Modeling 3D Fiber Reinforced Foam Core Sandwich Structures Using a Multi-Scale Finite Element Approach

by

Zachary T. Kier

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
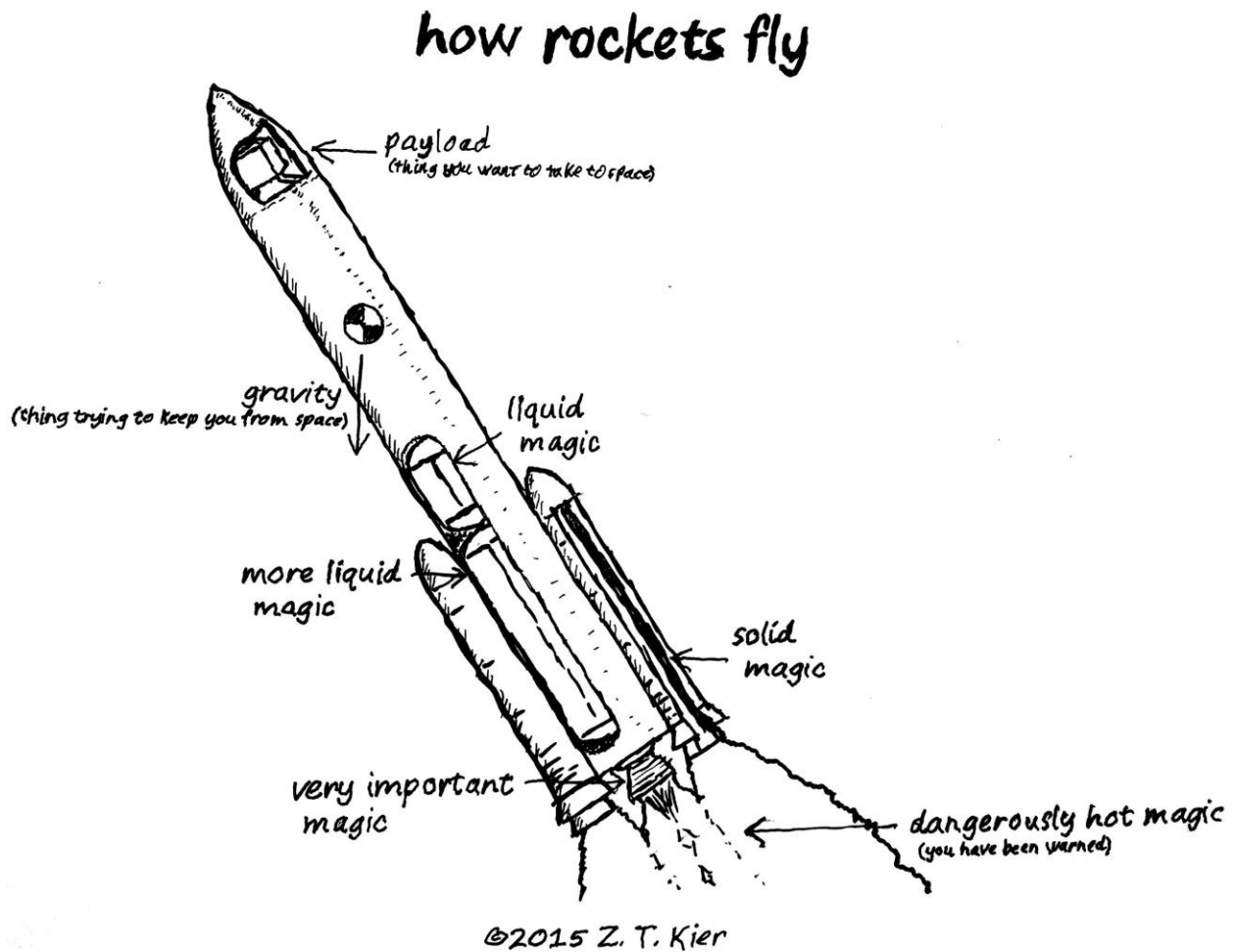in the University of Michigan
2015

Doctoral Committee:
      Professor Anthony M. Waas, Chair
      Senior Engineering Specialist Vinay K. Goyal, The Aerospace Corporation
      Assistant Professor Ann E. Jeffers
      Engineering Specialist Jacob I. Rome, The Aerospace Corporation
      Associate Professor Veera Sundararaghavan

how rockets fly

payload
(thing you want to take to space)

gravity
(thing trying to keep you from space)

liquid magic

more liquid magic

solid magic

very important magic

dangerously hot magic
(you have been warned)

©2015 Z. T. Kier

For Lyndsey

ACKNOWLEDGMENTS

First, I would like to thank my wife, Lyndsey for all her love and support. Without your tolerant encouragement through my late nights and increasing grey hair none of this would be possible. Second, I would like to thank my family for their continued support throughout my education and my life, always encouraging me to venture out into the unknown. I would like to thank Prof. Anthony Waas for providing the inspiration and backing to conduct my research and the independence to find a research area that I can be passionate about. Your contagious enthusiasm for understanding advanced materials drove me to pursue this research and I will forever be grateful for your guidance throughout this process.

I am grateful for Prof. Anthony Waas, Prof. Ann Jeffers, Prof. Veera Sundararaghavan, Dr. Vinay Goyal, and Dr. Jacob Rome for serving on my dissertation committee. Your guidance and feedback on the research has been invaluable.

I want to thank the members of the Aerospace Department Administrative Staff for helping with all the paperwork and logistics. I want to thank the Aerospace Department Technical Staff for their help running the labs as well as the often needed witty banter: Tom Griffin, Chris Chartier, Terry Larrow, Dave McLean, Martin Stenzel, and Aaron Borgman. A very special thanks to my graduate coordinator Denise Phelps; your continuous, selfless, and relentless dedication to your student's success is like a guiding light in the darkness. I owe a large part of my success to your sustaining support.

I am grateful to my many colleagues from The Aerospace Corporation that helped in various stages of this research including the manufacture, testing, and inspection of the composite panels necessary to support the effort: Jacob Rome, Vinay Goyal, Chad Foerster, John Klug, Pete Pollock, James Tuck-Lee, Thomas Albright, Jim Nokes, Neil Ives, Patrick Schubel, Yong Kim, and Gary Steckel.

I want to extend a very special thanks to Dhruv Patel from The Aerospace Corporation. Many dissertations focused on new materials have limited test data to help validate their methods

Finally, I would like to thank Mrs. Brewer; I can't imagine where my life would be without your encouragement. Your love for science inspired my own.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ACRONYMS

3DFRFC	Three-Dimensionally Reinforced Foam Core
BDCB	Bonded Double Cantilever Beam
BENF	Bonded End Notched Flexure
CCM	Concentric Cylinder Model
cUDG	clamped-Uniform Deformation Gradient (Homogenized Model)
DCB	Double Cantilever Beam
DCZM	Discrete Cohesive Zone Method
DIC	Digital Image Correlation
DoF	Degree of Freedom
ELS	End-loaded Split
ENF	End Notched Flexure
EWC	Edgewise Compression
FWC	Flatwise Compression
FWT	Flatwise Tension
IM7/8552	Carbon/Epoxy prepreg system
MicroCT	Micro X-ray Computed Tomography (μCT)
NDE	Non-destructive Evaluation
PFA	Progressive Failure Analysis
PTFE	Polytetrafluoroethylene (i.e. Teflon®)
RUC	Repeating Unit Cell (Geometric Unit)
RVE	Representative Volumetric Element
SCA	Smeared Crack Analysis
SERR	Failure variable form Abaqus user element (failed if SERR=1)
UDCB	Unsymmetrical Double Cantilever Beam
UENF	Unsymmetrical End Notched Flexure
VCCT	Virtual Crack Closure Technique
Voxel	Volumetric Pixel

LIST OF SYMBOLS

α	Pin inclination angle
β	Global in-plane rotation angle
δ_L	deviation in lateral/transverse pin spacing
δ_S	deviation in pin spacing
δ_α	deviation in pin inclination angle
ε_{11}	Axial strain in 1-direction
$\overline{\varepsilon}_{11}$	Average axial strain in 1-direction
γ_{12}	Shear strain in 1-2 plane
$\overline{\gamma}_{12}$	Average shear strain in 1-2 plane
$\vec{U}_{x1i}$	Displacement vector for ith node on surface X1
$\vec{U}_{Rx}$	Displacement vector for reference node on X axis
$\vec{\kappa}_{x1i}$	Rotation vector for ith node on surface X1
A_{x1_i}	Nodal weight for ith node on surface X1
$\vec{F}_{Rx}$	Reaction force vector for reference node on X axis
$\overline{\sigma}_{11}$	Average axial stress in 1-direction
$\overline{\tau}_{12}$	Average shear stress in 1-2 plane
S	Pin spacing
L	Lateral pin spacing
L_e	Effective edge length
t_c	Thickness of Core
L_x	RVE length along X axis
z	height of layer through thickness
P	Applied load
Δ	Applied load-point displacement
$[\overline{C}]$	Volume average 3D stiffness matrix

$[\overline{S}]$	Volume average 3D compliance matrix
E_{11}	Axial modulus in 1-direction
ν_{12}	Poisson's ratio in 2-direction for load applied in 1-direction
G_{12}	Shear modulus in 1-2 plane
N_x	Applied force for classical lamination theory in x-direction
M_x	Applied moment for classical lamination theory on x-normal surface
$\begin{bmatrix} A & B \\ B & D \end{bmatrix}$	Composite plate stiffness
ε_x°	Axial strain in x-direction about z=0 plane
$\varepsilon_{x_{Upper}}^\circ$	Axial strain in x-direction about z=0 plane in BDCB upper arm
$\varepsilon_{x_{Lower}}^\circ$	Axial strain in x-direction about z=0 plane in BDCB lower arm
$B'_{11_{Upper}}$	Composite plate compliance term for BDCB upper arm
$B'_{11_{Lower}}$	Composite plate compliance term for BDCB lower arm

ABSTRACT

An engineering challenge of composite sandwich structures is quantifying their ability to tolerate damage, particularly in launch vehicles and spacecraft, where mission assurance is critical. Recently, there has been a development of new core materials that may alter their damage tolerance through the use of a three-dimensional, truss-like network of reinforcing fibers inside a lightweight foam core. This research focuses on the testing and developing a multi-scale approach to model 3D Fiber Reinforced Foam Core (3DFRFC) sandwich composites with defects across typical operating temperatures.

Details of the 3DFRFC microstructure are examined through extensive use of microCT scans. The architecture measured directly from the microstructure was utilized to develop a parametric code for generating detailed embedded element models. These models were used for direct detailed modeling of fracture, edgewise compression, flatwise tension, flatwise compression, and three point bending test specimens. The embedded element models were also used as the cornerstone of a new method of developing effective homogenized properties for 3DFRFCs based on the details of the microstructure. Improved homogenization techniques developed by including the local interaction between the facesheet and the core are also included. Part of this required the development of a generalized six degree-of-freedom periodic boundary condition code which is included in the appendix of the dissertation.

Additionally, the design, development, and initial failure of an interface fracture test for 3DFRFCs is presented. The understanding gained by using Digital Image Correlation on the failed tests allowed for a different approach to be utilized in designing a new bonded double cantilever beam specimen for testing the Mode I fracture of a 3DFRFC sandwich structure. This method resulted in a successful interface fracture test. The bonded DCB specimens exhibited relatively smooth crack propagation and produced G_{Ic} values similar to honeycomb sandwich structures and significantly higher than comparable foam structures.

A full fabrication, testing, and evaluation of 3DFRFC specimens with differing sizes of facesheet-to-core interface debonds is also presented. The analysis methods presented were able to predict the failure load and modes within 5%. The 3DFRFC proved to be tolerant to the

presence of facesheet to core debonds with only the largest debond demonstrating a statistically significant reduction of 22%.

Finally, a detailed investigation of the through thickness behavior of a 3DFRFC composite under ambient and cold conditions is presented. This includes detailed microstructure modeling of the different loading configurations, modeling of thermal stresses, identification of failure modes, and a thorough study of the effects of discrete specimen size and edge effects. MicroCT interrogation of tested specimens was then used to confirm the modes of failure in the tested specimens. The 3DFRFC specimens demonstrated better through thickness ambient performance than unreinforced cores of comparable density: >30% increase in tension, >100% increase in compression, and >5% increase in shear. The 3DFRFC's also demonstrated relatively small reductions in strength at cold temperatures: <2% reduction compression, <5% reduction tension, 23% reduction shear.

The investigation into the performance of 3DFRFC composite structures highlights the robust behavior of the structure to cold environments while underscoring the importance of loading direction on the structural response of these highly orthotropic composites. Future efforts will be focused on incorporating the detailed structural effects of the 3DFRFC microstructure into system level models, providing increased confidence in the design of structures with reinforced foam cores without requiring micromechanics-based detailed modeling.

CHAPTER 1

Introduction

1.1 Motivation

Sandwich composites offer key advantages in automotive and aerospace applications including reduction in weight over metals currently used. A current engineering challenge of utilizing sandwich composite structures is quantifying their ability to tolerate damage, particularly in launch vehicles and spacecraft, where mission assurance is critical to mitigating cost from loss or failure, technological set-backs, and potential risk to human life. The strength of sandwich composites can be reduced through many mechanisms, including impact damage, embedded foreign objects, use of poor bonding agents, or surface preparation issues.

1.2 3D Fiber Reinforced Foam Core

Recently, new core materials have been developed that have the potential to affect the damage tolerance of sandwich composites. One class of core material being considered may alter its damage tolerance through the use of three-dimensional, truss-like networks of reinforcing fibers inside a lightweight foam core. This overall class of 3D Fiber Reinforced Foam Core (3DFRFC) is defined to be quite broad and encompasses a substantial design envelope of core material with freedom for the tailorability of the effective core properties. The design space of 3DFRFCs can be explored through the selection of the base foam (material, density, thickness), the geometry of the truss structure, the type of reinforcing fibers used (carbon, glass, Kevlar®, Spectra®, etc.), and the method of manufacture (resin infusion, prepreg fiber insertion). Examples of this emerging class of core materials include NidaFusion [1,2], TYCOR® [3], and K-Cor® [4], Figure 1.1.
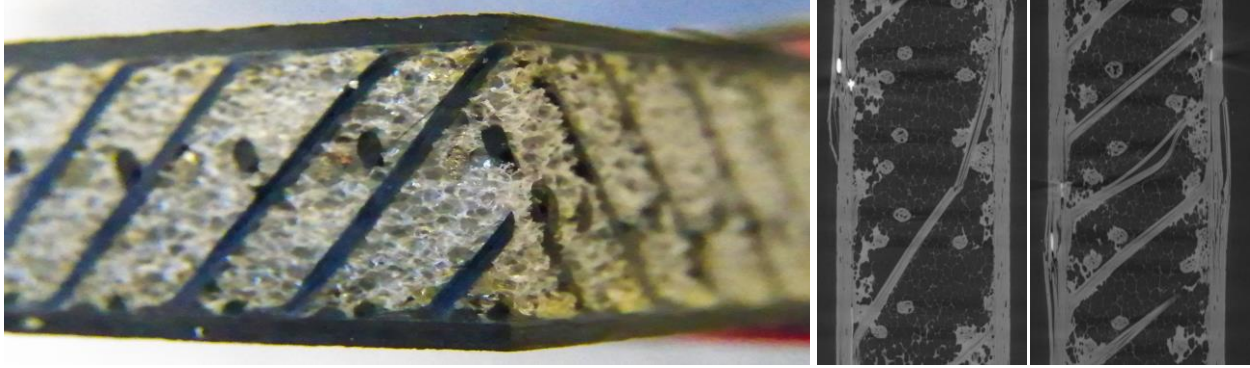
Figure 1.1. A 3D fiber reinforced foam core, left, and microCT scans of failed reinforcement, right. (Not to scale. Images used with the permission of The Aerospace Corporation.)

The truss structure of the 3D fiber network provides added paths for load transfer and acts to impede crack propagation within the foam core. The stiffness and strength of these three-dimensionally reinforced sandwich composites become challenging to predict as a result of the added load paths. In addition, the relatively coarse architecture of 3DFRFCs can lead to local variations due to the interactions of the facesheet and the reinforcements within the core and can cause challenges in quantifying the strength using standard coupons due to free-edge effects. Prior research has been limited to non-standard test specimens and z-pinned sandwich structures (reinforcement orthogonal to facesheets) [5–10]. Z-pinned sandwich structures can be thought of as a specialized case of the 3DFRFC architecture where the inclination angle of the reinforcement is zero.

1.3 Research Objectives

The goals attained by this research effort include:

- Development of periodic microstructure model to obtain homogenized core properties.
- Development of test method to measure the fracture properties of the facesheet-to-core adhesive bonding interface.
- Predict and experimentally verify the effect of facesheet-to-core debonds.
- Predict and verify the effect of specimen size on the effective structural properties.
- Experimentally evaluate the effect of cold temperatures on the through thickness failure of 3DFRFCs.
- Determine viability of 3DFRFCs to replace standard foam cores in space structures.

1.4 Thesis Organization

Much of the information contained in this dissertation is a collection of past or planned publications; however effort has been made to reduce the amount of repetition between chapters for the sake of brevity. While all of the chapters are inherently interconnected it is possible to read just the chapter of interest.

- Chapter 2 discusses the methods utilized to develop detailed models of the 3DFRFC microstructure that will be subsequently utilized to investigate the discrete behavior of the experimental investigations in the subsequent chapters. The fidelity of the models used is largely a function of what is being investigated and the detail required.

- Chapter 3 discusses the design and development of interface fracture tests to allow for measurement of the fracture properties of the facesheet-to-core bonding interface in 3DFRFCs. Additional challenges arose with the initial design and those details are included in the chapter as it uncovered additional design considerations for testing fracture in 3DFRFCs.

- Chapter 4 details a thorough investigation into the effect of facesheet-to-core delaminations subject to edgewise compression loading.

- Chapter 5 discusses the through thickness failure of the 3DFRFC. This includes discussion of the inherent size dependent behavior that results from the number of repeating unit cells present the specimen.

- Chapter 6 extends the investigation presented in Chapter 5 to include the cold temperature performance of a 3DFRFC sandwich structure. Chapter 6 contains the most overlap as it was published in publication #2 on following page with minor changes for continuity within the dissertation and is designed to be completely independent.

- Chapter 7 includes some suggestions for areas of continued research into 3DFRFCs.

1.5 Significant Contributions

Overall the research demonstrated the viability of 3DFRFCs for use in space structures whilst developing the engineering tools and methods necessary to support ongoing design and future use. An overview of the significant contributions contained in the thesis:

- Development of a hierarchical multi-scale method and attained effective homogenized core properties that incorporate the localized interactions between the composite facesheet and truss-core microstructure.
- Automated generation of reinforcement geometry for modeling arbitrarily large 3DFRFC sandwich panels.
- Predicted and experimentally verified increased tolerance of 3DFRFC structures to facesheet-to-core debonds subjected to edgewise compression loading.
- Predicted and experimentally verified free edge effects in 3DFRFC structures.
- Experimentally evaluated the through thickness performance of 3DFRFC structures demonstrating increased performance compared to a standard foam core of the same density.
- Development of a successful Mode I fracture test of the facesheet-to-core interface in 3DFRFC sandwich structures. Results indicate a significant increase in resistance to crack propagation as compared to unreinforced foam cores.

1.6 Publications

The following related publications were available at the time of the dissertation defense:

1. Kier, Z. T., Waas, A. M., Rome, J. I., Goyal, V. K., Patel, D., and Steckel, G., "Through-thickness Failure of 3D Fiber Reinforced Foam Core Sandwich Structures," SAMPE Journal, vol. 50, 2014, pp. 32–38.
2. Kier, Z. T., Patel, D. N., Goyal, V. K., Rome, J. I., Steckel, G. L., and Waas, A. M., "Response of 3D Fiber Reinforced Foam Core Sandwich Structures at Cold Temperatures," 29th American Society for Composites Conference, San Diego, CA: 2014, p. 624.
3. Rome, J. I., Goyal, V. K., Patel, D. N., and Kier, Z. T., "Foam Heat Treatment and Its Effects on Strength of Sandwich Composites," 29th American Society for Composites Conference, 2014, p. 627.
4. Kier, Z. T., Waas, A. M., Rome, J. I., Goyal, V. K., Patel, D., and Steckel, G., "Through-thickness Failure of 3D Fiber Reinforced Foam Core Sandwich Structures," SAMPEtech 2014, Seattle, WA: 2014.

5. Kier, Z. T., Waas, A. M., Foerster, C., Rome, J., and Goyal, V. K., "Effects of Ply Stacking Sequence in 3D Fiber Reinforced Foam Core Sandwich Structures with Defects," 55th AIAA/ASMe/ASCE/AHS/SC Structures, Structural Dynamics, and Materials Conference, 13-17 January 2014, National Harbor, Maryland, AIAA 2014-0504.
6. Kier, Z. T., Waas, A. M., Rome, J. I., and Goyal, V. K., "Specimen Size and Effective Compressive Stiffness of 3D Fiber Reinforced Foam Core Sandwich Structures," 28th Annual Technical Conference of the American Society for Composites, Boston, MA: 2013.
7. Kier, Z. T., and Waas, A. M., "Determining Effective Properties of 3D Fiber Reinforced Foam Core Sandwich Structures (US Only)," SAMPE 2013, Long Beach, CA: 2013.
8. Kier, Z. T., and Waas, A. M., "Determining Effective Interface Fracture Properties of 3D Fiber Reinforced Foam Core Sandwich Structures," 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2013.
9. Kier, Z. T., Waas, A. M., Rome, J. I., Goyal, V. K., Schubel, P., Steckel, G., Patel, D., and Kim, Y., "Modeling Failure of 3D Fiber Reinforced Foam Core Sandwich Structures with Defects," 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2012, pp. 1–11.
10. Kier, Z. T., Rome, J. I., Goyal, V. K., Schubel, P., Steckel, G., Patel, D., and Kim, Y., "Predicting Strength Reduction of Sandwich Structures with Interfacial Debonds (Presentation Only)," 25th American Society for Composites Conference, Dayton, OH: 2010.

CHAPTER 2
3DFRFC Microstructure Modeling

2.1 Introduction

The focus of this research is a 3D fiber reinforced core (3DFRFC) composite sandwich structure that consists of a truss network of carbon composite beams in lightweight foam, Figure 2.1. This chapter focuses on developing general modeling methods and tools for this material.

Figure 2.1. Sandwich composite made with a 3D fiber reinforced foam core.
(Not to scale.)

Composite materials inherently bridge multiple length scales. Some materials such as concrete or asphalt consist of relatively large constituents and provide an easily relatable example of these length scales. From a distance these materials appear homogeneous but if you look at a broken piece of concrete you can easily see the discrete pieces of aggregate (10-20mm) contained within the cement. Upon closer inspection you may be able to see even smaller sand particles (<1mm) within the cement between the larger aggregate. Similarly, 3DFRFC sandwich structures can be investigated at many length scales, ranging from the atomistic level up to large scale launch vehicle components that could be as large as 10m (32.8 ft), the diameter for the

Block 2 fairing on NASA's Space Launch System [11]. There are multiple ways to look at these scales. In the broadest sense multi-scale modeling refers to any method that takes the information from one length scale and uses it to inform the behavior of another. One of the simplest and most common forms is hierarchical multi-scale modeling where the effective constituent behavior at a smaller scale is used to formulate the behavior of a larger scale. This may be performed multiple times, over various lengths to incorporate the behavior from several subscales into the global structure. Conversely, stresses from a larger, global model can be passed to a smaller scale model to investigate the localization or constituent level behavior due to the far field stresses. Coupling of two or more of these scales is also possible and is referred to as concurrent multi-scale modeling; although this comes with additional computational costs [12].

Proper context for the analysis of 3DFRFC structures can be given by going backward through the hierarchical multiscale analysis flowchart in Figure 2.2; zooming in progressively further into the details of the structure. A launch vehicle or satellite can be thought of as a complex series of large scale structural components. These components may be on the order of 1m-10m and are often modeled using shells. The shells can be divided into a layered continuum on the order of 10-30mm. For the 3DFRFC, the core of this layered structure is actually made of the repeating composite truss structure embedded within lightweight foam, Figure 2.3. This discrete microstructure model can be further subdivided into the individual truss members for the core and textile composite tows for the facesheets. These individual members (single pin/tow) can be viewed locally as transversely isotropic unidirectional composites of fiber and matrix. While it is possible to further subdivide the fiber and matrix down to the atomistic level, for this work we stop at the fiber/matrix scale using the concentric cylinder model to attain the transversely isotropic tow properties [13,14].
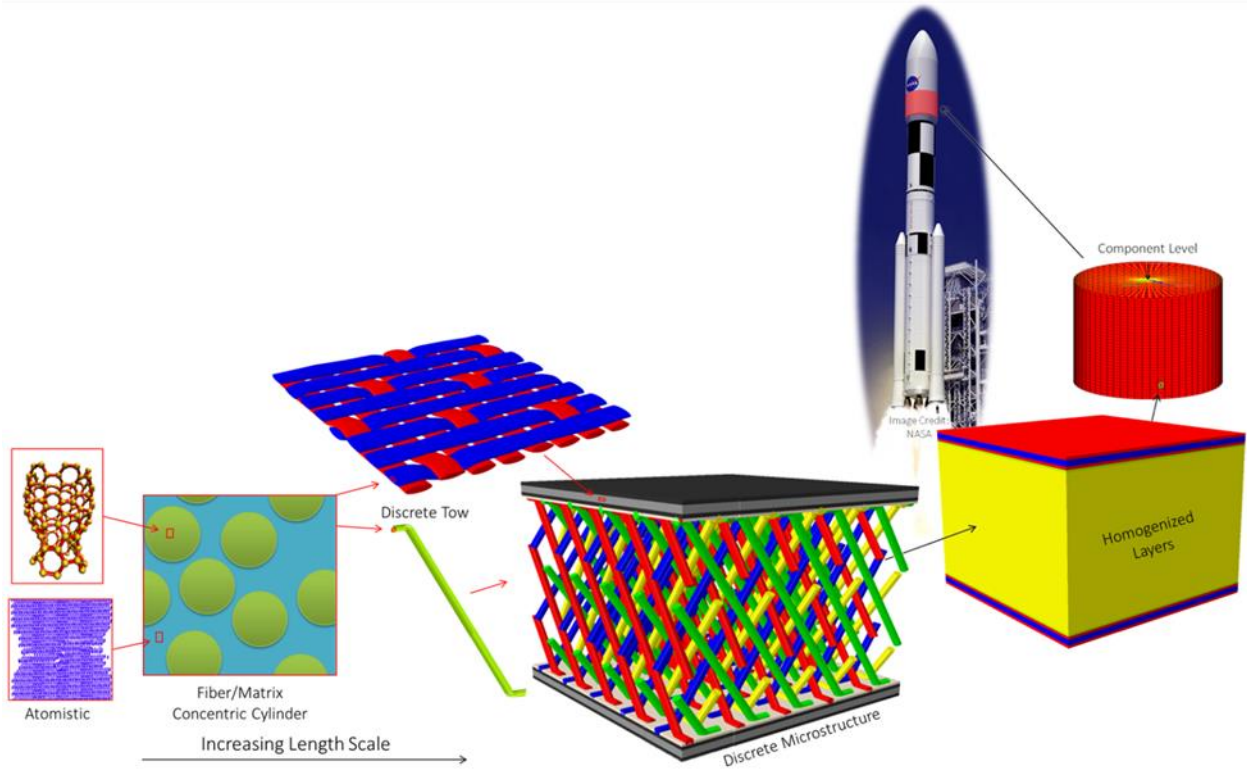
Figure 2.2. Illustration of length scales for hierarchical multiscale modeling in 3DFRFC structures.
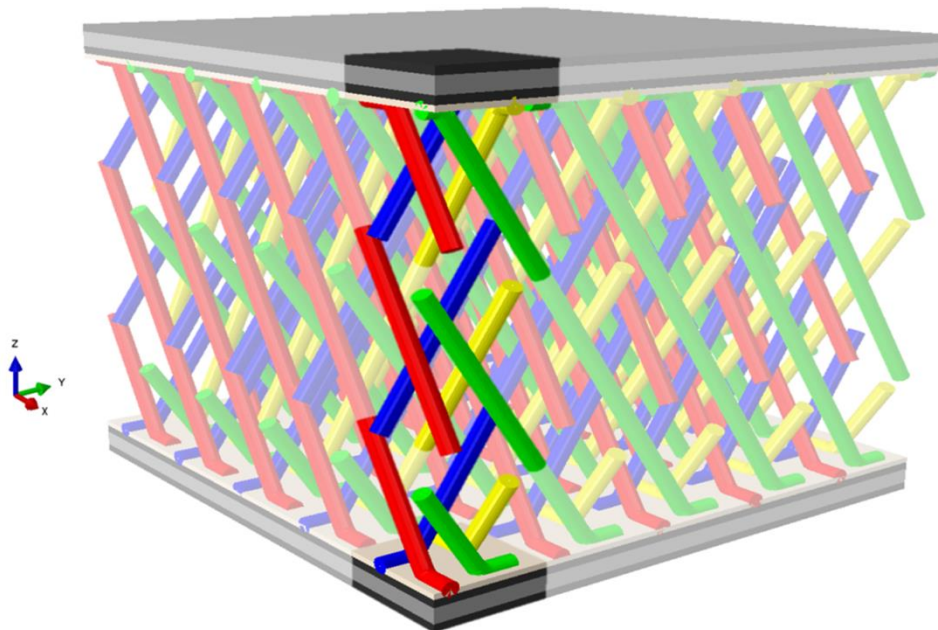
Figure 2.3. Model of 3DFRFC sandwich composite highlighting geometric repeating unit cell. (Not to scale. Foam removed for clarity)

2.2 MicroCT of 3DFRFC Microstructure

Micro X-ray Computed Tomography (MicroCT) was used to interrogate the details of the 3DFRFC composite sandwich structures after final fabrication. MicroCT allows for high resolution visualization of the inside of a specimen or material without disturbing its internal structure, unlike sectioning which can damage or alter the internal structures being measured. All microCT scans in this dissertation were conducted by The Aerospace Corporation and post processed using the open source image processing package Fiji [15,16]. Resolutions for the scans range from 15µm/voxel to 100µm/voxel depending on the size of the sample and the equipment used. An example of a raw x-ray image of the 3DFRFC is given in Figure 2.4 showing the network of reinforcing pins. A 3D reconstruction generated from this microCT data with most of the foam removed is shown in Figure 2.5.
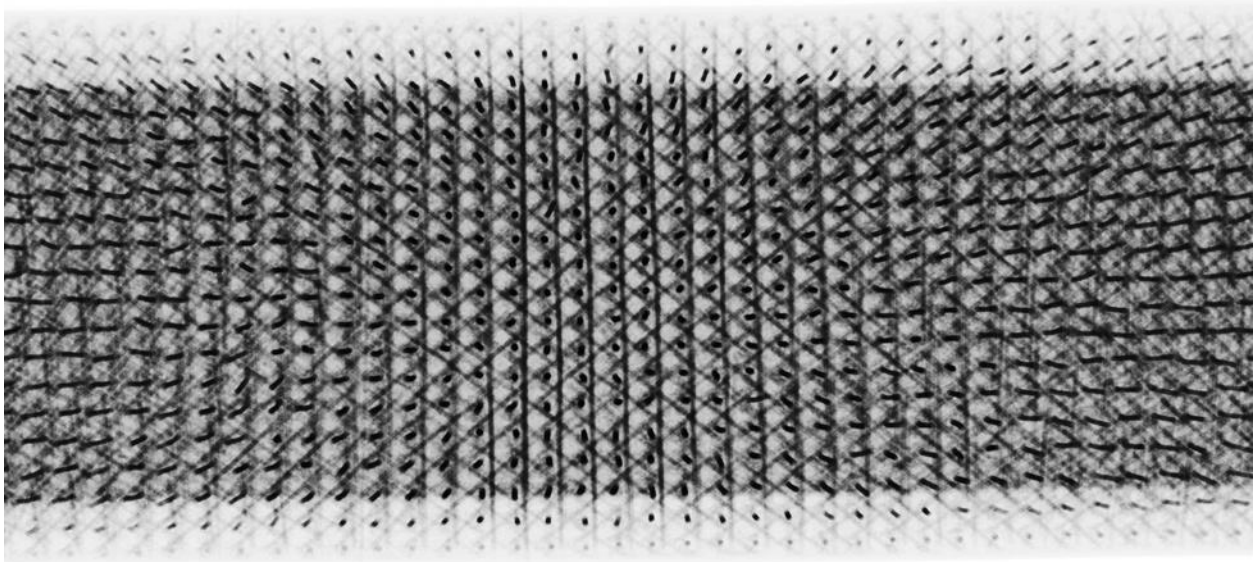
Figure 2.4. X-ray image of a 3DFRFC sandwich composite.
(Not to scale. Image used with the permission of The Aerospace Corporation.)
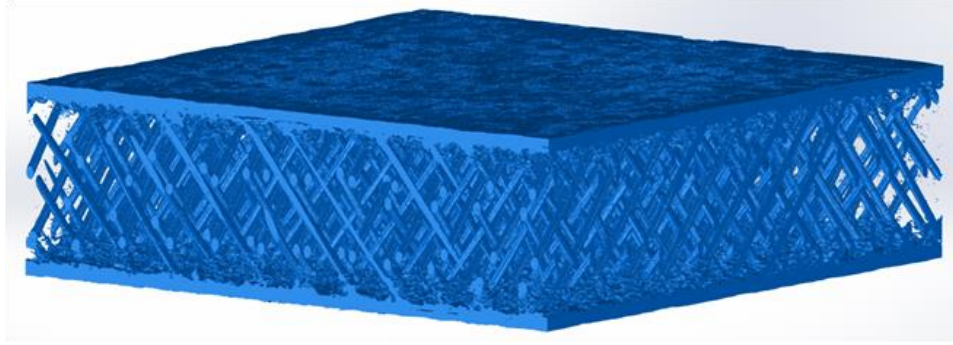
Figure 2.5. MicroCT reconstruction of a 3DFRFC sandwich composite.
(Not to scale.)

2.2.1   Features of 3DFRFC

There are a couple key features to the 3DFRFC used, the most notable of which is the repeating rows of coplanar angled pins, Figure 2.6. The other is the bonding interface between the facesheets and the truss members. The pins are folded over at the ends to create feet with additional surface area for bonding, Figure 2.7 and Figure 2.8. It is important to note that the pins are contained within the core and do not penetrate the facesheet. Some of the adhesive does surround the foot of the pin effectively embedding the pin within the adhesive. This is a feature that is investigated in Section 2.4.2.
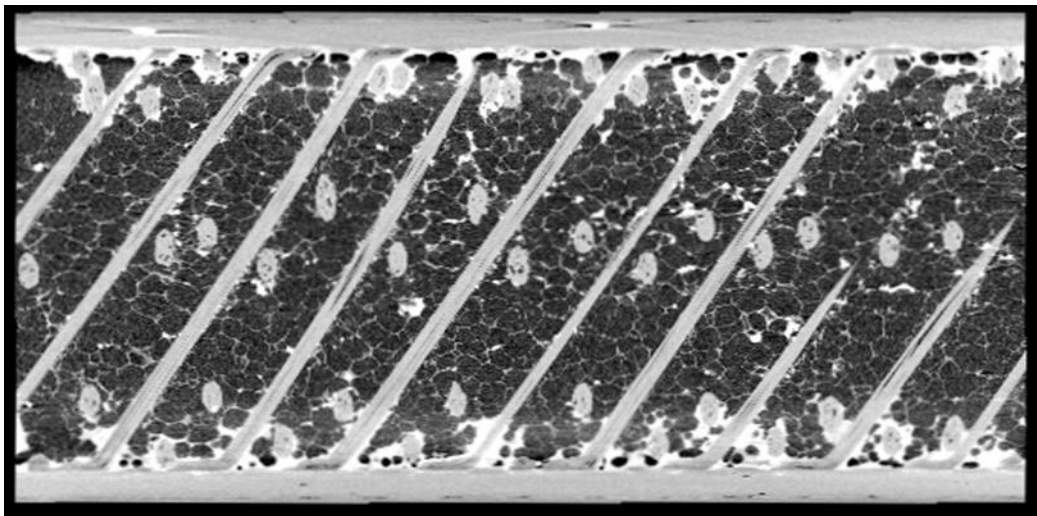
Figure 2.6. MicroCT scan showing side view of a row of reinforcement within the 3DFRFC.
(Not to scale. Image used with the permission of The Aerospace Corporation.)
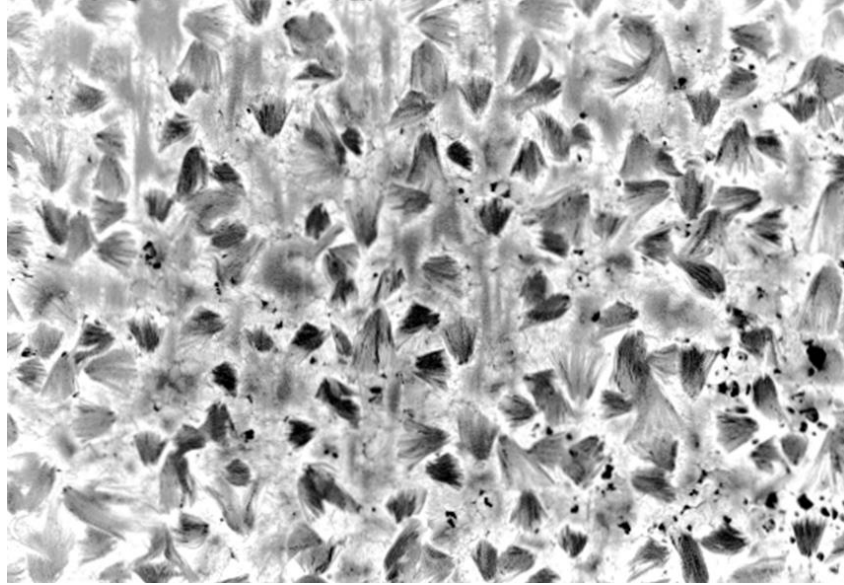
Figure 2.7. MicroCT scan of the reinforcement ends within the adhesive layer of a 3DFRFC sandwich structure.
(Not to scale. Image used with the permission of The Aerospace Corporation.)
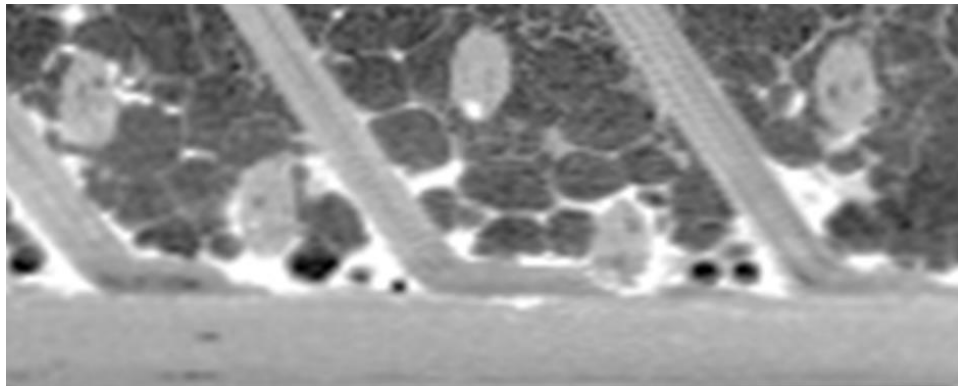
Figure 2.8. MicroCT side view of pin feet within a 3DFRFC sandwich structure.
(Not to scale. Image used with the permission of The Aerospace Corporation.)

2.2.2 Measurement of As-Manufactured Microstructure

The spacing and angle of the reinforcement in a 3DFRFC sandwich composite were measured from the microCT scans using the GNU Image Manipulation Program (GIMP) open-source software [17,18]. All microstructure measurements were taken along the through thickness centerline to ensure consistency and to decouple the spacing and angle measurements. The mean and standard deviation of the centerline spacing (S) and angle (α) were measured as well as the lateral spacing (L, perpendicular to inclined pin, not shown), Figure 2.9. The means

11

from these measurements as well as the detailed geometry of the bonded reinforcement ends were incorporated into the development of the detailed microstructure modeling. Example measurements from one of the microCT scans are given in Table 2.1. This is only from one 3DFRFC sample and is not meant to be representative of the 3DFRFC system as a whole.
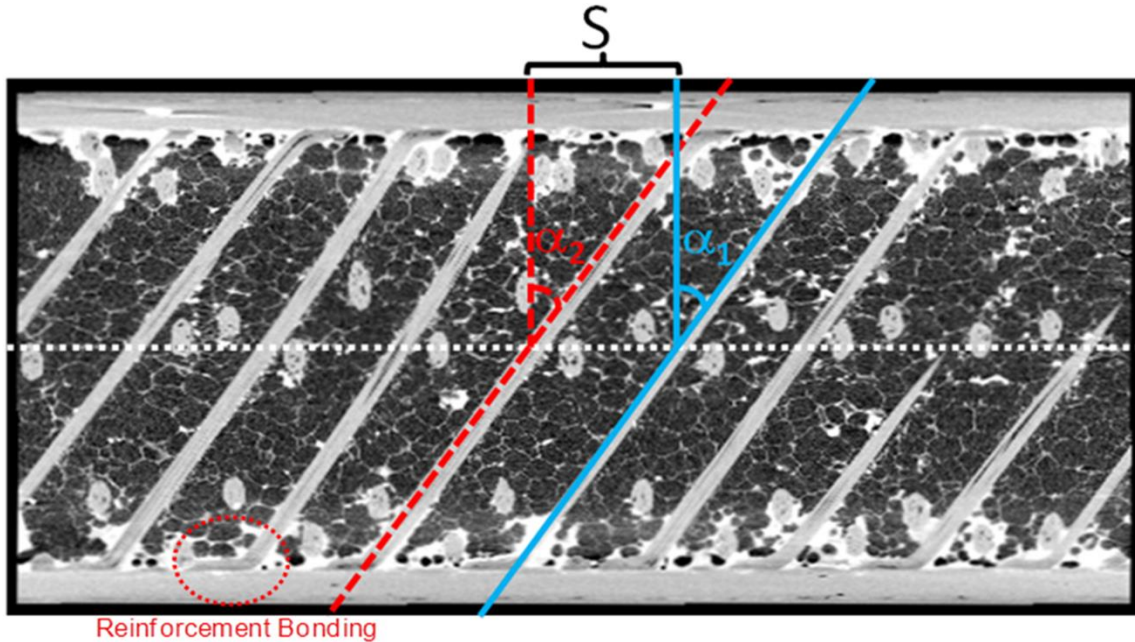
Figure 2.9. Measurements within of the reinforcement within a 3DFRFC sandwich structure. (Not to scale. Image used with the permission of The Aerospace Corporation.)

Table 2.1. Measured variability within a 3DFRFC sample.

