*M(i+1,2)*Lseg...
        +(3*(1-t)^2-2*(1-t)^3)*M(i,1)...
        -((1-t)^3-(1-t)^2)*M(i,2)*Lseg);
    slope(1+j,1) = ...
        1/Lseg*((6*t-6*t^2)*M(i+1,1)...
        +(3*t^2-2*t)*M(i+1,2)*Lseg...
        +(6*t^2-6*t)*M(i,1)...
        -(-3*t^2+4*t-1)*M(i,2)*Lseg);
    curvature(1+j,1) = ...
        1/Lseg^2*((6-12*t)*M(i+1,1)...
        +(6*t-2)*M(i+1,2)*Lseg...
        +(12*t-6)*M(i,1)...
        -(-6*t+4)*M(i,2)*Lseg);
    j = j+1;
end

%% Plots
t = 0+(i-1):.01:1+(i-1);
% DISPLACMENT
```

```
d_plot = plot(t',displacment,'k');
d_plot.LineWidth = 2; grid on; set(gca,'FontSize',15)
ylabel('in')
title(DISPLACMENT)
hold on
end
sgtitle('              4 Point Bending Test')
sgt.FontSize = 20;
sgt.LineWidth = 2;
```

material-properties.m:

```
% This scrip holds variables for the material propories and dimensions of

e_num = 9;                          % number of elements
force = -5000/15;                   % lbf/width
%% Material
E = 29007547;%.53;                    % psi
%% Dimensions
L = 20;                             % in
h = 10;                             % in
t_face = 2;                         % in
w_face = 10;                        % in
t_web = 1;                          % in
w_web = w_face;                     % in

%% Normilzation
L_n=20;                             % total length of the beam # enter whatever
Lstar=1;                            % Char. length for non-dim
Lbar = L_n/Lstar;                   % total length after Lstar
L0 = Lbar/e_num;                    % box length
ntt = L0/10;                        % top face plate thickness
ntb = L0/10;                        % bottom face plate thickness
ntw = L0/20;                        % web thickness
h=L0/2;
top_EI= E*ntt^3/12; %EIt
bot_EI= top_EI; %EIb
web_EI= E*h^2*(ntt+ntb)/4; %Ew
E_sum= 1;%web_EI+top_EI+bot_EI; % sum of stifneess for non-dim
top_EI= top_EI/E_sum; %EIt
bot_EI= top_EI; %EIb
web_EI= web_EI/E_sum; %EIs
% Shear terms
G1=top_EI*12/L0^2;                  % top face plate shear term
G2=G1;                              % bottom face plate shear term
EIs=E*ntw^3/12/E_sum/h/L0;          % web shear term
```

```
ter = 1;                          % 1 if WBC 1000 if tradisonal
k_vec = [top_EI,bot_EI,web_EI,EIs*1.5*ter,G1*ter,G2*ter]*L0;
% common factors
z = 1/L0;
f = 12;
```

wcb.m:

```
function [energy,g] = wcb(x)
material_properties
%loc = 10+5*(e_num-1)-4; % cant. on end
loc = (10+5*(e_num-1))/2-4; % mid force
e = wcb_energy();
energy = e(x)/2 - force/2*x(loc)- force/2*x(loc+5);
g = zeros(1,10+5*(e_num-1));
j = 1;
for i = 1:5:5*e_num
    ii = i+9;
    ge = wcb_grad(j);
    g(1,i:ii) = g(1,i:ii) + ge(x);
    j = j + 1;
end
g(1,loc) = g(1,loc) - force/2;
g(1,loc+5) = g(1,loc+5) - force/2;
end
```

wcb-energy.m:

```
function [p_energy] = wcb_energy()
p_energy = @(x) 0; % intial energy
material_properties
for j = 0:e_num-1
%% Variables
s = j*5;
a1 = @(x) x(1 + s);                    % vertical deflection left
b1 = @(x) x(2 + s);                    % plate slope left
phiA1 = @(x) x(3 + s);                 % top plate slope left
phiB1 = @(x) x(4 + s);                 % bot plate slope left
psi1 = @(x) x(5 + s);                  % web plate slope bot
a2 = @(x) x(6 + s);                    % vertical deflection right
b2 = @(x) x(7 + s);                    % plate slope right
phiA2 = @(x) x(8 + s);                 % bot plate slope right
phiB2 = @(x) x(9 + s);                 % bot plate slope right
psi2 = @(x) x(10 + s);                 % web plate slope top
%% calculations
cal
% Large Groups [EIt,EIb,Ew,EIs,G1,G2]
```

```
% Energy
p_energy = @(x) k_vec(1)*(top_bending(x)^2 + (dface_bending(x)^2)/12)...
    + k_vec(2)*(bot_bending(x)^2 + (dface_bending(x)^2)/12)...
    + k_vec(3)*(axial(x)^2 + (dface_bending(x)^2)/12)...
    + k_vec(4)*(web_bending(x)^2 + (dweb_bending(x)^2)/12 ...
    + 12*(web_bending2(x)^2 - dweb_bending2(x)^2))...
    + k_vec(5)*(phiA_avg(x)^2 + (dphiA(x)^2)/12)...
    + k_vec(6)*(phiB_avg(x)^2 + (dphiB(x)^2)/12)...
    + p_energy(x); % summation of energy
end
end
```

wcb-grad.m:

```
function [g] = wcb_grad(j)
material_properties
%% Variables
s = (j-1)*5; % used to index design variables
a1 = @(x) x(1 + s);                          % vertical deflection left
b1 = @(x) x(2 + s);                          % plate slope left
phiA1 = @(x) x(3 + s);                       % top plate slope left
phiB1 = @(x) x(4 + s);                       % bot plate slope left
psi1 = @(x) x(5 + s);                        % web plate slope bot
a2 = @(x) x(6 + s);                          % vertical deflection right
b2 = @(x) x(7 + s);                          % plate slope right
phiA2 = @(x) x(8 + s);                       % bot plate slope right
phiB2 = @(x) x(9 + s);                       % bot plate slope right
psi2 = @(x) x(10 + s);                       % web plate slope top
%% calculations
cal
%% Gradent
% a1 b1 phiA1 phiB1 psi1 a2 b2 phiA2 phiB2 psi2
g_matrix = [0,-z,-z,0,0,0,z,z,0,0;...                % top bending
            0,-z,0,-z,0,0,z,0,z,0;...                % bot bending
            0,-z,0,0,-z,0,z,0,0,z;...                % axial
            12*z^2,6*z,0,0,0,-12*z^2,6*z,0,0,0;...  % dface_bending
            0,0,-.5,.5,0,0,0,-.5,.5,0;...            % web_bending
            0,0,.25,.25,-.5,0,0,.25,.25,-.5;...      % web_bending2
            0,0,1,-1,0,0,0,-1,1,0;...              % dweb_bending
            0,0,-.5,-.5,1,0,0,.5,.5,-1;...           % dweb_bending2
            0,0,.5,0,0,0,0,.5,0,0;...                % phiA
            0,0,0,.5,0,0,0,0,.5,0;...                % phiB
            0,0,-1,0,0,0,0,1,0,0;...                 % dphiA
            0,0,0,-1,0,0,0,0,1,0];                   % dphiB
g_kvec = @(x)...
    [top_bending(x)*k_vec(1),...                          % top_bending
```

```
    bot_bending(x)*k_vec(2),...                                  % bot_bending
    axial(x)*k_vec(3),...                                        % axial happens
    dface_bending(x)/f*(k_vec(1)+k_vec(2)+k_vec(3)),...          % dface_bending
    web_bending(x)*k_vec(4),...                                 % web_bending
    web_bending2(x)*f*k_vec(4),...                               % web_bending2
    dweb_bending(x)/f*k_vec(4),...                               % dweb_bending
    dweb_bending2(x)*k_vec(4),...                                % dweb_bending2
    phiA_avg(x)*k_vec(5),...                                     % phiA
    phiB_avg(x)*k_vec(6),...                                     % phiB
    dphiA(x)/f*k_vec(5),...                                      % dphiA
    dphiB(x)/f*k_vec(6)];                                        % dphiB
g(1,:) = @(x) g_kvec(x)*g_matrix; % GRAD ---- Might need to change location
   of force
end
```

cal.m:

```
%% Average
a_avg = @(x) (a1(x) + a2(x))/2;
b_avg = @(x) (b1(x) + b2(x))/2;
phiA_avg = @(x) (phiA1(x) + phiA2(x))/2;
phiB_avg = @(x) (phiB1(x) + phiB2(x))/2;
psi_avg = @(x) (psi1(x) + psi2(x))/2;
%% Diffrenace
da = @(x) a2(x) - a1(x);
db = @(x) b2(x) - b1(x);
dphiA = @(x) phiA2(x) - phiA1(x);
dphiB = @(x) phiB2(x) - phiB1(x);
dpsi = @(x) psi2(x) - psi1(x);
%% Componets of Total Strain
top_bending = @(x) (db(x)+dphiA(x))*z;
bot_bending = @(x) (db(x)+dphiB(x))*z;
axial = @(x) (db(x)+dpsi(x))*z;
dface_bending = @(x) (b_avg(x)*z-da(x)*z*z)*12;
web_bending = @(x) phiB_avg(x)-phiA_avg(x);
dweb_bending = @(x) dphiB(x)-dphiA(x);
web_bending2 = @(x) (phiB_avg(x)+phiA_avg(x))/2 - psi_avg(x);
dweb_bending2 = @(x) (dphiB(x)+dphiA(x))/2 - dpsi(x);
```

Python Code Created by Dr. Srinivasa:
WCBeamConstrained2.py:

Listing C.1: Python example

```
# -*- coding: utf-8 -*-
Created on Wed Nov 11 15:54:27 2020@author:  Arun
import numpy as np
```

```
from scipy import optimize as opt
from scipy.optimize import LinearConstraint
import matplotlib.pyplot as plt
import ars_MYPLOT as myplt
from matplotlib.ticker import (MultipleLocator, FormatStrFormatter,
                               AutoMinorLocator)