Normalized Mean Pin Angle, α	0.979 deg./deg.
Pin Spacing Deviation, δ_S	5.73%
Pin Angle Deviation , δ_α	4.31%

2.3 Developing Finite Element Models of Microstructure

2.3.1 Parametric Script

A Python script was developed to automate the generation of the detailed microstructure geometry for the 3DFRFC within Abaqus CAE. The script is used to generate line geometry in order to create a beam mesh of the reinforcement that can then be used in conjunction with the embedded element method to model the 3DFRFC. The script allows for the definition of arbitrary reinforcement angle, α and spacing, S and L with the addition of parameterized input

for the global panel level rotation as well as the reinforcement bonding end length and radius, Figure 2.10 and Figure 2.11. Side-by-side illustrations of the details of the 3DFRFC model and actual structures are given in Figure 2.12 and Figure 2.13. A key area of deviation can be noted in the details of the bondline. The adhesive layer in the microstructure is assumed uniform; however, the as-manufactured bondline varies in thickness, Figure 2.14.
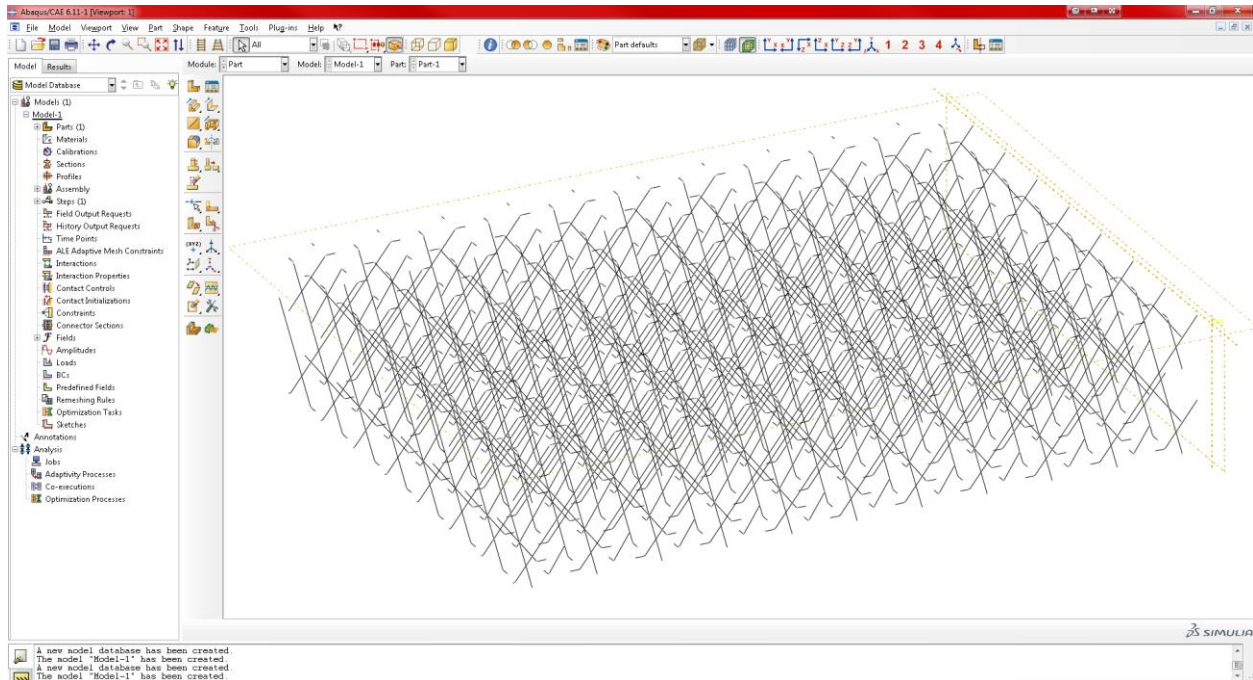
Figure 2.10. A screen capture of geometry output by parametric Abaqus script for a 7x10 unit cell model with 38º reinforcement.
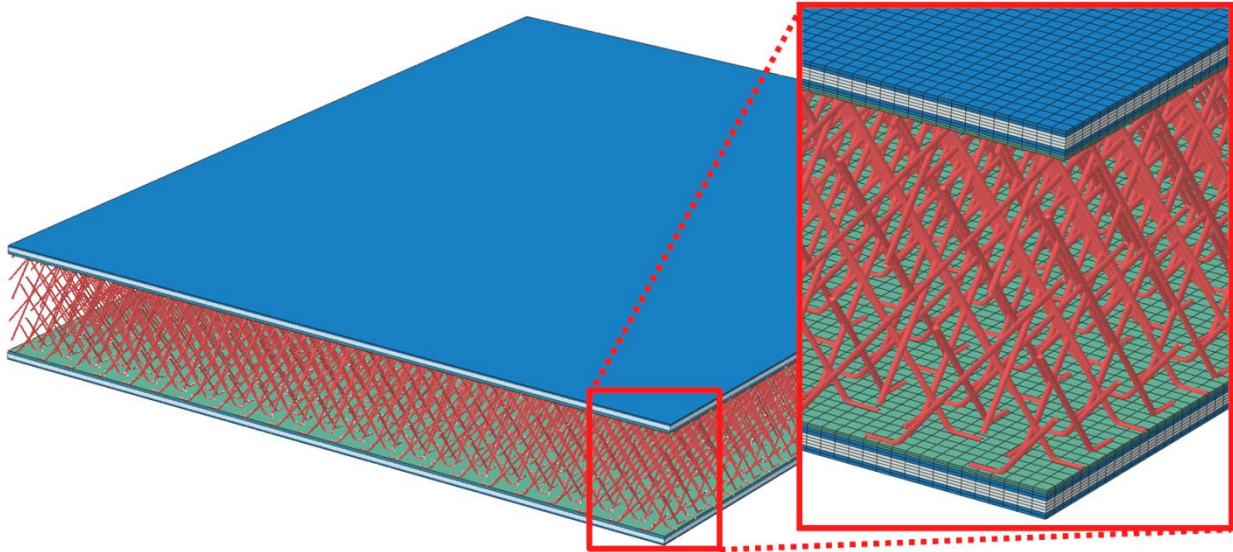
Figure 2.11. 27 x 22 unit cell model of a 3DFRFC sandwich composite with foam removed. (Not to scale.)
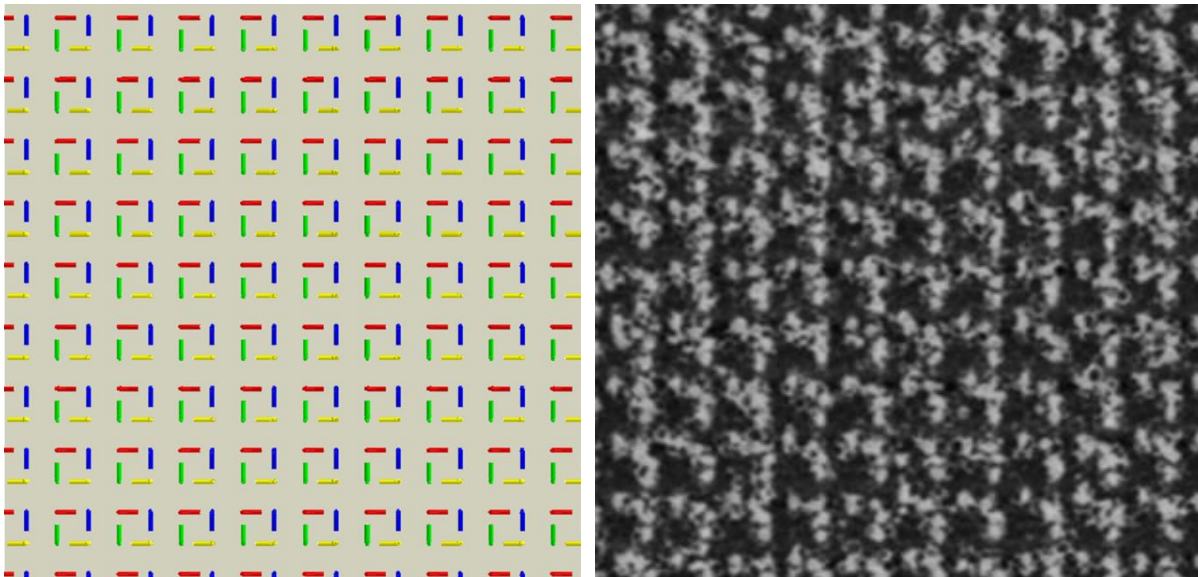
Figure 2.12. Pin bonding surface pattern generated from model, left, microCT, right. (MicroCT image used with the permission of The Aerospace Corporation.)
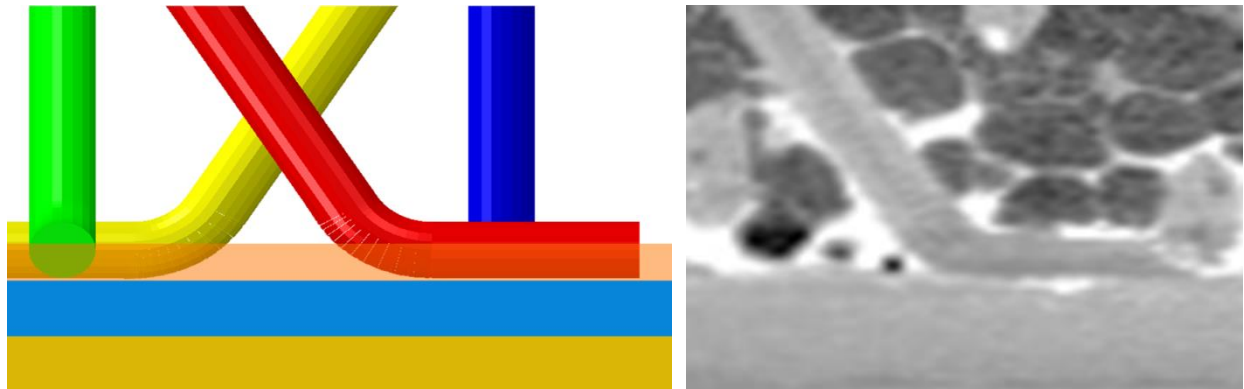
Figure 2.13. Pin end detail from model, left, microCT, right.
(MicroCT image used with the permission of The Aerospace Corporation.)
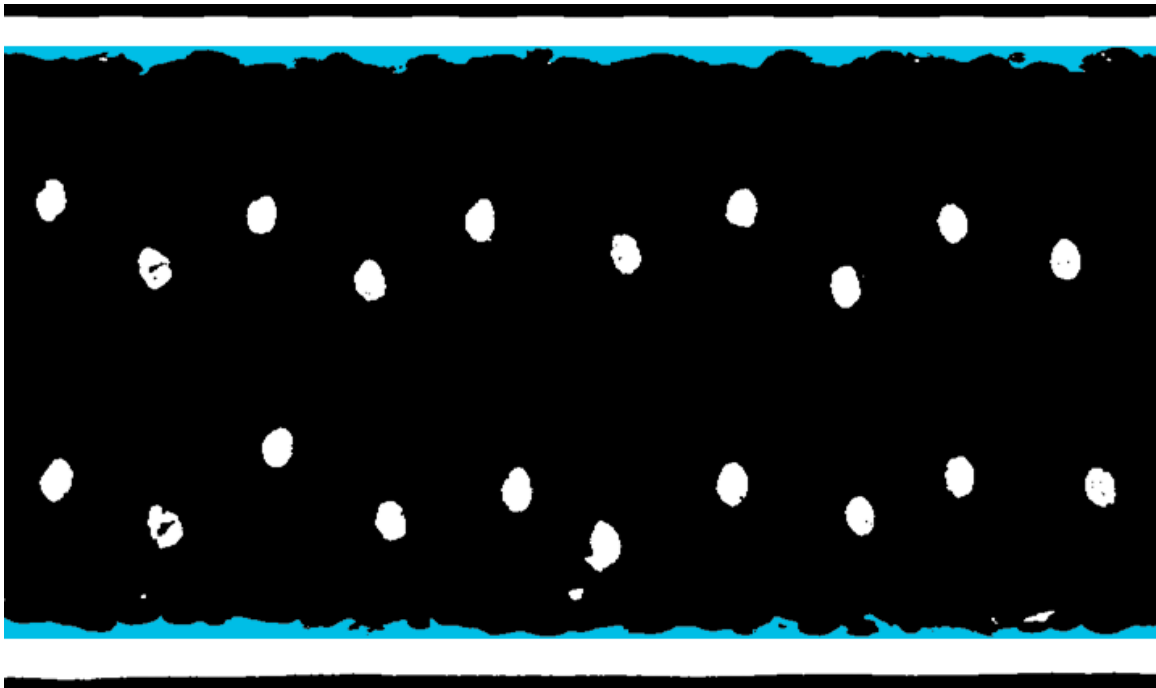
Figure 2.14. Cross-section in a 3DFRFC showing thickness variation of bondline (blue).
(Not to scale.)

The modeling capabilities of this code were expanded to incorporate variations in the microstructure geometry based on the assumptions of independent Gaussian distribution for the pin angle, in-line, and lateral spacing. An illustration of the effect of these three variations is shown independently in Figure 2.15 along with a side-view example of a 3DFRFC microstructure model incorporating all three of these variations. The code also allows for arbitrary global rotation of the core, Figure 2.16. Where β is the rotation of the core within the global x-y plane. These models can be used to investigate the effect of manufacturing variability on the material response of 3DFRFC and the relative sensitivity of the structure to each of these contributing variations. This Python script is made available in Appendix A.
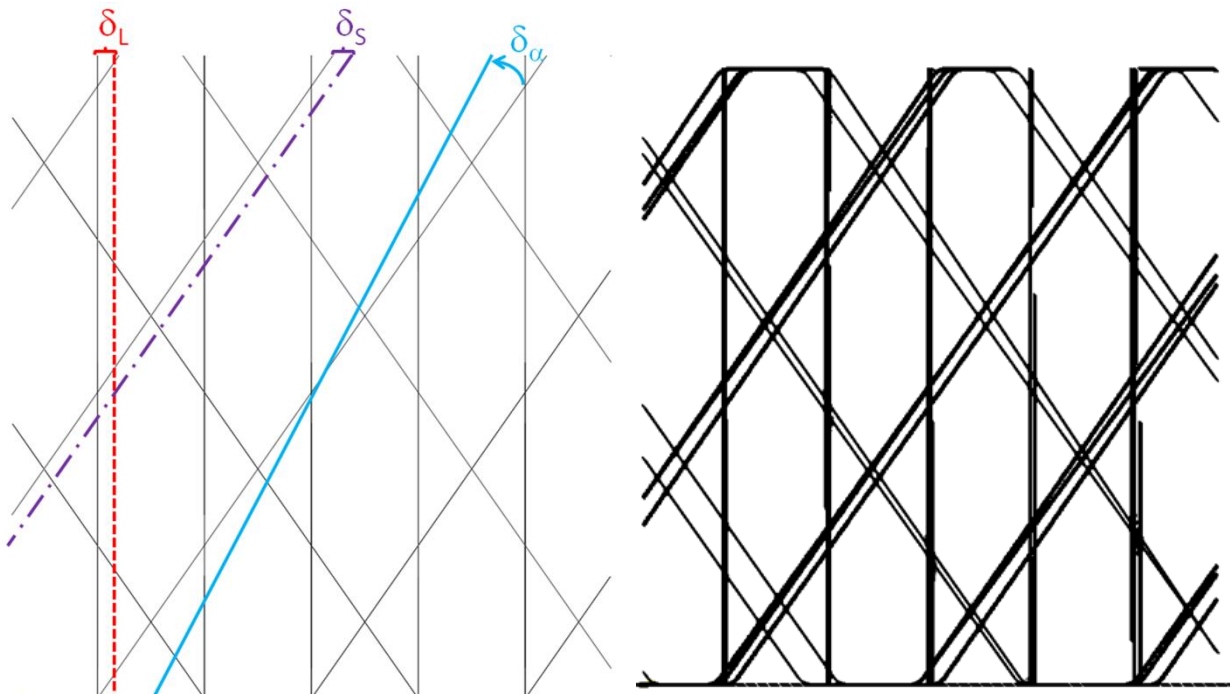
Figure 2.15. Side-view illustration of modeled Gaussian perturbations to microstructure geometry (left) and detailed microstructure with Gaussian variations (right).
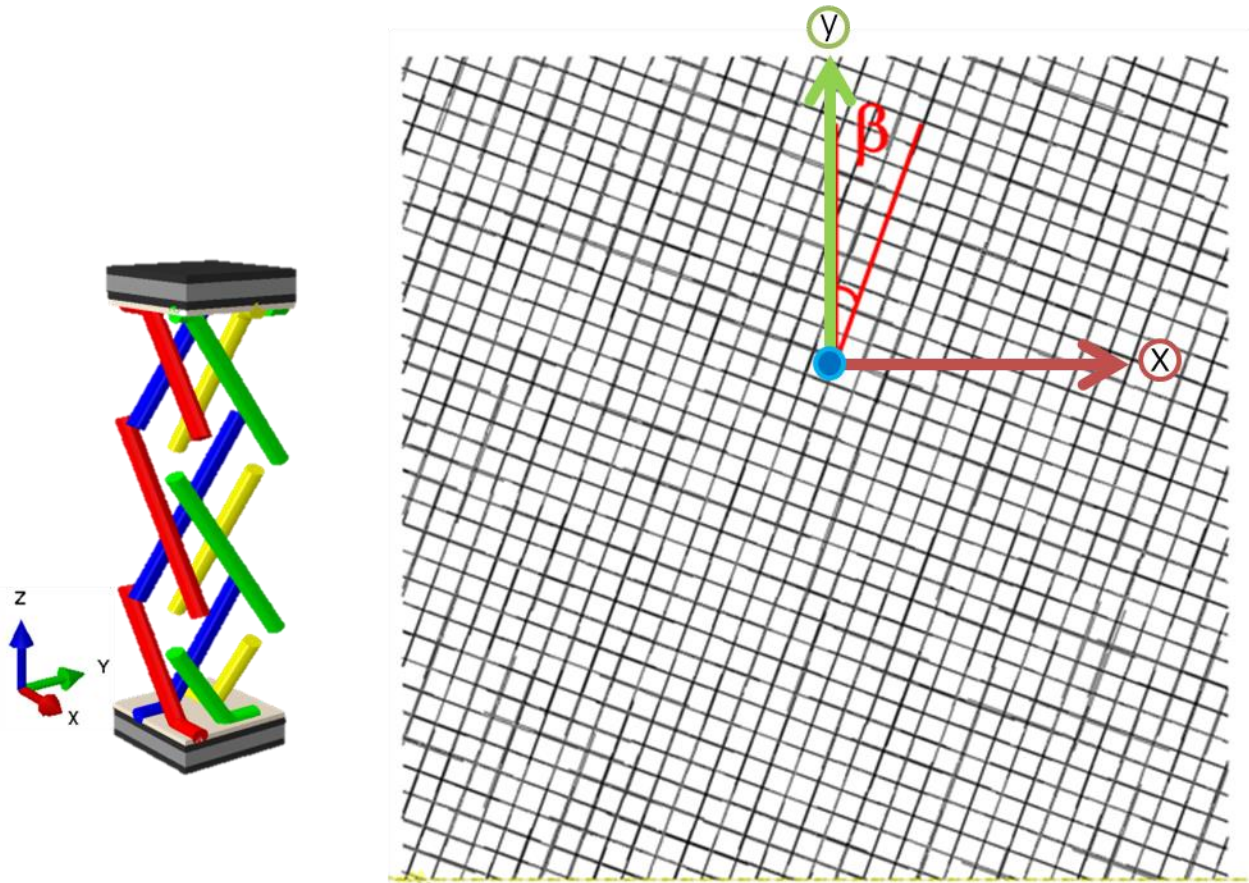
Figure 2.16. Top-down view Illustration of capability to model panels with arbitrary in-plane angle.

2.3.2 Representative Volumetric Element

Given the complex and discrete nature of the 3DFRFC it is important to have an adequate understanding of the bulk behavior of the material. The embedded method can be used to investigate the bulk material behavior. This is accomplished through the use of a representative volume element of the 3DFRFC modeled with periodic boundary conditions. Periodic boundary conditions have been used commonly in investigations into fiber-matrix interactions within unidirectional polymer matrix composites [19] and more recently in an investigation into the interaction of woven fiber tows with polymer matrix during the cure of triaxial braided composites [20]. The material in both of these examples and the 3DFRFC are heavily dependent on the interaction of the discrete constituents that make up the microstructure and so it follows that the use of periodic boundary conditions could be applied to the embedded element model of the detailed microstructure of the 3DFRFC. The representative volume element (RVE) is a cut of the 3DFRFC that captures the entirety of the microstructure. The smallest RVE that captures the geometric detail of the microstructure is called the repeating unit cell (RUC). An example of the unit cell for a 3DFRFC is shown in Figure 2.17 alongside a side view illustrating how the unit cell repeats to make the larger structure.
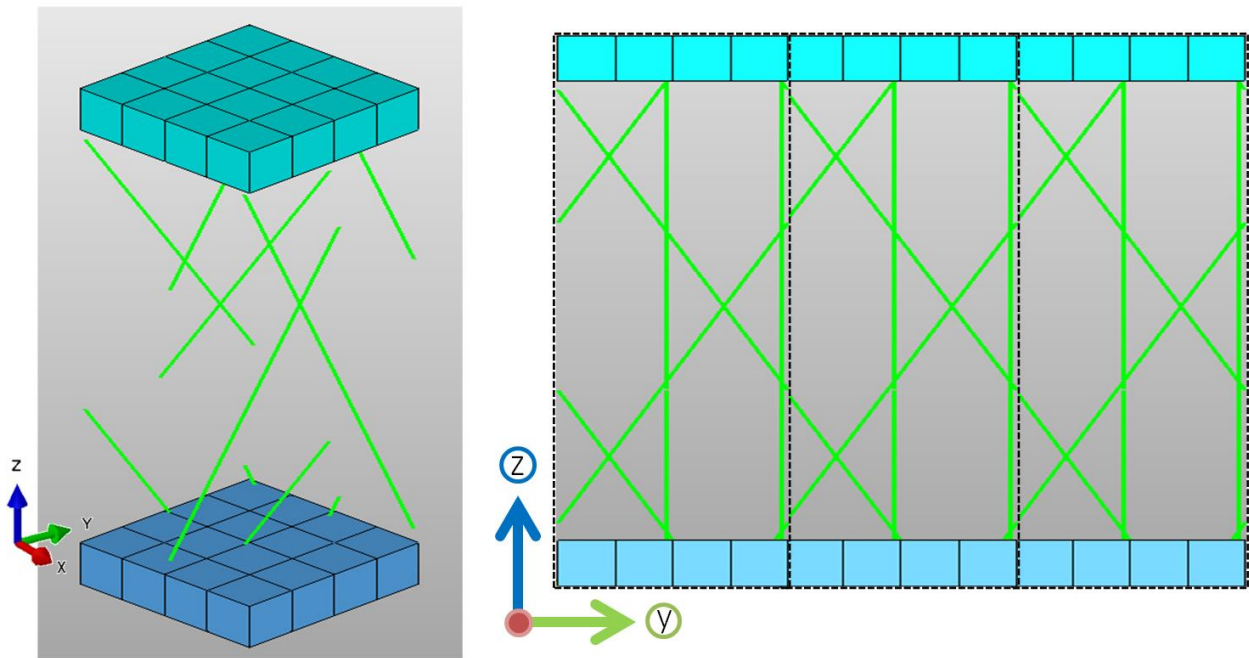
Figure 2.17: Illustration of a 3DFRFC repeating unit cell, left, and a side view of three repeating unit cells, right.
(Not to scale. Foam constituent removed for clarity.)

18

One additional consideration in applying periodic boundary conditions to the embedded element model for the 3DFRFC is the additional rotational degrees of freedom (DoFs 4-6) associated with the nodes on the beam elements modeling the reinforcement. This can be avoided in some 3DFRFC architectures by modeling the RVE with material cuts along planes within the material where the reinforcement does not cross; however, in many 3DFRFC architectures all planes cut through the material intersect the reinforcement in some way. This can be seen in the architecture shown in Figure 2.17. This additional feature of the 3DFRFC RVE is handled by linking the rotations of the nodes on one side to those of the corresponding node on the opposing face, represented by the blue and red squares in Figure 2.18. Much of this work focused on developing a code for automating the generation of these periodic boundary conditions for generic RVE models that contain continuum and/or six degree of freedom nodes on the RVE boundaries. This, in theory, will benefit the modeling of other highly discretized materials that are commonly modeled with beam or shell elements such as honeycomb, corrugated, and truss cores in addition to 3DFRFCs.
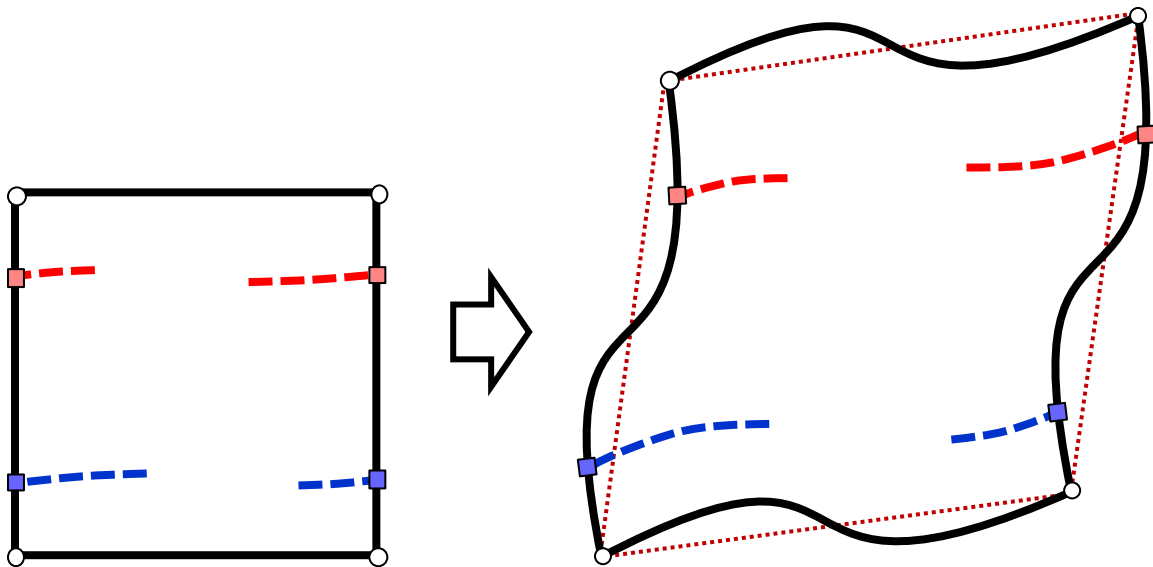
Figure 2.18: Example of a hypothetical two-dimensional periodic unit cell with embedded beam elements in undeformed, left, and deformed configuration, right.

2.3.3 Generalized Periodic Boundary Conditions

Periodic boundary conditions allow for the modeling of a small portion of an infinite material continuum. For this research this was achieved through the development of a 6 degree of freedom (DOF) periodic boundary condition code. This code is critical to capturing the bulk material behavior while only modeling a small portion of the 3DFRFC that represents an infinite solid. The code, written in MATLAB, reads an Abaqus input file for the representative volumetric element (RVE). The RVE can be made up of multiple unit cells; the minimum geometry to describe the structure of the 3DFRFC. The code writes an equation file based on the node information and the periodicity desired. The code is not restricted to only the 3DFRFC models and has been developed to be flexible allowing for generation of periodic boundary conditions for various models including 1D, 2D, and 3D periodic solid, shell, beam, and embedded models and is available in Appendix B.

The truss structure of the 3D fiber network of the 3DFRFC provides added paths for load transfer, but relies on the facesheets to transfer load between the reinforcing truss members. As a result, the stiffness of the facesheet plays an integral role in the ability of the core to resist load. A similar interaction between facesheet and core has been observed in honeycombs and is commonly referred to as the skin effect [21–24]. The through thickness (non-periodic) strain is controlled in an average sense in order to better understand the interaction of the truss structure with the facesheets and how the effective core properties vary with facesheet stiffness and relative location of the pin end. A cross-section showing the differing boundaries is given in Figure 2.19. This allows for application of the global strain on the core in the most generalized method while allowing for the local interaction of the core and facesheet.
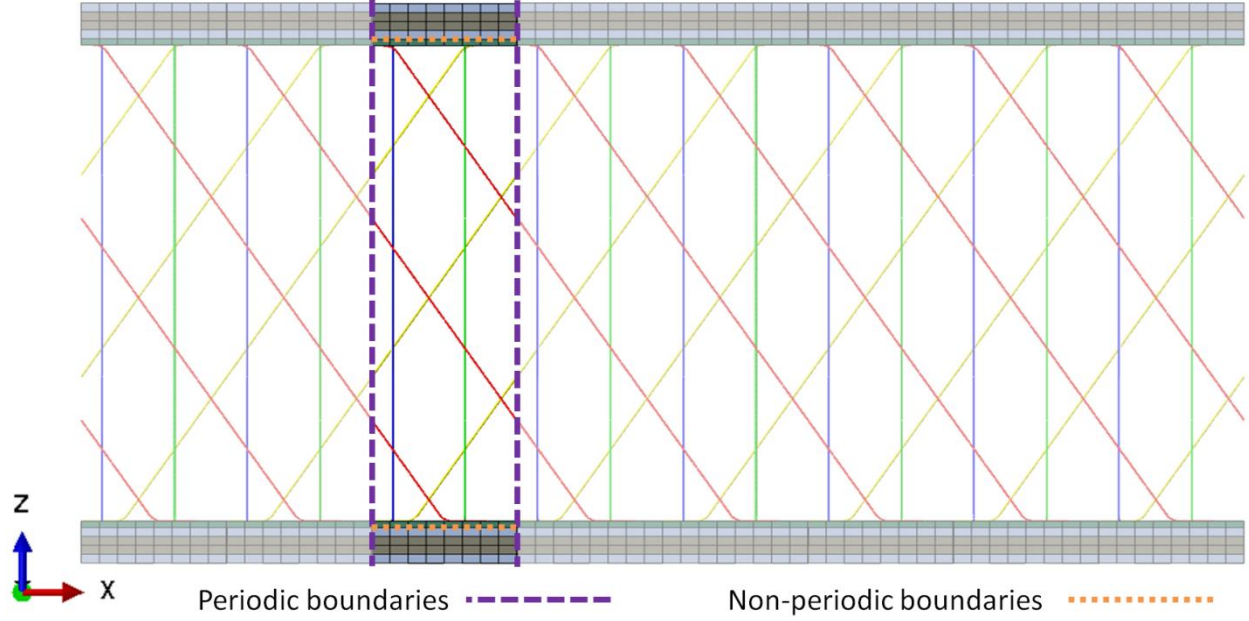
**Figure 2.19: Side view of 3DFRFC with periodic and non-periodic boundaries.
(Not to scale. Foam removed for clarity)**

In general the periodic boundary conditions are given for the x, y, and z planes:

$$\vec{\overline{U}}_{x1i} - \vec{\overline{U}}_{x0i} = \vec{\overline{U}}_{Rx}$$

$$\vec{\overline{U}}_{y1i} - \vec{\overline{U}}_{y0i} = \vec{\overline{U}}_{Ry} \tag{2.1}$$

$$\vec{\overline{U}}_{z1i} - \vec{\overline{U}}_{z0i} = \vec{\overline{U}}_{Rz}$$

Where $\vec{\overline{U}}_{x1i}$ is the displacement vector for the ith node on surface X1, $\vec{\overline{U}}_{x0i}$ is the displacement vector for the corresponding ith node on surface X0, and $\vec{\overline{U}}_{Rx}$ is the displacement vector for the reference node on the X axis. The equivalent terms for the Y and Z planes are also given in Eq. 2.1. The point by point linking of the nodes on opposing surfaces through the displacement of the reference nodes results in the required periodicity. For nodes with rotational degrees of freedom the necessary additional coupling equations are given by:

$$\vec{\kappa}_{x1i} - \vec{\kappa}_{x0i} = 0$$

$$\vec{\kappa}_{y1i} - \vec{\kappa}_{y0i} = 0 \tag{2.2}$$

$$\vec{\kappa}_{z1i} - \vec{\kappa}_{z0i} = 0$$

21

Where $\vec{\kappa}_{x1i}$ is the rotation vector for the ith node on surface X1 and $\vec{\kappa}_{x0i}$ is the rotation vector for the corresponding ith node on surface X0. The equivalent terms for the Y and Z planes are also given in Eq. 2.2. The direct point by point linking of the nodal rotation on opposing surfaces enforces the required periodicity for the rotation degrees of freedom.

For a strain controlled, non-periodic, direction the translational boundary conditions are relaxed to a surface averaged formulation given in Eq. 2.3.

$$
\begin{aligned}
\sum_{i=1}^{n} A_{x1_i}\vec{U}_{x1_i} - \sum_{i=1}^{m} A_{x0_i}\vec{U}_{x0_i} &= A_{Rx}\vec{U}_{Rx} \\
\sum_{j=1}^{n} A_{y1_i}\vec{U}_{y1_i} - \sum_{j=1}^{m} A_{y0_i}\vec{U}_{y0_i} &= A_{Ry}\vec{U}_{Ry} \\
\sum_{k=1}^{n} A_{z1_i}\vec{U}_{z1_i} - \sum_{k=1}^{m} A_{z0_i}\vec{U}_{z0_i} &= A_{Rz}\vec{U}_{Rz}
\end{aligned}
\tag{2.3}
$$

Where the A terms are the weights for the individual nodes. The calculation of these terms is one of the keys to evaluating the surface averaged displacements allowing for the global strain to be proscribed with the minimum constraint. The surface average coefficients are calculated using one of two methods. If the nodes belong to continuum elements, a nodal area of influence is calculated based on the element definitions, dividing the area of each element equally between each node on the surface. In the case of higher order continuum elements a set of two equations are used to decouple the corner and midpoint nodes. This method has been verified using differing element types (linear and higher order) with both structured and unstructured meshes. Plots of the nodal areas of influence calculated for an example mesh surface is shown in Figure 2.20. A secondary method for calculating the area of influence is based on Dirichlet-Voronoi cells [25–27] and is used for non-continuum elements (non-embedded trusses, beams, & shells). For simple structured meshes these two methods produce the same results. The Dirichlet-Voronoi cell method can generate errors for unstructured meshes with differing elements attached at differing nodes on the surface. For this reason it is included only for the use with non-continuum elements where an area cannot be defined from the element definition. For this

dissertation the nodal area of influence based on the element definitions is used exclusively and is recommended for all continuum meshes for its consistent results.
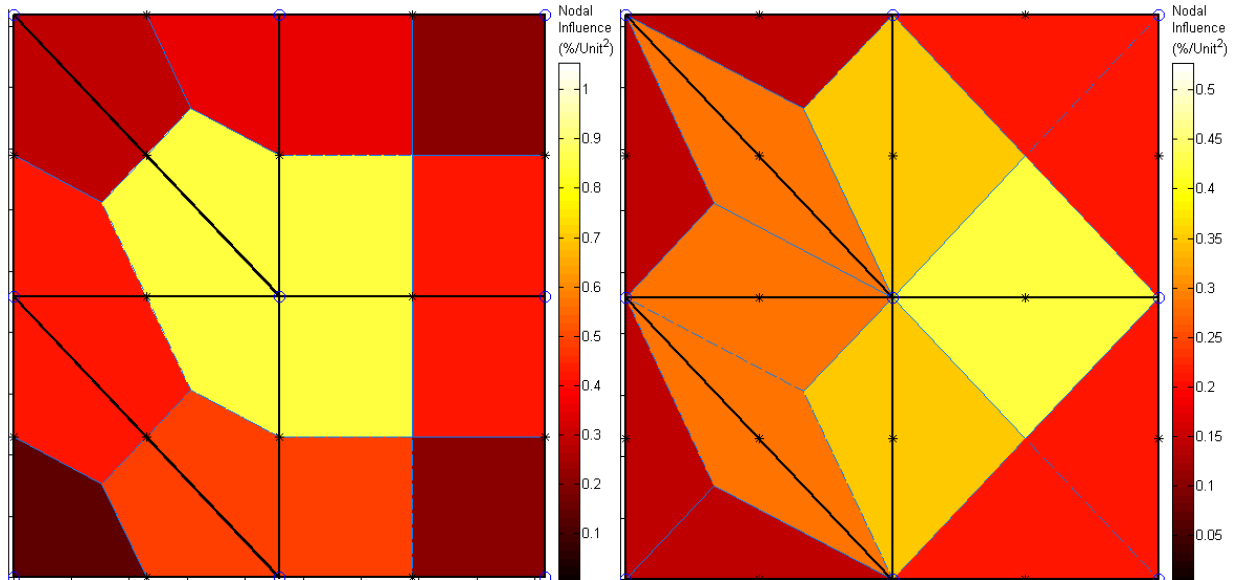
Figure 2.20: Example nodal area of influence output for surface with various higher-order Elements. Corner nodes, left, midpoint nodes, right.

2.4 Effective Bulk 3DFRFC Properties

In order to determine the effective bulk properties of the 3DFRFC the periodic and generalized boundary conditions discussed in the last section are used and the six global strain components are applied to the RVE independently. The displacements are only applied through the three reference nodes and distributed to the model through the generalized periodic boundary conditions. The resultant forces averaged over the RVE yield a global, or bulk, stress that can be used to determine effective moduli that inherently includes the effect of the faceshseet interaction with the core. A similar effective elastic moduli approach was discussed by Achenbach at the fiber/matrix level [28]. For the 3DFRFC analysis we take this a step further and use the theory of superposition to separate the core response within the global model from the facesheet response, [29–31] Figure 2.21. An illustration depicting the components of the force and displacement vectors at the reference nodes in the x-y plane is given in Figure 2.22 to clarify the equations that follow.
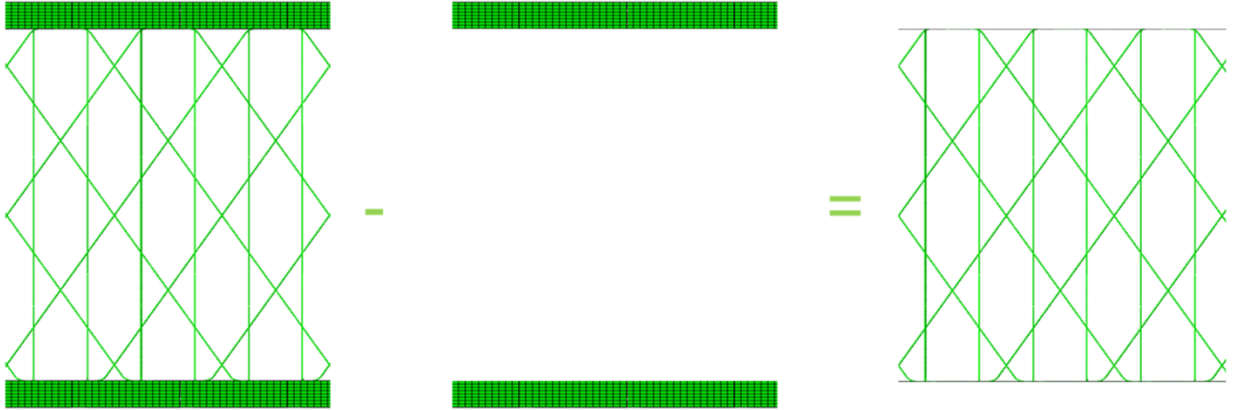
Figure 2.21: Illustration of use of superposition to get effective core properties.
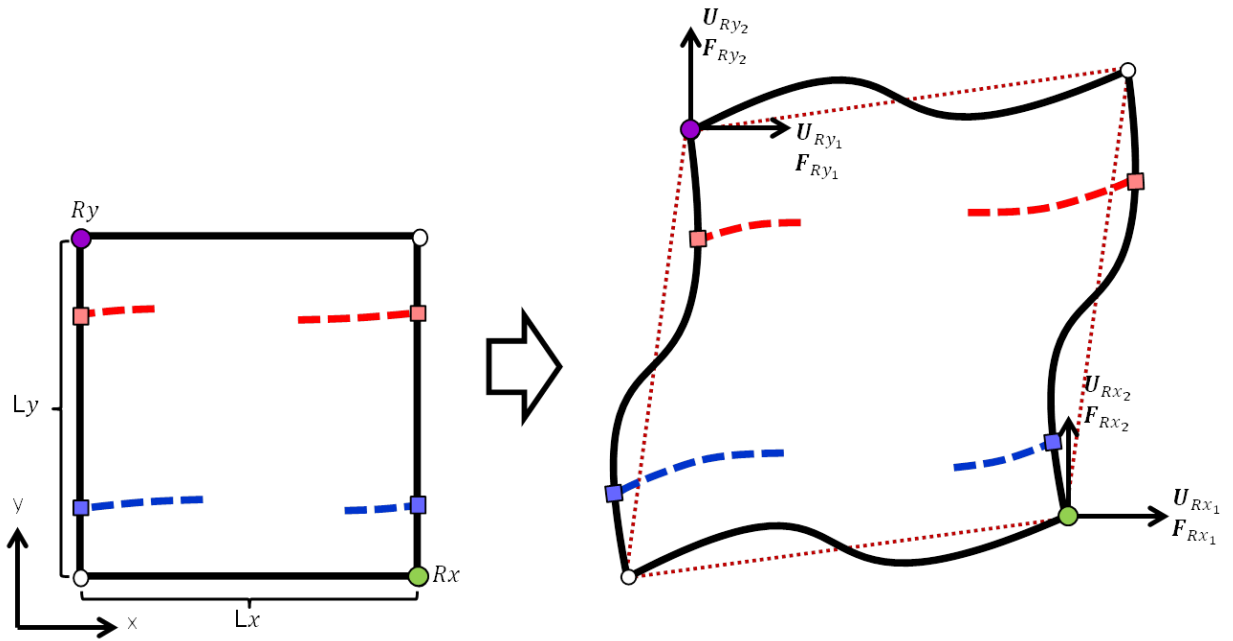
Figure 2.22: Illustration of force and displacement components at reference nodes in x-y plane.

The six strains that are applied globally are given by

$$
\begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{23} \\ \gamma_{31} \\ \gamma_{12} \end{pmatrix} = \underbrace{\begin{pmatrix} \varepsilon_{11} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{case\ 1}, \underbrace{\begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{case\ 2}, \underbrace{\begin{pmatrix} 0 \\ 0 \\ \varepsilon_{33} \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{case\ 3}, \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ \gamma_{23} \\ 0 \\ 0 \end{pmatrix}}_{case\ 4}, \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \gamma_{31} \\ 0 \end{pmatrix}}_{case\ 5}, \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \gamma_{12} \end{pmatrix}}_{case\ 6} \qquad (2.4)
$$

where the strains are given in terms of the reference node displacements and RVE dimensions as

$$\bar{\varepsilon}_{11} = \frac{U_{Rx_1}}{L_x}, \qquad \bar{\varepsilon}_{22} = \frac{U_{Ry_2}}{L_y}, \qquad \bar{\varepsilon}_{33} = \frac{U_{Rz_3}}{L_z}$$

$$\bar{\gamma}_{23} = \frac{U_{Rz_2}}{L_z} + \frac{U_{Ry_3}}{L_y}, \quad \bar{\gamma}_{31} = \frac{U_{Rz_1}}{L_z} + \frac{U_{Rx_3}}{L_x}, \quad \bar{\gamma}_{12} = \frac{U_{Rx_2}}{L_x} + \frac{U_{Ry_1}}{L_y}$$

(2.5)

The same boundary conditions are applied to the sandwich and facesheet model, effective core-only forces are then calculated as

$$\vec{F}_{Rx} = \vec{F}_{Rx_{Sandwich}} - \vec{F}_{Rx_{Facesheet}}$$

(2.6)

The core-only effective stresses are then given by

$$\bar{\sigma}_{11} = \frac{F_{Rx_1}}{L_y L_z}, \qquad \bar{\sigma}_{22} = \frac{F_{Ry_2}}{L_x L_z}, \qquad \bar{\sigma}_{33} = \frac{F_{Rz_3}}{L_x L_y}$$

$$\bar{\tau}_{23} = \frac{F_{Rz_2}}{2L_x L_y} + \frac{F_{Ry_3}}{2L_x L_z}, \quad \bar{\tau}_{31} = \frac{F_{Rz_1}}{2L_x L_y} + \frac{F_{Rx_3}}{2L_y L_z}, \quad \bar{\tau}_{12} = \frac{F_{Rx_2}}{2L_y L_z} + \frac{F_{Ry_1}}{2L_x L_z}$$

(2.7)

Since the stresses and strains are now known for the six cases we can solve for the stiffness matrix in the form

$$\begin{pmatrix} \bar{\sigma}_{11} \\ \bar{\sigma}_{22} \\ \bar{\sigma}_{33} \\ \bar{\tau}_{23} \\ \bar{\tau}_{31} \\ \bar{\tau}_{12} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\ c_{12} & c_{22} & c_{23} & 0 & 0 & 0 \\ c_{13} & c_{23} & c_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{66} \end{pmatrix} \begin{pmatrix} \bar{\varepsilon}_{11} \\ \bar{\varepsilon}_{22} \\ \bar{\varepsilon}_{33} \\ \bar{\gamma}_{23} \\ \bar{\gamma}_{31} \\ \bar{\gamma}_{12} \end{pmatrix} = [\bar{C}] \begin{pmatrix} \bar{\varepsilon}_{11} \\ \bar{\varepsilon}_{22} \\ \bar{\varepsilon}_{33} \\ \bar{\gamma}_{23} \\ \bar{\gamma}_{31} \\ \bar{\gamma}_{12} \end{pmatrix}$$

(2.8)

The effective orthotropic engineering properties can then be obtained from the compliance matrix, $\bar{S}$, which is found by inverting the stiffness matrix [32]:

$$
[\overline{C}]^{-1} = [\overline{S}] = \begin{pmatrix} \dfrac{1}{E_{11}} & \dfrac{-\nu_{21}}{E_{22}} & \dfrac{-\nu_{31}}{E_{33}} & 0 & 0 & 0 \\[2.2ex] \dfrac{-\nu_{12}}{E_{11}} & \dfrac{1}{E_{22}} & \dfrac{-\nu_{32}}{E_{33}} & 0 & 0 & 0 \\[2.2ex] \dfrac{-\nu_{13}}{E_{11}} & \dfrac{-\nu_{23}}{E_{22}} & \dfrac{1}{E_{33}} & 0 & 0 & 0 \\[2.2ex] 0 & 0 & 0 & \dfrac{1}{G_{23}} & 0 & 0 \\[2.2ex] 0 & 0 & 0 & 0 & \dfrac{1}{G_{13}} & 0 \\[2.2ex] 0 & 0 & 0 & 0 & 0 & \dfrac{1}{G_{12}} \end{pmatrix} \tag{2.9}
$$

For simplicity the stiffness and compliance matrices are shown assuming the form of an orthotropic solid, but in general this is not required. For the 3DFRFC studied this assumption was found to have a negligible impact on the effective properties (~0.1%).