E=29007547
Ltot=180  # total length of the beam # enter whatever
Lstar=1 # Charactersitic length for non-dimensionalization: put Ltot for non dim,  put 1 if you dont want non-dim
P0=5000/15   # force on the bar
Pstar=1 # Characterstic Force  for non-dimensionalization: put P0 for non-dim put 1 if you dont want non-dim
Lbar=Ltot/Lstar
L0=Lbar/9 # box length
tt=L0/10 # top thickness
tb=L0/10 # botton thickness
ts=L0/20 #
H=L0/2
EIt=E*tt**3/12
EIb=E*tb**3/12
Ew=E*H**2*(tt+tb)/4
print(EIt,EIb, Ew)
Estar=1 #Ew+EIt+EIb # totl
Ew/=Estar
EIt/=Estar
EIb/=Estar
# These are the additional terms
EIs=E*ts**3/12/Estar/H/L0
G1=EIt*12/L0**2
G2=EIb*12/L0**2



#EXT_F[6]=1

#_________HELPER FUNCTIONS FOR DRAWING_____________________________________
def draw_hermite(ax,curve_start ,curve_end,N=10, color='k'):
    xl,yl,ml=curve_start
    xr,yr,mr=curve_end
    dx=xr-xl
    x=np.linspace(xl,xr,N)
```

```
    t=(x-xl)/dx
    mlbar=ml*dx
    mrbar=mr*dx
    y=(3*t**2-2*t**3)*yr+(t**3-t**2)*mrbar+(3*(1-t)**2-2*(1-t)**3)*yl-((1-t)
       **3-(1-t)**2)*mlbar
    ax.plot(x,y,color=color,linewidth=2)

def draw_beam(ax,data,N=10,color='k'):
    for i in range(N_ELEMENTS):
        yl,ml,phi1l,phi2l,phi3l,yr,mr,phi1r,phi2r,phi3r=data[N_DOFS_PER_NODE
           *i:N_DOFS_PER_NODE*(i+2)]
        xl,xr=NODAL_LOCATIONS[i:i+2]
        draw_hermite(ax,[xl,yl,ml],[xr,yr,mr],N,color)
#
   ------------------------------------------------------------------------


#
   ------------------------------------------------------------------------


LT=Lbar  #total length
N_ELEMENTS=21
N_NODES=N_ELEMENTS+1
N_DOFS_PER_NODE=5
N_DOFS=N_DOFS_PER_NODE*N_NODES
data=np.zeros(N_DOFS)
NODAL_LOCATIONS=np.linspace(0,LT,N_NODES)
MATL=np.array([[EIt,EIb,Ew,EIs,G1, G2],]*N_ELEMENTS)
#MATL[:,1]*=100
EXT_F=np.zeros(N_ELEMENTS)
EXT_F[9]=P0/Pstar
#BOUNDARY  CONSTRAINTS
N_BCS=10

A=np.zeros([N_BCS,N_DOFS])
# A[0,0]=A[1,1]=A[2,1]=A[3,1]=A[4,-5]=A[5,-4]=A[6,-4]=A[7,-4]=1
# A[1,2]=A[2,3]=A[3,4]=A[5,-3]=A[6,-2]=A[7,-1]=-1
# A[0,0]=A[1,1]=A[2,1]=A[3,1]=1
# A[1,2]=A[2,3]=A[3,4]=-1
A[0,0]=1  # V(0)
A[1,1]=1
A[2,2]=1
A[3,3]=1
A[4,4]=1  # V(0)
A[5,-1]=1
A[6,-2]=1
```

```
A[7,-3]=1
A[8,-4]=1
A[9,-5]=1
#lin_const=LinearConstraint(A,[0,0,0,0,0],[0,0,0,0,0])
lin_const=LinearConstraint(A,[0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0])

#
    ------------------------------------------------------------------------

def compute_WCBeam_energy(data):
    # boundary conditions
    energy=0
    for i in range(N_ELEMENTS):

        f=EXT_F[i]
        xl,xr=NODAL_LOCATIONS[i:i+2]
        l=xr-xl
        a=1/l
        D=MATL[i,:]*l
        j=N_DOFS_PER_NODE*(i)
        k=N_DOFS_PER_NODE*(i+1)
        v_ave,m_ave,p0_ave,p1_ave,p2_ave=0.5*(data[k:k+N_DOFS_PER_NODE]+data[
            j:j+N_DOFS_PER_NODE])
        dv,dm,dp0,dp1,dp2=data[k:k+N_DOFS_PER_NODE]-data[j:j+N_DOFS_PER_NODE]

        bend0=(dm+dp0)*a
        bend1=(dm+dp1)*a
        bend2=(dm+dp2)*a
        dbend=(m_ave*a-dv*a*a)*12
        bend_s1=(p1_ave-p0_ave)
        dbend_s1=(dp1-dp0)
        bend_s2=((p1_ave+p0_ave)/2-p2_ave)
        dbend_s2=((dp1+dp0)/2-dp2)


        energy+=((bend0*bend0+dbend*dbend/12)*D[0]+
            (bend1*bend1+dbend*dbend/12)*D[1]+
            (bend2*bend2+dbend*dbend/12)*D[2]+
            (bend_s1*bend_s1+dbend_s1*dbend_s1/12+12*bend_s2*bend_s2+dbend_s2
                *dbend_s2)*D[3]+
            (p0_ave*p0_ave+dp0*dp0/12)*D[4]+
            (p1_ave*p1_ave+dp1*dp1/12)*D[5]
            )/2-f*v_ave