2.4.1 Model Configurations

The current state of the art for homogenized material models that can be used for 3DFRFC materials is a clamped-Uniform Deformation Gradient (c-UDG) model [33]. The published c-UDG model builds on a truss-core (no foam) model that was derived using simple axial tension or compression (direct action) of the individual pins and the assumption of a uniform deformation gradient [34]. In [33] Liu incorporated the supporting foam into the model and included lateral forces on the pins caused by the foam. Both models assume a uniform deformation gradient and Liu add the assumption that the pin ends are clamped at the boundary with the facesheet. The enforcement of a uniform deformation gradient within the core limits the local deformation of the individual pins and does not allow for interaction between neighboring pins and coupling with the facesheet and adhesive layers. The more generalized effective elastic approach discussed in the previous section relaxes these constraints to allow for these local interactions. Throughout this dissertation the clamped-Uniform Deformation Gradient (c-UDG) homogenized model published by Liu is used as a basis for comparison [33].

The effective elastic approach can be utilized to investigate the impact of changes to the microstructure or modeling assumptions. The influence of pin placement and facesheet thickness

on the effective properties can now be investigated. This is accomplished using 3 models both with and without the adhesive layer (6 total), Figure 2.23. The reinforcing elements are modeled as beams but are rendered with thickness for clarity. The reason for modeling both with and without the adhesive layer is due to the unevenness seen in the actual bondline, Figure 2.14. A uniform bondline is likely to be stiffer than the actual bondline and modeling that area with foam-only properties should provide a lower bound on the interaction thus using both models should bound the actual material. The three models investigated are the full detailed embedded model, a simplified straight pin model, and the published c-UDG model.
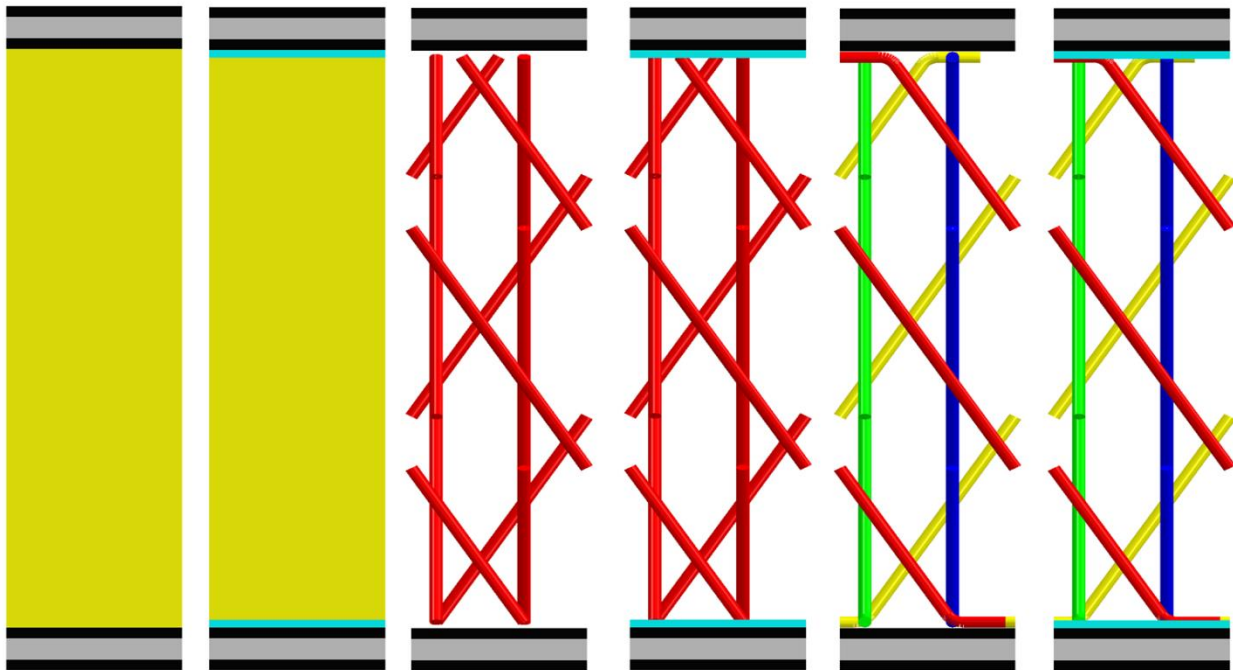
**Figure 2.23: Side view illustrating the six 3DFRFC configurations evaluated.
(Not to scale. Foam removed for clarity)**

2.4.2   Pin Placement within Adhesive Layer

Analysis was done to investigate the effect of the location of the end of the reinforcing pins within the adhesive layer. The location was varied from 0% (inner surface of the adhesive) to 100% (completely embedded in the adhesive, touching the facesheet), Figure 2.24. Since the pins are modeled with beam elements this corresponds with the location of the centroid of the reinforcing pin. Measurements of the actual microstructure using the microCT scan data

27

corresponded to a 25% embedded pin. As a result 25% is used for the other two analyses in this section.
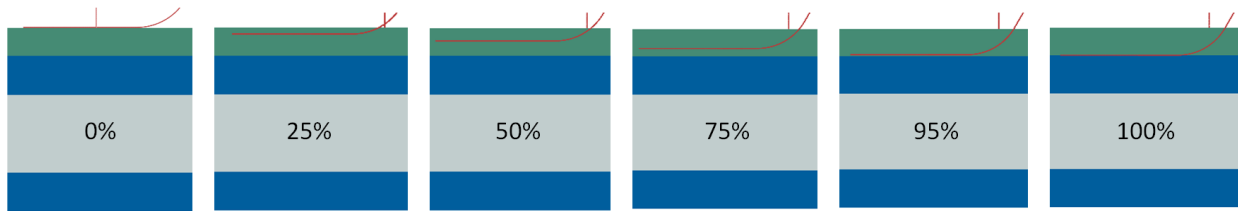
Figure 2.24: Side view of 3DFRFC with varying pin placement within the adhesive layer (green). (Not to scale. Foam removed for clarity)

The effective engineering properties plotted as a function of the location placement of the pin end within the adhesive layer are given in Figures 2.25-2.30. For ease of comparison the moduli are normalized by the through thickness moduli for the c-UDG model without adhesive. The pin placement location is not relevant for the c-UDG cases and hence their values are constant. As expected the presence of the adhesive layer increases stiffness in all cases. Of greater interest is the relatively high sensitivity of the through thickness axial and shear moduli to the location of the pin placement. This finding makes some sense as the primary mechanism of load transfer between pins is through the facesheet. As the distance between the pin and facesheet is decreased the shear-lag effect is also reduced [35–39]. The higher compliance of the foam causes the values for the models without adhesive remain relatively constant for pin locations ≤ 75% as the faceheet becomes decoupled from the truss members within the core. Also note that the through thickness axial modulus for the c-UDG homogenized model is considerably higher than the other models. This discrepancy is confirmed in Chapter 5 where the size effect of discrete samples is investigated.
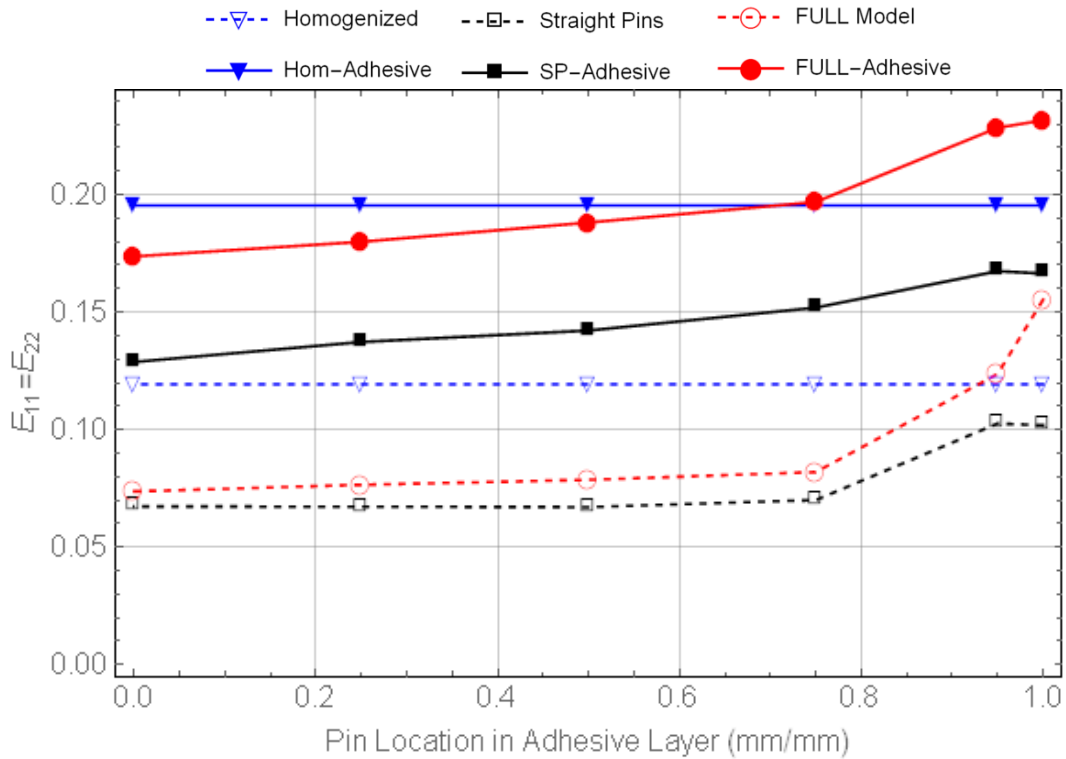
Figure 2.25: Plot of normalized in-plane axial modulus (E11 = E22) as a function of pin location.
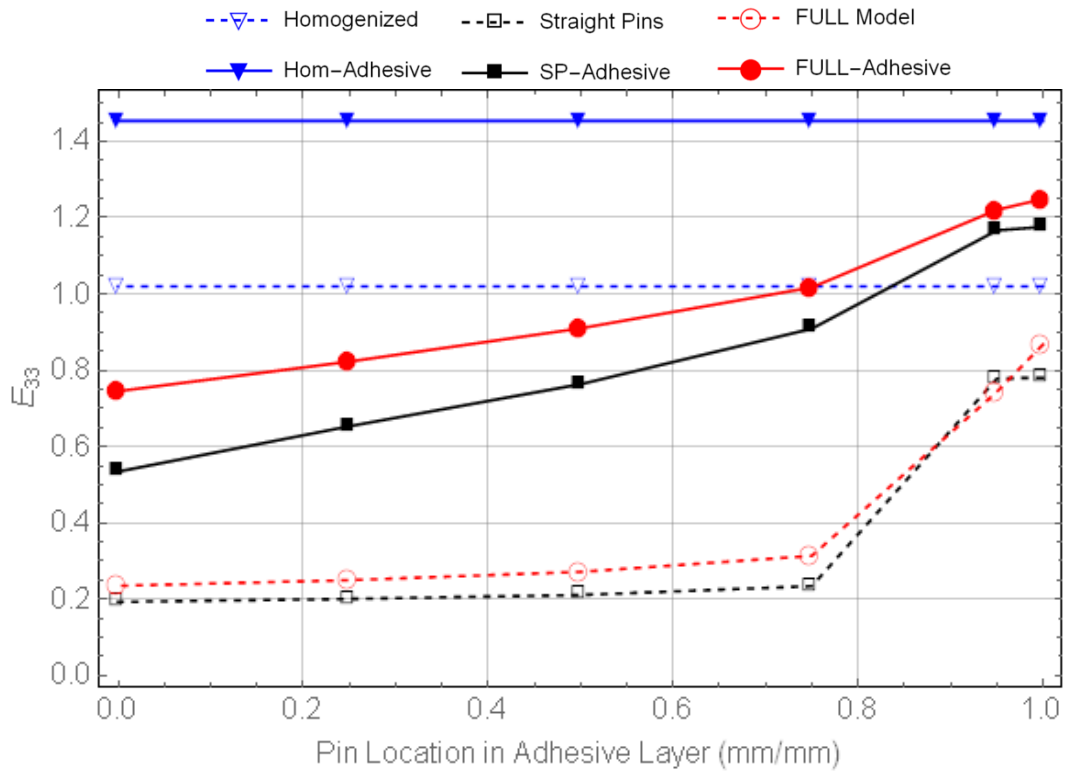
Figure 2.26: Plot of normalized through thickness axial modulus (E33) as a function of pin location.
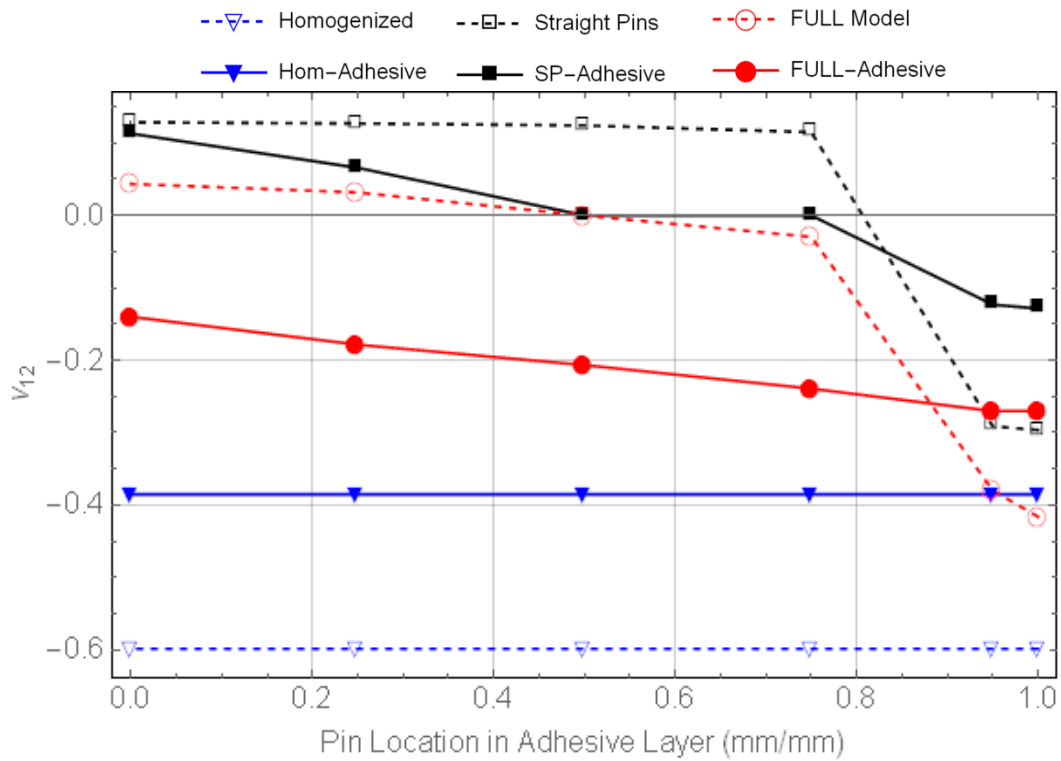
Figure 2.27: Plot of in-plane Poisson's ratio (ν12) as a function of pin location.
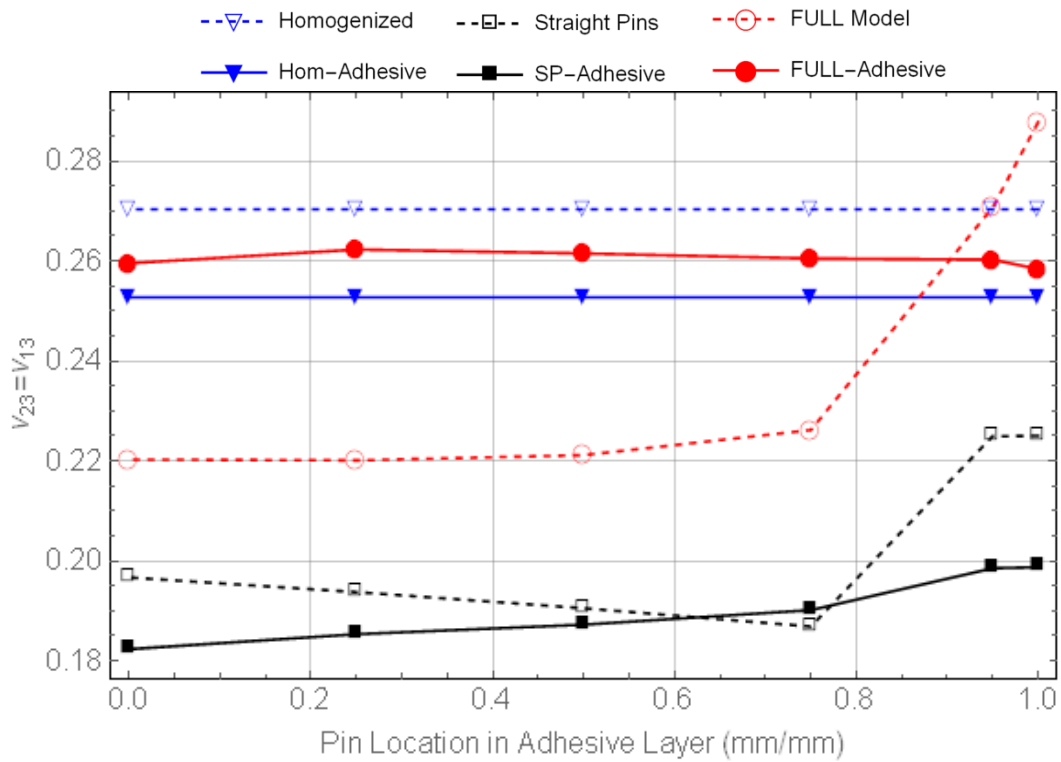
Figure 2.28: Plot of through thickness Poisson's ratio (ν23 = ν13) as a function of pin location.

30
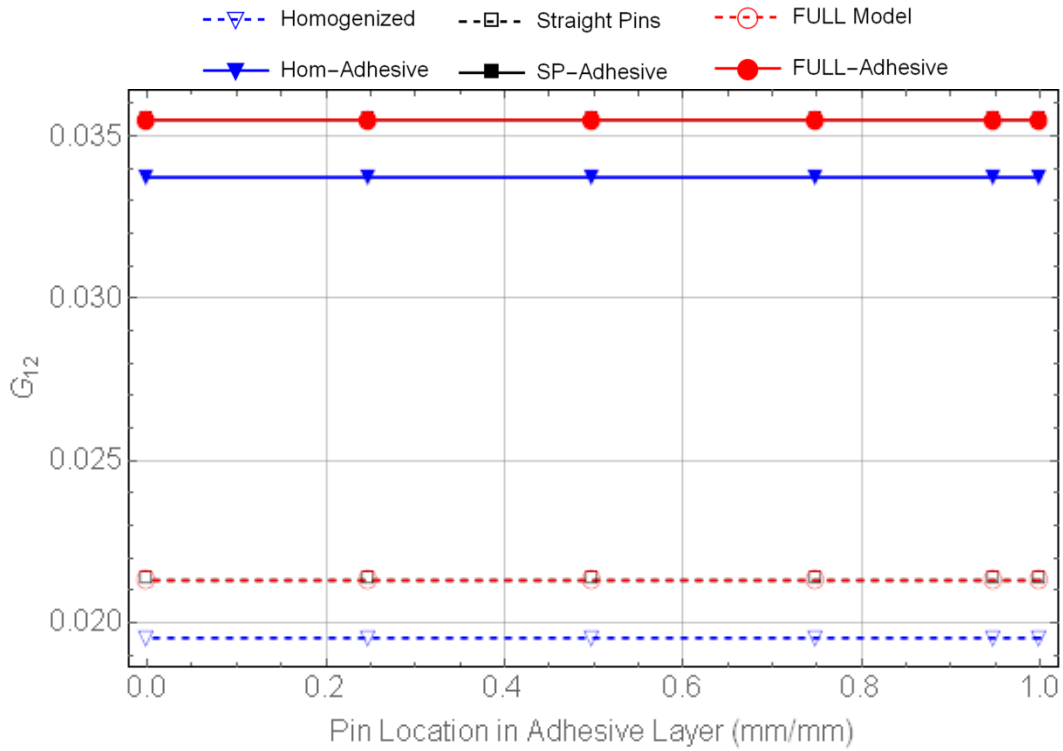
Figure 2.29: Plot of normalized in-plane shear modulus (G12) as a function of pin location.
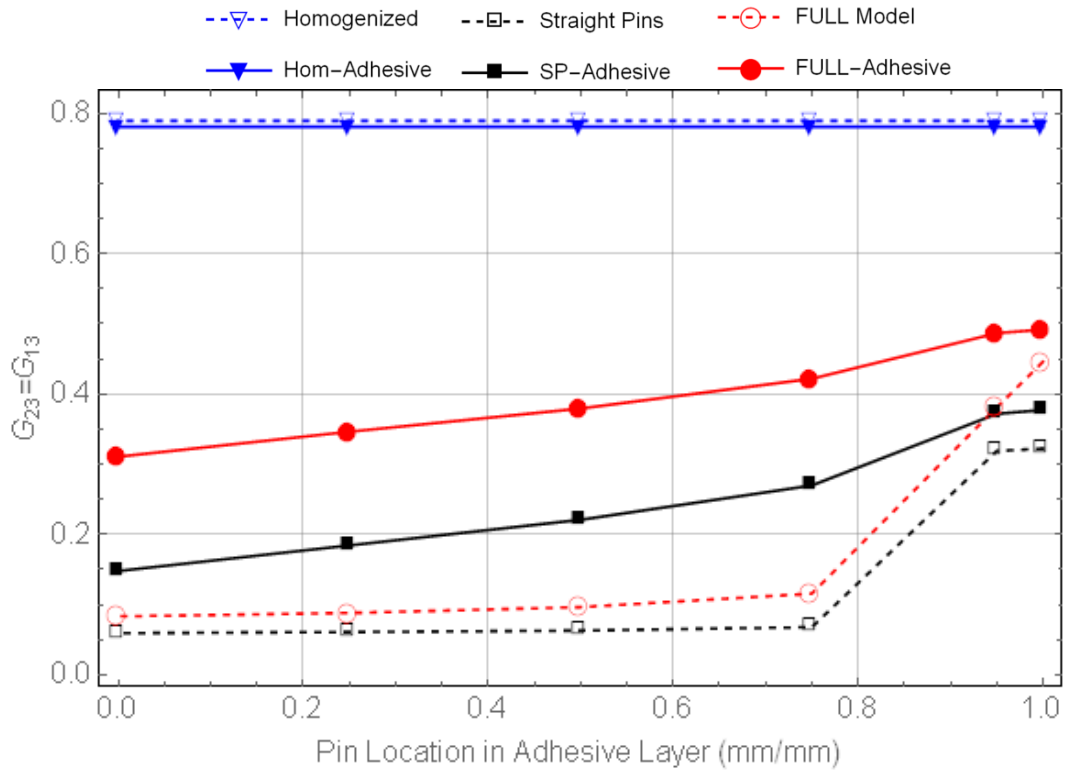
Figure 2.30: Plot of normalized through thickness shear modulus (G23 = G13) as a function of pin location.

31

2.4.3 Skin Effects in 3DFRFC

In order to evaluate the effect of facesheet stiffness on the effective bulk properties of the 3DFRFC the number of facesheet plies in the RVE was varied from 1 to 32 plies, Figure 2.31. This was selected over increasing the ply stiffness as this is a more realistic evaluation of the variations in 3DFRFC sandwich composites. While the type of facesheet material could be changed it is more likely that most of the variation that would occur in an actual structure would be due to regions designed with thicker facesheets. The baseline facesheet is a 4 ply woven facesheet. For the thicker facesheets the facesheet stacking $[(0°/90°)/\pm45°/\mp45°/(90°/0°)]$ is repeated as necessary. For thinner facesheets the outer two or three plies are simply removed.
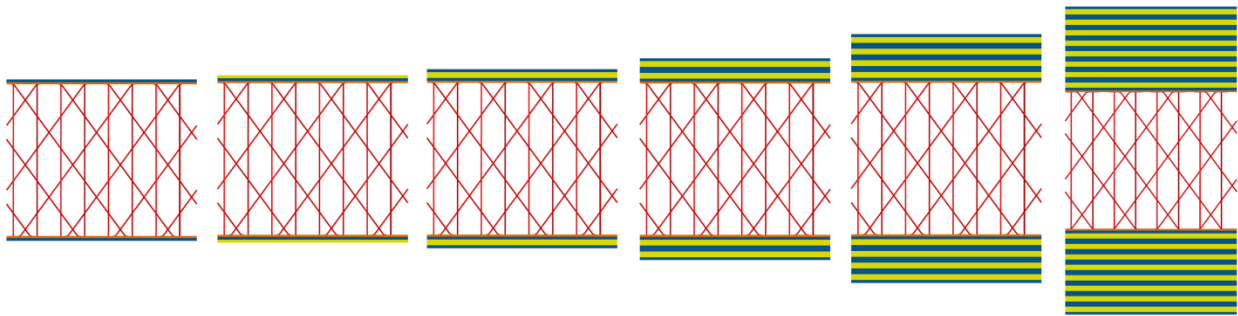
Figure 2.31: Side view of 3DFRFC with varying facesheet plies.
(Not to scale. Foam removed for clarity)

The effective engineering properties plotted as a function of the facesheet thickness are given in Figures 2.32-2.37. For easy of comparison the moduli are again normalized by the through thickness moduli for the c-UDG model without adhesive. Overall the sensitivity of the engineering properties to the facesheet thickness was found to be relatively minimal. The only real exception was the out of plane shear modulus which exhibited ~25% variation in stiffness. The previous analysis showed that the properties were highly sensitive to the placement of the pine end. It is possible that placing the pins closer to the facesheet could result in a stronger facesheet-pin coupling and thus a stronger skin effect.
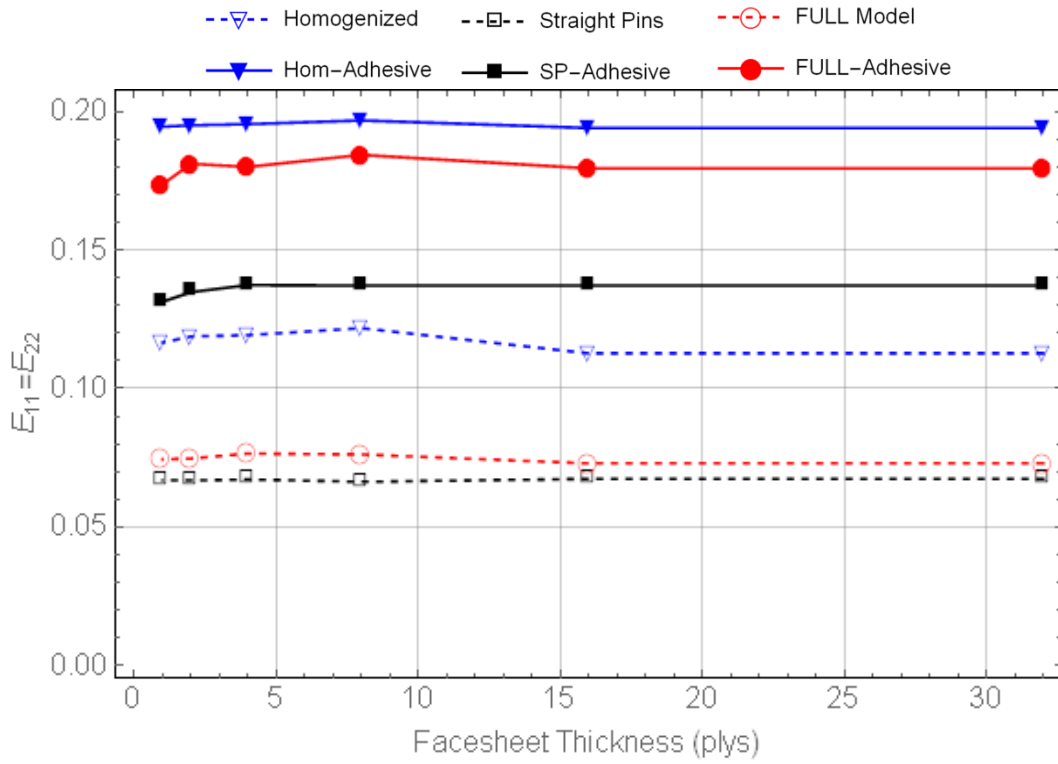
Figure 2.32: Plot of normalized in-plane axial modulus (E11 = E22) as a function of facesheet thickness.
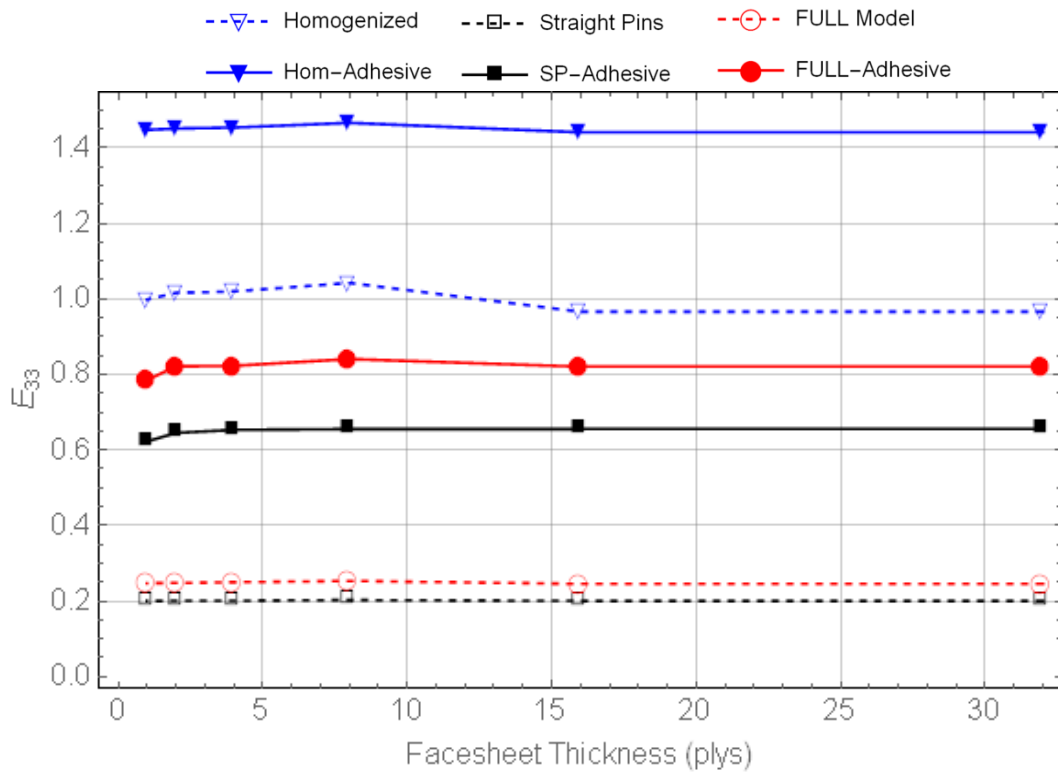
Figure 2.33: Plot of normalized through thickness axial modulus (E33) as a function of facesheet thickness.
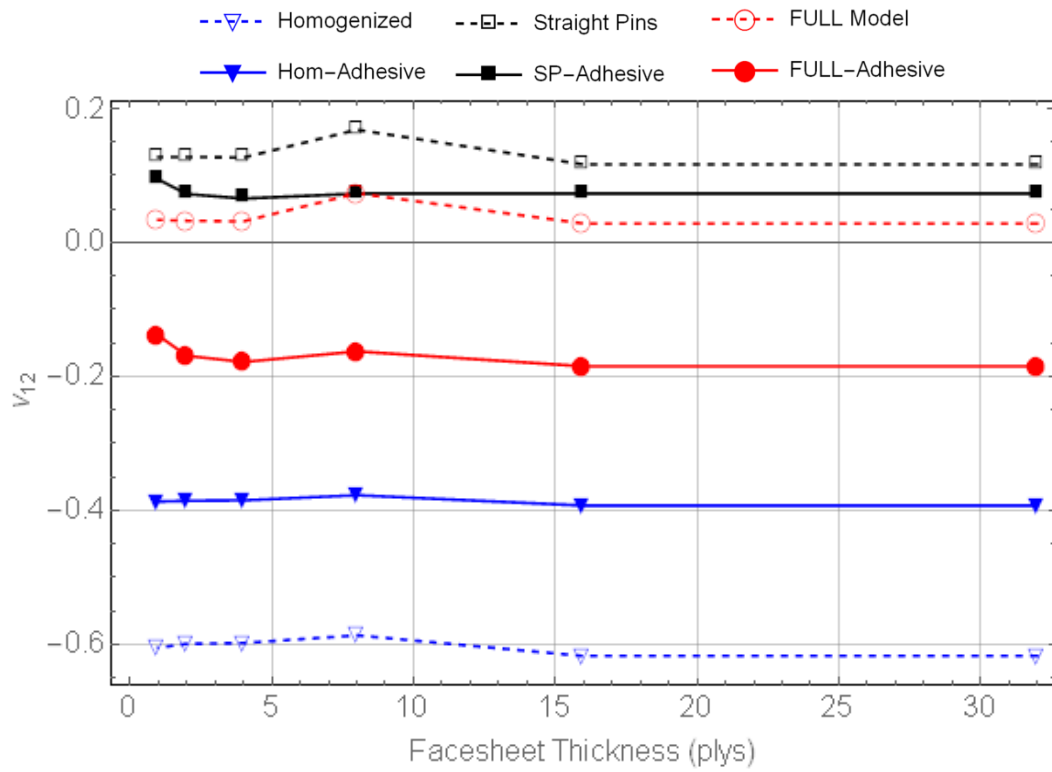
Figure 2.34: Plot of in-plane Poisson's ratio (ν12) as a function of facesheet thickness.
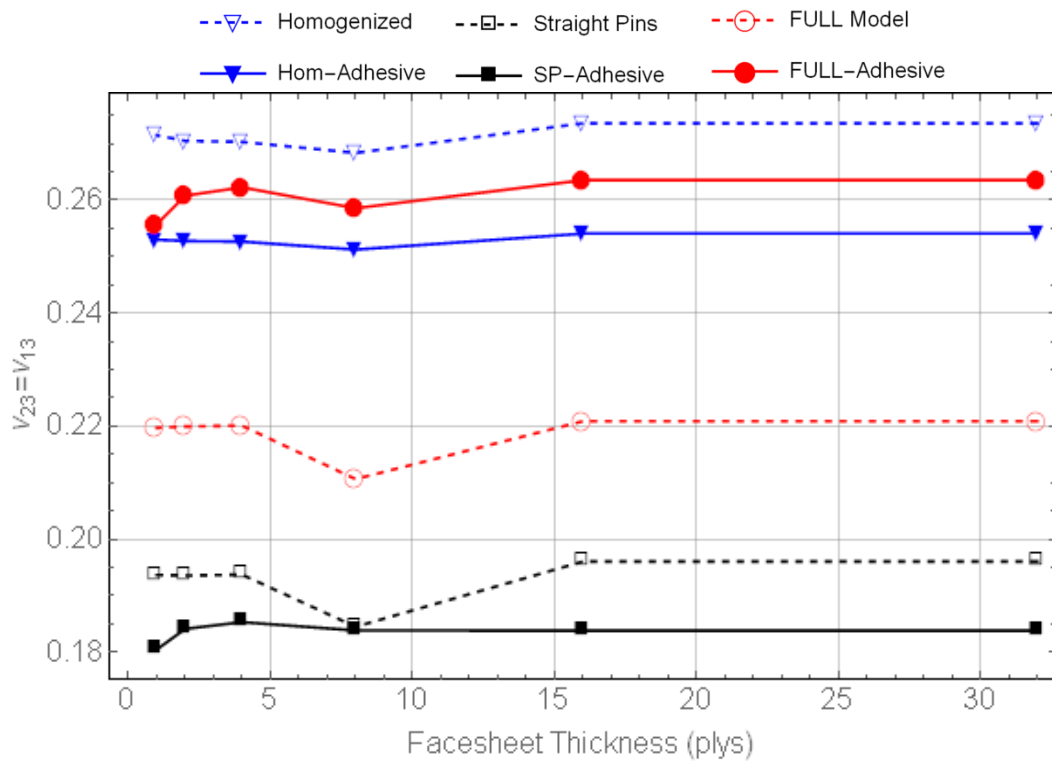
Figure 2.35: Plot of through thickness Poisson's ratio (ν23 = ν13) as a function of facesheet thickness.
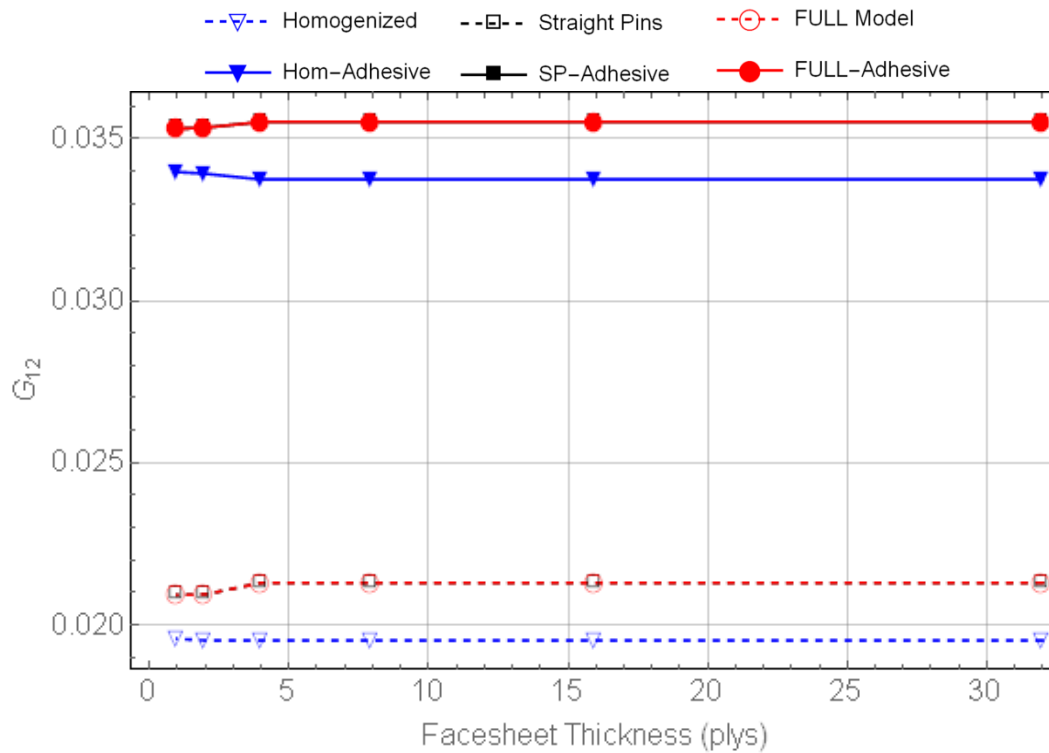
Figure 2.36: Plot of normalized in-plane shear modulus (G12) as a function of facesheet thickness.
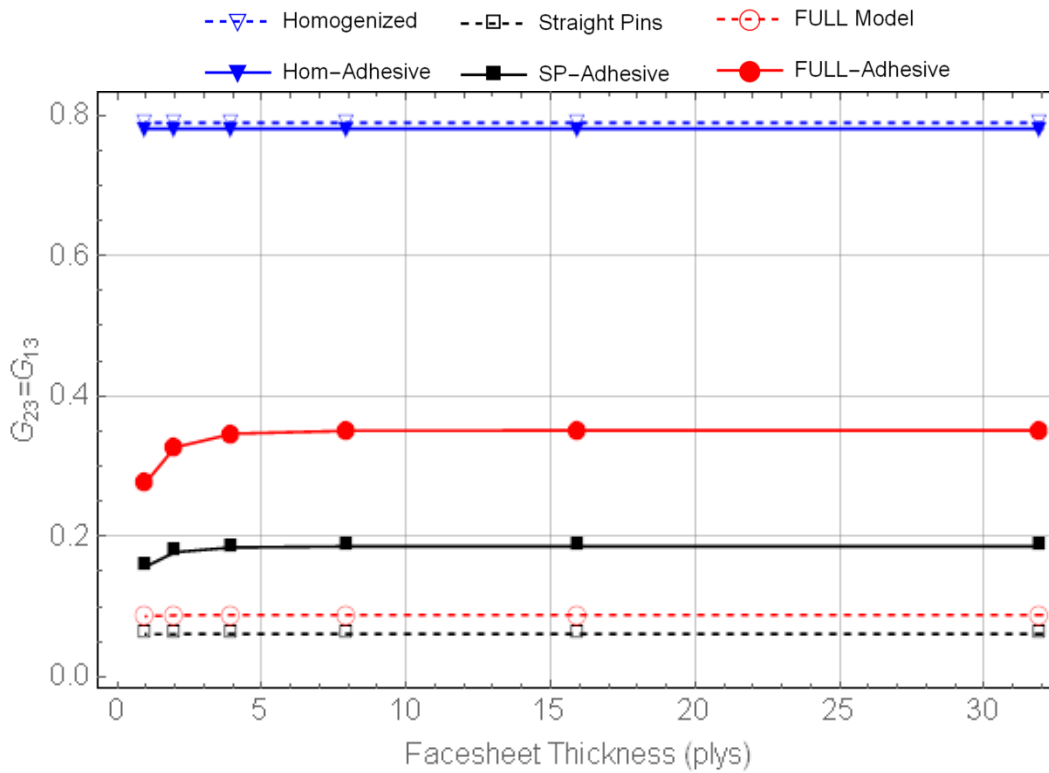
Figure 2.37: Plot of normalized through thickness shear modulus (G23 = G13) as a function of facesheet thickness.