    return energy
```

```
126
127 #    ------------------------------------------------------------------------
128
129 def compute_WCBeam_gradient(data):
130     g_e=np.zeros(N_DOFS)
131     grad=np.zeros([13,10])
132     for i in range(N_ELEMENTS):
133
134         f=EXT_F[i]
135         xl,xr=NODAL_LOCATIONS[i:i+2]
136         l=xr-xl
137         a=1/l
138         D=MATL[i,:]*l
139         j=N_DOFS_PER_NODE*(i)
140         k=N_DOFS_PER_NODE*(i+1)
141         v_ave,m_ave,p0_ave,p1_ave,p2_ave=0.5*(data[k:k+N_DOFS_PER_NODE]+data[
                j:j+N_DOFS_PER_NODE])
142         dv,dm,dp0,dp1,dp2=data[k:k+N_DOFS_PER_NODE]-data[j:j+N_DOFS_PER_NODE]
143
144         # kinematical quantities that go into the
                energy_________________________
145         bend0=(dm+dp0)*a
146         bend1=(dm+dp1)*a
147         bend2=(dm+dp2)*a
148         dbend=(m_ave*a-dv*a*a)*12
149         bend_s1=(p1_ave-p0_ave)
150         bend_s2=((p1_ave+p0_ave)/2-p2_ave)
151         dbend_s1=(dp1-dp0)
152         dbend_s2=((dp1+dp0)/2-dp2)
153
154         #_________gradient of the energy with respect to the KE
                variables_______
155         de=([bend0*D[0],bend1*D[1],bend2*D[2],dbend/12*(D[0]+D[1]+D[2]),
156             bend_s1*D[3],12*bend_s2*D[3], dbend_s1/12*D[3], dbend_s2*D[3],
157             p0_ave*D[4],dp0*D[4]/12,p1_ave*D[5],dp1*D[5]/12,-f])
158         a2=a*a
159
160         #______gradient of the kinematics with respect to the
                DOFS_____________
161         grad[:,:]=[[   0,  -a,  -a,   0,   0,      0,   a,   a,   0,   0],
162                    [   0,  -a,   0,  -a,   0,      0,   a,   0,   a,   0],
163                    [   0,  -a,   0,   0,  -a,      0,   a,   0,   0,   a],
164                    [12*a2,6*a,   0,   0,   0, -12*a2, 6*a,   0,   0,   0],
```

```
                  [   0,   0, -0.5,0.5,   0,      0,   0,-0.5, 0.5,   0],
                  [   0,   0,0.25,0.25,-0.5,      0,   0,0.25,0.25,-0.5],
                  [   0,   0,  1,   -1,   0,      0,   0,  -1,   1,   0],
                  [   0,   0,-0.5,-0.5,   1,      0,   0, 0.5, 0.5,  -1],
                  [   0,   0, 0.5,   0,   0,      0,   0, 0.5,   0,   0],
                  [   0,   0,  -1,   0,   0,      0,   0,   1,   0,   0],
                  [   0,   0,   0, 0.5,   0,      0,   0,   0, 0.5,   0],
                  [   0,   0,   0,  -1,   0,      0,   0,   0,   1,   0],
                  [ 0.5,   0,   0,   0,   0,    0.5,   0,   0,   0,   0]]
        #
            ------------------------------------------------------------------

        g_e[j:j+2*N_DOFS_PER_NODE]+=de@grad

    return g_e

#
    ------------------------------------------------------------------------

#
    ------------------------------------------------------------------------

def compute_WCBeam_hessian(data,datadot):
    g_e=np.zeros(N_DOFS)
    grad=np.zeros([12,10])
    for i in range(N_ELEMENTS):
        xl,xr=NODAL_LOCATIONS[i:i+2]
        l=xr-xl
        a=1/l
        D=MATL[i,:]*l
        j=N_DOFS_PER_NODE*(i)
        k=N_DOFS_PER_NODE*(i+1)
        v_ave,m_ave,p0_ave,p1_ave,p2_ave=0.5*(datadot[k:k+N_DOFS_PER_NODE]+
            datadot[j:j+N_DOFS_PER_NODE])
        dv,dm,dp0,dp1,dp2=datadot[k:k+N_DOFS_PER_NODE]-datadot[j:j+
            N_DOFS_PER_NODE]

        # kinematical quantities that go into the
            energy_________________________
        # kinematical quantities that go into the
            energy_________________________
        bend0=(dm+dp0)*a
        bend1=(dm+dp1)*a
        bend2=(dm+dp2)*a
        dbend=(m_ave*a-dv*a*a)*12
```