2.4.4 Effective Edge Property in 3DFRFC

The highly discrete nature of the 3DFRFC coupled with its angled reinforcement results in a sizeable region near the edge of the specimen that is not fully bonded. While it is assumed that as a component gets larger that most of it should behave as the bulk 3DFRFC there will always be a region near the edge that will not behave as the bulk material. Fortunately, we can calculate the size of this area and determine an effective "edge" property for the material in this region. The length of the region affected by the cut edge can be determined geometrically based on the details of the specific 3DFRFC. The effective edge length is calculated as

$$t \tan \alpha \leq L_e \leq S + t \tan \alpha \tag{2.10}$$

Where t is the core thickness, α is the pin angle, and S is the pin spacing. These are illustrated in Figure 2.38. Based on the geometry of the 3DFRFC any material at least $L_e$max from an edge should behave like the bulk 3DFRFC material and can be assigned the bulk 3DFRFC properties. The microstructure behavior of the edge can then be accounted for by developing an effective edge property that already has the discrete nature near the edge in its formulation.
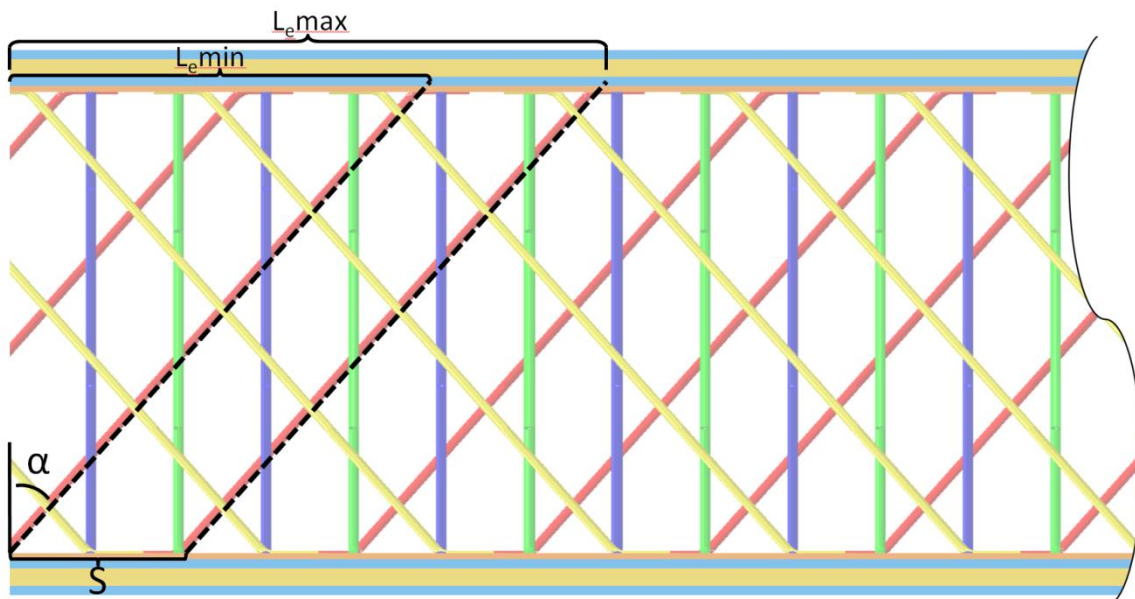
Figure 2.38: Side view of 3DFRFC illustrating effective edge length (L_e).
(Not to scale. Foam removed for clarity.)

36

In order to develop an effective edge property that can be implemented the generalized PBC code can be applied to the RVE with the cut reinforcement removed, Figure 2.39. This allows determination of the effective 3D engineering properties near an edge where only the reinforcement parallel to the cut contributes to carrying the load. For the 3DFRFC in this study $L_e$max is approximately equal to four RUCs so there is potential to account for the effect of the cut panel edges at the component level by replacing the material within four RUCs of an edge with the effective edge property. This allows for some of the effect due to the discrete edge to be accounted for without discreetly modeling every individual fiber in the component which quickly becomes untenable. To underscore this, the example 10m SLS fairing given earlier would contain $\sim 10^7$ RUCs requiring a minimum of $\sim 5 \times 10^8$ DOF just to model the truss members with single beam elements.
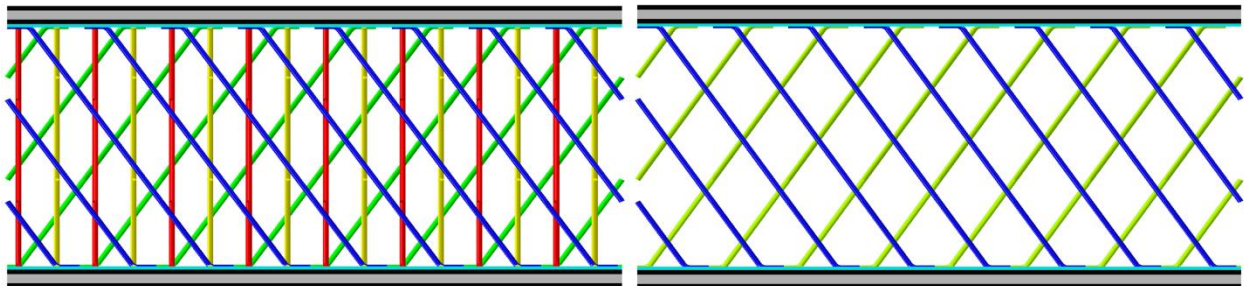
Figure 2.39: 3DFRFC model with full microstructure, left, and edge microstructure, right.
(Not to scale. Foam removed for clarity)

The effective edge properties are given in Table 2.2. The moduli are normalized by the bulk through thickness modulus and the 1-direction parallels the cut edge. Overall the moduli are reduced for the edge properties as to be expected due to the reduced number of members available for load transfer. The exception to this being G_{13} which increased due to the decoupling of the 1 and 2 directions. For the bulk case both of the out of plane shear directions load all of the pins; this is not the case for the edge model, resulting in a higher G_{13} and significantly lower G_{23}. The change in the structural coupling is mirrored by the shifting of the negative Poisson's ratio from ν_{12} to ν_{13}, albeit at a smaller magnitude.

Table 2.2. Effective normalized edge properties for 3DFRFC.

Property	Bulk 3DFRFC	Edge 3DFRFC
E_{11}(Pa/Pa)	0.219	0.159
E_{22}(Pa/Pa)	0.219	0.130
E_{33}(Pa/Pa)	1.000	0.685
ν_{12}	-0.178	0.303
ν_{23}	0.262	0.379
ν_{13}	0.262	-0.067
G_{12}(Pa/Pa)	0.043	0.043
G_{23}(Pa/Pa)	0.421	0.027
G_{13}(Pa/Pa)	0.421	0.642

2.5 Summary

The development of modeling methods to obtain the effective elastic properties of 3DFRFC sandwich composites was presented. Key findings included:

- Details of the 3DFRFC microstructure were measured using microCT.
- Detailed embedded element models of the 3DFRFC microstructure were created using a parametric python script developed to automate this task.
- Six degree-of-freedom periodic boundary conditions were developed that couple rotational degrees of freedom across periodic boundaries allowing for periodic models of 3DFRFCs with beam elements to be developed.
- Surface averaged boundary conditions were developed for the through thickness direction in 3DFRFCs allowing for determination of full effective three-dimensional properties of the core without introducing additional constraints.
- The effective engineering properties of 3DFRFC were found to be highly sensitive to the location of the pin within the adhesive layer.
- Facesheet thickness was found to have a minimal impact on the effective engineering properties with the caveat that the findings may vary if the pin placement was also varied.
- An effective edge property was introduced by looking at the limiting case of a thin strip of 3DFRFC where there are no fibers carrying load orthogonal to the edge.

CHAPTER 3
Design of Interface Fracture Test

3.1 Introduction

Composite sandwich structures provide distinct advantages in aerospace, automotive, and construction industries, affording high specific stiffness compared to metallic components. A particular challenge of utilizing sandwich structures is their sensitivity to manufacturing induced defects, damage, and core-to-facesheet delamination. The ability to assess the residual load carrying capability of sandwich components with such features requires extensive full-scale test programs, detailed and thorough analysis, or likely some combination of the two. The current emphases on cost-reduction tends to shift focus toward less cost-intensive simulation; however, the heavy reliance on simulation and computational analysis requires more careful thought into designing the coupon level tests, which are conducted to acquire the material properties necessary to perform the desired full-scale analysis. This has long been a challenge in determining the fracture properties for composite structures and is further compounded with the addition of bonded cores in sandwich structures. Determining the appropriate method for measuring the facesheet-to-core interface fracture properties of sandwich composites continues to be a challenge as the development of new types of sandwich core materials persists.

One emerging class of materials has been developed with the potential to affect the ability of sandwich structures to tolerate manufacturing induced defects, damage, and core-to-facesheet delamination. 3D Fiber Reinforced Foam Core (3DFRFC) represents a class of sandwich core materials that consist of low density structural foam reinforced with a three-dimensional, truss-like fiber composite structure that provides added load paths between the facesheets of the composite sandwich and acts to impede crack propagation within the foam. The 3DFRFC architecture can be quite varied through the selection of the reinforcing fiber (glass, carbon, Kevlar®, Spectra®, etc.), foam material, foam density, and matrix material; in addition to the

overall geometry of the reinforcing truss itself. Some examples of 3DFRFCs include NidaFusion[1,2], TYCOR®[3], and K-Cor [4]. The added complexity of 3DFRFC sandwich structures makes the prediction of the global response of full-scale components exceedingly difficult particularly when those structures contain manufacturing defects or damage. The ability to design structures with this class of sandwich materials and predict their performance requires an adequate understanding of the constituent interaction and an ability to quantify their damage tolerance.

This research aims to develop test methods to experimentally quantify the effective fracture properties of the bonding interface between the core and facesheet in a 3DFRFC sandwich composite. Due to the complexity of the 3DFRFC, a thorough investigation of the test geometry of the test specimens is performed in support of the experimental investigation of the fracture properties of the 3DFRFC sandwich specimens.

3.2 Development of Interface Fracture Tests

The bulk mechanical properties of 3DFRFC sandwich structures are dependent on a variety of factors including the facesheet properties, the foam core properties, the geometry and mechanical properties of the reinforcement within the foam core, and the properties of the adhesive that bonds the facesheets to the core. The interaction of these constituents at the bonding interface is critical to the understanding of the limits of 3DFRFC sandwich structures and is necessary for the development of predictive failure models. The quantification of the fracture properties of the bonded interface is key to this process. There has been substantial effort into quantifying the Mode I and Mode II fracture properties of the bonding interface in foam and honeycomb core sandwich structures [40–47]. Many of these approaches are based on various modifications to the Double Cantilever Beam (DCB) test and the End-Notched Flexure (ENF) test to measure the Mode I and Mode II fracture properties, respectively. Many of these methods focus on modifications to the loading boundary conditions in an attempt to compensate for the inherent mode mixity that arises from having an offset between neutral axis of the sandwich beam and the intended crack path along the bonding interface. Another approach has been to retain the standard boundary conditions and account for the mode mixity due to the lack of symmetry in the specimen in order to get the relative Mode I and Mode II contributions as is done with the Unsymmetrical Double Cantilever Beam (UDCB) [48] and the Unsymmetrical

40

End Notch Flexure (UENF) [49]. One key limitation of the UDCB and UENF tests is that they do not allow for direct measurement of the Mode I and Mode II critical energy release rates.

A simpler approach published by Davidson et al.[50] is used in this study to design test specimens to determine the effective fracture properties. This is accomplished by designing the specimens so that the neutral axis is coincident with the bonded interface between the facesheet and the core. The specimens are designed such that the neutral axis of the sandwich is collocated with the adhesive interface by bonding an aluminum facing to the facesheet nearest the interface to be tested. Illustrations of the resulting Bonded Double Cantilever Beam (BDCB) and Bonded End-notched Flexure (BENF) samples can be seen in Figure 3.1 and Figure 3.2 where the initial delamination is shown on the right of the samples at the interface between the upper composite facesheet and the 3DFRFC.
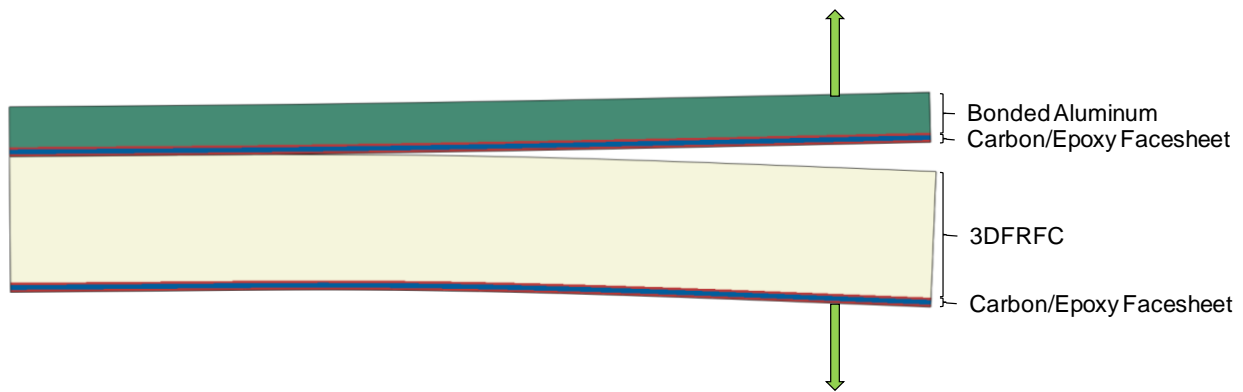
Figure 3.1. Illustration of 3DFRFC BDCB sample.
(Not to scale.)
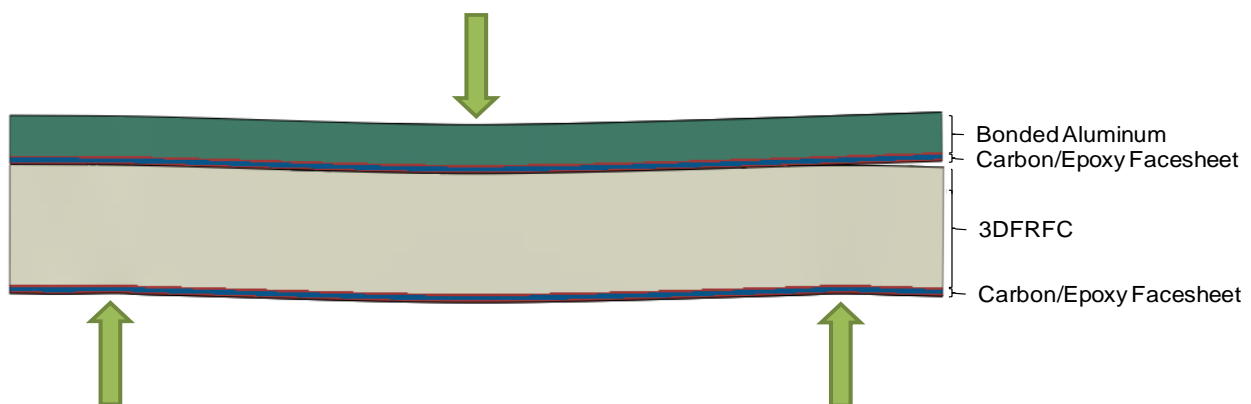
Figure 3.2. Illustration of 3DFRFC BENF sample.
(Not to scale.)

The BDCB and BENF samples were analyzed in order to determine the validity of the experimental method once applied to sandwich composites taking into account the highly

orthotropic core properties unique to the 3DFRFC. This was accomplished using finite element analysis using the commercial finite element software ABAQUS. The facesheet plies were individually modeled with brick elements, and the adhesive layer between the facesheet and the 3DFRFC was modeled with decohesion elements. The initial strength and fracture properties for the interface were based on measured properties for a foam core with equivalent bulk density to the 3DFRFC as an approximate lower bound [51]. The facesheet properties were measured through testing at The Aerospace Corporation [52], while the mechanical properties of the film adhesive were obtained from vendor data [53]. The effective orthotropic core properties were determined utilizing a micromechanics model recently developed for metallic and pin reinforced foams [33] that model the reinforcements within the foam as beams on an elastic foundation to derive the effective orthotropic elastic properties for the 3DFRFC based on its specific microstructure and material composition.

One challenge of applying methods developed for foam core sandwich structures to the 3DFRFC is that decohesive zones yield accurate results when the crack plane is well defined [54–56], but lose fidelity as the material becomes more discretized and the crack path can no longer be inferred a priori. The added paths for load transfer in the 3D fiber reinforced foam core act to impede crack propagation within the foam and the use of decohesive elements to model this material will likely become less accurate and unable to capture the highly discretized nature of the 3DFRFC particularly in situations with more complex loading. In the current study, the discrete cohesive zone model (DCZM) pioneered by Xie and Waas [56] is used for modeling delamination between the 3DFRFC and the facesheets due to its increased modeling efficiency.

Finite element analysis of the Mode I modified double cantilever beam specimen using the homogenized orthotropic 3DFRFC material properties resulted in stable crack propagation in Mode I, as intended, Figure 3.3. In Figure 3.3 SERR is the state variable showing the amount of failure in the element where SERR=1 corresponds to complete failure of the element. While this first step analysis did not account for any additional effects as a result of the discrete nature of the core reinforcement it did illustrate that the global specimen design had the capability to create the conditions for the desired Mode I propagation and was a viable candidate for preliminary testing and further analysis.
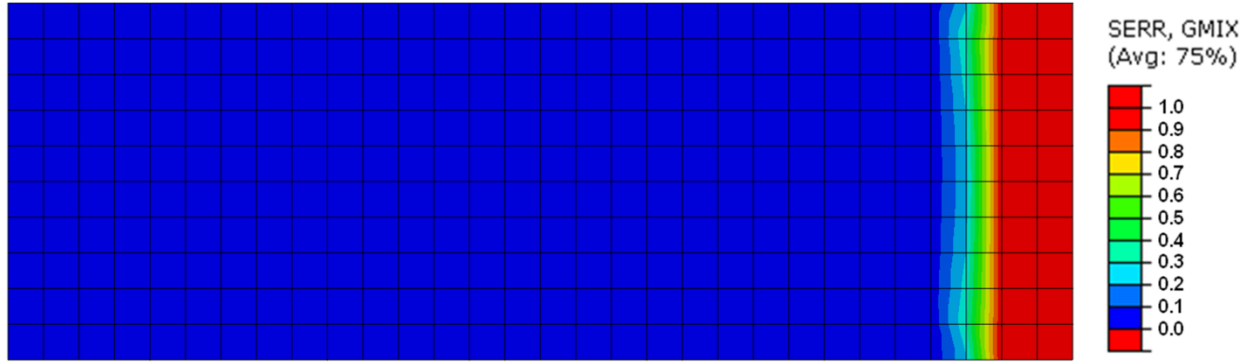
Figure 3.3: Illustration of fracture surface in BDCB model.
(Not to scale.)
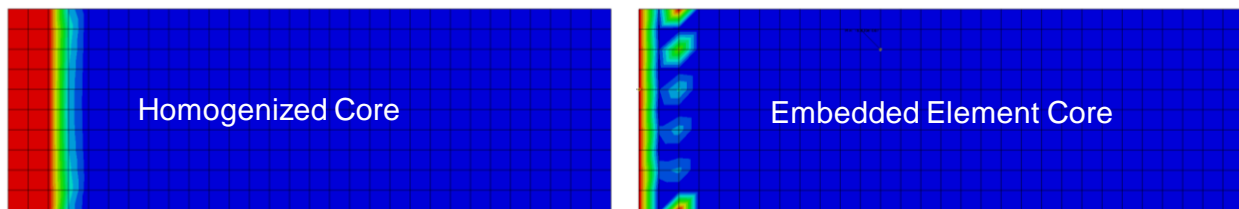
Figure 3.4: Illustration of fracture surface in BDCB models with homogenized orthotropic core,
left, and the discrete embedded element core, right.
(Not to scale.)

The embedded element modeling methods within Abaqus were also utilized to gain a greater understanding of the behavior of the 3DFRFC fracture specimens. Initial work in this area has been focused on Mode I DCB specimens, given their more predictable and stable behavior exhibited with the homogenized orthotropic core. Initial work replaced the homogenized orthotropic core with an isotropic foam core and embedded beam elements while still using DCZM elements for the interface between the facesheet and core. The homogenized core model exhibits smooth crack propagation as is expected for homogenized materials but would not be expected experimentally in a material with coarse discrete reinforcement such as 3DFRFC, Figure 3.4. The embedded element model captures the damage localization due to the stress concentrations caused by the discrete reinforcements within the core. This method, however, has convergence challenges and has limited ability to capture the interaction of the failure of the various constituents within the 3DFRFC.

Initial results for the Mode II BENF specimen highlighted some additional challenges. Unlike the BDCB model, the analysis of the BENF specimen did not yield crack propagation in the desired shear mode. The behavior of the 3DFRFC near the crack front was more complex in the BENF model, resulting in a large region failing in Mode I near the center of the specimen

ahead of the initial crack, Figure 3.5. Additional analysis was conducted to verify whether this phenomenon was a result of the unique orthotropic properties of the 3DFRFC, or if the basic specimen design or modeling parameters were faulty. Additional analyses were conducted by replacing the orthotropic 3DFRFC core properties in the BENF model with an equivalent isotropic core having the same effective in-plane stiffness as the 3DFRFC. This resulted in crack propagation with the failure occurring primarily in Mode II, as intended, thus exposing the interaction of the highly orthotropic properties of the 3DFRFC as the underlying cause for the change in failure mode from design. The unique interaction of the 3DFRFC material near the crack tip is most clearly highlighted by a comparison of the lateral deformation of the BENF model with the full equivalent (homogenized) orthotropic properties to that of the BENF model with an equivalent isotropic core at the same load point deflection, Figure 3.6. This comparison shows that the isotropic core exhibits a small amount of lateral expansion near the crack tip (yellow) whereas the model with the full orthotropic properties demonstrates a significant amount of lateral contraction (blue). This structural response is a direct result of the inherent truss structure of the 3DFRFC and results in the localized Mode I behavior of the material despite the global loading conditions. As a result of these findings, additional analysis was conducted to investigate other loading methods as well as the effect of material orientation on the local material response.
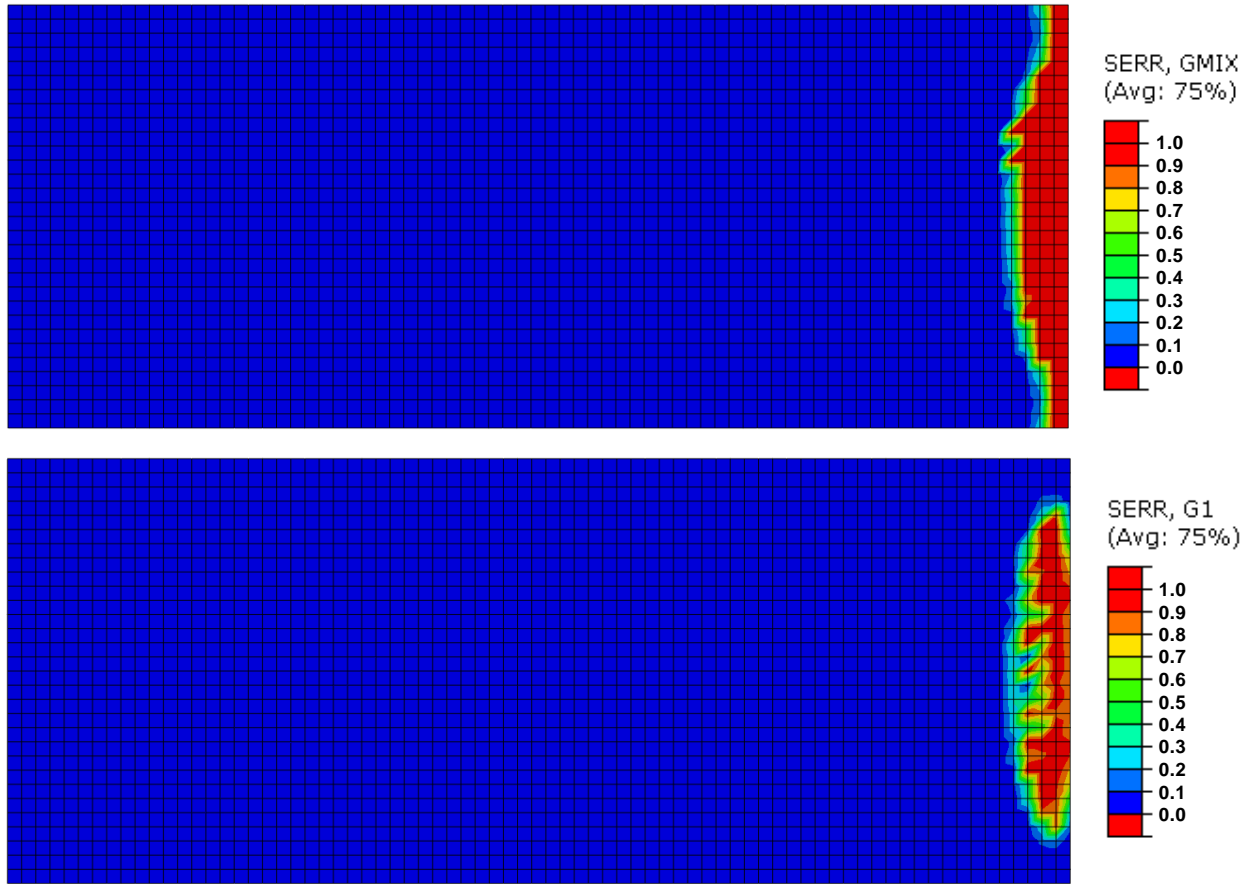
Figure 3.5: Illustration of fracture surface of BENF model with full orthotropic 3DFRFC properties, top, and area failing in Mode I, bottom.
(Not to scale.)
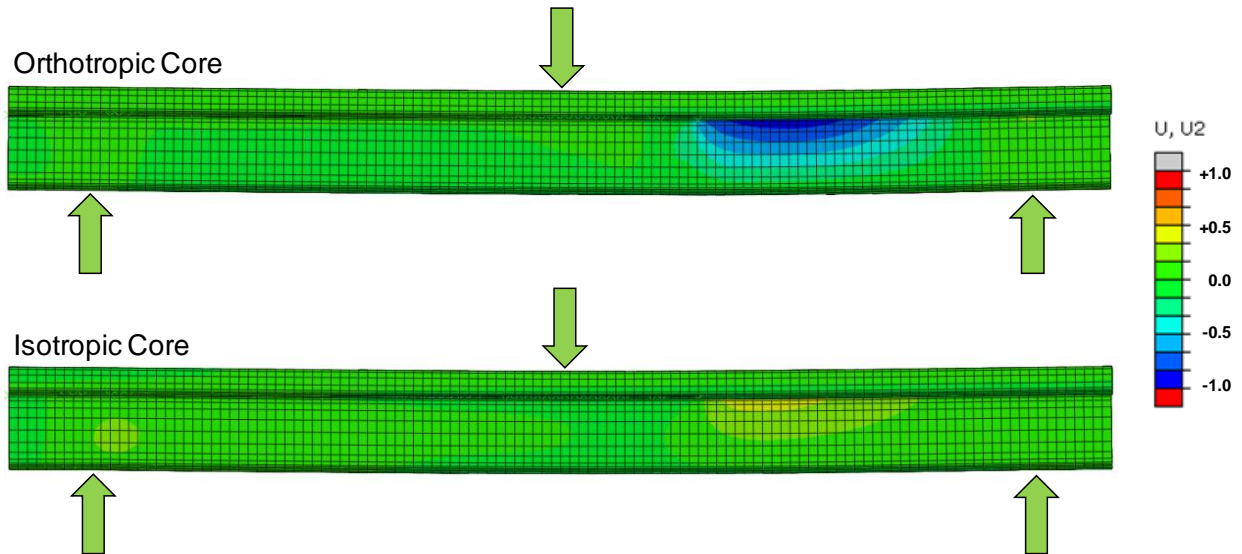
Figure 3.6: Normalized lateral displacement of BENF models with fully orthotropic 3DFRFC properties and equivalent isotropic properties at the same load-point displacement.
(Not to scale.)

Several possible solutions to alleviate the Mode I contribution in the Mode II tests have been considered including changes in material orientation in the BENF configuration or changing the global boundary conditions to introduce the shear loading through a different loading geometry such as that used in the end-loaded split (ELS) test [57]. A simpler solution was investigated by retaining the 3-point loading configuration but flipping the BENF sample such that the central loading point is contacting the facesheet opposite of the fracture surface and the side supports are contacting the aluminum facing, Figure 3.7. Initial finite element analysis of this flipped BENF configuration has shown the desired Mode II fracture propagation with essentially no Mode I contribution, Figure 3.8. This result illustrates that the flipped BENF specimen design has the capability to create the conditions for the desired Mode II propagation and is a viable candidate for preliminary testing and further analysis. Additional analysis is needed to investigate the sensitivity of the configuration to the interface fracture parameters. This configuration does present additional challenges due to the central loading point being located directly on the facesheet. This will have to be addressed in order to insure that localized core crushing and/or facesheet wrinkling does not affect experimental work based on the flipped BENF configuration.
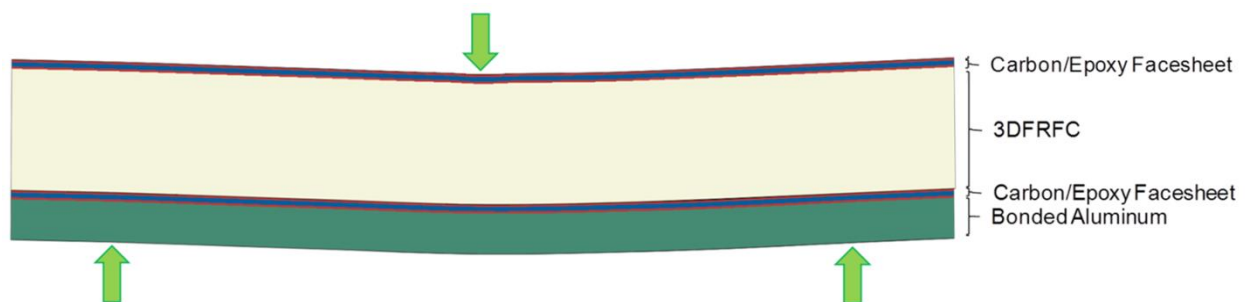
Figure 3.7: Illustration of 3DFRFC flipped bonded end–notched flexure sample.
(Not to scale.)
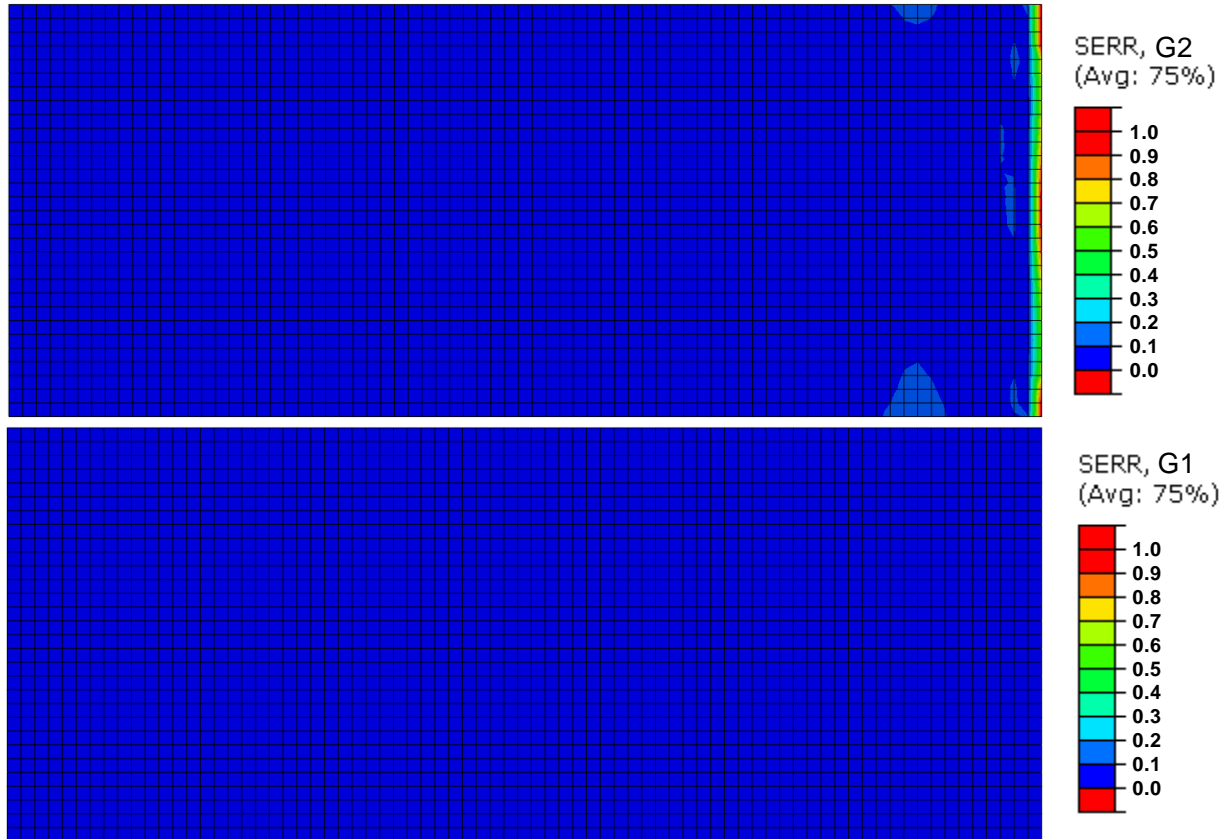
Figure 3.8: Illustration of fracture surface of flipped BENF model with full orthotropic 3DFRFC properties, Mode II top, and Mode I, bottom.
(Not to scale.)

Additionally, a smeared crack model [58] was investigated for its ability to handle the more complex fracture interaction due to the presence of the discrete fiber reinforcement. This method provides additional benefit for modeling failure under more complex loading where the failure may not be restricted to a specific material plane. Due to the relatively high fidelity inherent in modeling the discrete microstructure of the 3DFRFC's it was quickly discovered that using this method to attempt to model failure within the entire sample becomes computationally prohibitive for models with multiple hundreds of RUCs. This method is explored in a limited context in Chapter 5; however, it does show promise for developing an effective failure envelope when coupled with the smaller periodic models used in Chapter 2. This is part of the future work discussed in Chapter 7.

3.3 Manufacture of Interface Fracture Samples

The general manufacturing procedure for the 3DFRFC interface fracture samples is similar to the one discussed previously for the manufacture of edgewise compression samples with defects [59] but is included here for completeness. The material system chosen for this investigation is IM7/8552 carbon epoxy for the facesheets and a 19mm (0.75inch) thick 192kg/m^3 (12lb/ft^3) 3DFRFC for the core. FM® 300 film adhesive is used to bond the facesheets to the core. The desired debonds are manufactured by removing a region of adhesive and replacing it with a PTFE insert. The panels are inspected via NDE to ensure panel quality and to verify debond placement in the cured sandwich panel prior to removing the desired samples from the fabricated panels, Figure 3.9. The manufactured 3DFRFC sandwich panels are then cut into samples for aluminum bonding prior to fracture testing, Figure 3.10. Note the use of FM® 300 in these samples is a departure from the other sandwich composites made in this dissertation. FM® 300 was the only material that could be acquired at the time of the fracture specimen fabrication. FM® 300 has similar properties to the AF-191 used throughout the rest of the dissertation and it was determined to be more advantages to proceed with fracture testing to prove the viability of the test.
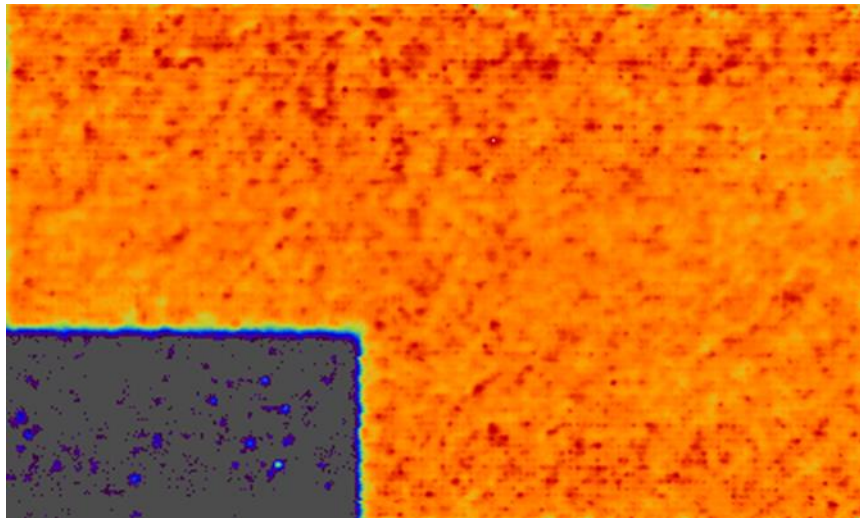
Figure 3.9: Ultrasonic through transmission of 3DFRFC panel containing Mode I fracture test specimens.
(PTFE insert in grey. Image used with the permission of The Aerospace Corporation.)

Figure 3.10. Fracture sample for measuring 3DFRFC properties.
(Not to scale.)

During testing, additional insight into the material behavior and the interaction of the discrete constituents of the 3DFRFC was gained through the use of the digital image correlation (DIC) capabilities of the Composite Structures Laboratory at the University of Michigan. This capability allows for the mapping of the two dimensional strain fields on the surface of the specimen throughout the test. This capability provides critical insight and allow for a deeper understanding of the microstructure interaction of the 3DFRFC and aid in the development of models capable of capturing this interaction.

Due to the complexity of the 3DFRFC, there are additional challenges to consider when considering test methods for determining the bulk fracture properties. The highly discretized nature of the reinforcement within the core results in a significant region of the material near the edges with partially bonded fibers. These severed reinforcements can no longer transfer load and are not representative of the bulk material. In order to determine the effective bulk properties, it is recommended that tests be conducted on three specimen sizes and the effective fracture properties determined using two methods. The first method uses the first two test sizes and backs out the critical energy release rate of the bulk by assuming that the total energy release rate can be calculated as an area weighted average of the energy release rate values for the partially bonded region and the fully bonded region. The third test size is used to validate this measurement. The second method bases the calculations on number of pins fully bonded for each sample to determine an effective area.

3.4 Test Results – Original Design

Preliminary testing conducted on the bonded Double Cantilever Beam (BDCB) specimens highlighted an additional failure mode that was not captured in the analysis, Figure 3.11. The specimens failed with the formation of a through thickness fracture followed by crack propagation within the adhesive interface between the facesheet and the opposing side of the core. Detailed analysis of the local strain fields within the core just prior to failure show the location of the failure initiation away from the crack tip, Figure 3.12. This failure highlights the weakness of the 3DFRFC under bending when the reinforcement ends are unbonded. This is an important consideration when analyzing structures with defects where a region of the core may contain unbonded reinforcement. Additional analysis was conducted to evaluate methods for eliminating this failure mode and obtaining the fracture properties.
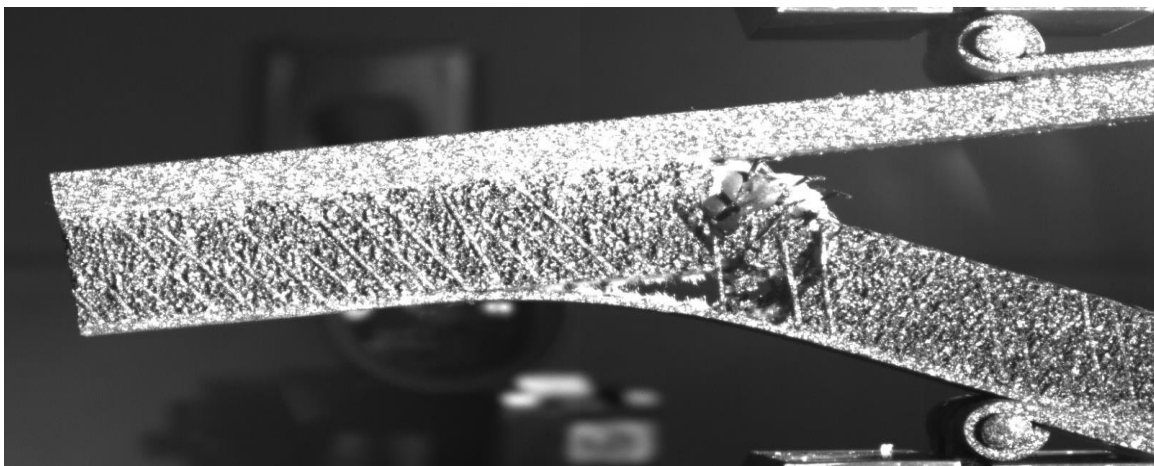
Figure 3.11. Fracture propagation in bonded DCB samples.
(Not to scale.)
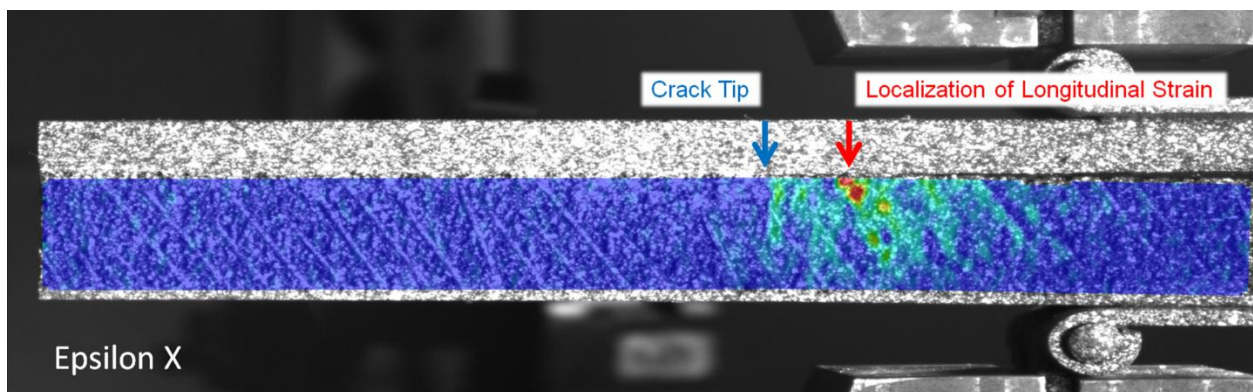
Figure 3.12. Longitudinal strains in bonded DCB sample prior to failure.
(Not to scale.)

3.5 Redesign of Interface Fracture Test

The original BDCB specimen design was based on matching the bending rigidity of the two beams as had successfully been done for other material systems [50]. A couple corrective options were considered in light of the through-the-thickness failure observed. First the stiffener could simply be placed on the opposite facesheet. Once the crack reached the opposite surface in the previous tests it was shown to propagate within the adhesive interface as desired, but at the opposing interface. The problem with this approach is that the crack will not necessarily be a pure Mode I crack. In order to reduce the likelihood of tensile failure of the core and reduce or eliminate any Mode II contribution, the specimen was redesigned to match the longitudinal strains in the upper and lower arms within the cracked region, effectively resulting in the changes depicted in Figure 3.13
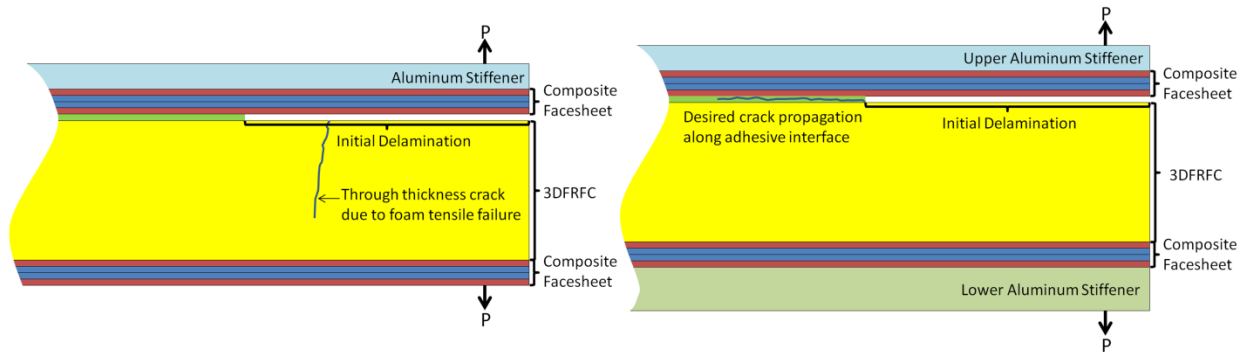
Figure 3.13. Changes between two BDCB configurations.

3.5.1 Designing Specimen to Match Axial Strains in Cracked Region

In order to determine the required upper and lower stiffeners to achieve the desired matching strain states classical lamination theory is used to calculate the axial strain in each arm of the BDCB specimen [60]. The composite plate stiffness (ABD) is given in the form

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \qquad (3.1)$$

The ABD matrices are derived for the upper and lower arms by setting z=0 to be coincident with the crack plane which means that ε_x° is specifically defined as the longitudinal strain at the crack surface. Solving the equation for when the axial strain in both the upper and lower arms are equal can easily be done since the stiffness equations were derived about the crack plane. Inversion yields

$$\begin{Bmatrix} \varepsilon \\ \kappa \end{Bmatrix} = \begin{bmatrix} A & B \\ B & D \end{bmatrix}^{-1} \begin{Bmatrix} N \\ M \end{Bmatrix} = \begin{bmatrix} A' & B' \\ B' & D' \end{bmatrix} \begin{Bmatrix} N \\ M \end{Bmatrix} \qquad (3.2)$$

where from [32]

$$[A'] = [A]^{-1} + [A]^{-1}[B]([D] - [B][A]^{-1}[B])^{-1}[B][A]^{-1}$$
$$[B'] = -[A]^{-1}[B]([D] - [B][A]^{-1}[B])^{-1} \qquad (3.3)$$
$$[D'] = ([D] - [B][A]^{-1}[B])^{-1}$$

3.2 can be written expanded as

$$\begin{Bmatrix} \varepsilon_x^\circ \\ \varepsilon_y^\circ \\ \gamma_{xy}^\circ \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} = \begin{bmatrix} A'_{11} & A'_{12} & A'_{16} & B'_{11} & B'_{12} & B'_{16} \\ A'_{12} & A'_{22} & A'_{26} & B'_{12} & B'_{22} & B'_{26} \\ A'_{16} & A'_{26} & A'_{66} & B'_{16} & B'_{26} & B'_{66} \\ B'_{11} & B'_{12} & B'_{16} & D'_{11} & D'_{12} & D'_{16} \\ B'_{12} & B'_{22} & B'_{26} & D'_{12} & D'_{22} & D'_{26} \\ B'_{16} & B'_{26} & B'_{66} & D'_{16} & D'_{26} & D'_{66} \end{bmatrix} \begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{Bmatrix} \qquad (3.4)$$

Just behind the crack (in the cracked region) the only external load on the system is the applied moment due to the external load applied to the BDCB specimen, M_x, yielding

$$
\begin{Bmatrix} \varepsilon_x^{\circ} \\ \varepsilon_y^{\circ} \\ \gamma_{xy}^{\circ} \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} = \begin{bmatrix} A'_{11} & A'_{12} & A'_{16} & B'_{11} & B'_{12} & B'_{16} \\ A'_{12} & A'_{22} & A'_{26} & B'_{12} & B'_{22} & B'_{26} \\ A'_{16} & A'_{26} & A'_{66} & B'_{16} & B'_{26} & B'_{66} \\ B'_{11} & B'_{12} & B'_{16} & D'_{11} & D'_{12} & D'_{16} \\ B'_{12} & B'_{22} & B'_{26} & D'_{12} & D'_{22} & D'_{26} \\ B'_{16} & B'_{26} & B'_{66} & D'_{16} & D'_{26} & D'_{66} \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ M_x \\ 0 \\ 0 \end{Bmatrix}
\tag{3.5}
$$

Subsequently the longitudinal strain is directly given for the upper and lower arm as

$$
\varepsilon_{x\,Lower}^{\circ} = B'_{11\,Lower} M_x
$$
$$
\varepsilon_{x\,Upper}^{\circ} = B'_{11\,Upper} M_x
\tag{3.6}
$$

Solving for when the longitudinal strain in upper and lower arms is equal yields

$$
\varepsilon_{x\,Upper}^{\circ} = \varepsilon_{x\,Lower}^{\circ}
$$
$$
\therefore \ B'_{11\,Upper} = B'_{11\,Lower}
\tag{3.7}
$$

Inherently $B'_{11\,Upper}$ and $B'_{11\,Lower}$ are functions of the sandwich constituents and the upper and lower stiffeners. For constant sandwich values $B'_{11\,Upper}$ and $B'_{11\,Lower}$ can be plotted as functions of the upper and lower stiffener thickness. The desired solutions are given as the intersection between the two surfaces, Figure 3.14
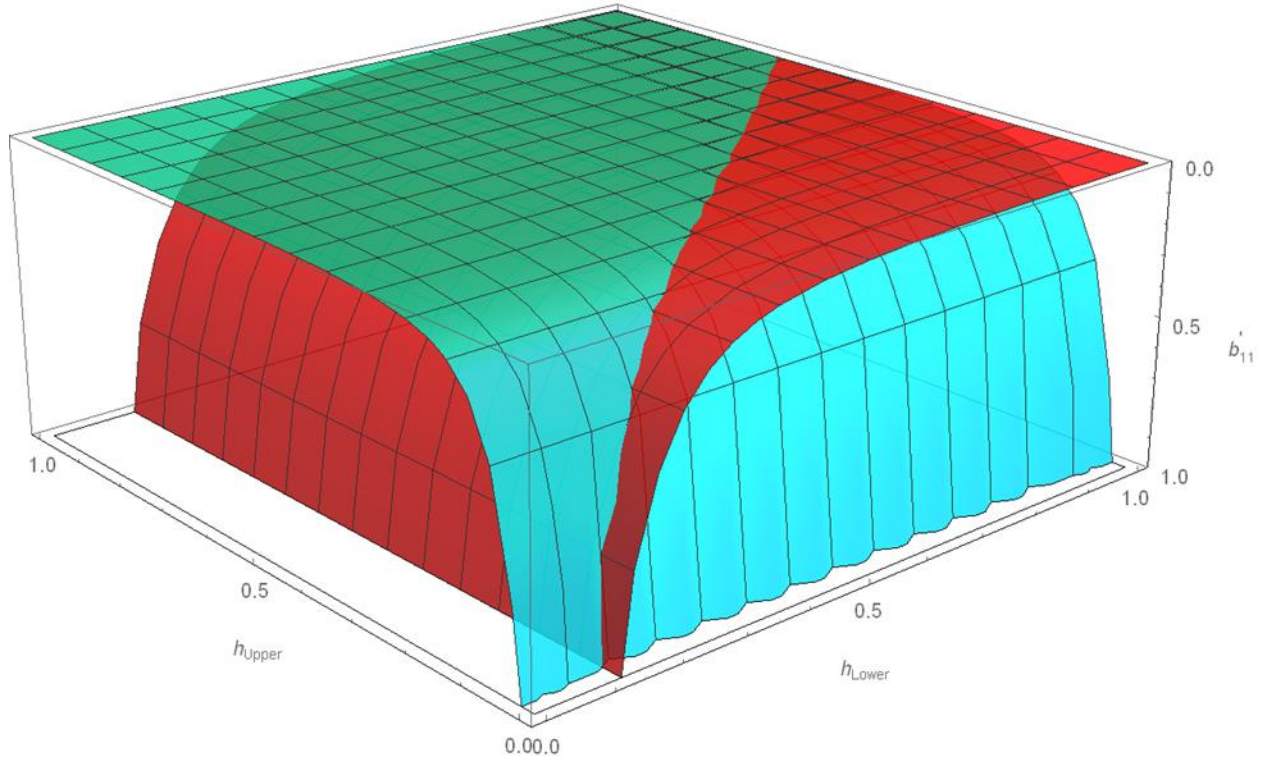
Figure 3.14. Plot showing intersection of B'$_{11_{Upper}}$ and B'$_{11_{Lower}}$ surfaces.

3.5.2   Specimen Fabrication

Manufacturing of the new version of the BDCB specimens included a couple key changes based on lessons learned from the previous version. First the exposed edges of the 3DFRFC were taped with high temp flash tape to prevent a buildup of excess adhesive on the sides of the sample, Figure 3.15. Second, after the aluminum was bonded to the composite with Loctite® H8000 additional tape was placed laterally (shown perpendicular to specimen in Figure 3.15). This was to prevent the aluminum from shifting on the sandwich once the vacuum was applied, Figure 3.16. The finished specimens did have some adhesive that seeped under the tape and into the edge cells of the 3DFRFC, but this was shown to not significantly impact the specimen behavior, Figure 3.17. This could be minimized if a film adhesive is used or eliminated if the stiffeners are added during the original fabrication, but this can introduce additional fabrication difficulties, chiefly delamination during cutting.

Figure 3.15. Image from BDCB fabrication showing taping of 3DFRFC edges.
(Not to scale.)

Figure 3.16. Vacuum bagged BDCB specimens.
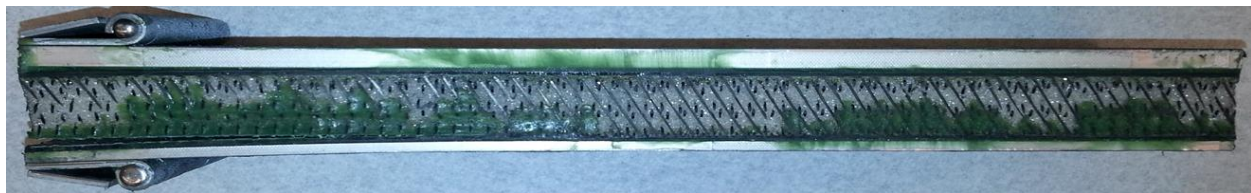
Figure 3.17. Finished BDCB specimen.
(Not to scale.)

3.5.3 Test Results

The redesigned BDCB specimens were shown to produce the desired crack propagation within the adhesive layer, Figure 3.19. While four specimens were fabricated using this new design philosophy they represented two possible solutions. The first two represented the minimum solution with only one stiffener required. Both of these designs produced stable crack propagations within the adhesive layer, however the first two samples were found to be entirely too compliant and could not be completely failed using the entire stroke of the test frame. While these specimens did fail as designed they did not provide useful data. The other two test specimens represented a stiffer solution and demonstrated a useful Mode I test for the 3DFRFC. All of the specimens exhibited some degree of pin pullout, Figures 3.20 and 3.21. This behavior varied from less than 0.1% to 2% and is likely do to a combination of variance in the bond quality on the two ends of the pin and the actual bonding area of each pin end itself. The facesheet side of the fracture surface exhibited some of the carbon fibers from the pin feet and clusters of fractured foam cell walls, Figure 3.22.
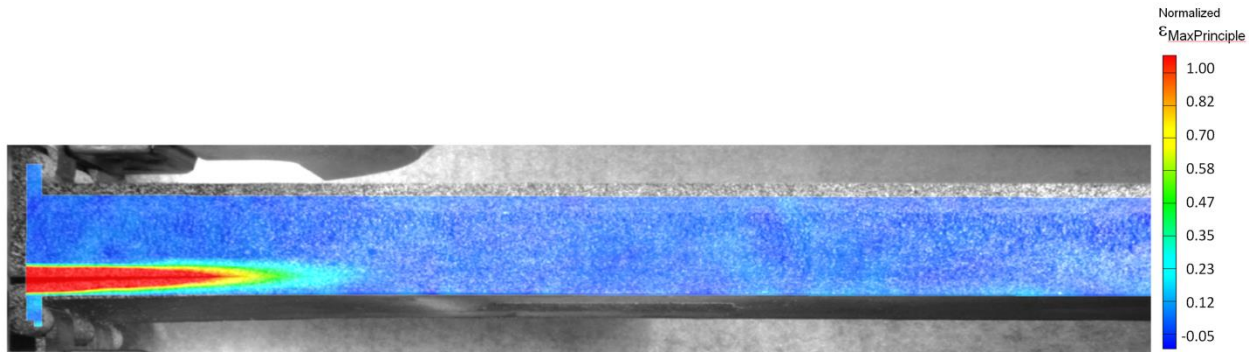
Figure 3.18. Image from BDCB test showing major strain from DIC prior to crack propagation. (Not to scale.)
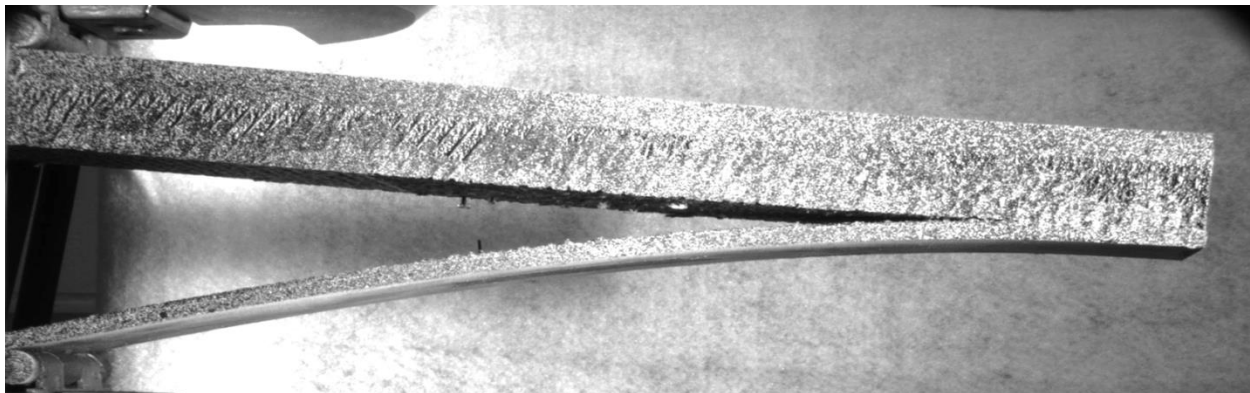
Figure 3.19. Image from BDCB test showing crack propagation along the adhesive interface. (Not to scale.)

Figure 3.20. BDCB fracture surface showing pin pulling out during test.
(Not to scale.)

Figure 3.21. BDCB fracture surface (core side) showing higher percentage of partial pin pullout.
(Not to scale.)
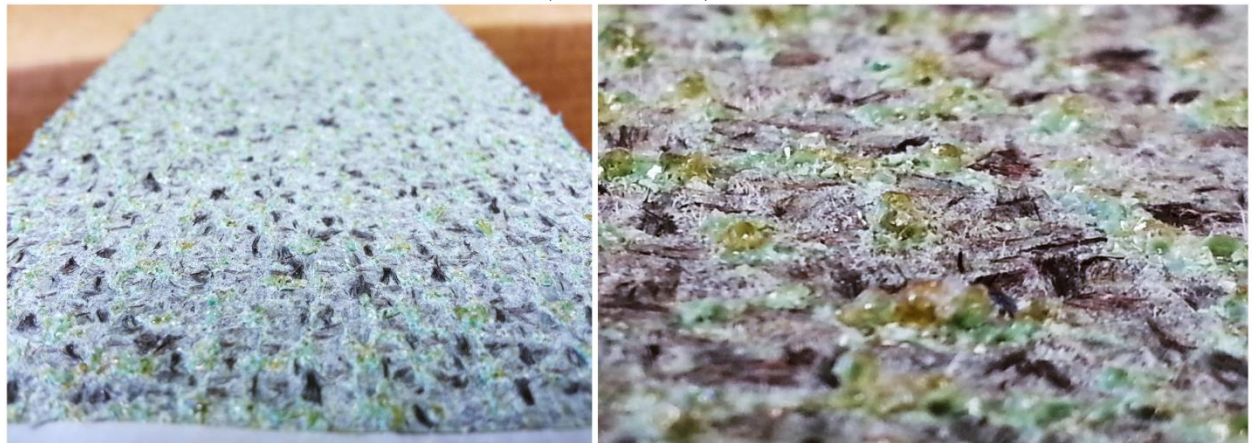
Figure 3.22. BDCB fracture surface (facesheet side) showing fractured foam cells.
(Not to scale.)

The load vs displacement and crack length vs displacement for the two redesigned BDCB specimens are given in Figures 3.23-3.24. The displacement and crack lengths are given in terms of RUC length. The vertical green line indicates when the crack is at the end of the Teflon insert. Specimen B had more adhesive flow under the protective flash tape resulting in more nonlinearity before the crack reached the end of the insert. The crack and load curves were used to calculate the delamination resistance curves using the Compliance Calibration (CC), Modified Beam Theory (MBT), and Modified Compliance Calibration (MCC) methods per ASTM D5528 [61], Figures 3.25-3.26. The measured fracture toughness from these preliminary tests was found to fall in the high range of the scatter of previously tested FM300 honeycomb sandwich structures [62].
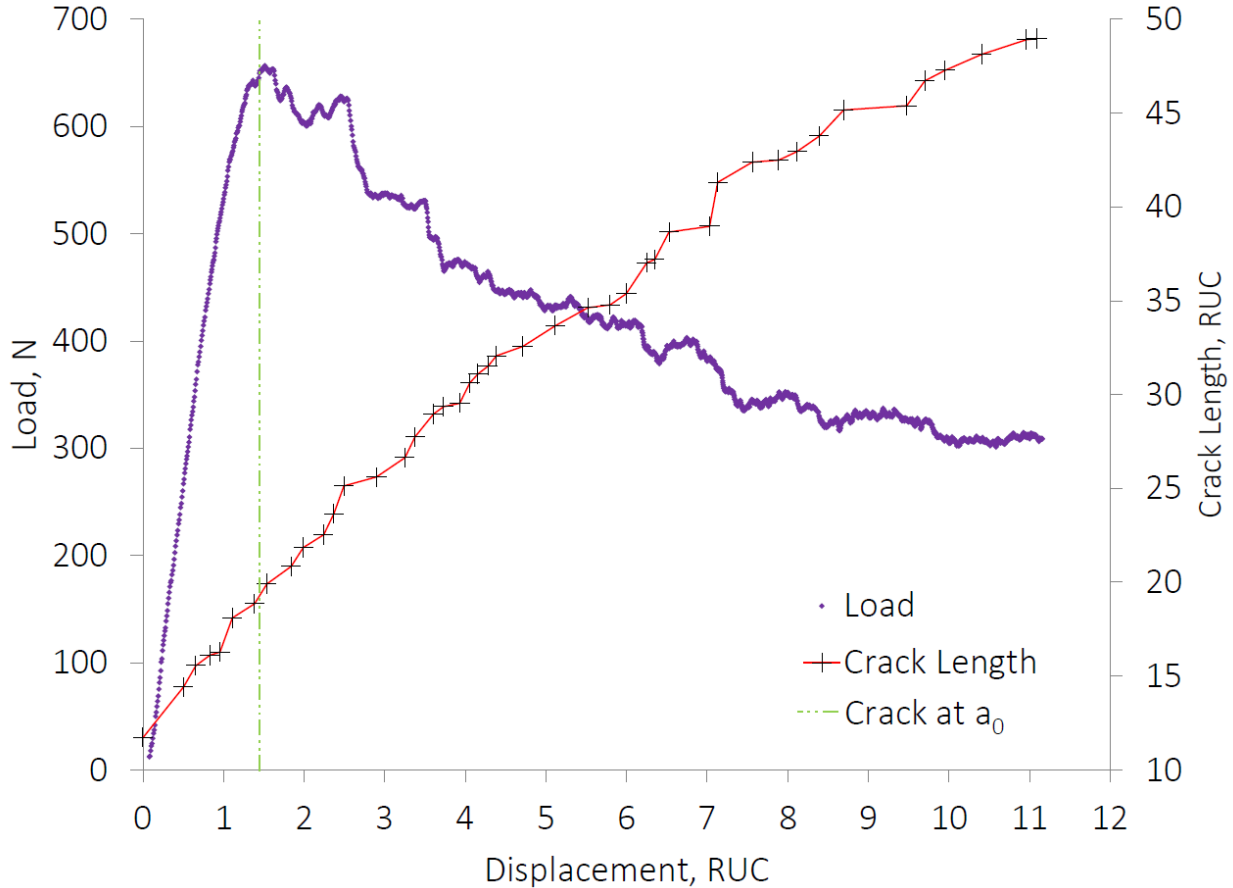
Figure 3.23. Load vs displacement and crack length vs displacement for BDCB specimen A.
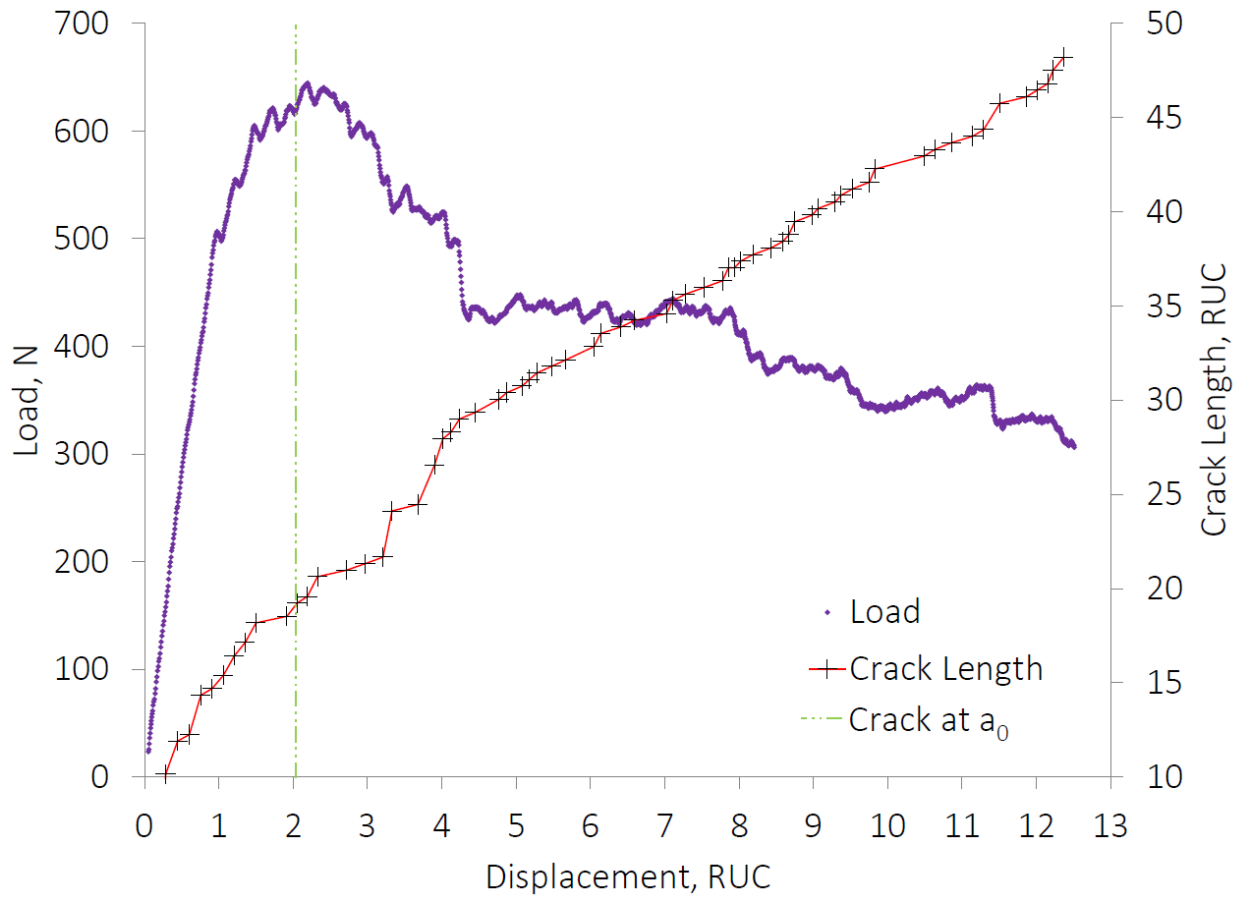
Figure 3.24. Load vs displacement and crack length vs displacement for BDCB specimen B.
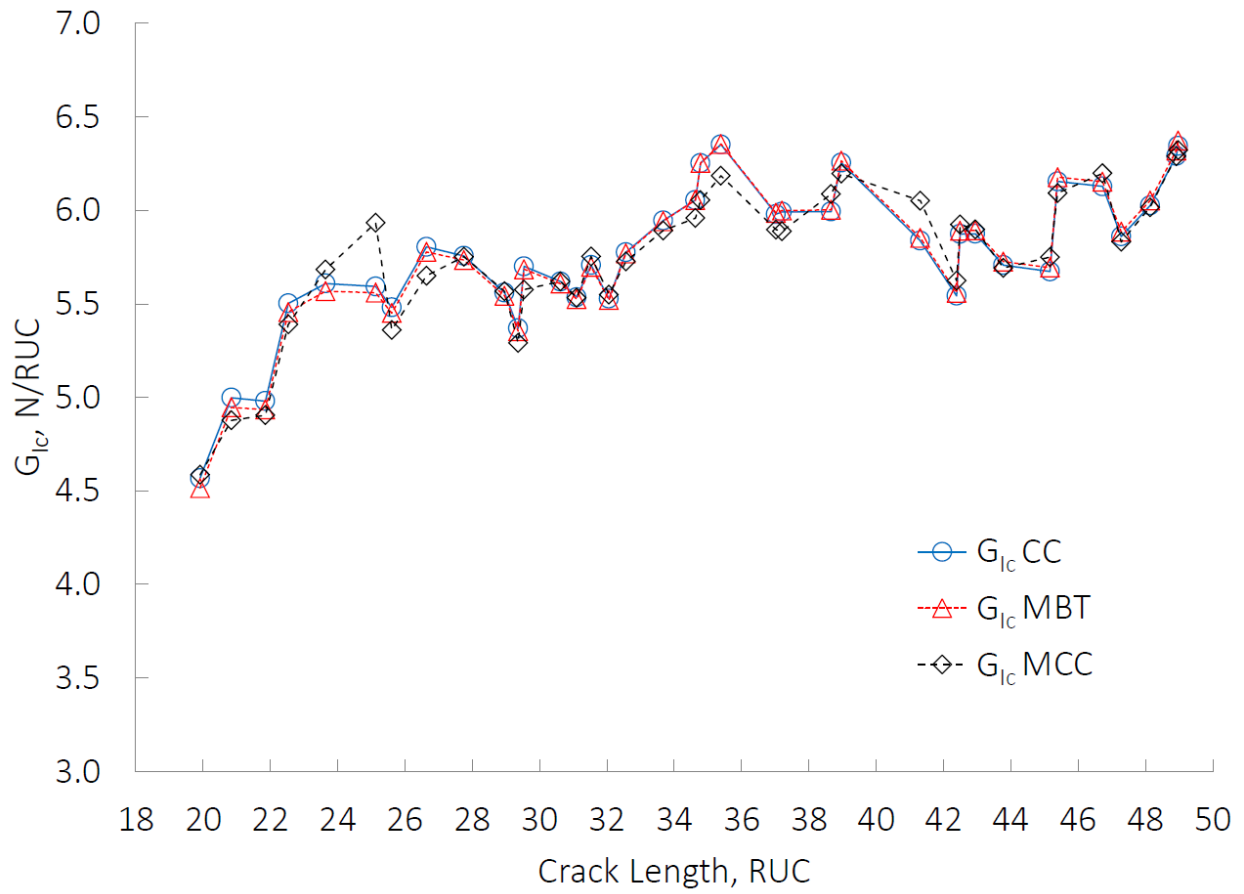
Figure 3.25. Delamination resistance curve for BDCB specimen A.
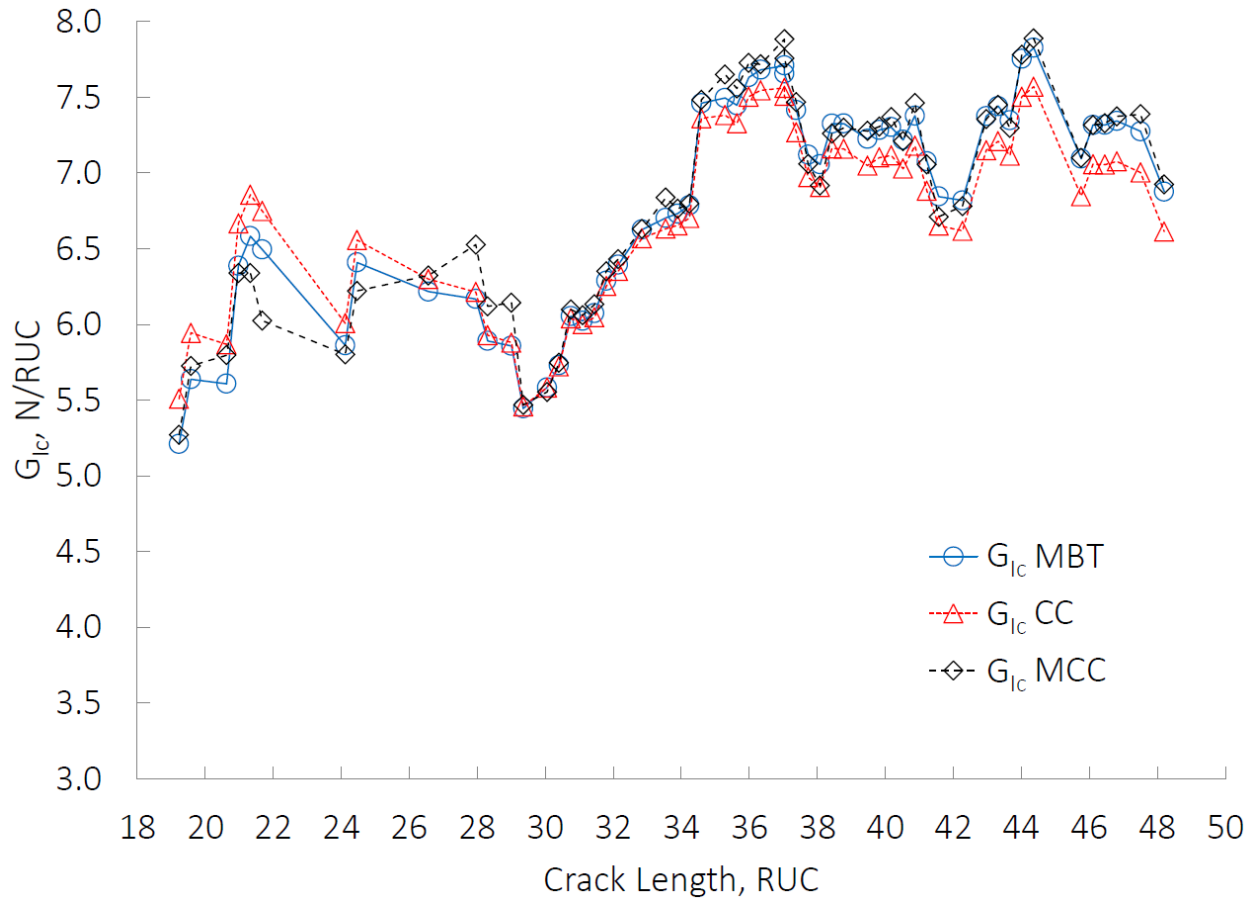
Figure 3.26. Delamination resistance curve for BDCB specimen B.

3.6 Summary

The development of interface fracture tests for 3DFRFC sandwich composites was presented. Key findings included:

- Mode I and Mode II fracture tests were developed for 3DFRFCs.
- Original Mode I 3DFRFC specimens exhibited transverse core cracking in unbonded region.
- A redesigned bonded double cantilever beam specimen was developed, fabricated, and tested.
- G_{Ic} values for the 3DFRFC were found to be comparable to FM300 honeycomb samples which provide significantly higher performance than standard unreinforced foam core.

The experimental investigation of the Mode I and Mode II fracture of 3DFRFC sandwich composites is an ongoing area of research and the BDCB shows great potential for quantifying the bulk interface fracture behavior of 3DFRFCs. The quantification of the effective bulk critical energy release rate is important to the modeling of 3DFRFC sandwich structures with manufacturing induced defects, damage, and core-to-facesheet delamination that can reduce the strength of sandwich composites. The outcomes of this research provide critical understanding and engineering tools required to fully exploit the benefits of advanced three-dimensionally reinforced sandwich structures in current and future spacecraft and launch vehicles, while having transformative impacts to the ability to utilize advanced materials in commercial aerospace and non-aerospace applications.

CHAPTER 4

Strength Reduction of Edgewise Compression with Defects

4.1 Introduction

A current engineering challenge of utilizing sandwich composite structures is quantifying their ability to tolerate damage, particularly in launch vehicles and spacecraft, where mission assurance is critical to mitigating cost from loss or failure, technological set-backs, and potential risk to human life. The strength of sandwich composites can be reduced through many mechanisms, including debonds between a facesheet and the core. Debonds of sufficient sizes, which are typically introduced during manufacturing of foam core composite sandwich structures, could become critical and lead to catastrophic failure. These debonds can result from impact damage, embedded foreign objects, use of poor bonding agents, or surface preparation issues.

Previous studies have highlighted the role of localized facesheet buckling in the failure of reinforced and unreinforced foam core sandwich structures with facesheet-to-core debonds subjected to edgewise compression (EWC) loading [59,63]. These studies demonstrated that for larger defects the specimen failure was driven by the local instability of the facesheet in the unbonded region. 3DFRFC's are next generation materials consisting of foam reinforced with rigid composite rods. One goal of these materials is to increase the performance of sandwich structures in extreme temperature ranges. 3DFRFC materials have garnered significant attention from the aerospace industry, and they are already being used in other industries. When primary structural components of launch vehicles are manufactured using this new technology it is expected that defects will occur inadvertently during manufacturing and handling as commonly occurs when manufacturing large composite structures.

The following chapter details an investigation into the effect of the local interactions on facesheet stability in reinforced and unreinforced foam core composite sandwich structures with

facesheet-to-core debonds. This includes the implementation of a detailed microstructure model to better understand the interaction and behavior of the 3DFRFC material.

4.2 Experimental Setup and Specimen Manufacturing

The material system chosen for this investigation was IM7/8552 carbon epoxy for the facesheets and 19mm (0.75inch) 192kg/m^3 (12lb/ft^3) 3DFRFC. AF191 film adhesive was used to bond the facesheets to the core. Testing methods were selected based on an investigation of typical loads on a launch vehicle [63]. This study found that in-plane compression was the dominant stress, and as a result the edgewise compression test was selected, ASTM C 364-99 [64]. This load case is of particular interest for the study of facesheet-core debonds because the compressive stress in the facesheet is capable of causing localized buckling of the unbonded region of the facesheet. The ASTM guidelines for specimen sizing were utilized to ensure that failure would not result from a global buckling mode resulting in a specimen containing roughly 600 Repeating Unit Cells (RUC) of the 3DFRFC. The specimens were designed to contain circular debonds that were sized to encompass roughly 25, 50, and 100 RUCs of the 3DFRFC, Figure 4.1. Debond sizes were selected based on typical criteria used in the launch industry. The desired debonds were manufactured by removing a circular region of adhesive and replacing it with a PTFE insert, Figure 4.2. The panels were inspected via NDE to ensure panel quality and to verify debond placement in the cured sandwich panel prior to removing the desired samples from the fabricated panels, Figure 4.3. A microCT cross-section of a failed EWC sample shows the relative placement of the PTFE through the specimen thickness, Figure 4.4
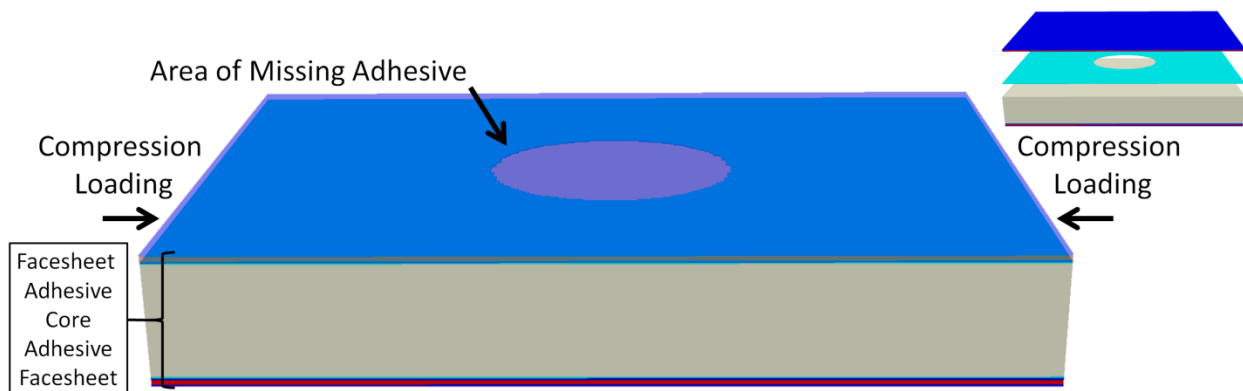
Figure 4.1. Schematic of core-facesheet debond specimen.
(Not to scale.)
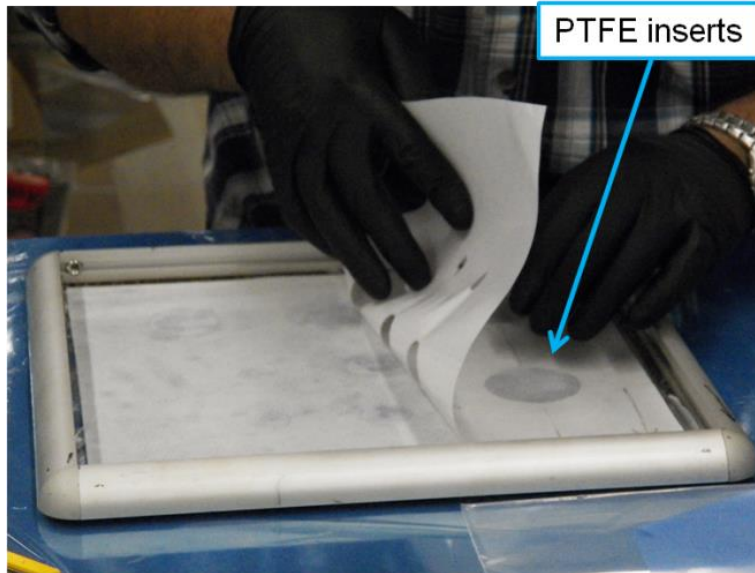
Figure 4.2. Film adhesive with various PTFE inserts.
(Image used with the permission of The Aerospace Corporation.)
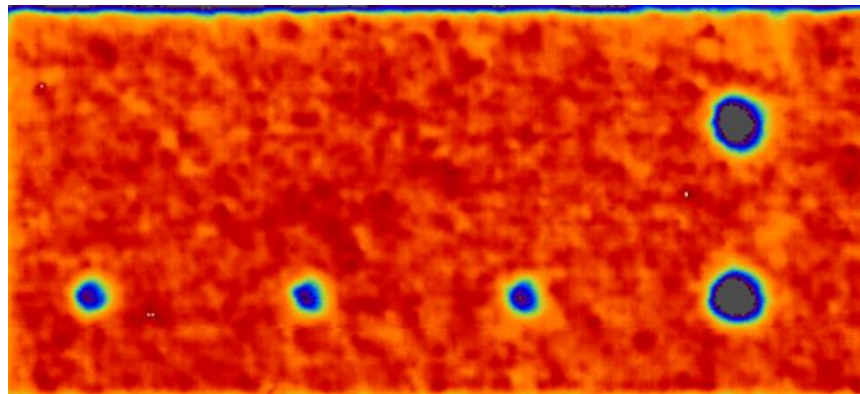
Figure 4.3. Ultrasonic through-transmission of 3DFRFC sandwich panel with various debonds.
(Not to scale. Images used with the permission of The Aerospace Corporation.)
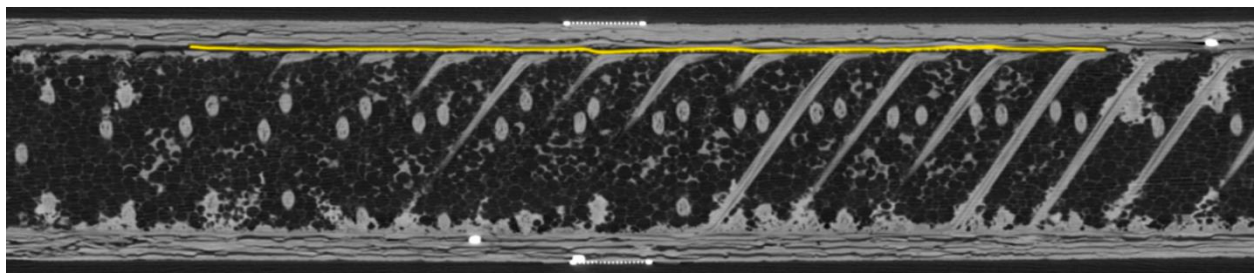
Figure 4.4. MicroCT of failed EWC specimen showing PTFE insert (highlighted in yellow).
(Not to scale. Image used with the permission of The Aerospace Corporation.)

The composite sandwich coupons with varying sized defects were tested according to ASTM C 364-99 [64] using an Instron Universal Testing Machine, Figure 4.5. The tests were conducted at a displacement rate of 5 mm/min until catastrophic specimen failure. Either strain gages were placed in the center of each facesheet or digital image correlation (DIC) was utilized to capture the surface strains. Some samples both contained strain gages and had DIC conducted on the region surrounding the strain gauge. The load and axial displacement were recorded from the Instron. 4-5 samples were tested for each debond size. It is worth noting that all of the specimens were potted in machined aluminum test fixtures after localized end crushing was observed when using the standard clamping method on the baseline samples, Figure 4.6.

Figure 4.5. Edgewise compression test fixture.

Figure 4.6. Edgewise Compression sample with potted ends (top) and with end failure (bottom). (Not to scale.)

4.3 Modeling EWC samples

Modeling efforts have been focused on developing tools to allow for the detailed modeling of the microstructure of the 3DFRFC. Using the commercial code Abaqus, this effort has resulted in a parametric Abaqus script that automates the creation of the microstructure geometry for the reinforcing fibers based on the geometric parameters of the microstructure and the size of the specimen to be modeled. Note the reinforcing geometry is modeled using beams; however, some images show the fibers with rendered thickness for clarity. This script incorporates details of the bonding ends of the reinforcement into the geometry generated. The geometry generated by this script was utilized to develop the embedded element models of the edgewise compression sample configurations. See Chapter 2 for further details on the microstructure modeling.

The evaluation of the EWC configurations was performed with a finite element analysis using Abaqus. The facesheet plies and adhesive layers were individually modeled with brick elements. The facesheet and unreinforced foam core properties were measured through testing at The Aerospace Corporation [52], while the mechanical properties of the film adhesive were obtained from vendor data [65], Table 4.1. The 3DFRFC is modeled utilizing two different methods: homogenized orthotropic and embedded element. The homogenized orthotropic model takes advantage of recently developed micromechanics models [33] for metallic and pin reinforced foams to derive the homogenized orthotropic elastic properties for the 3DFRFC based on its specific microstructure and material composition. The embedded element method utilizes the aforementioned Abaqus script to model the discrete reinforcing pins as beams within the solid foam mesh.

Table 4.1. Properties of sandwich constituent materials.