```
200         bend_s1=(p1_ave-p0_ave)
201         dbend_s1=(dp1-dp0)
202         bend_s2=((p1_ave+p0_ave)/2-p2_ave)
203         dbend_s2=((dp1+dp0)/2-dp2)
204 
205         #_________gradient of the energy with respect to the KE
                variables_______
206         de=([bend0*D[0],bend1*D[1],bend2*D[2],dbend/12*(D[0]+D[1]+D[2]),
207             bend_s1*D[3],12*bend_s2*D[3], dbend_s1/12*D[3], dbend_s2*D[3],
208             p0_ave*D[4],dp0*D[4]/12,p1_ave*D[5],dp1*D[5]/12])
209         a2=a*a
210 
211         #______gradient of the kinematics with respect to the
                DOFS_____________
212         grad[:,:]=[[   0,  -a,  -a,   0,   0,      0,   a,   a,   0,   0],
213                    [   0,  -a,   0,  -a,   0,      0,   a,   0,   a,   0],
214                    [   0,  -a,   0,   0,  -a,      0,   a,   0,   0,   a],
215                    [12*a2,6*a,   0,   0,   0, -12*a2, 6*a,   0,   0,   0],
216                    [   0,   0, -0.5,0.5,   0,      0,   0,-0.5, 0.5,   0],
217                    [   0,   0,0.25,0.25,-0.5,      0,   0,0.25,0.25,-0.5],
218                    [   0,   0,  1,   -1,   0,      0,   0,  -1,   1,   0],
219                    [   0,   0,-0.5,-0.5,   1,      0,   0, 0.5, 0.5,  -1],
220                    [   0,   0, 0.5,   0,   0,      0,   0, 0.5,   0,   0],
221                    [   0,   0,  -1,   0,   0,      0,   0,   1,   0,   0],
222                    [   0,   0,   0, 0.5,   0,      0,   0,   0, 0.5,   0],
223                    [   0,   0,   0,  -1,   0,      0,   0,   0,   1,   0]]
224         #
                ------------------------------------------------------------------
                -

225         #
                ------------------------------------------------------------------
                ----

226         g_e[j:j+2*N_DOFS_PER_NODE]+=de@grad
227     return g_e
228 
229 #
    ------------------------------------------------------------------------
    ----

230 
231 
232 
233 # res = opt.minimize(compute_WCBeam_energy, np.zeros(N_DOFS), method='nelder
    -mead',
234 #                options={'xatol': 1e-8, 'disp': True,'maxfev':1e8})
235 
```

```
236
237 # res = opt.minimize(compute_WCBeam_energy, np.zeros(N_DOFS), method='BFGS',
        jac=compute_WCBeam_gradient,
238 #                    options={'gtol':1e-8, 'disp': True})
239
240 # # res = opt.minimize(compute_TBeam_energy, np.zeros(N_DOFS), method='
        Newton-CG', jac=compute_TBeam_gradient,
241 # #                      hessp=compute_TBeam_hessian,
242 # #                    options={'xtol':1e-8, 'disp': True})
243
244 res = opt.minimize(compute_WCBeam_energy, 0.1*np.random.rand(N_DOFS), method
        ='trust-constr',
245                    jac=compute_WCBeam_gradient,
246                    hessp=compute_WCBeam_hessian,
247                    constraints=lin_const,
248                  options={'xtol':1e-10, 'disp': True})
249
250 data[:]=res.x
251 max_y=np.max(data[::N_DOFS_PER_NODE])
252 min_y=np.min(data[::N_DOFS_PER_NODE])
253
254
255 fig,ax=plt.subplots(2,2)
256
257
258 myplt.set_myplot(ax[0,0],[0,LT],[min_y*0.9,max_y*1.1])
259
260 draw_beam(ax[0,0],data)
261
262
263 max_y=np.max(data[2::N_DOFS_PER_NODE])
264 min_y=np.min(data[2::N_DOFS_PER_NODE])
265
266 myplt.myplot(ax[0,1],NODAL_LOCATIONS,data[2::N_DOFS_PER_NODE],[0,LT],[min_y,
        max_y])
267
268
269
270 max_y=np.max(data[3::N_DOFS_PER_NODE])
271 min_y=np.min(data[3::N_DOFS_PER_NODE])
272
273 myplt.myplot(ax[1,0],NODAL_LOCATIONS,data[3::N_DOFS_PER_NODE],[0,LT],[min_y,
        max_y])
274
275
```

```

max_y=np.max(data[4::N_DOFS_PER_NODE])
min_y=np.min(data[4::N_DOFS_PER_NODE])

myplt.myplot(ax[1,1],NODAL_LOCATIONS,data[4::N_DOFS_PER_NODE],[0,LT],[min_y,
    max_y])
```

ars-MYPLOT.py

```
# -*- coding: utf-8 -*-
Created on Wed Sep 9 11:10:26 2020@author:  Arun

import numpy as np
from matplotlib import pyplot as plt
from scipy.linalg import toeplitz
from scipy.linalg import solve_banded
from itertools import permutations
from random import sample
import scipy.sparse
from matplotlib.ticker import (MultipleLocator, FormatStrFormatter,
                               AutoMinorLocator)

#Get the color-wheel
Nlines = 200
color_lvl = 8
rgb = np.array(list(permutations(range(0,256,color_lvl),3)))/255.0
colors = sample(rgb.tolist(),Nlines)

#Get the color-wheel
Nlines = 200
color_lvl = 8
rgb = np.array(list(permutations(range(0,256,color_lvl),3)))/255.0
colors = sample(rgb.tolist(),Nlines)

def myplot(ax,x,y,xrange=[0,1], yrange=[0,1],color='k',linestyle='-',xaxis='
    ',yaxis='',legend=''):
    ax.plot(x,y,color=color,linewidth=2)
    #plt.axis('equal')
    #plt.set_axisbelow(True)
    ax.set_xlim(xrange)
    ax.set_ylim(yrange)
    #ax.set_minorticks_on()
    dx=(xrange[1]-xrange[0])/10
    ddx=dx/5
    dy=(yrange[1]-yrange[0])/10
    ddy=dy/5
```