Facesheet Lamina Properties, SGP370-8H/8552	
E_1 (GPa)	76.5
E_2 (GPa)	80.0
ν_{12}	0.05
Tensile Strength F_{1t} (GPa)	1.06
F_{2t} (GPa)	1.03
Strain to Failure ε_{1t} (%)	1.35
Compressive Strength F_{1c} (GPa)	0.525
Film Adhesive Properties, AF191	
Elastic Modulus, E (GPa)	2.206
Poisson's Ratio, ν	0.40

The same structured base mesh was utilized for all of the cases and was sufficiently refined to provide a converged solution and adequate detail for the embedded element model resulting in a base mesh of approximately 5 million degrees-of-freedom. The model used mirrored the test sample and contained approximately 600 representative unit cells (RUCs). It is important to note that RUC in this analysis is defined strictly from the minimum geometric unit needed to capture the repeating structure of the 3DFRFC (Chapter 2). For all cases the facesheet-to-core debond was created by removing a circular region of the adhesive layer, Figure 4.7. Three different circular defect sizes were investigated for each of the three core models. Linear buckling analysis showed that all models exhibited localized faceheet buckling as the primary mode, similar to the one shown in Figure 4.7. The mode corresponding to the smallest linear buckling load had a positive out-of-plane displacement amplitude, thus precluding the possibility of contact buckling; contact buckling may occur for higher modes or differing boundary conditions. Issues related to contact buckling of debonds are addressed in work by Comiez et al. [66] as well as Shahwan and Waas [67].
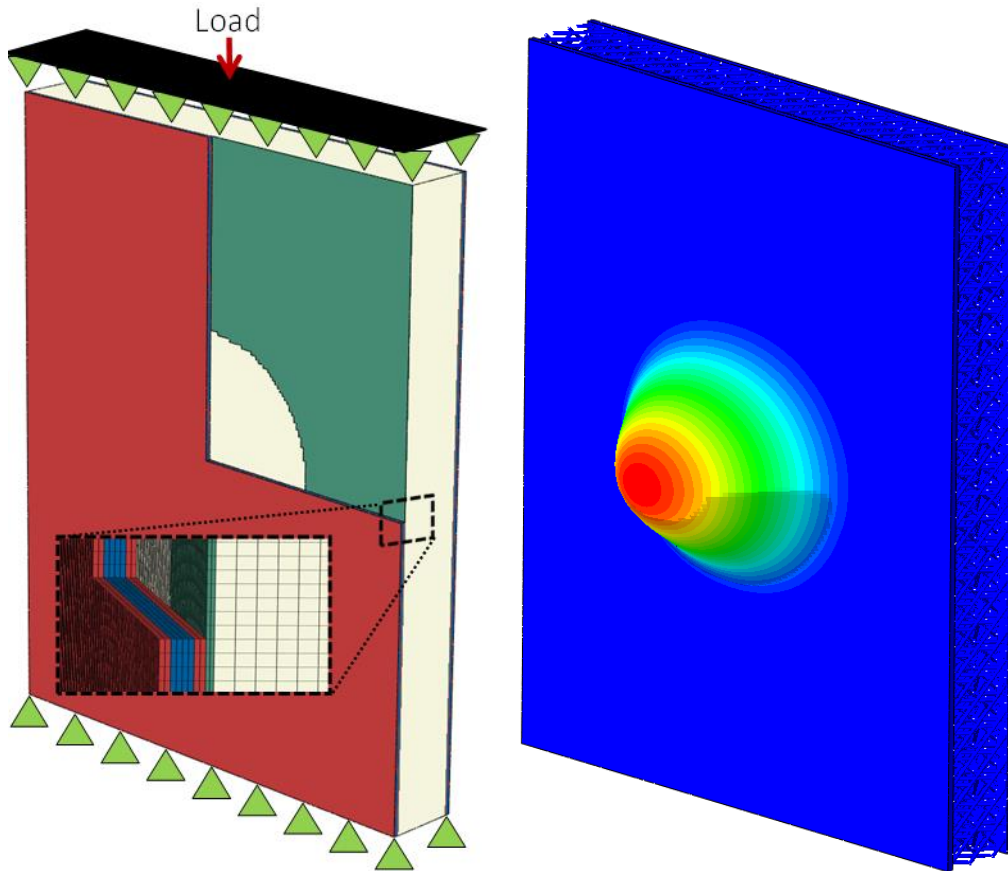
Figure 4.7. Model configuration shown with portion of facesheet removed, left, and representative buckling mode shape shown with foam removed, right.
(Not to scale.)

4.3.1 Linear Buckling Analysis

Linear buckling analysis was conducted on the unreinforced foam core model first to provide a foundation for evaluating the behavior of the 3DFRFC models. Both the homogenized and embedded 3DFRFC models predict an increase in buckling load as compared to the unreinforced foam core models. The embedded model exhibits a lower value compared to the homogenized model likely due to its ability to account for the interaction between the discrete reinforcement and the specimen edges; those effects are not captured by the simple homogenized model.

Additional insight into the local interaction of the 3DFRFC can be garnered from comparing the buckling mode displacement field of the embedded element model to that of the unreinforced foam core, Figure 4.8. While the overall buckling shape is similar, the 3DFRFC embedded

model exhibits deformation that is highly constrained to the region of the debond. Conversely, the unreinforced foam core model exhibits a more widespread interaction with the debonded region. The higher degree of localization demonstrated by the 3DFRFC models effectively constrains the boundary of the debonded region. The local constraint provided by the reinforcement near the boundary of the debond is more clearly illustrated by the deformation of the pins shown in Figure 4.9. This local interaction will be important to investigating the nonlinear buckling response of the EWC samples and modeling the initiation of failure.
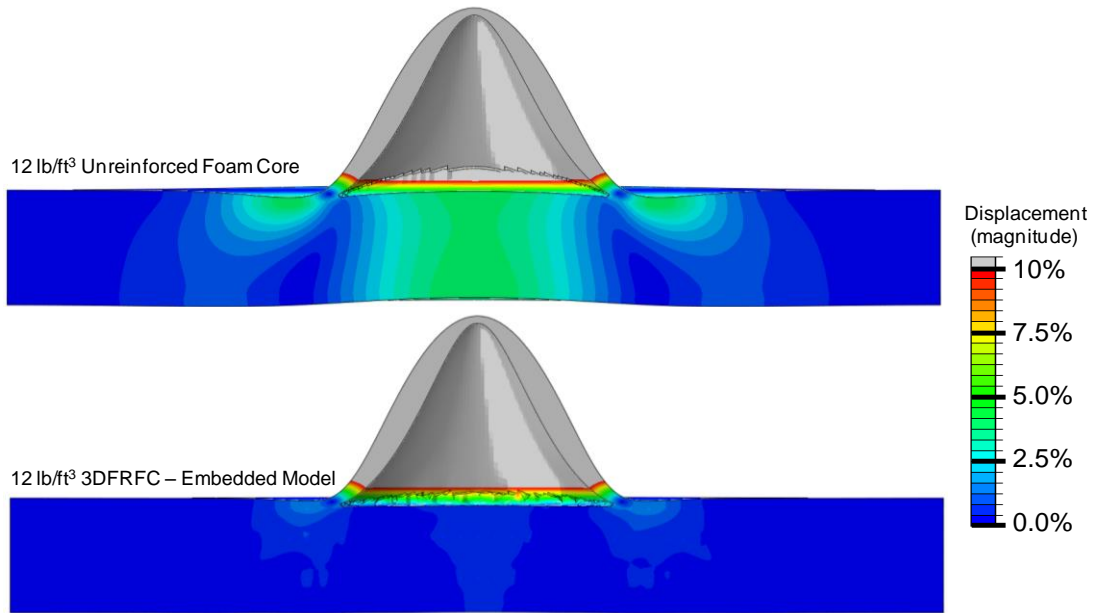
Figure 4.8. Buckling mode in 12 lb/ft^3 3DFRFC and unreinforced foam core with 50 RUC^2 defect. (Not to scale.)
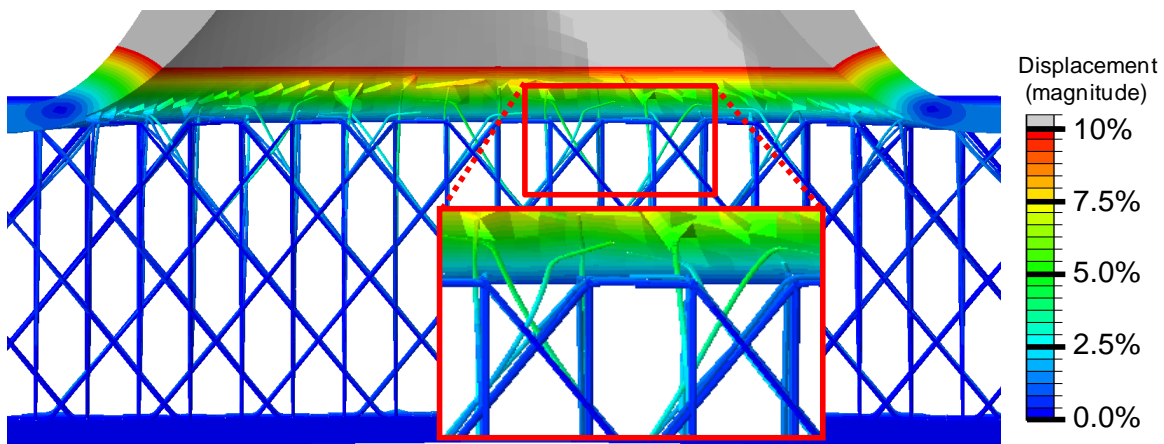
Figure 4.9. Local displacement fields in bucking mode for 3DFRFC with 50 RUC^2 debond. (Not to scale.)

4.3.2 Analysis of Delamination Growth

A quarter symmetry finite element model of the experimental setup was developed and solved using the finite element software Abaqus to evaluate debonds between the facesheet and the core. The facesheet plies were individually modeled with brick elements and the adhesive layer between the facesheet and the foam was modeled with decohesion elements. The strength and fracture properties for the interface were based on measured foam properties, as the foam is the weaker of the constituents at the interface. The properties of the facesheet lamina, foam core and film adhesive are provided, Table 4.1.

The value for the lamina compression strength was determined from prior edgewise compression tests. The value used for the Mode I critical strain energy release rate of the 0.19g/cm^3 (12lb/ft^3) unreinforced foam is an assumed lower bound based on preliminary fracture testing and is supported as a lower bound by published results [51]. Due to the lack of testing, the Mode II and Mode III fracture values for the foam were taken as equal to the mode I values for the analysis in the interest of maintaining a lower bound on the fracture properties. The remainder of the facesheet and unreinforced foam properties were measured through in-house testing [52], while the mechanical properties of the film adhesive were obtained from vendor data [65]. No published experimental fracture properties exist for the 3DFRFC sandwich structures and the fracture testing discussed in Chapter 3 had not been developed at the time of this work. As a result, the strength and fracture properties for the interface were modeled by using the previously measured values for the 0.19g/cm^3 (12lb/ft^3) unreinforced foam as an approximate lower bound since it has the same bulk density. This modeling effort also takes advantage of the recently developed micromechanics models [33] for metallic and pin reinforced foams that models the reinforcements within the foam as beams on an elastic foundation to derive the effective orthotropic elastic properties for the 3DFRFC based on its specific microstructure and material composition.

The failure load was predicted using a progressive failure methodology including nonlinear geometry. Displacement control was used to simulate test boundary conditions and enable the numerical simulation of failure progression. Progressive failure analysis (PFA) did not account for matrix-cracking and fiber failure, because it was observed in prior no-defect experiments that the structural response was linear and failure of the samples occurred suddenly in the form of catastrophic facesheet compression failure. There was no indication that matrix-cracking

preceded the ultimate failure, but most likely it occurred concurrently with the onset of fiber failure. While PFA was used to simulate delamination propagation, the facesheet compression failure was predicted by identifying the load at which the facesheet stress exceeded its compression strength. In the current study, the discrete cohesive zone model (DCZM) pioneered by Xie and Waas [56] is used for modeling delamination between the core and the facesheets due to its increased modeling efficiency. One challenge of applying this method is that decohesive zones yield accurate results when the crack plane is well defined [54–56], but lose fidelity as the material becomes more discretized and the crack path can no longer be inferred a priori. This is an important caveat that needs to be considered in using this method (See Chapter 3). The added paths for load transfer in the 3D fiber reinforced foam core act to impede crack propagation within the foam and the use of decohesive elements to model this material will likely become less accurate and unable to capture the highly discretized nature of the 3DFRFC as the loading becomes more complex.

The modes from the linear buckling analysis were scaled to introduce a geometric imperfection into the nonlinear model. The magnitude of the imperfection introduced was varied from $0.01\% - 1.0\%$ of the facesheet thickness. The analysis showed that the 50 RUC^2 debond model was highly sensitive to imperfections, resulting in a transition from pure facesheet compression failure to a buckling driven compression failure. For the 50 RUC^2 and 100 RUC^2 debonds, the failure involves facesheet buckling induced compression failure that occurs concurrently with delamination initiation, Figure 4.10. The load drop shown at higher displacements in Figure 4.10 is due to the delamination propagating across the specimen width; however, the facesheet is predicted to fail before this can occur. The slight delamination that corresponds with the anticipated fiber failure is expected to open slower with the higher fracture properties measured in Chapter 3; however, since the delamination propagation is not the primary predicted failure mechanism this effect should be negligible. Additionally, the values for the Mode II and mode III critical strain energy release rates were varied from 1 to 10 times the Mode I value and the predicted failure was found to be insensitive to the changes. This finding may not be applicable for a different composite system.
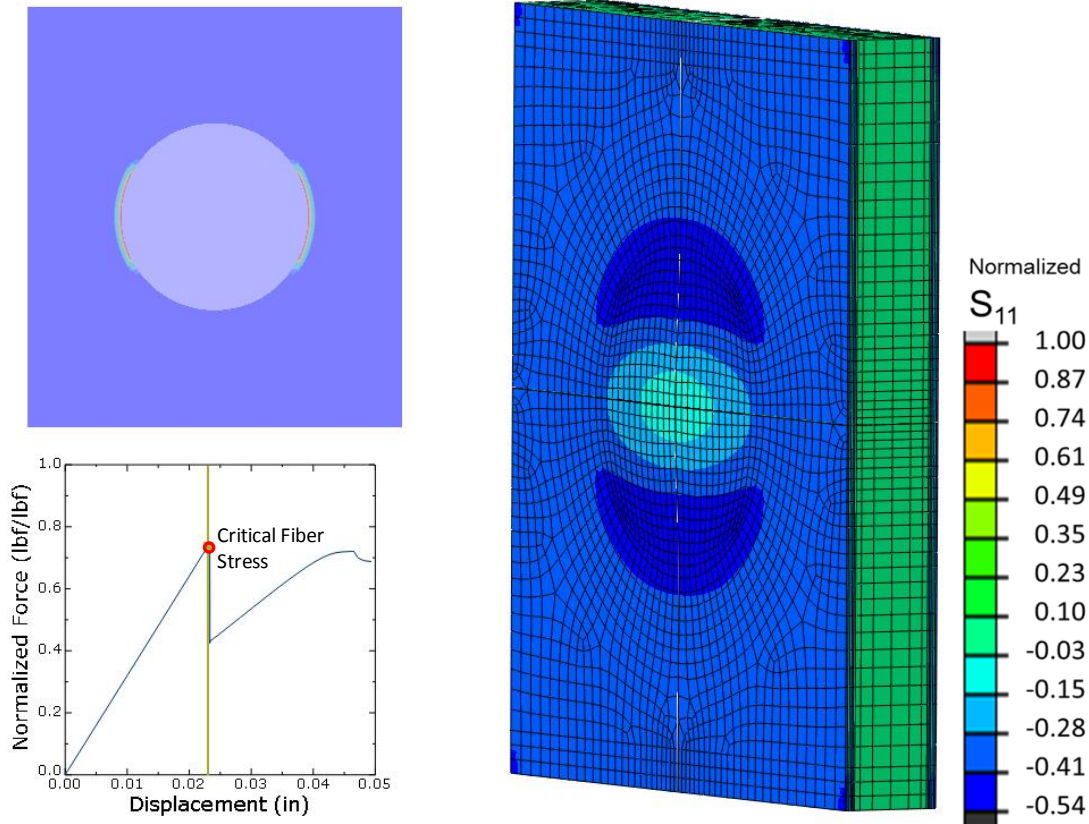
**Figure 4.10. Failure analysis for 100 RUC debond in 3DFRFC.
(Not to scale.)**

4.4 EWC Results and Discussion

Insight into the material behavior and the interaction of the discrete constituents of the 3DFRFC sandwich is gained through the use of digital image correlation (DIC). This capability allows for the mapping of the two dimensional strain fields on the surface of the specimen throughout the test. This capability provides critical insight and allow for a deeper understanding of the microstructure interaction of the 3DFRFC and aids in the development of models capable of capturing this interaction.

The strength for the samples with debonds was reduced when compared to the sample without defect, especially for the 100 RUC^2 debond sample where a 19 percent strength reduction was observed, Table 4.2. The load displacement curves from the median sample for each defect size is given in Figure 4.11. The load versus displacement behavior is similar between all samples with only the discernable difference being the failure load. Fiber failure initiated with the onset of buckling for samples with 100 RUC^2 debonds. The analytical model

73

for the 3DFRFC sandwich with the 100 RUC^2 debond predicts a larger effect due to delamination growth. This is likely a function of the discrete material propensity to turn the crack diverting it away from the facesheet keeping failure localized, Figure 4.13 and Figure 4.12 Additionally, a small increase in strength of 3.8% was observed for 25 RUC^2 debond; however, with p=0.29 it is not clear weather this is a real physical phenomenon or a result of chance [68,69]. It is possible that this is due to changes in stress distribution due to the transition between fully bonded and partially bonded pins. This is supported by the strain distribution seen in the DIC images of the 25 RUC^2 debond samples prior to failure, Figure 4.14. A larger sample size is needed to confirm this behavior.

Table 4.2. Observed and predicted strength reduction with facesheet-core debonds.

Debond Area (RUC^2)	Number of Samples	Standard Deviation (kN)	Failure Load (kN)	Measured Strength Reduction (%)	Predicted Strength Reduction (%)
-	4	3.3	110.8	--*	0.0*
25	5	6.3	115.1	-3.8*	0.0*
50	5	2.7	109.6	$1.1^{*,++}$	$0.0^* - 9.5^{++}$
100	5	4.7	86.4	22.0^{**}	$22.6^{++} - 32.0^{++}$

Failure Mode: Facesheet Compression*   *Localized Buckling of Facesheet*$^{++}$
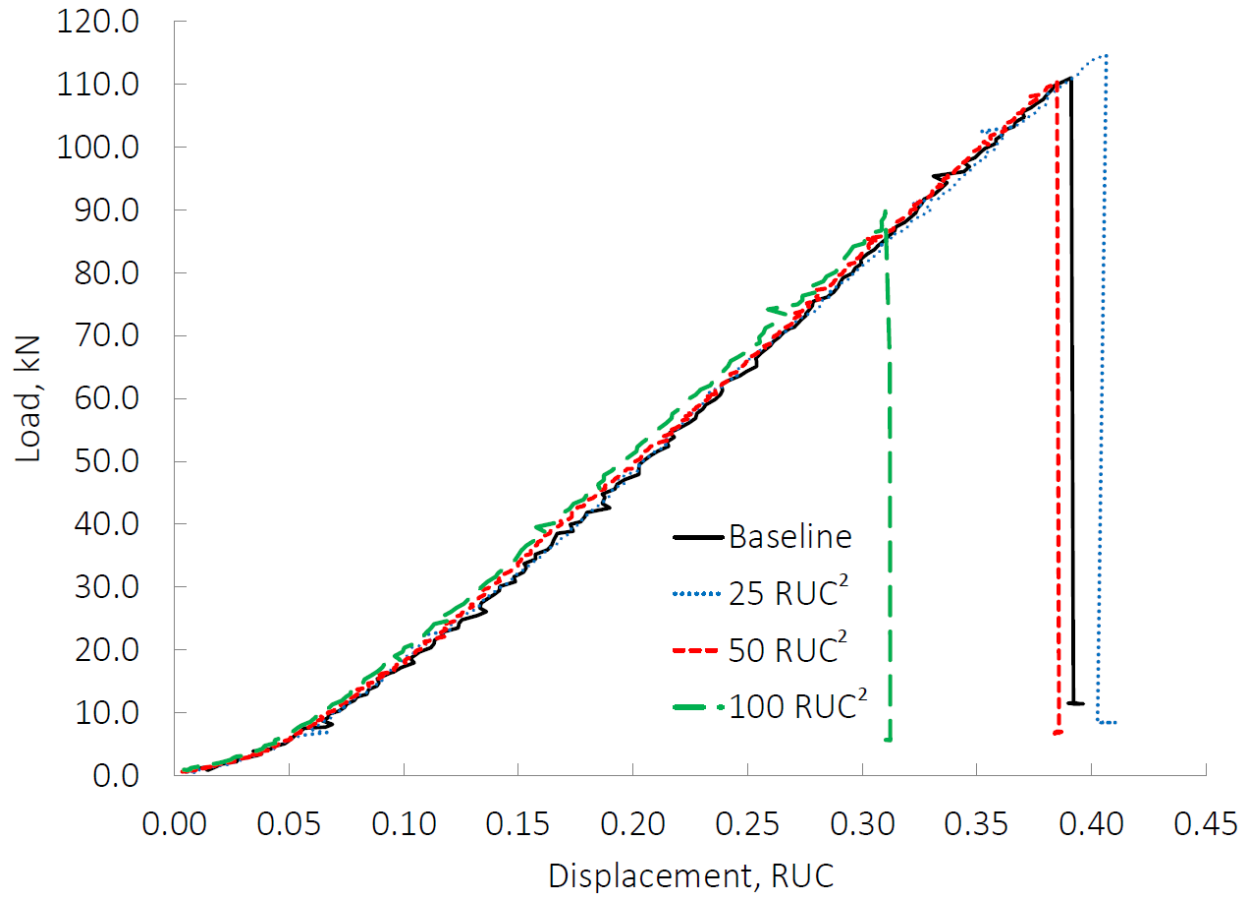
Figure 4.11. Load vs displacement for median EWC samples.
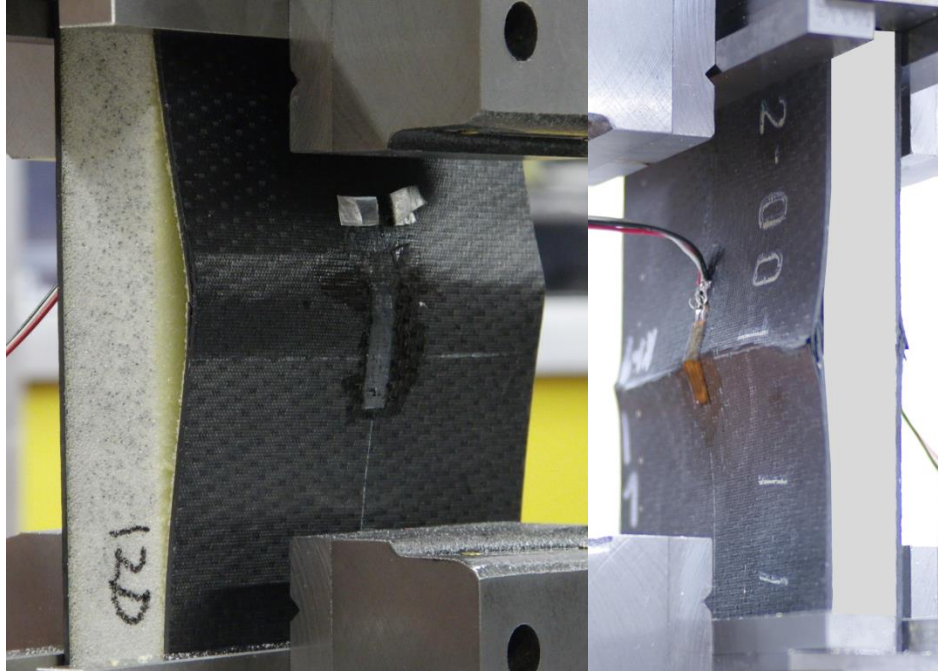
Figure 4.12. Failed EWC samples with unreinforced foam (left) and reinforced foam (right). (Details of reinforcement geometry removed. Images used with the permission of The Aerospace Corporation.)
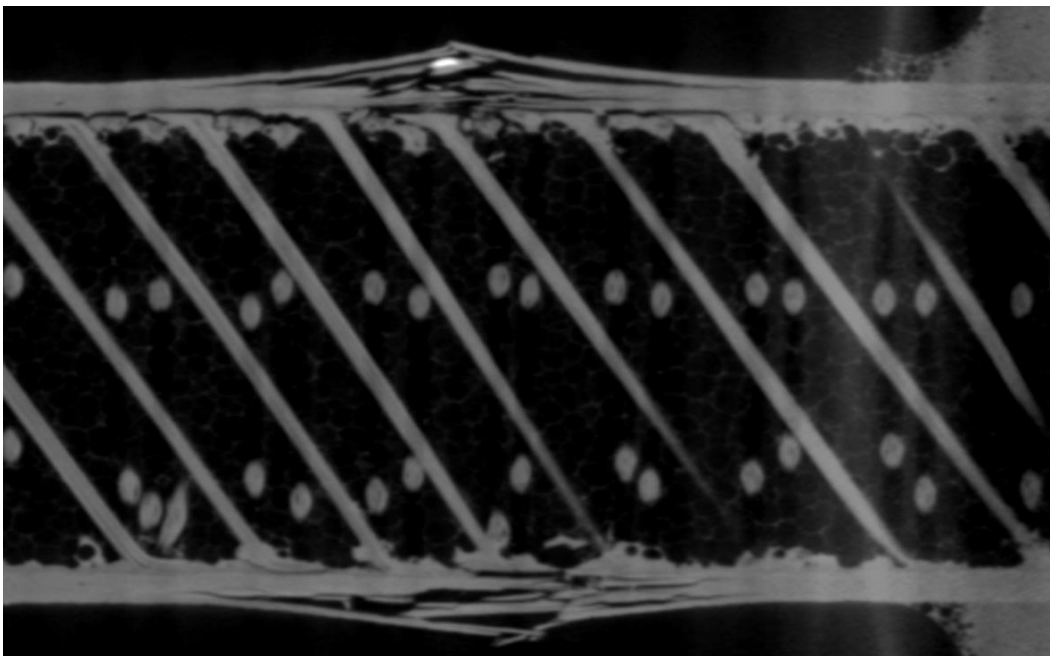
Figure 4.13. MicroCT of failed EWC specimen. (Not to scale. Image used with the permission of The Aerospace Corporation.)
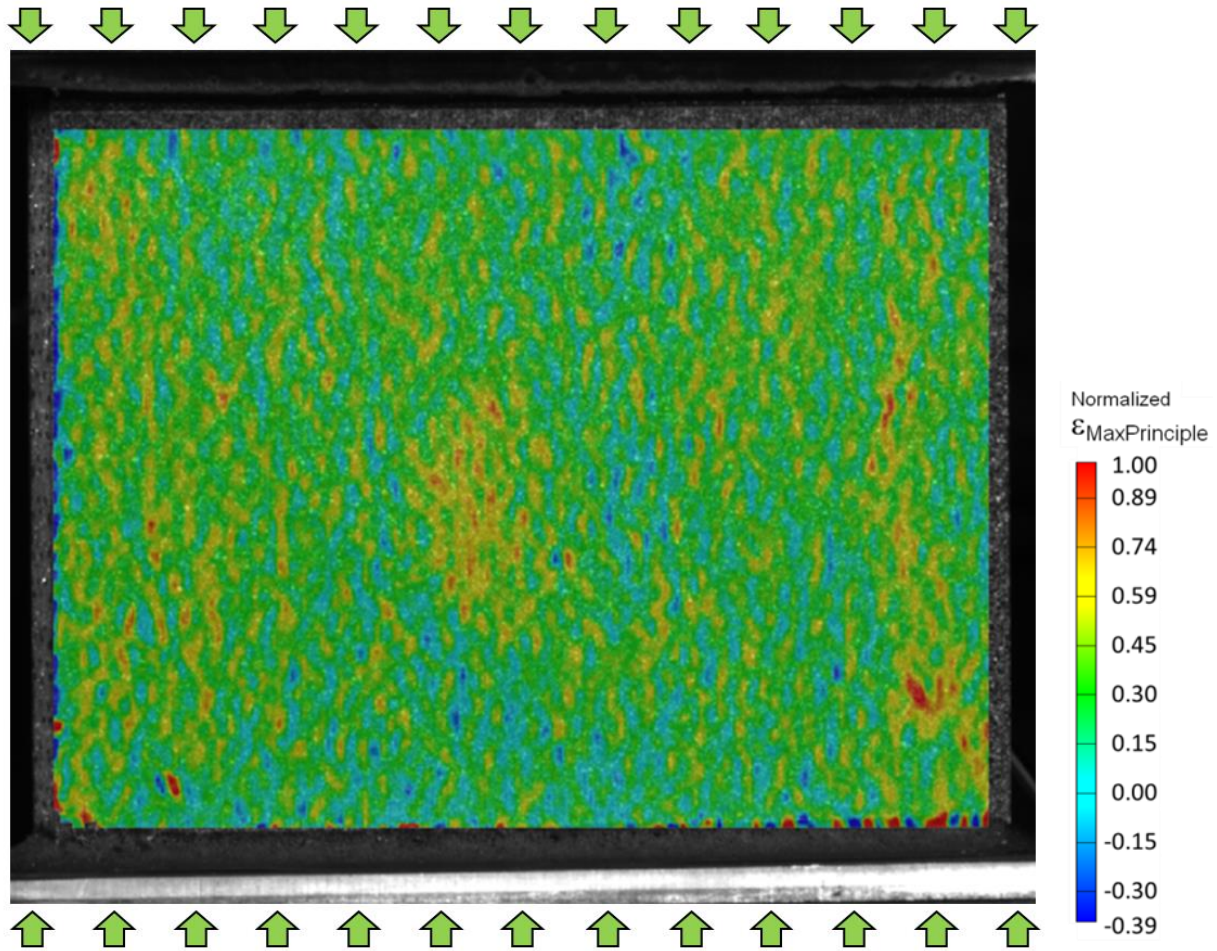
Figure 4.14. Strain distribution in 3DFRFC EWC with ~25 RUC^2 defect. (Not to scale.)

Additional interest surrounds the observations of the failure in the 50 RUC^2 debond specimens. The analysis for this case demonstrated a high sensitivity of the predicted failure mode to the initial imperfection in the system changing from facesheet compression to buckling driven facesheet compression. While most of the 50 RUC^2 debond specimens exhibited facesheet buckling just prior to failure, Figure 4.15, one of the tests actually failed progressively in facesheet compression without any buckling, Figure 4.16. Furthermore, all specimens with a 100 RUC^2 debond exhibited facesheet buckling with some demonstrating slight delamination growth before fiber failure as predicted by the analysis, Figure 4.17.
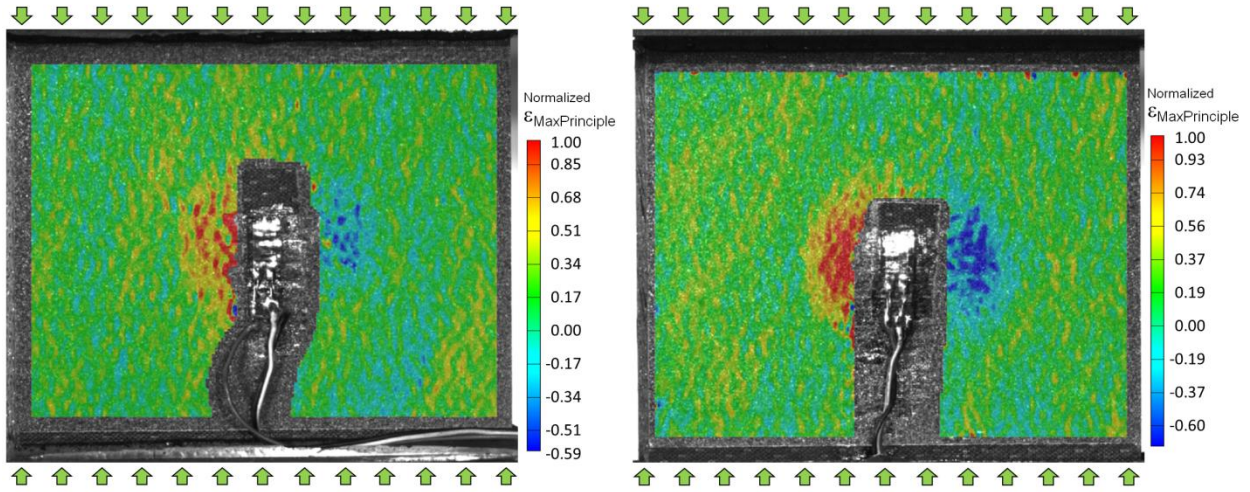
Figure 4.15. Buckling driven failure in 2 3DFRFC EWC specimens with ~50 RUC^2 debond. (Not to scale.)
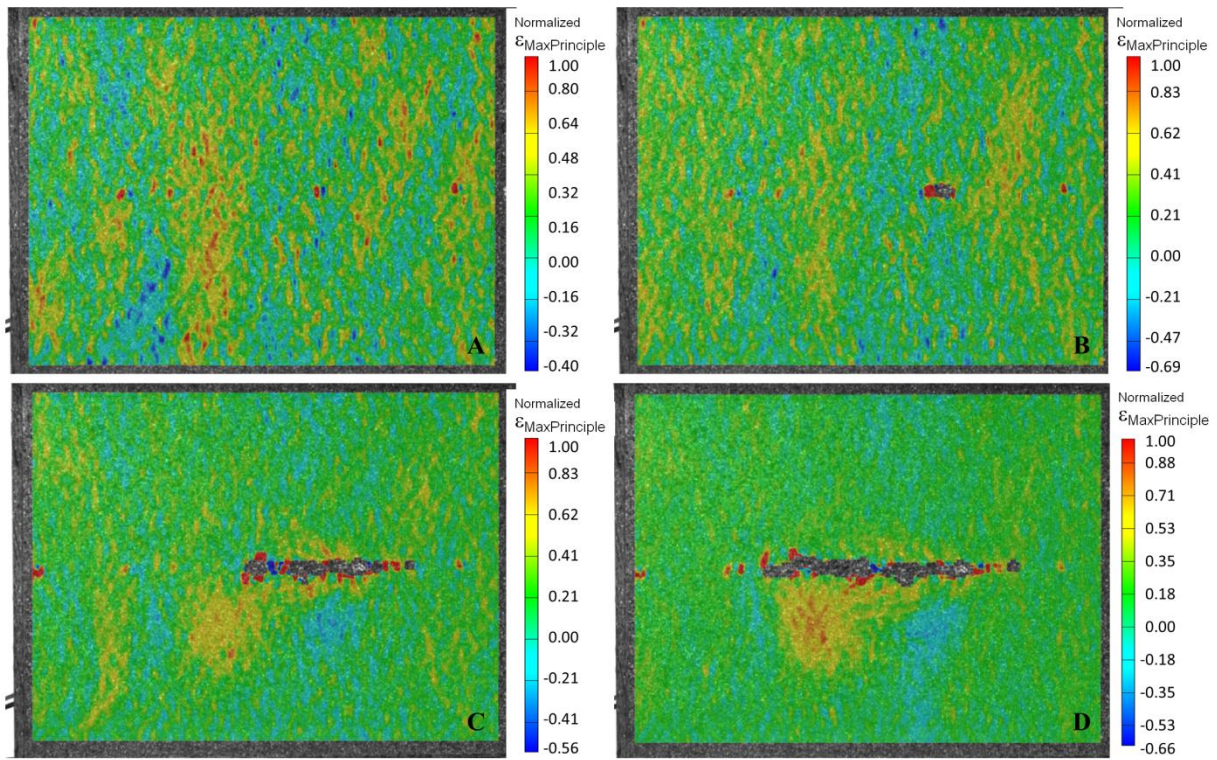
Figure 4.16. Failure progression in 3DFRFC EWC with ~50 RUC^2 defect. (Not to scale.)
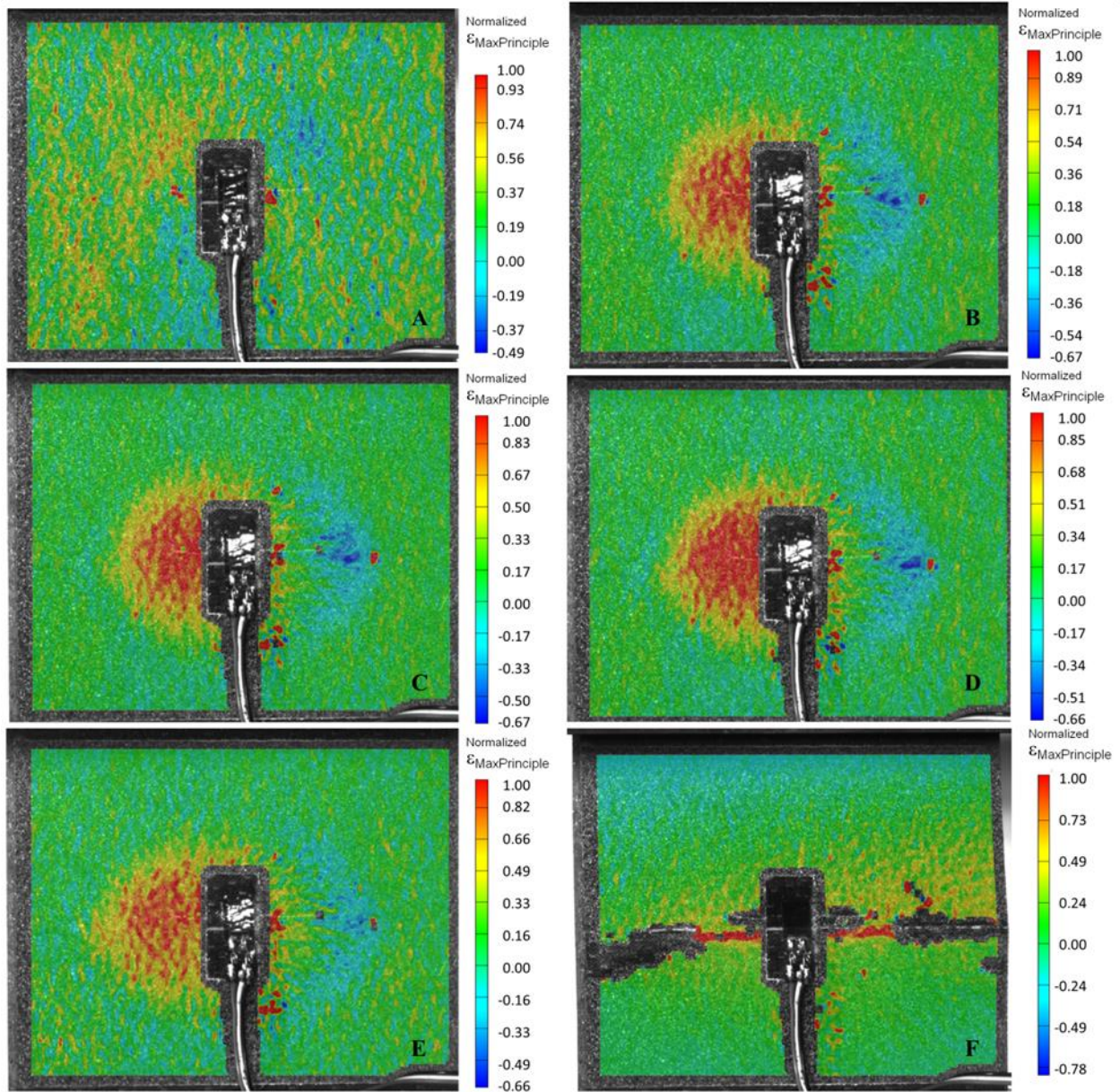
Figure 4.17. Failure progression in 3DFRFC EWC with ~100 RUC^2 defect. (Not to scale.)

4.5 Summary

The development of modeling methods and results of experimental investigation into the failure of 3DFRFC sandwich composites under edgewise compression loading was presented. Key findings included:

- 3DFRFC specimens demonstrated a high tolerance to facesheet-to-core debonds with the 100 RUC^2 debonds providing the only statistically significant reduction in strength, 22%.

- Nonlinear finite element analysis predicted the magnitude of strength reduction as well as the change in failure mode observed for the 50 RUC^2 debonds specimens.

- Digital image correlation was used to confirm the failure modes in the EWC samples. The failure mode cannot be determined based on the load displacement behavior.

CHAPTER 5

Through Thickness Failure of 3DFRFC

5.1 Introduction

The truss structure of the 3D fiber network in a 3DFRFC provides added paths for load transfer and acts to impede crack propagation within the foam core. The stiffness and strength of these three-dimensionally reinforced sandwich composites become exceedingly difficult to predict as a result of the added load paths. The relatively coarse architecture of 3DFRFCs can cause challenges in quantifying the strength using standard coupons due to free-edge effect. This chapter focuses on modeling and experimentally investigating the through-the-thickness failure of a 3D Fiber Reinforced Foam Core (3DFRFC) sandwich composite. This includes the development of various modeling methods to better understand the constituent interaction and behavior of this material. An investigation of the unreinforced foam is performed in parallel to the experimental investigation of the 3DFRFC sandwich specimens.

5.1.1 3DFRFC Specimen Fabrication

The general manufacturing procedure for the 3DFRFC sandwich samples is similar to the one published previously for the manufacture of edgewise compression samples with defects [59] but is included here for completeness. The material system chosen for this investigation is an 8-harness satin weave IM7/8552 carbon/epoxy prepreg for the facesheets and a 19 mm (0.75 inch) thick 0.19g/cm^3 (12lb/ft^3) 3DFRFC for the core. 3M™ Scotch-Weld™ AF191 film adhesive is used to bond the facesheets to the core. The panels are inspected via non-destructive evaluation to ensure panel quality of the cured sandwich panel prior to removing the desired samples from the fabricated panels. The manufactured 3DFRFC sandwich panels are then cut into samples for tensile testing. During testing, additional insight into the material behavior and the interaction of the discrete constituents of the 3DFRFC is gained through the use of the digital

image correlation (DIC) capabilities of the Composite Structures Laboratory at the University of Michigan. This capability allows for the mapping of the two dimensional strain fields on the surface of the specimen throughout the test. This capability provides critical insight, allows for a deeper understanding of the microstructure interaction of the 3DFRFC, and aids in the development of models capable of capturing this interaction.

5.1.2 Microstructure Modeling

Modeling efforts have been focused on developing tools to allow for the detailed modeling of the microstructure of the 3DFRFC. Accurate modeling of the 3DFRFC microstructure was facilitated through interrogation of the as-manufactured microstructure using X-ray microtomography (microCT). The microCT scans allowed for rapid measurement of the reinforcement angle (α) and spacing (S) without disturbing the reinforcing truss network as can occur with sectioning. The measured microstructure geometry was subsequently modeled using a parametric script in the commercial code Abaqus that automates the creation of the microstructure geometry for the reinforcing fibers based on the geometric parameters of the microstructure and the size of the specimen to be modeled. Note the reinforcing geometry is modeled using beams; however, some images show the fibers with rendered thickness for clarity. This script incorporates details of the bonding ends of the reinforcement into the geometry generated (see Chapter 2). The geometry generated by this script was utilized to develop the embedded element models of the flatwise tension (FWT), flatwise compression (FWC), and through thickness shear (3ptB) sample configurations.

5.2 Predicted Specimen Size Effects

The evaluation of the FWC configurations was performed using finite element analysis using the finite element software Abaqus. The facesheet plies and adhesive layers are individually modeled with brick elements. The facesheet properties were measured through testing at The Aerospace Corporation [52], while the mechanical properties of the film adhesive were obtained from vendor data [65].