```
    ax.xaxis.set_major_locator(MultipleLocator(dx))
    ax.xaxis.set_minor_locator(MultipleLocator(ddx))
    ax.yaxis.set_major_locator(MultipleLocator(dy))
    ax.yaxis.set_minor_locator(MultipleLocator(ddy))
    ax.grid(which='major', linestyle=linestyle, linewidth='0.5', color='k')
# # Customize the minor grid
    ax.grid(which='minor', linestyle=':', linewidth='0.5', color='grey')
    ax.set_xlabel(xaxis)
    ax.set_ylabel(yaxis)
    ax.set_title(legend)

    return

def set_myplot(ax,xrange=[0,1], yrange=[0,1],linestyle='-',xaxis='',yaxis=''
    ,legend=''):
    #ax.plot(x,y,color=color,linewidth=2)
    #plt.axis('equal')
    #plt.set_axisbelow(True)
    ax.set_xlim(xrange)
    ax.set_ylim(yrange)
    #ax.set_minorticks_on()
    dx=(xrange[1]-xrange[0])/10
    ddx=dx/5
    dy=(yrange[1]-yrange[0])/10
    ddy=dy/5
    ax.xaxis.set_major_locator(MultipleLocator(dx))
    ax.xaxis.set_minor_locator(MultipleLocator(ddx))
    ax.yaxis.set_major_locator(MultipleLocator(dy))
    ax.yaxis.set_minor_locator(MultipleLocator(ddy))
    ax.grid(which='major', linestyle=linestyle, linewidth='0.5', color='k')
# # Customize the minor grid
    ax.grid(which='minor', linestyle=':', linewidth='0.5', color='grey')
    ax.set_xlabel(xaxis)
    ax.set_ylabel(yaxis)
    ax.set_title(legend)

    return
```

APPENDIX D

ANSYS SIMULATIONS

D.1 Data Gathering

A total of 8805 element 29893 nodes were used. All elements were square 2D elements with a size of .25 inches and were uniform thought out. See Figure 3.7 for mesh.

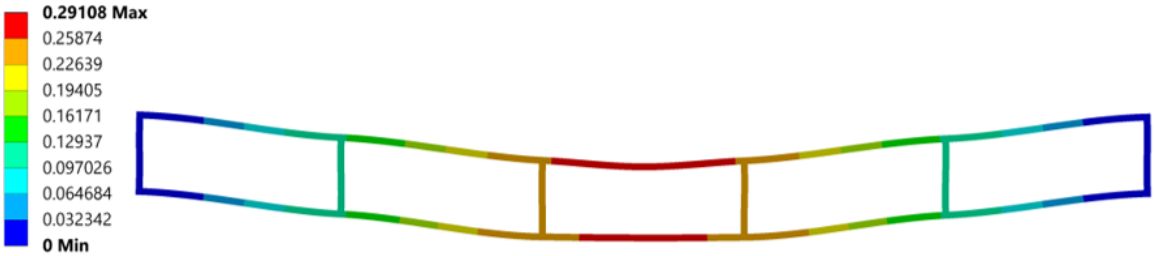


Figure D.1: Front View of Total Deflection (Unit: in) for SVD Structure

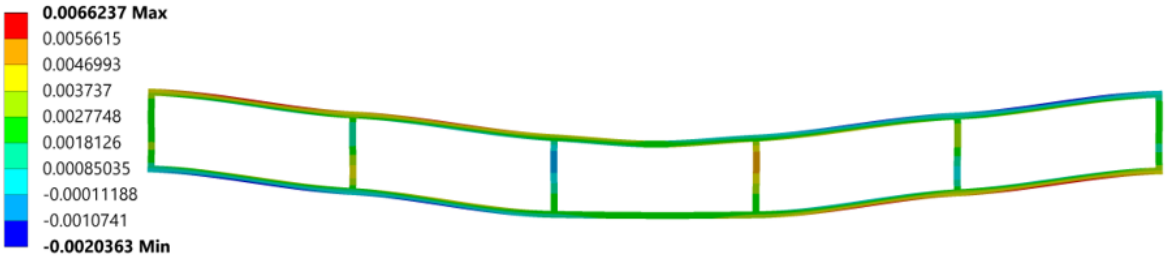


Figure D.2: Front View of X-Axis Deflection (Unit: in) for SVD Structure

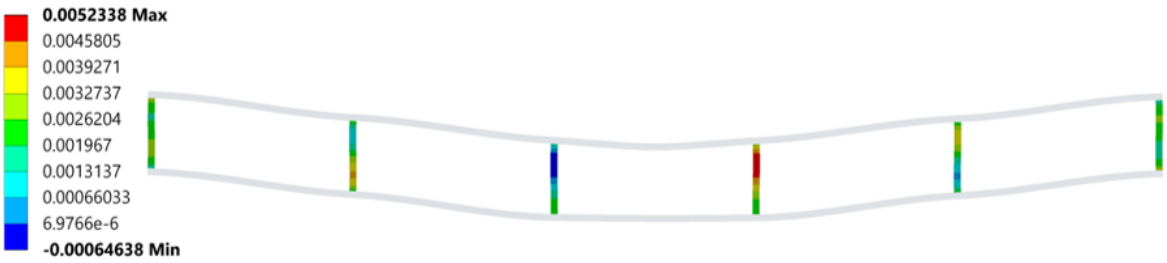


Figure D.3: Front View of X-Axis Deflection, Webs Only (Unit: in) for SVD Structure

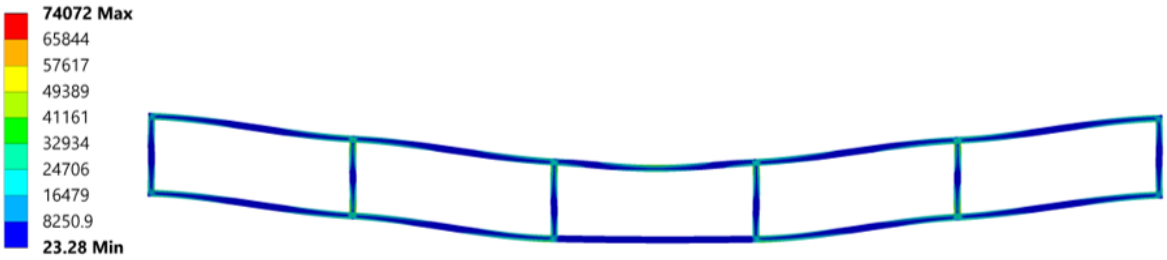


Figure D.4: Front View of Von Mises Stress (Unit: psi) for SVD Structure

D.2 Comparing Results

A total of 777600 element 3526011 nodes were used. All elements were square 3D elements with a size of .25 inches and were uniform thought out.