The initial FWC models were used to investigate the size effects and the role of the foam surrounding the reinforcement. Figure 5.1 shows normalized load deflection curves for 5 x 5

RUCs (repeating unit cell), 10 x 10 RUCs, and 15 x 15 RUCs models each with reinforcement within the base foam (embedded) and without the support foam (pin only). It is important to note that RUC in this analysis is defined strictly from the minimum geometric unit needed to capture the repeating structure of the 3DFRFC (see Chapter 2). While the foam contributes a small amount to the initial stiffness of the specimens, the largest role of the foam is to prevent pin buckling that results in the load plateau seen in Figure 5.1. A side view of both models illustrates the buckling suppression that is provided by the presence of the foam, Figure 5.2. Additionally, the increase in stiffness as the number of unit cells increases can be attributed to the higher percentage of fully bonded reinforcement and an effective reduction in the edge effects caused by the severed pins located at the material boundary (dark blue pins in lower image, Figure 5.2).
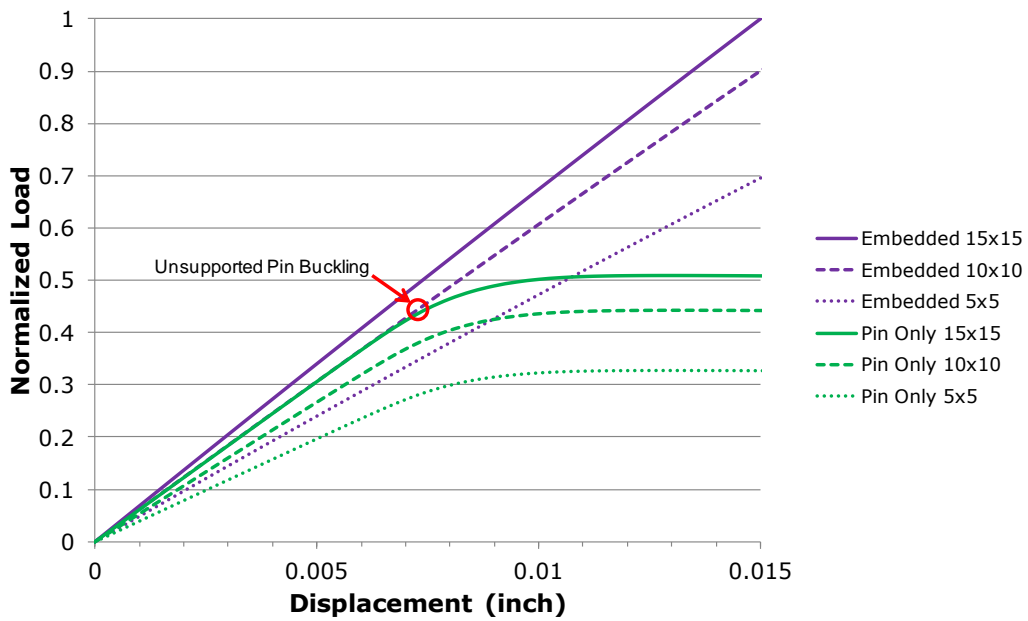
Figure 5.1. Normalized load-displacement plot from FWC models highlighting the primary function of the foam in suppressing buckling.
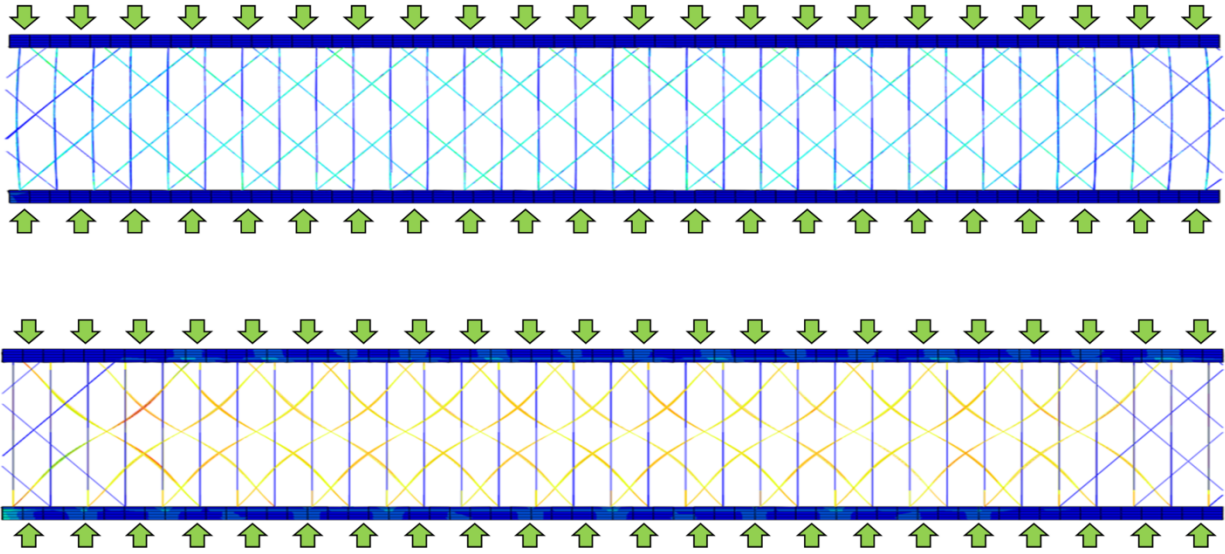
Figure 5.2. Illustration of pin buckling in FWC models with embedded elements (top, foam not shown) and unsupported pins (bottom) at the same displacement.
(Not to scale.)

In order to investigated the effect of specimen size on the compression response of 3DFRFC samples the aforementioned modeling method was extended to 30 sample sizes ranging from less than 1 to over 260 RUCs. This modeling effort uses the recently developed clamped-Uniform Deformation Gradient (c-UDG) micromechanics model [33] for metallic and pin reinforced foams as a basis of comparison with the embedded element model. All 30 models were analyzed using three methods for modeling the 3DFRFC: discrete microstructure (embedded elements), homogenized c-UDG, and isotropic (same through-thickness stiffness as c-UDG). The purpose of utilizing these three methods is to help differentiate the contributions due to the cut reinforcement, specimen geometry, and constraining interaction of the orthotropic properties that will be present during the testing of finite specimens. The results of these analyses are presented in Figure 5.3, where the effective compression stiffness is normalized by the through-thickness modulus given by the micromechanics model.
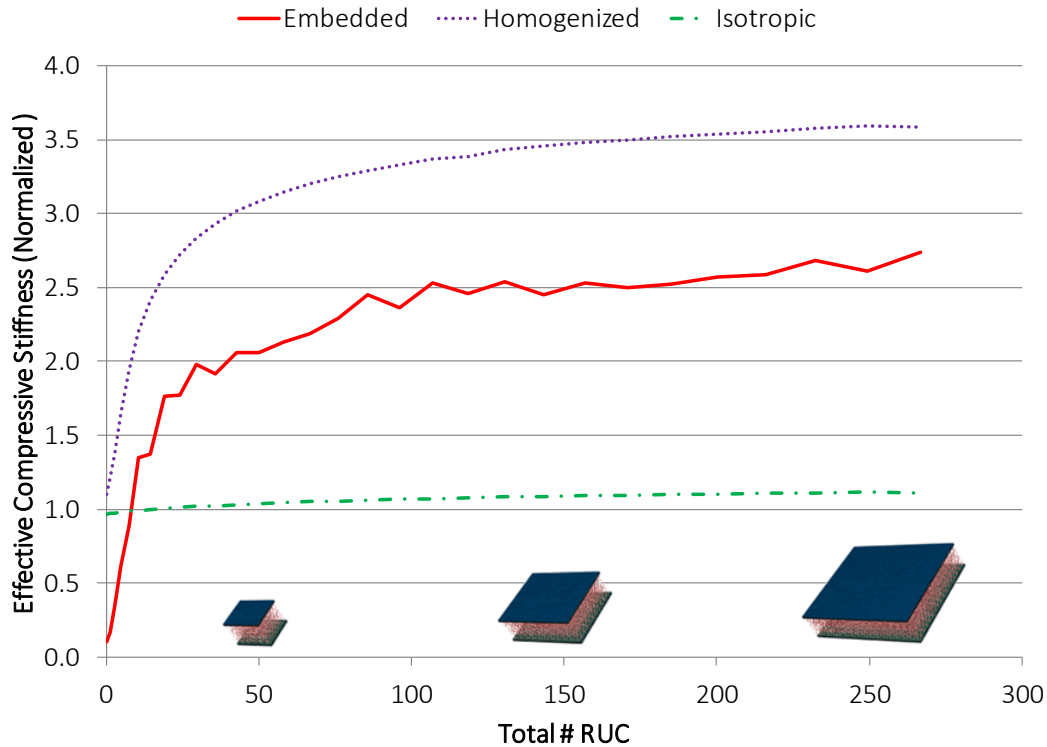
Figure 5.3. Effect of specimen size on effective compression stiffness with various core models.

A comparison of the isotropic and homogenized c-UDG models helps to highlight the effect of specimen geometry and the interaction of the orthotropic properties as both models have the same through-thickness modulus. While both models have nearly the same effective stiffness for the smallest size, the c-UDG models rapidly increases in effective stiffness up to 3.6 for the largest sample. In contrast, the isotropic models only increase to approximately 1.1 for the largest specimen size highlighting the importance of the orthotropic material interaction on the measured structural level stiffness. The embedded element models help to highlight the third contributor to the structural response: the severed reinforcement pins. The stiffness for the smallest embedded model is much lower than the isotropic and homogenized models as the response is dominated by the softer foam (no fibers directly connect to both of the facesheets in the smaller sizes). The inherent roughness in the embedded curve is a result of variation in where a sample edge occurs within a RUC and is most pronounced for the smaller sizes. The trend of the embedded models is similar to the homogenized orthotropic models; however, even the largest embedded model has regions of cut reinforcement near the edges, Figure 5.4. An

illustration of how the affected edge is consistent between different specimen sizes is given in Figure 5.5.
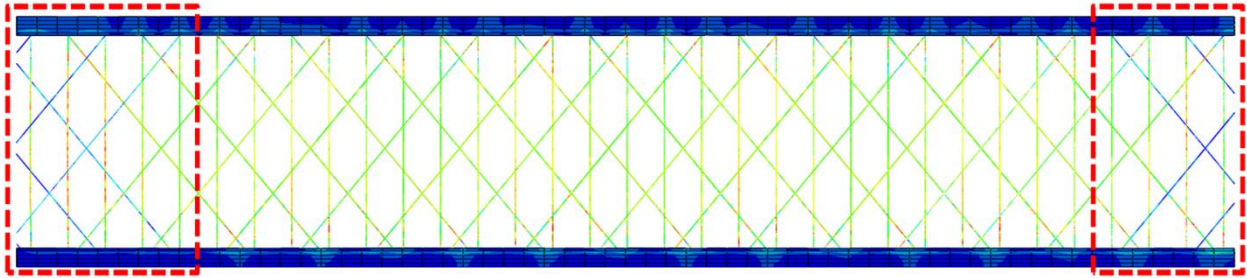
Figure 5.4. Partially bonded regions highlighted in slice of 267 RUC FWC model.
(Not to scale.)
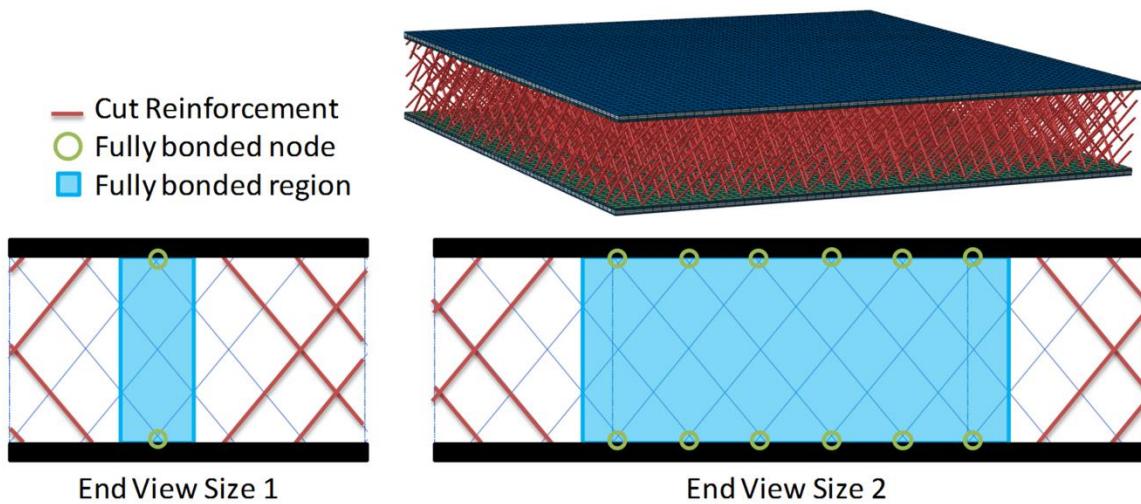
Figure 5.5. Example of region affected by cut edges in two different sample sizes.
(Not to scale.)

A closer look at the cut fiber regions of the FWC samples show that approximately 50% of the fibers shown outside the fully bonded region are cut, Figure 5.4. It is important to note that for a given cross section of the sample only fibers parallel to the viewing plane are affected by the cut edge. This can be seen in Figure 5.4 where the out of plane fibers in the highlighted regions (vertical lines) are still carrying load. This equates to approximately 25% of the total fibers within the partially bonded regions being cut. An effective area can then be used based on the total number of fibers fully bonded. When the effective area is applied to the embedded model results the stiffness plateaus much sooner and to a higher value; however the inherent roughness is still present and may be reduced through more careful measurement of the effective area at each model scale, Figure 5.6. The plateau values for the embedded and homogenized

methods differ even with using the effective area approach. This difference is supported by the deviation observed in effective through thickness stiffness calculated using the multiscale approach in Chapter 2.
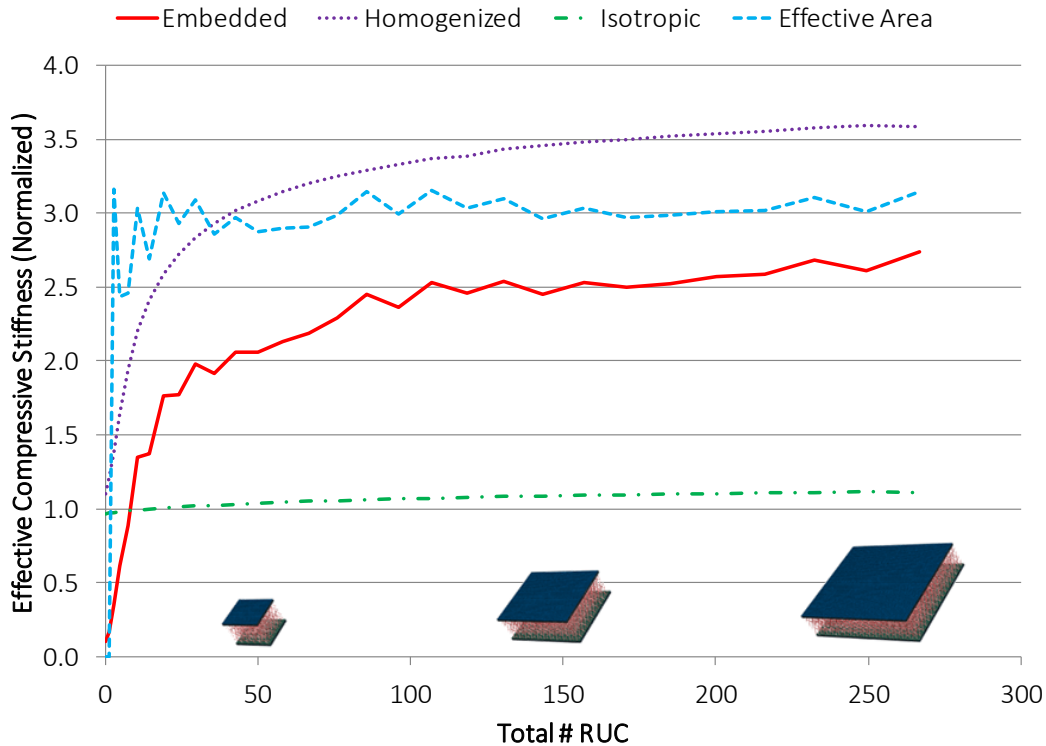
Figure 5.6. Effective compression stiffness vs. specimen size with effective area.

5.3 Flatwise Tension Testing

5.3.1 Experimental Results

Testing on 3DFRFC sandwich and unreinforced foam specimens in flatwise (through-thickness) tension was conducted in accordance with ASTM C297 [70] (Figure 5.7). Unreinforced foam specimens were tested of one size (Size 1), whereas, the 3DFRFC sandwich specimens were tested in two sizes: denoted Size 1 and Size 2. Note all specimens were of the same thickness. Size 1 3DFRFC specimens contained approximately 30 RUCs while Size 2 specimens contained approximately 130 RUCs. Testing two sizes of the 3DFRFC specimens

allows a study into the free-edge effect, which is caused by the severed reinforcing fibers at the sides of the test specimens. The measured strengths for the flatwise tension specimens are given in Table 5.1. The strengths are normalized to the average strength measured for the Size 2 FWT specimens to allow for direct comparison. The larger Size 2 3DFRFC specimens exhibit higher failure strength due to the increase in proportion of fully bonded through-thickness fibers. Similar behavior has been discussed for the analysis of size effects in flatwise compression 3DFRFC specimens, Section 5.2. Another key finding of the flatwise tensile testing is the difference in observed failure modes between the unreinforced foam and 3DFRFC sandwich samples. The unreinforced foam samples exhibit tensile failure within the foam (Figure 5.8). Unlike the foam, both sizes of 3DFRFC sandwich samples exhibit failure primarily within the adhesively bonded region between the facesheet and the ends of the through-thickness reinforcing fibers (Figures 5.9 & 5.10). Failure in the 3DFRFC samples is accompanied by coincident pullout of some of the partially bonded reinforcing fibers near the edge of the specimen.

Figure 5.7: Flatwise tension specimen in experimental setup.
(Image used with the permission of The Aerospace Corporation.)

Table 5.1. Normalized strength for flatwise tension specimens; Size 2 FWT=1.0.

Specimen	Unreinforced Foam	3DFRFC - Size 1	3DFRFC - Size 2
1	0.141	0.754	0.995
2	0.135	0.726	1.037
3	0.128	0.770	0.967
4	0.123	0.748	0.962
5	0.130	0.761	1.048
6	-	-	0.991
Average (Pa/Pa)	0.131	0.752	1.000
Standard Deviation (Pa/Pa)	0.007	0.016	0.035
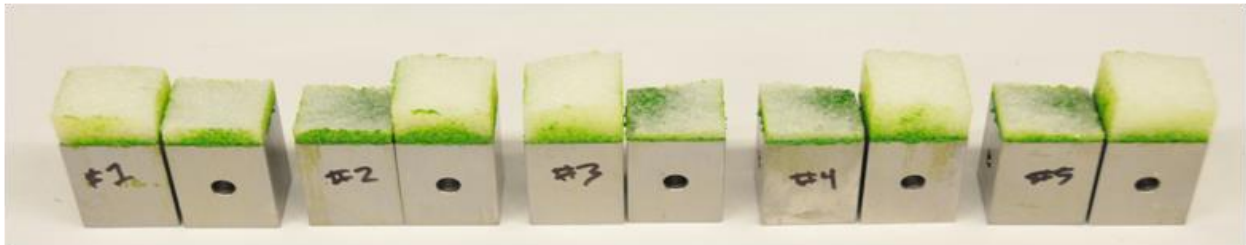
Figure 5.8. Failed unreinforced foam flatwise tension specimens.
(Not to scale. Image used with the permission of The Aerospace Corporation.)
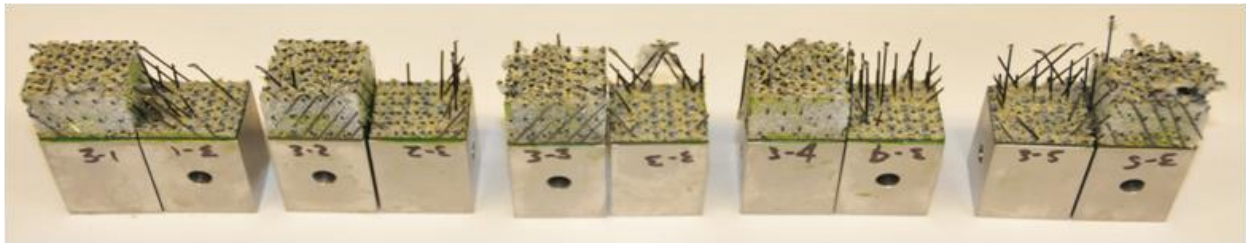
Figure 5.9. Failed Size 1 3DFRFC sandwich flatwise tension specimens.
(Not to scale. Image used with the permission of The Aerospace Corporation.)
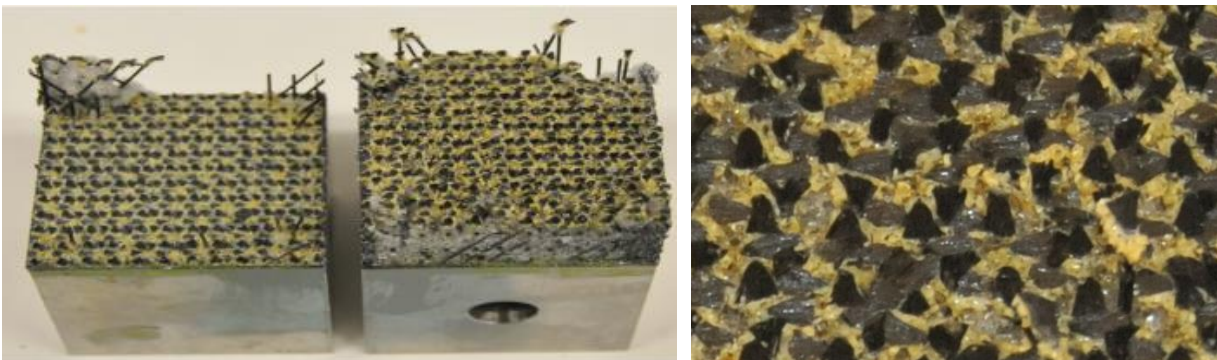
Figure 5.10. Typical failure observed in Size 2 3DFRFC FWT specimen, left, close-up of failure surface, right.
(Not to scale. Image used with the permission of The Aerospace Corporation.)

A side view of a Size 2 3DFRFC sample before and after failure is given in Figure 5.11 with the fracture edges circled in red. The side of the sample has a speckle pattern to allow for surface

analysis of the strain field using digital image correlation (DIC). DIC analysis of this test highlights the localized strain fields present at the edge of these samples (Figure 5.12). The red arrows indicate two faint bands of localized strains in the sample that correspond to the location of severed out-of-plane reinforcing fibers at the specimen edge. Clearer banding was observed in another test specimen, Figure 5.13. The interaction of the constituents within the microstructure including the failure of the adhesive layer is a key to modeling the complex failure in the 3DFRFC. The following section highlights methods for modeling the failure within these samples.
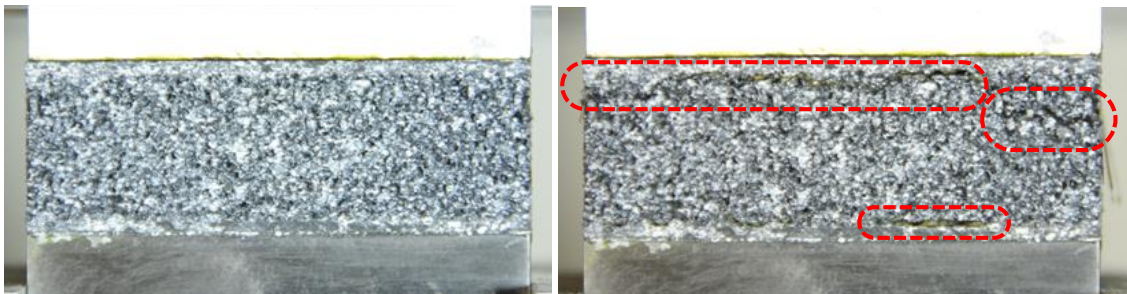
Figure 5.11. Side view of the 3DFRFC Size 2 FWT before failure (left) and after failure (right). (Not to scale. Image used with the permission of The Aerospace Corporation.)
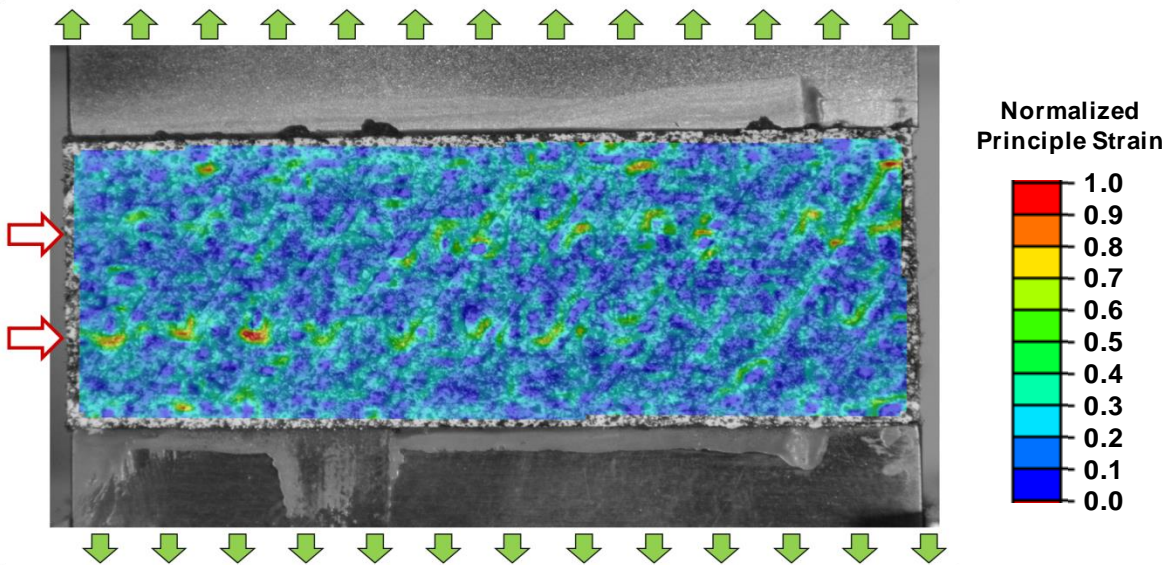
Figure 5.12. DIC image from FWT test of Size 2 3DFRFC. (Not to scale.)
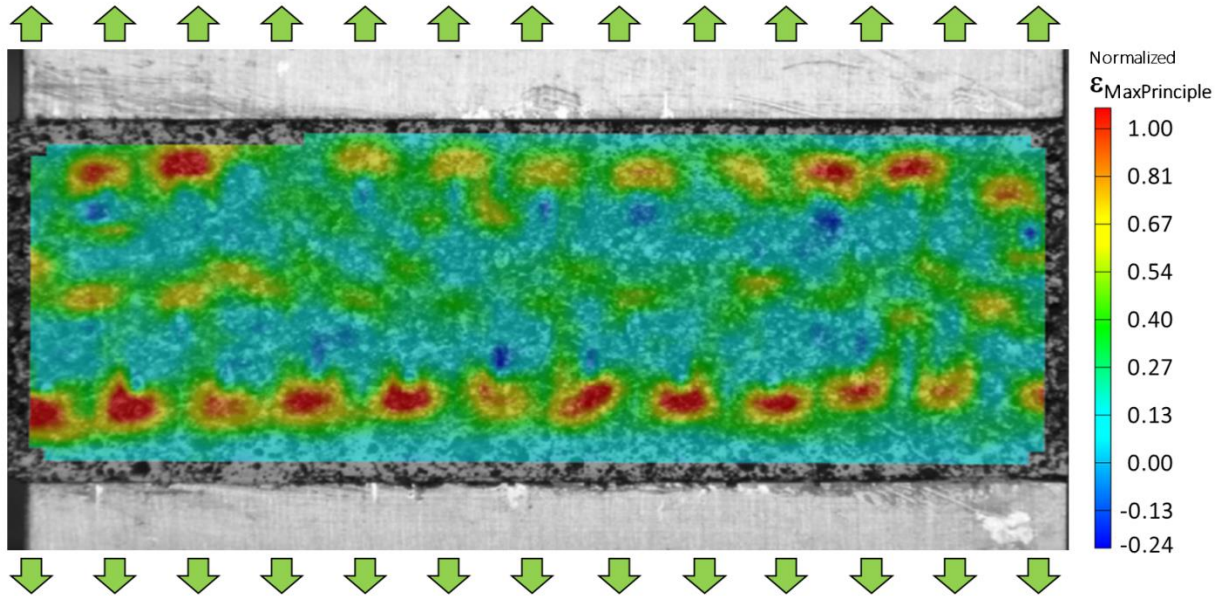
Figure 5.13. DIC image from FWT test of Size 2 3DFRFC.
(Not to scale.)

5.3.2 Microstructure Modeling

The evaluation of the flatwise tension configurations was performed with a finite element analysis using Abaqus. The models for the FWT samples were generated in the same fashion as the models discussed in the size effects section. A detailed view of the reinforcement within the flatwise tension specimen is given alongside an internal view of the displacement field within the Size 1 3DFRFC model in Figure 5.14. There is a clear gradient in the displacement field in the area of foam surrounding the severed reinforcing fibers near the edge of the specimen. Similar behavior is demonstrated by looking at the strain field along the specimen edge (Figure 5.15). The red arrows indicate bands of localized strains in the model that correspond to the location of severed out-of-plane reinforcing fibers at the specimen edge similar to the behavior observed in testing. It is worth noting that this banded behavior is strictly due to the severed pin ends and as such will be a function of the location of the cut within the specimen. For example another side of the same sample might exhibit only one row of banding. Additional insight into the local interaction of the 3DFRFC can be garnered by interrogating the stress field within the adhesive layer between the 3DFRFC and the facesheets (Figure 5.16). In the absence of any failure, stress concentrations are clearly present as a result of the bonding to the through-thickness reinforcing

91

fibers. The red boxed area in Figure 5.16 shows the one and only RUC within the sample with all 4 pins clearly transferring load. This further supports the information presented in the size effects section and gives a clearer illustration of the necessity to test larger samples.
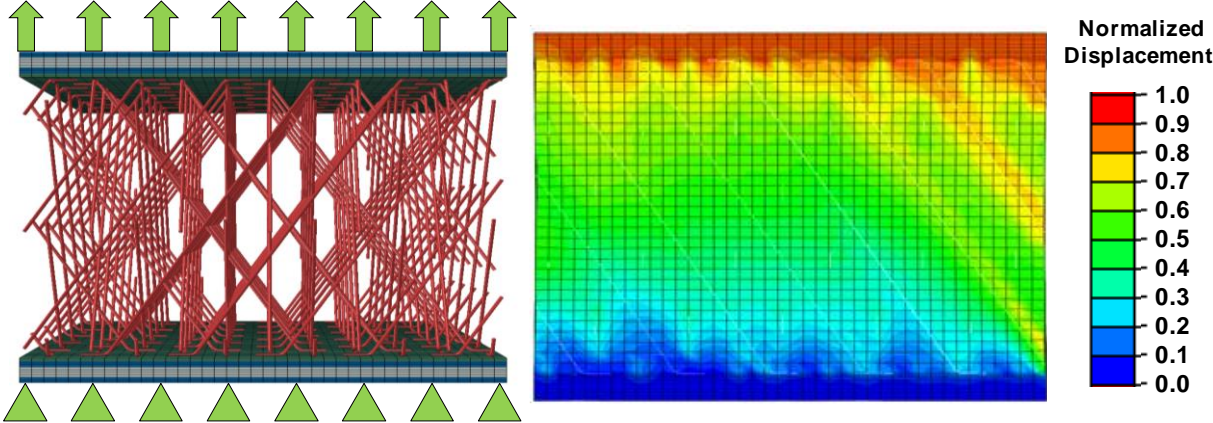
Figure 5.14. Side view of Size 1 3DFRFC FWT model with foam removed, left, and displacement field interior to model, right.
(Not to scale.)
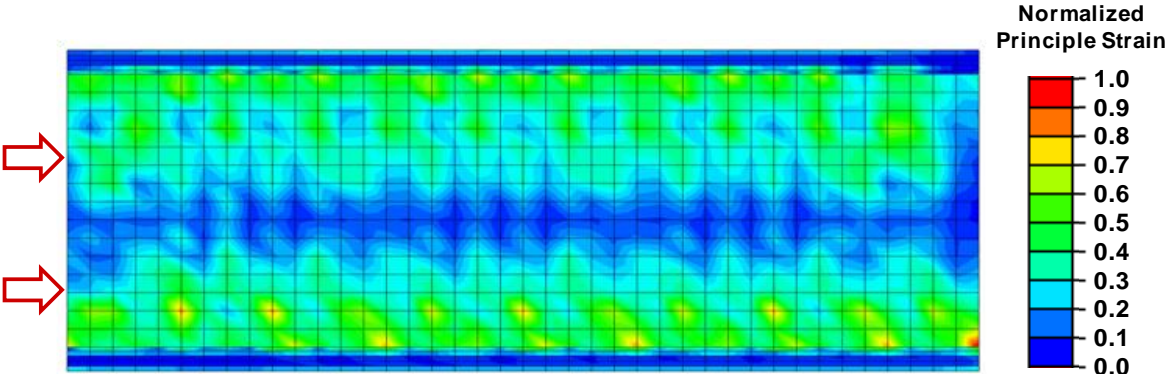
Figure 5.15. Strain distribution on specimen edge in Size 2 3DFRFC FWT model.
(Not to scale.)
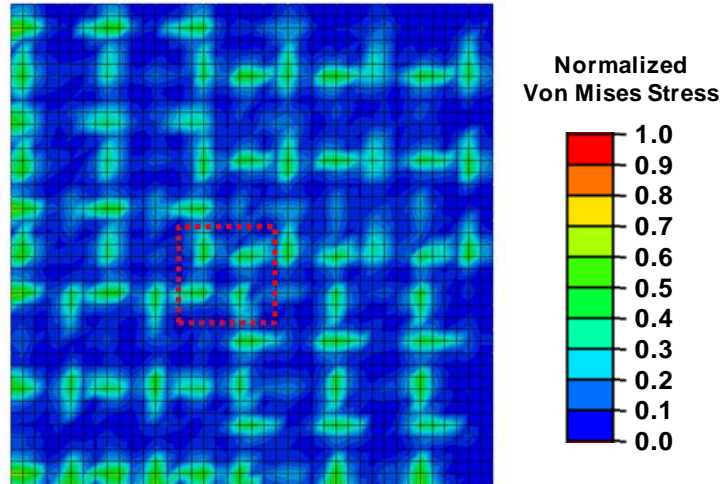
Figure 5.16. Stress concentrations in adhesive layer of Size 1 3DFRFC FWT model. (Fully bonded RUC in red.)

Failure modeling was facilitated through the implementation of the Smeared Crack Approach (SCA) [58] through a user defined material model within Abaqus. The SCA facilitates the modeling of complex failure paths that are not well defined and cannot be inferred a priori [71,72]. The SCA is used for modeling failure within the foam and the adhesive layer utilizing fracture toughness values published for each constituent [51,73]. Preliminarily, the critical stress for the foam was determined from modeling the Size 1 foam only tests, and the critical stress for the adhesive is determined from the Size 1 3DFRFC tests. Failure within the through-thickness reinforcing fibers is not observed in this loading configuration and was not modeled within the analysis.

The analysis of the unreinforced foam samples using SCA shows failure initiation near the bonding interfaces of the specimen (Figure 5.17). This is similar to the failure observed in the experiments; however, the analysis initially shows failure initiating at both sides of the specimen with eventual localization to one side. The real system is imperfect and will result in failure preferentially initiating at one end over the other; however, in the current model the side of eventual localization is solely a result of numerical variation. The properties determined from modeling the Size 1 unreinforced foam tests were implemented with the corresponding published material data into the Size 2 3DFRFC model. Results using these properties qualitatively show the capability of SCA to demonstrate failure initiation within the adhesive layer due to the stress concentration near the bonding interface with the through-thickness reinforcement (Figure 5.18).

The SCA shows promise for modeling more complex failure in the 3DFRFC; however, for it becomes cost prohibitive as specimen size increases. For many cases with failure more or less constrained to the adhesive layer, utilizing cohesive type elements [56] in conjunction with the effective orthotropic properties derived in Chapter 2 will result in a more efficient analysis (Chapter 4). Regardless of modeling method used, additional effort is needed to accurately quantify the properties of the bonding between the 3DFRFC and the adhesive layer, that effort was the focus of the interface fracture testing development given in Chapter 3.
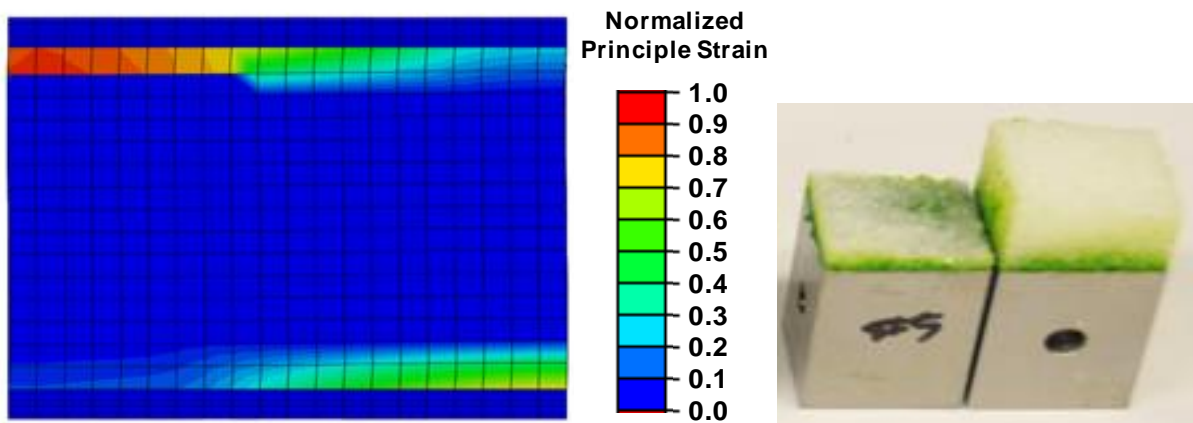
Figure 5.17. Failure localization in FWT unreinforced foam model using SCA, left, and failure location in unreinforced foam sample, right.
(Not to scale. Image used with the permission of The Aerospace Corporation.)
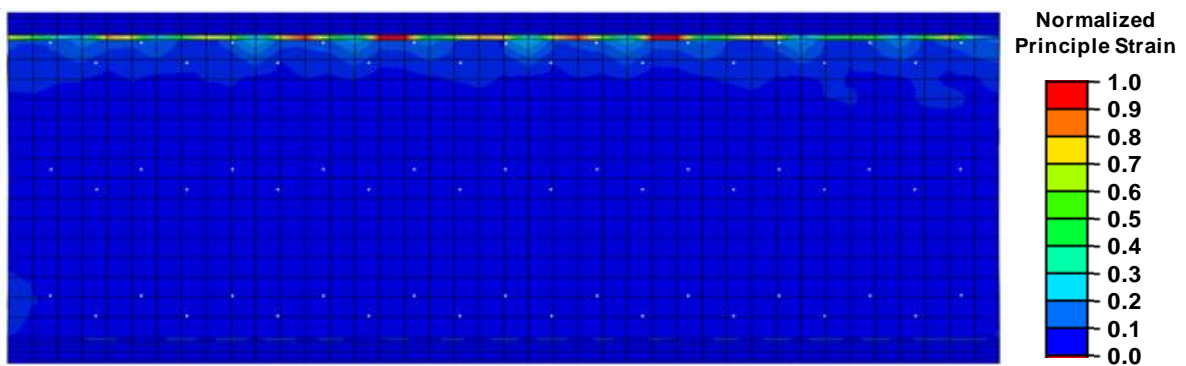
Figure 5.18. Internal failure localization in adhesive layer for FWT Size 2 embedded element 3DFRFC sandwich model using SCA.
(Not to scale.)

94

5.4 Flatwise Compression Testing

5.4.1 Experimental Results

Testing on 3DFRFC sandwich and unreinforced foam specimens in flatwise (through-thickness) compression was conducted in accordance with ASTM C365 [74] (Figure 5.19). Similarly to the FWT testing unreinforced foam specimens were tested of one size (Size 1), whereas, the 3DFRFC sandwich specimens were tested in two sizes: denoted Size 1 (30 RUC) and Size 2 (130 RUC). The measured strengths for the flatwise compression specimens are given in Table 5.2. The strengths are normalized to the average strength measured for the Size 2 *FWT* specimens to allow for direct comparison between configurations. Similar to the FWT results the larger Size 2 3DFRFC specimens exhibit higher failure strength due to the increase in proportion of fully bonded through-thickness fibers. This behavior was discussed for the analysis of size effects in the FWC 3DFRFC specimens [75]. An additional challenge for the FWC configuration is that the failure of the core is inherently hidden from view. Some of the specimens were observed to exhibit pin buckling as evident, Figure 5.20 and Figure 5.21, but it was expected that this failure mode was only applicable to pins that were not fully supported by the foam and would not be representative of the bulk behavior of the 3DFRFC.
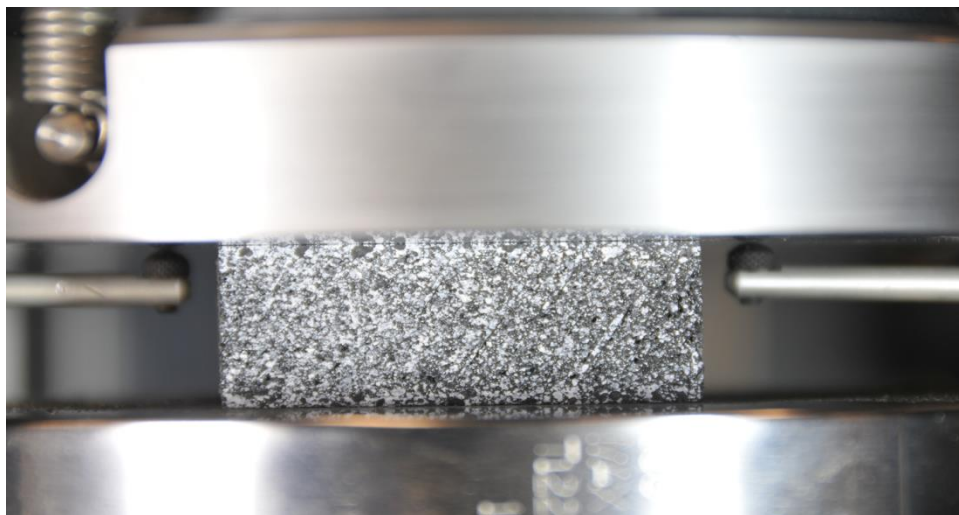
Figure 5.19. Flatwise compression specimen in experimental setup.
(Not to scale. Image used with the permission of The Aerospace Corporation.)

Table 5.2. Normalized strength for flatwise compression specimens; Size 2 FWT=1.0.

Specimen	Unreinforced Foam	3DFRFC - Size 1	3DFRFC - Size 2
1	0.0628	1.993	1.986
2	0.0627	1.328	2.163
3	0.0635	1.568	2.110
4	0.0633	1.764	2.275
5	0.0633	1.643	2.247
6	-	-	2.123
7	-	-	2.191
8	-	-	2.224
Average (Pa/Pa)	0.0631	1.659	2.165
Standard Deviation (Pa/Pa)	0.0003	0.245	0.092
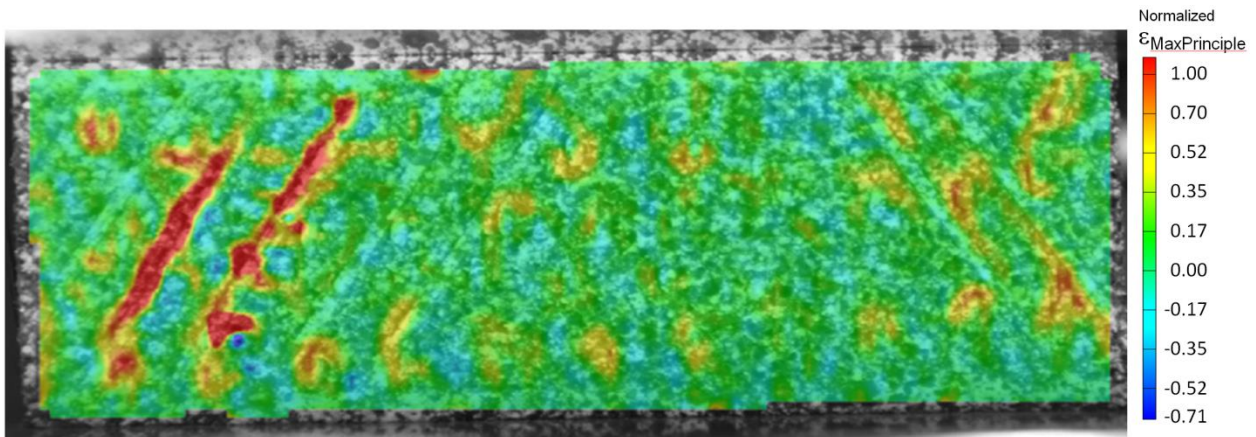
Figure 5.20. DIC image from FWC test of Size 2 3DFRFC.
(Not to scale.)
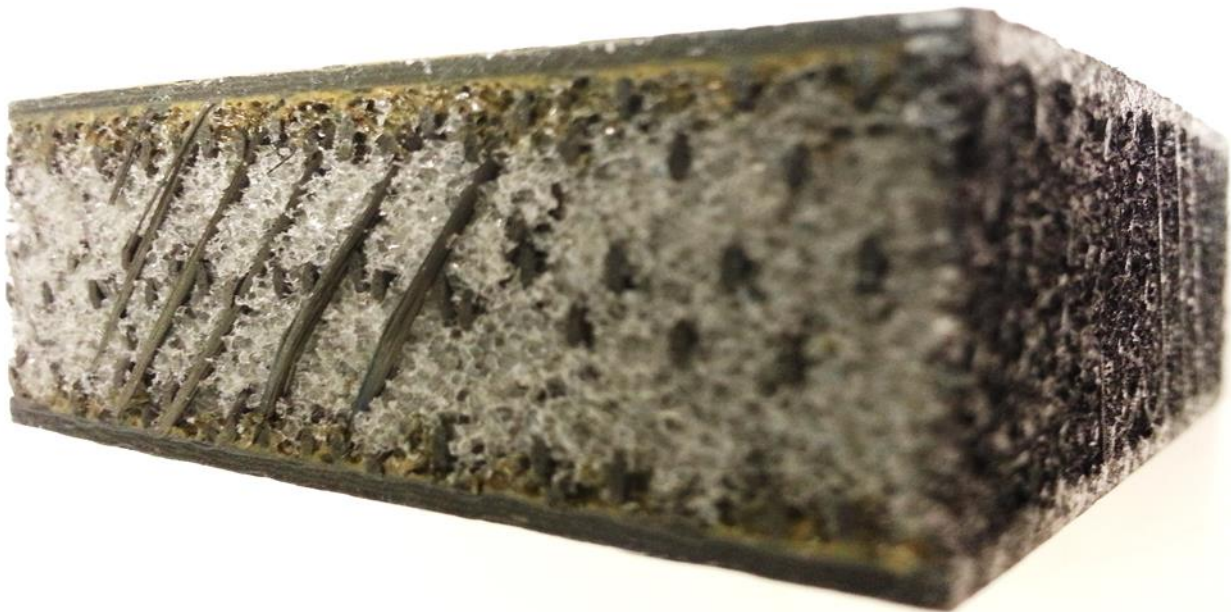
Figure 5.21. Externally visible pin buckling in size 2 3DFRFC FWC specimen.
(Not to scale.)