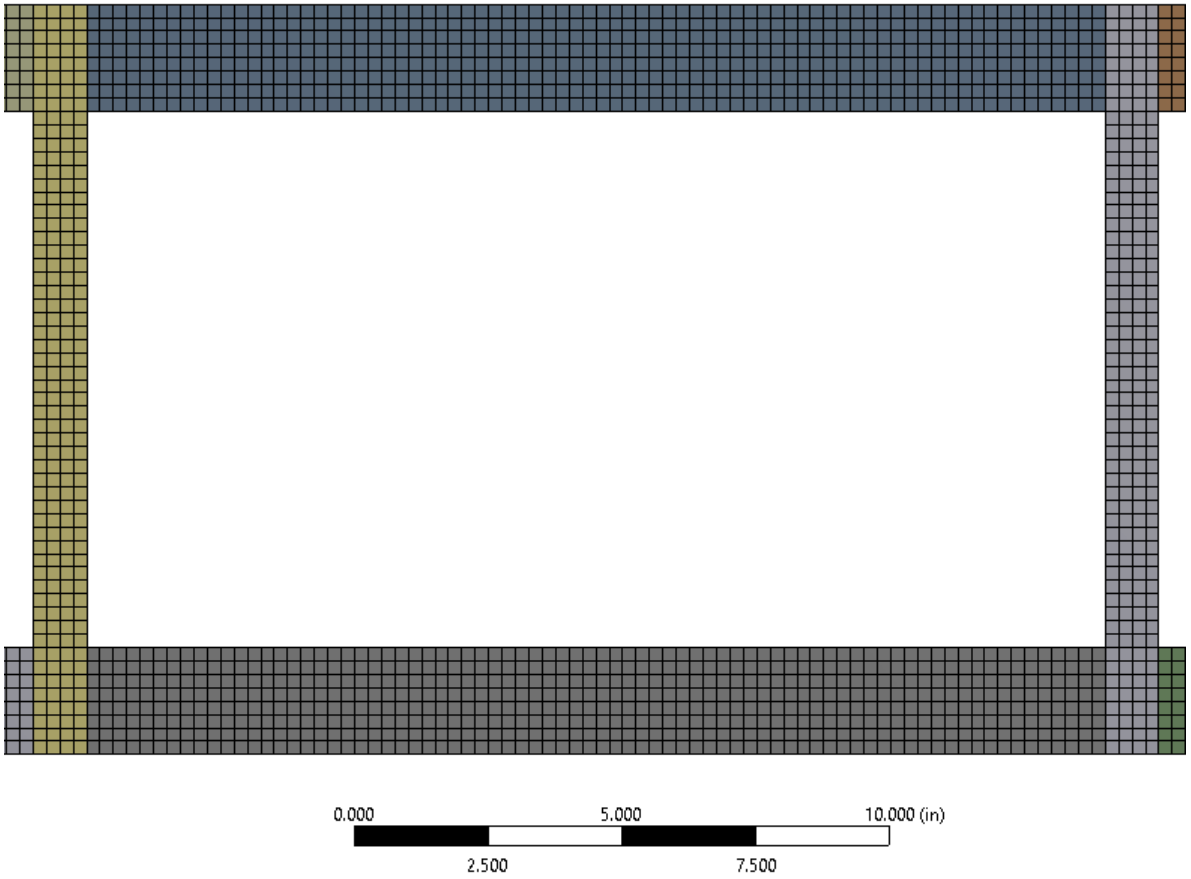


Figure D.5: Front View of Mesh Size for Simulation to Compare Results of Beam Model

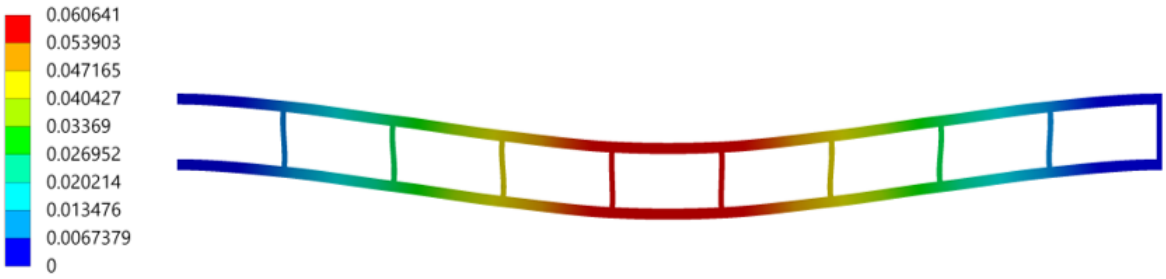


Figure D.6: Front View of Total Deflection (Unit: in) to Compare Results of Beam Model

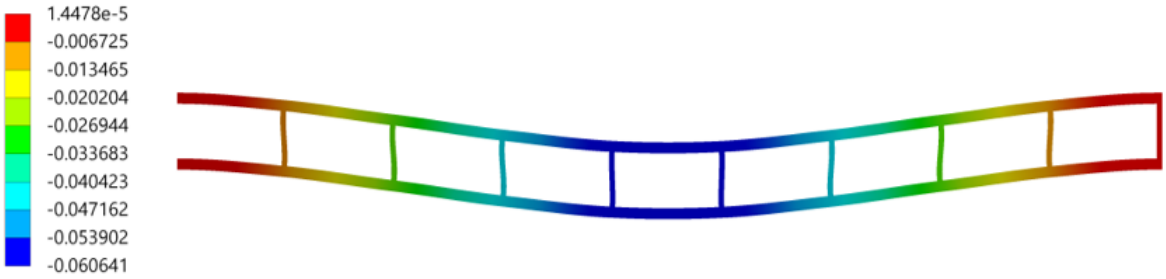


Figure D.7: Front View of Vertical Deflection (Unit: in) to Compare Results of Beam Model

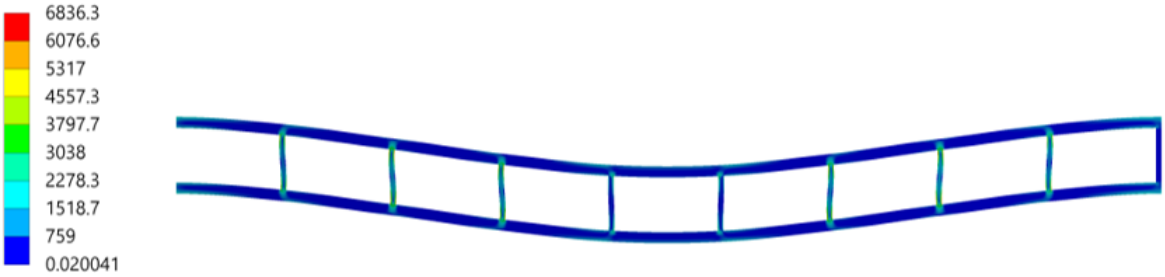


Figure D.8: Front View of Von Mises Stress (Unit: psi) to Compare Results of Beam Model

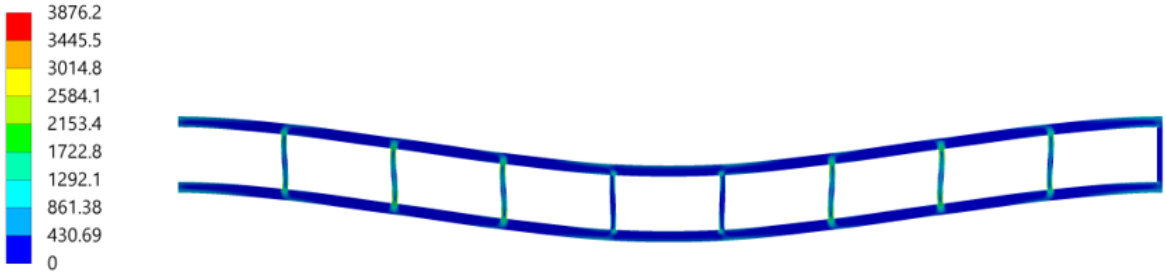


Figure D.9: Front View of Shear Stress Results (Unit: psi) to Compare Results of Beam Model

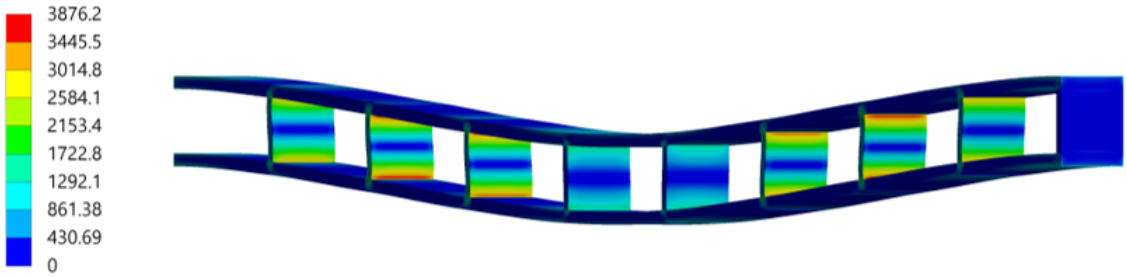


Figure D.10: Skewed View of Shear Stress Results (Unit: psi) to Compare Results of Beam Model