5.4.2 Internal Failure Mode

Most of the through thickness compression modeling was discussed in Section 5.2. An additional qualitative prediction was made to gain some understanding of the failure mode of the FWC configuration. One of the challenges with this loading configuration is that, unlike the tensile loading configuration, the mechanisms of failure away from the specimen edges are not immediately apparent. Investigation into the stress distributions within the FWC configuration showed high levels of compressive stress within the reinforcing fibers as to be expected, Figure 5.22. The prior buckling analysis demonstrated that the pins were not expected to buckle. Internal interrogation of failed FWC specimens was conducted using microCT to confirm compressive failure of the through thickness reinforcement, Figure 5.23.
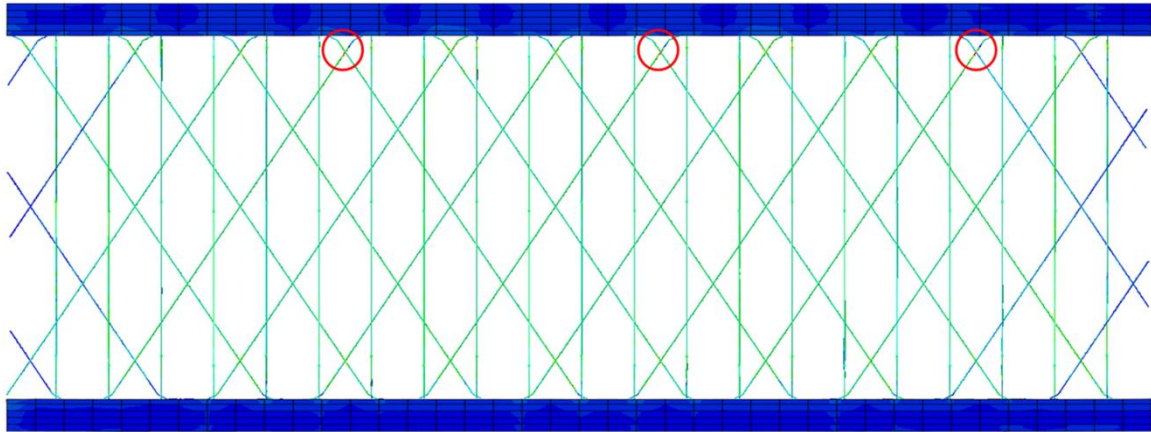
**Figure 5.22. Side view of Size 2 FWC model with foam removed. Areas of highest stress.
(Not to scale.)**
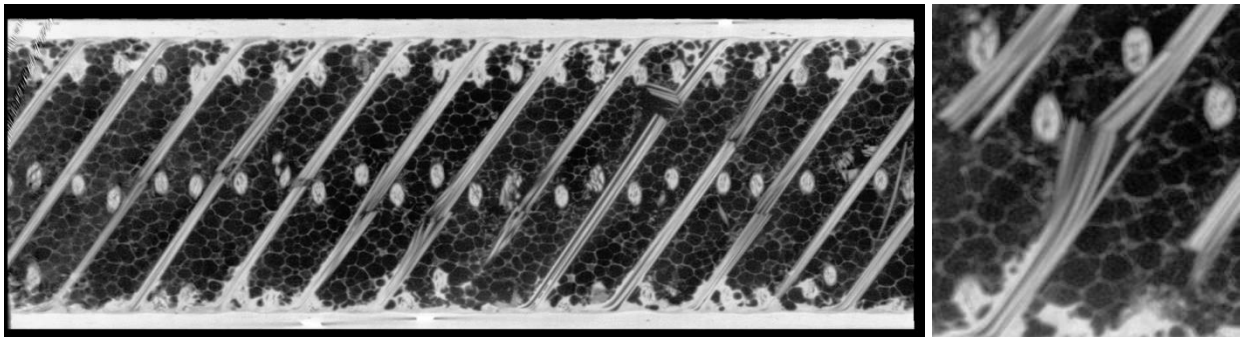
**Figure 5.23. MicroCT images from flatwise compression test of Size 2 3DFRFC.
(Not to scale.)**

The microCT scans of the interior of the specimen provided a couple useful pieces of additional information. First, they showed that none of the partially bonded pins near the specimen edges failed which supports the edge behavior observed in the size effect study. Second, some of the reinforcing fibers demonstrated a splitting type failure. This failure mechanism is likely due to the presence of internal voids in the reinforcement that run parallel to the pin. These features can clearly be seen in the high resolution microCT image, Figure 5.24.Similar behavior was observed for z-pinned sandwich structures, [76].
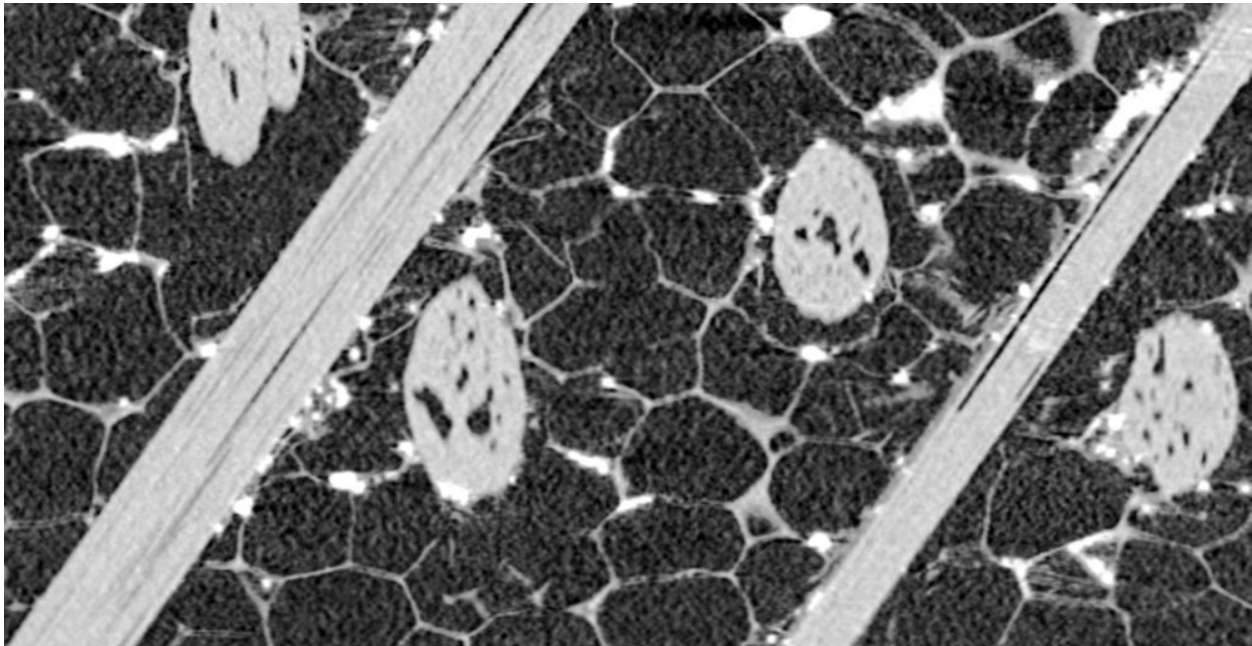
Figure 5.24. MicroCT image from flatwise compression test of Size 2 3DFRFC.
(Not to scale. Image used with the permission of The Aerospace Corporation.)

5.5 Through Thickness Shear Testing

5.5.1 Through Thickness Shear Experimental Results

Testing on 3DFRFC sandwich specimens in through-thickness shear (3-point bending) was conducted in accordance with ASTM C393 [77] (Figure 5.25). This flexure specimen is of the same material as the other tests; however it is significantly larger consisting of approximately 710 RUCs. As with the tensile and compression data, all measured strength data was again

normalized by the size 2 *FWT* value to allow for easier comparison with the other test configurations, Table 5.3. The key finding from the flexure test was that the measured shear strength was significantly lower than the values obtained in the pure tension or compression cases. Explanation for this behavior is given in the following modeling section. DIC was again used to gain additional insight into the specimen behavior prior to failure. An example showing strain localization just prior to failure is shown in Figure 5.26. As with the FWC samples the details of how the 3DFRFC shear samples failed internally is not immediately discernable. In the shear loading case half of the pins should be loaded in tension and the others in compression. An educated guess would point to tensile failure at the bond line since the measured compressive strength was twice the through thickness tensile strength. MicroCT was used to confirm this failure mode, Figure 5.27.
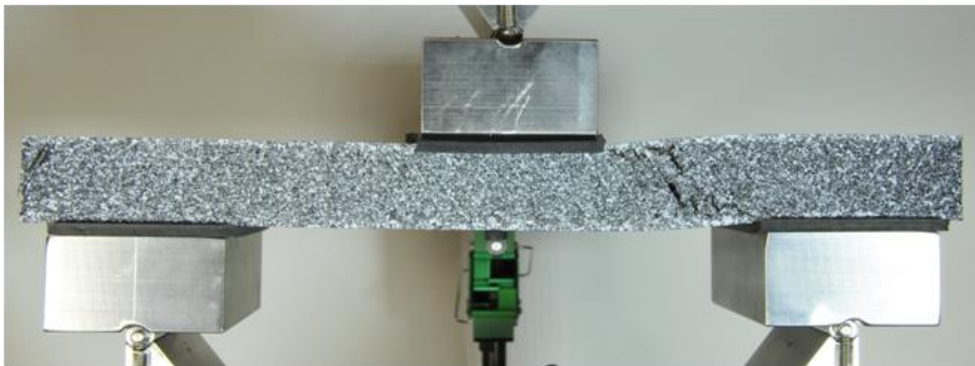
Figure 5.25. Experimental setup and typical failure observed in 3DFRFC flexure specimens. (Not to scale. Image used with the permission of The Aerospace Corporation.)

Table 5.3. Normalized Shear Strength for 3DFRFC; Size 2 FWT=1.0.

# of samples	Average (Pa/Pa)	Standard Deviation (Pa/Pa)
5	0.558	0.029
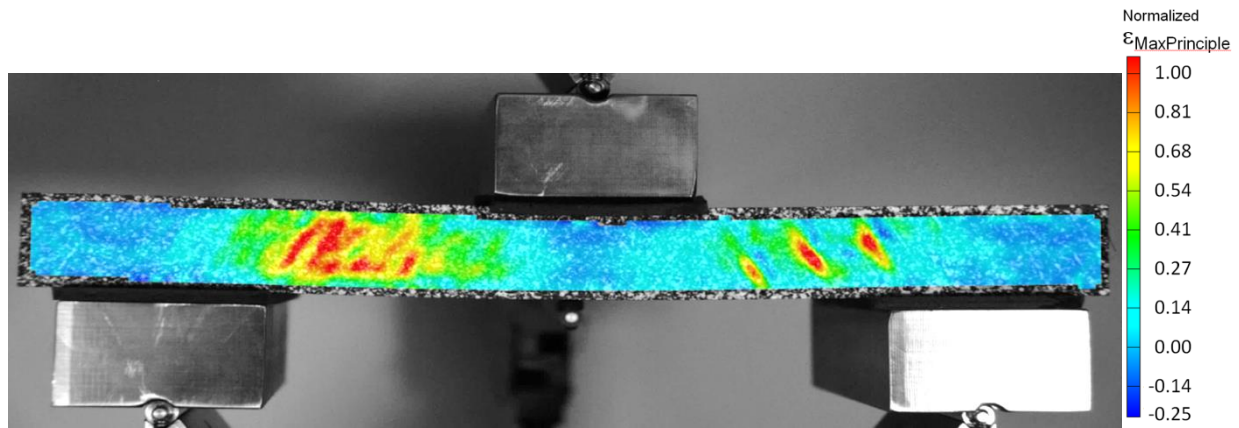
Figure 5.26: DIC image from flexure test of 3DFRFC.
(Not to scale.)
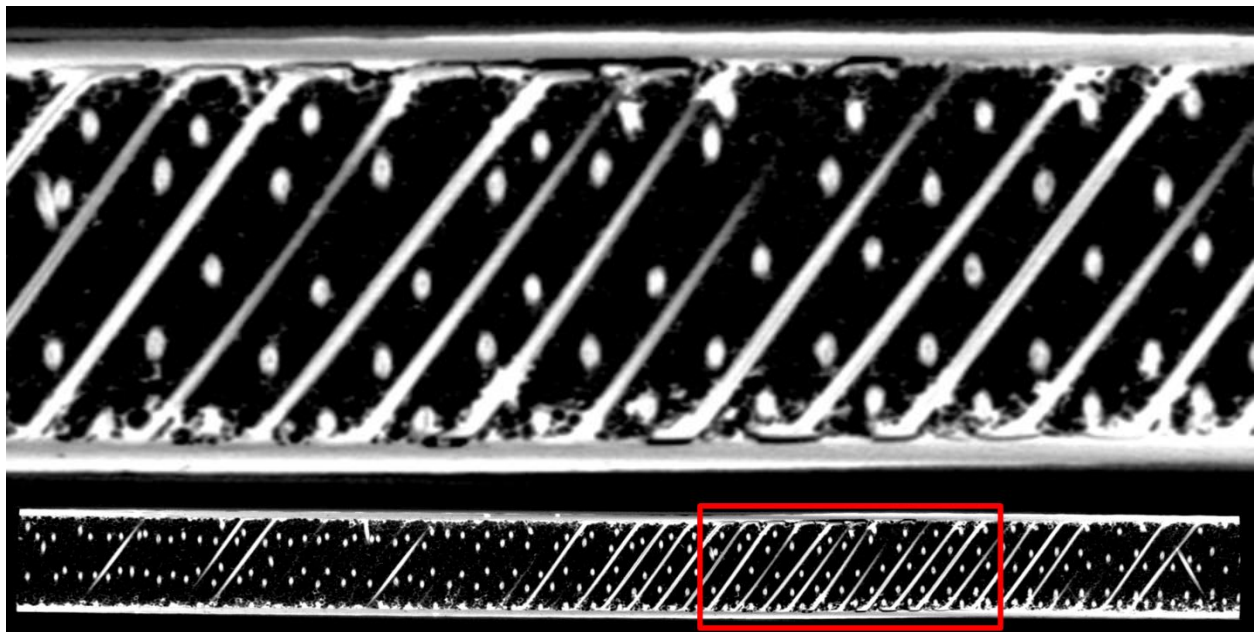
Figure 5.27. MicroCT image from flexure test of 3DFRFC.
(Not to scale. Image used with the permission of The Aerospace Corporation.)

5.5.2 Through Thickness Shear Microstructure Modeling

Embedded element modeling of the three-point bend configuration provided additional insight into the behavior of the specimens. While the Size 2 flatwise specimens contained approximately 130 RUCs, the much larger three-point bending specimens contained over 700 RUCs. An example of one of the embedded models showing a pattern due to variations in facesheet stress caused by the presence of the discreetly pins is given in Figure 5.28.

Examination of the stress distribution through the cross-section of the specimen between the loading points highlights the key cause of the observed strength reduction for the three-point bending specimen. While the truss members within the pure tension and compression cases result in all of the fully bonded fibers being loaded in a similar manner, this is not the case with the bending specimen. There is relative inactivity of the reinforcement orthogonal to the span direction for the bending specimen, dark blue pins in Figure 5.29. The resultant effect is that the reinforcing pins along the span direction are relied on almost entirely for load transfer effectively reducing the number of reinforcing fibers available for load transfer by 50%. This explains the relative reduction in shear strength compared to the pure tension or compression cases. The observed reduction was only 44%; however, in the 3-point bending specimens none of the load carrying fibers are severed since they are parallel to the cut plane. In Figure 5.29 the pins near the specimen edge are still capable of carrying load (light blue).
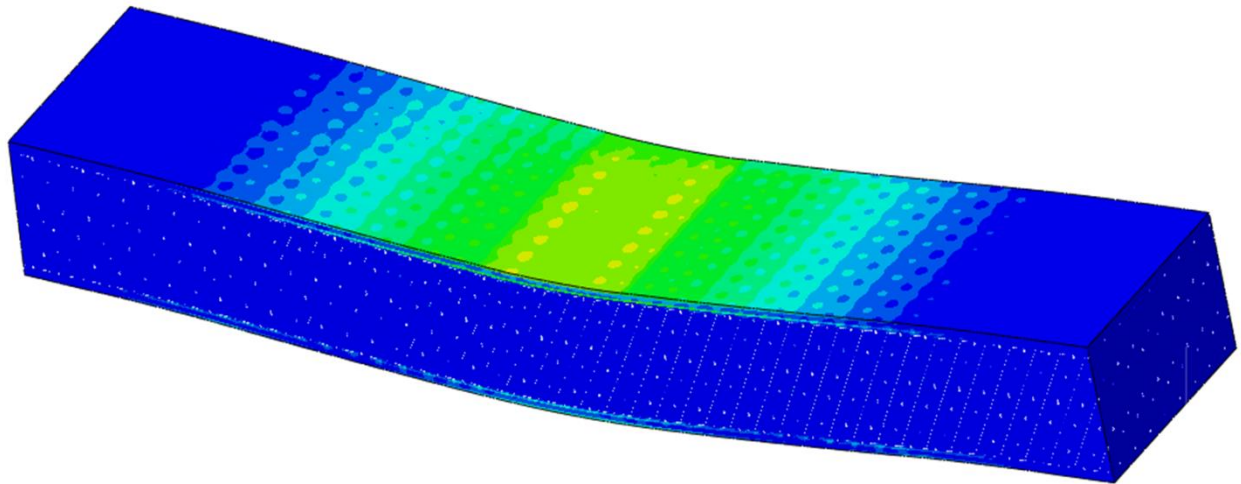
Figure 5.28. Cutaway view showing 3-point bend specimen cut along midline, Mises stress. (Not to scale.)
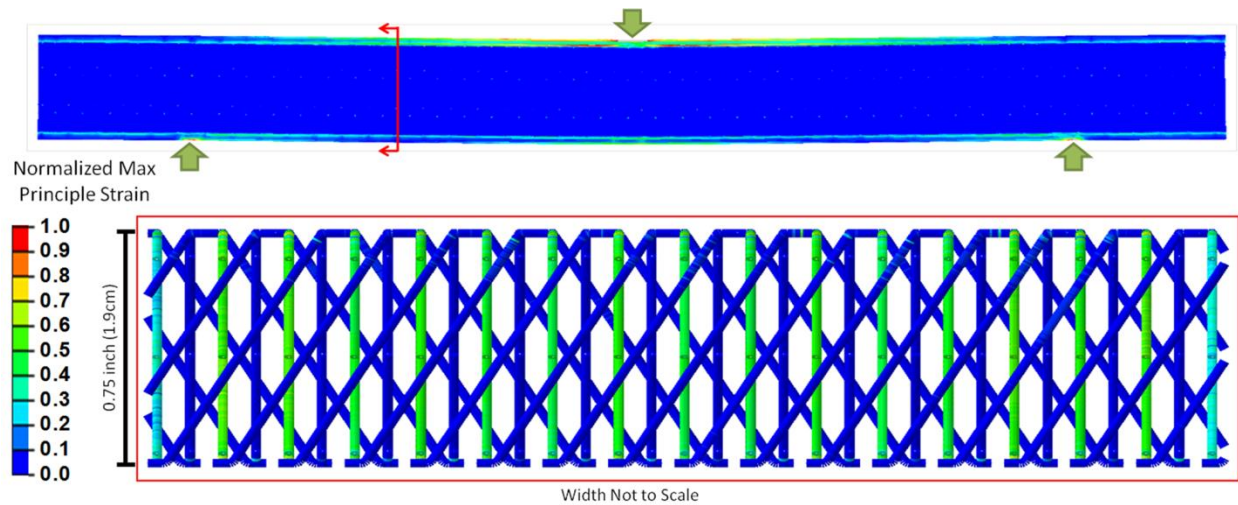
Figure 5.29. Normalized max principle strain in flexure specimen at failure load.
(Not to scale. Section view shown with foam removed.)

5.6 Summary

The results of experimental investigation into the through thickness failure of 3DFRFC sandwich composites and models to understand their behavior was presented. Key findings included:

- 3DFRFC specimens demonstrated considerable strength increase over similar density unreinforced foams: FWT >30%, FWC>100%, shear >5% [78].

- The effect of increased strength with increased specimen size was confirmed by through-thickness tension and compression testing.

- Detailed finite element analysis highlighted the lower percentage of load bearing fibers as the primary cause for the reduced strength in the three-point-bend tests.

- Thorough discussion of the specimen size effects was given highlighting the difficulty in directly measuring the through-thickness modulus of the 3DFRFC structures.

CHAPTER 6
Cold Temperature Testing of 3DFRFC

6.1 Introduction

A current engineering challenge for composite sandwich structures is to quantify their ability to tolerate damage at operating temperatures, particularly in launch vehicles and spacecraft where mission assurance is critical to mitigating cost from loss or failure. The strength of sandwich composites can be reduced through many mechanisms, including impact damage, embedded foreign objects, use of poor bonding agents, or surface preparation issues. Recently, new core materials have been developed that have the potential to affect the damage tolerance of sandwich composites particularly in cold temperature environments. One class of core material being considered may alter their damage tolerance through the use of a three-dimensional, truss-like network of reinforcing fibers inside a lightweight foam core. Examples of this emerging class of core materials include NidaFusion[1,2], TYCOR®[3], and K-Cor®[4], Figure 6.1.
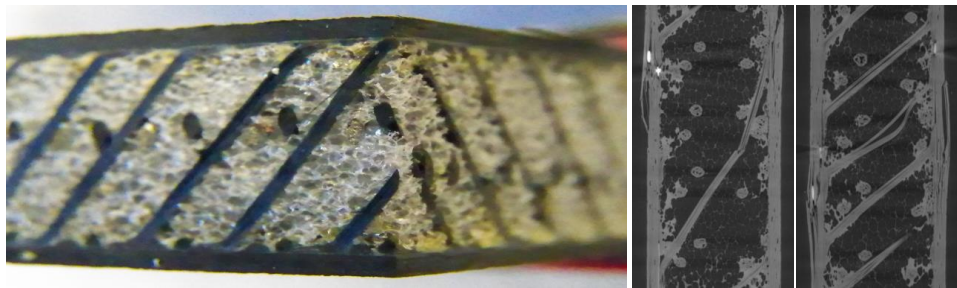
Figure 6.1. 3DFRFC sandwich, left, and microCT scans of failed reinforcement, right.
(Not to scale. Image used with the permission of The Aerospace Corporation.)

The truss structure of the 3D fiber network provides added paths for load transfer, acts to impede crack propagation within the foam, and affects the thermal interaction of the sandwich composite. As a result of the added load paths, the failure and strength of these three-dimensionally reinforced sandwich composites become difficult to predict. This research aims to

experimentally quantify the mechanical performance of a 3D Fiber Reinforced Foam Core (3DFRFC) sandwich composite at cold temperatures. Due to the complexity of the 3DFRFC, an investigation of the test geometry and thermal interaction is performed in parallel to the experimental investigation of the sandwich specimens.

The general manufacturing procedure for the 3DFRFC sandwich samples is similar to the one published previously for the manufacture of edgewise compression samples with defects [59], but it is included here for completeness. The material system chosen for this investigation is IM7/8552 carbon/epoxy prepreg for the facesheets and a 19 mm (0.75 inch) thick 0.192g/cm^3 (12lb/ft^3) 3DFRFC for the core. 3M™ Scotch-Weld™ AF191 film adhesive was used to bond the facesheets to the core. The manufactured 3DFRFC sandwich panels were inspected via nondestructive evaluation to ensure panel quality prior to removing the desired samples for cold temperature testing. All specimens were stored with a desiccant in sealed bags to prevent moisture ingress prior to testing. The cold temperature testing was conducted using an Instron machine fitted with an environmental chamber, Figure 6.2.

<center>(a)   (b)   (c)</center>
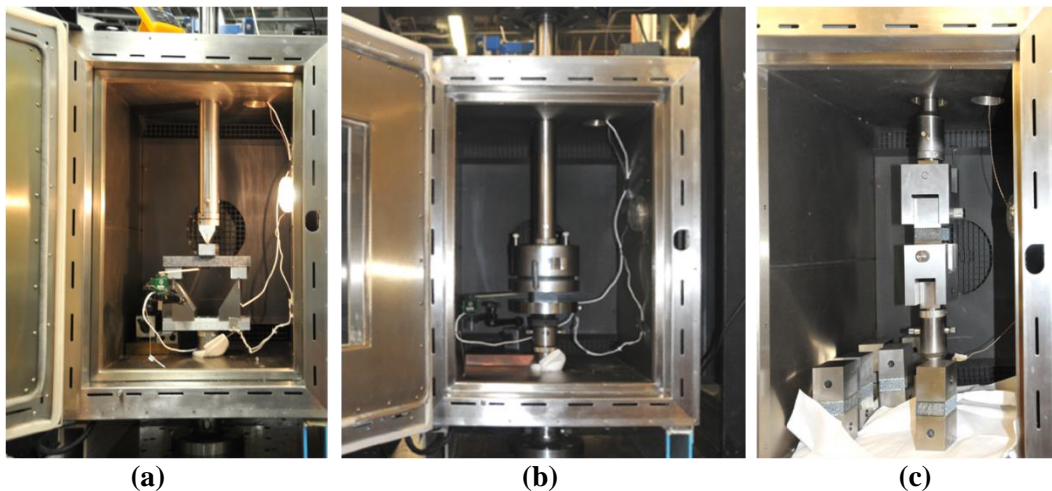
Figure 6.2. Test configurations for flexure (a), flatwise compression (b), and flatwise tension (c) .
(Not to scale. Images used with the permission of The Aerospace Corporation.)

Flatwise (through-thickness) tension and compression testing was conducted on unreinforced foam (lower density), Size 1 3DFRFC, and Size 2 3DFRFC at cold temperatures in accordance with ASTM C297 [70] and ASTM C365 [74]. Note all specimens were of the same thickness. However, the Size 2 specimen was twice the length and width of Size 1. Size 1 3DFRFC specimens contained approximately 30 representative unit cells (RUC) while Size 2

specimens contained approximately 130 RUCs. It is important to note that RUC in this context is defined strictly from the minimum geometric unit needed to capture the repeating structure of the 3DFRFC [75]. The three-point bend testing was performed on 3DFRFC composite sandwich coupons in accordance with ASTM C393 [77].

6.2 Thermo-mechanical Analysis

The cold temperature testing configurations were modeled using a finite element analysis. The analysis takes into account the effect of cool-down from the cure temperature as well as the thermal mismatch between the composite sandwich and the testing fixtures. A flowchart showing the basic procedure is given in Figure 6.3 for the flatwise tension configuration which includes the adhesive bonding of the loading blocks to the 3DFRFC sandwich at room temperature. The bonding of the specimen is only necessary for the flatwise tension case and this step is omitted for the compression and flexure analysis.

The thermal analysis is critical in isolating the thermal material performance of the 3DFRFC sandwich from the effects of the global interaction of the testing configuration. This analysis builds on the authors' previous work developing discrete modeling methods for these 3DFRFC sandwich composites [75,79,80]. The models utilized the embedded element method within the computational package Abaqus to model the discrete truss architecture of the 3DFRFC within the foam core, Figure 6.4. Note that the truss structure was modeled using beam elements, but many images show the beams with thickness for clarity.
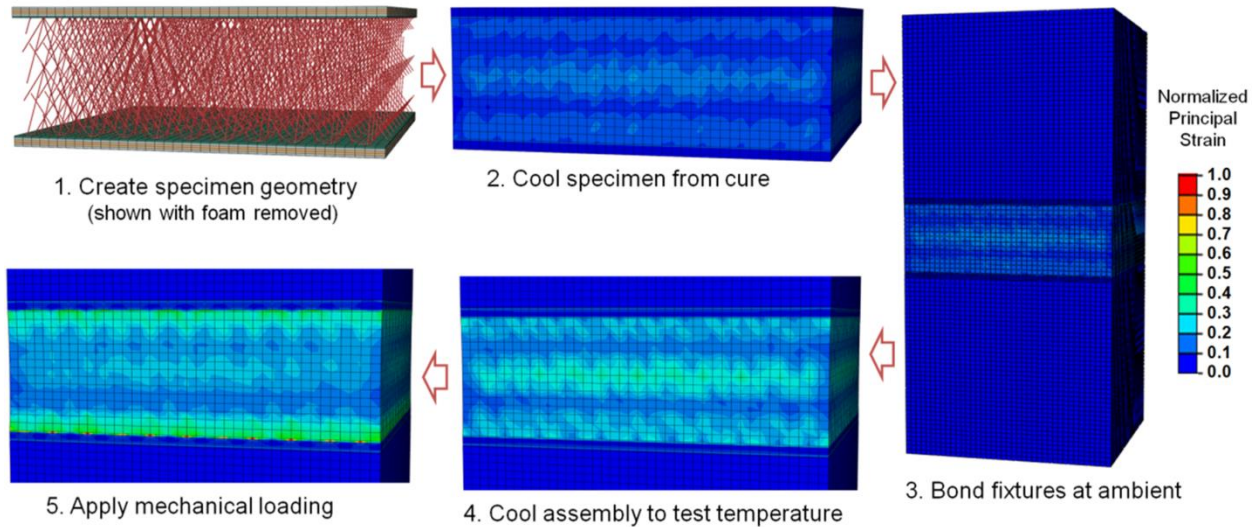
Figure 6.3. Flowchart for thermo-mechanical analysis for 3DFRFC sandwich specimens. (Not to scale.)
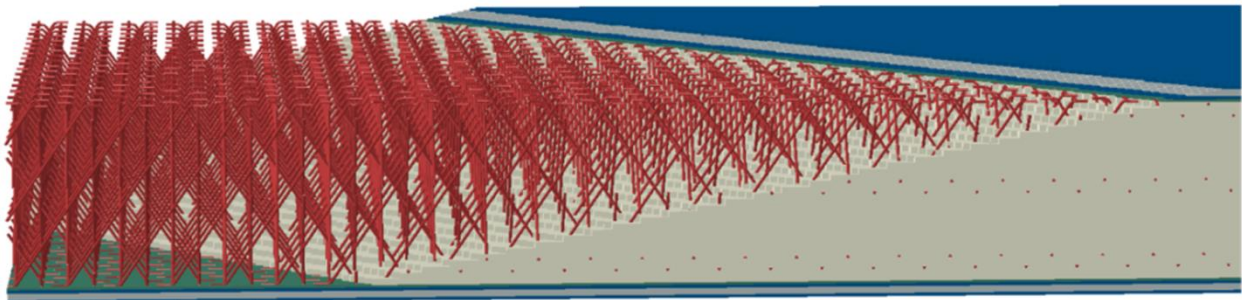
Figure 6.4. Cutaway view of embedded element model of a 3DFRFC sandwich structure. (Not to scale.)

All configurations modeled used solid elements for modeling the foam and individual adhesive and facesheet plies. Facesheet mechanical properties were measured during previous testing at The Aerospace Corporation [52], while the film adhesive properties were obtained from vendor data [65]. Thermal expansion properties were taken from internal testing when available. Thermal expansion for constituents without internal test data was obtained from published values for comparable constituents [81].

6.3 Flatwise Compression

The strengths measured from the compression specimens were normalized by the average tensile strength from the ambient Size 2 3DFRFC tests to allow for direct comparison across the three test configurations, Table 6.1. The compressive strength for the unreinforced (lower density) foam was actually observed to increase by 17%. The observed increase at cold temperature is consistent with behavior for polymers and polymeric foams reported previously [82–85].

Table 6.1. Normalized flatwise compressive strengths; ambient Size 2 FWT=1.0.

Specimen	Unreinforced Foam		3DFRFC - Size 1		3DFRFC - Size 2	
	Ambient	Cold	Ambient	Cold	Ambient	Cold
1	0.0628	0.0750	1.993	1.411	1.986	2.126
2	0.0627	0.0701	1.328	1.433	2.163	1.974
3	0.0635	0.0695	1.568	1.542	2.110	1.923
4	0.0633	0.0699	1.764	1.293	2.275	2.267
5	0.0633	0.0845	1.643	1.449	2.247	2.265
6	-	-	-	-	2.123	2.327
7	-	-	-	-	2.191	2.034
8	-	-	-	-	2.224	-
Average (Pa/Pa)	0.0631	0.0738	1.659	1.426	2.165	2.131
Standard Deviation (Pa/Pa)	0.0003	0.0064	0.245	0.089	0.092	0.159

The increase in compressive strength for the unreinforced foam did not correlate to increases in the 3DFRFC strength. In the 3DFRFC the relative increase in strength for the unreinforced foam was overshadowed by the increase in stresses induced by the thermal mismatch of the constituents at the lower temperature. This was supported by the finite element analysis that predicted the average residual stresses within the foam to already be 50% of the unreinforced strength at ambient and 100% under the cold conditions for the Size 1 specimens.

Testing two sizes of the 3DFRFC specimens also allowed insight into the free-edge effect, which are caused by the severed reinforcing fibers at the sides of the test specimens, Figure 6.5. Prior analytical work highlighted the importance of edge effects in 3DFRFC samples and how these effects reduce with increased specimen size [75]. The flatwise compression testing confirmed this trend. The dominant effect of the carbon fiber truss on the larger specimen size was highlighted by an increase in measured strength for the larger specimens: 30% increase in strength at ambient, 50% increase under cold conditions.
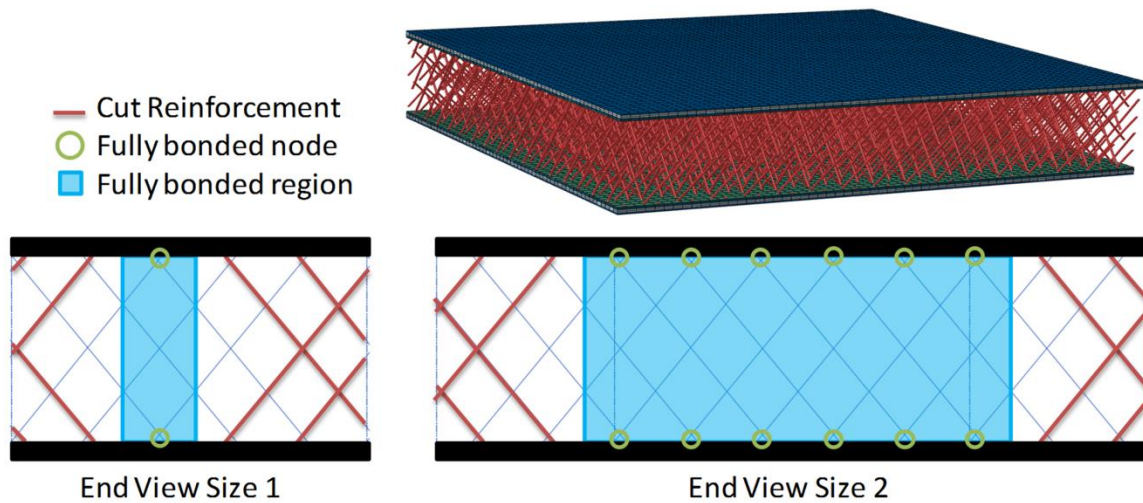
Figure 6.5. Example of region affected by cut edges in two different sample sizes. (Not to scale.) [75]

The reduction in strength under cold conditions observed in the Size 1 specimens can be attributed to the larger percentage of unbonded fibers in the smaller specimen coupled with tensile stresses that result from the thermal expansion mismatch. The presence of tensile stresses in the 3DFRFC foam is a key point and negates any benefit one might expect from the increase in foam compressive strength. The foam exhibited vastly differing temperature dependent behavior in tension versus compression. While the unreinforced foam exhibited an increase in compressive strength of 17% at cold temperature, the tensile strength was *reduced* by 58% under cold conditions (as presented in the Flatwise Tension section).

Investigation of the failure mode in the compressive samples through external examination is more difficult than for the tensile tests. Both the ambient and cold temperature samples exhibit relative displacement between the observable pins at the specimen edge and the foam, Figure 6.6, but the internal failure is not directly observable. The pins at the specimen edges are not fully bonded and constrained, whereas the reinforcement within the center of the specimen is fully bonded and constrained. The internal failure is likely a combination of local pin splitting and/or compressive kinking. A post-mortem investigation of the specimen interior using microCT is planned to verify the failure mode.
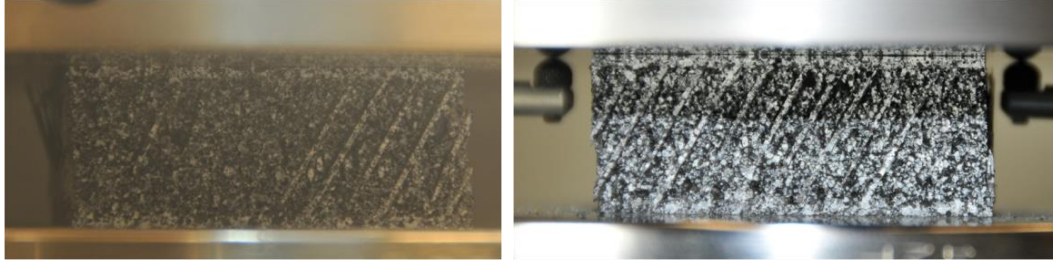
Figure 6.6. Failed Size 2 FWC specimens under cold, left, and ambient conditions, right. (Not to scale. Image used with the permission of The Aerospace Corporation.)

6.4 Flatwise Tension

All measured strength data for the flatwise tensile tests was normalized by the same value as the compression tests, Table 6.2. The ambient flatwise tensile data was published previously but is included for comparison purposes [80]. The test results show the largest decrease in strength for the unreinforced core (58%) with the Size 2 3DFRFC showing the most consistent performance across the two temperatures with only a 4% reduction in strength under cold conditions.

The tensile test exhibited the same size dependent behavior as the compression tests with increased measured strength for increased specimen size: 33% increase in strength at ambient, 103% increase at cold. The more pronounced increase in cold temperature strength from Size 1 to Size 2 in the tensile loading was likely a result of the higher percentage of fully bonded fibers available to transfer load as discussed in Section 6.3, Flatwise Compression.

Table 6.2. Normalized flatwise tensile strengths; ambient Size 2 FWT=1.0.

Specimen	Unreinforced Foam		3DFRFC - Size 1		3DFRFC - Size 2	
	Ambient	Cold	Ambient	Cold	Ambient	Cold
1	0.141	0.057	0.754	0.533	0.995	0.950
2	0.135	0.064	0.726	0.381	1.037	1.025
3	0.128	0.067	0.770	0.496	0.967	0.969
4	0.123	0.046	0.748	0.448	0.962	0.963
5	0.130	0.043	0.761	0.493	1.048	0.884
6	-	-	-	-	0.991	-
Average (Pa/Pa)	0.131	0.055	0.752	0.470	1.000	0.958
Standard Deviation (Pa/Pa)	0.007	0.011	0.016	0.058	0.035	0.050

The specimen failure modes for the cold temperature tension tests are the same as what was observed under ambient conditions [80]. Images of the failed cold temperature unreinforced

foam, Size 1 3DFRFC, and Size 2 3DFRFC samples are given in Figures 6.7, 6.8, and 6.9, respectively. Unlike the foam, the 3DFRFC specimens primarily exhibited failure inside the bonding interface between the reinforcing fibers of the core and the facesheet whereas the unreinforced foam fails *within* the foam near the bondline. The primary failure of the 3DFRFC inside the adhesive layer confirms the heavy reliance on the carbon truss as the primary means of load transfer. While the thermally induced stress predicted within the foam was similar to the compression specimens (~50% of strength at ambient, ~100% at cold), there did not appear to be a significant effect on the global failure of the specimens.
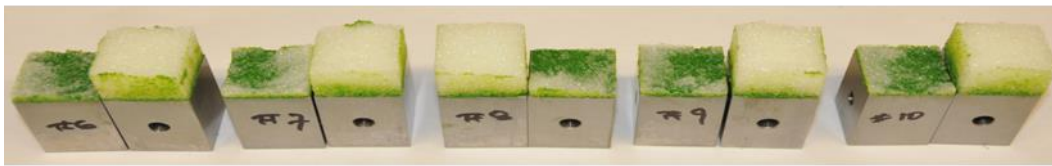
Figure 6.7. Failed cold temperature unreinforced foam FWT specimens.
(Not to scale. Image used with the permission of The Aerospace Corporation.)

Figure 6.8. Failed cold temperature Size 1 3DFRFC FWT specimens.
(Not to scale. Image used with the permission of The Aerospace Corporation.)
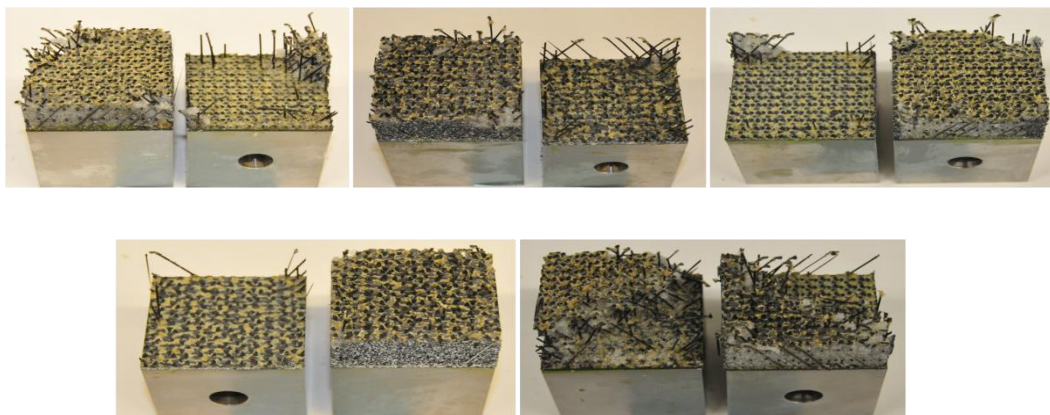
Figure 6.9. Failed cold temperature Size 2 3DFRFC FWT specimens.
(Not to scale. Image used with the permission of The Aerospace Corporation.)

6.5 Three-Point Bending

As with the tensile and compression data, all measured strength data was again normalized by the same value to allow for easier comparison with the other test configurations, Table 6.3. There were two key findings from the flexure tests. First, the measured shear strengths for both ambient and cold conditions were significantly lower than the values obtained in the pure tension or compression cases. Second, the flexure test showed a more significant strength reduction at cold temperature compared to either of the Size 2 through thickness tests: 23% for flexure versus 2% for compression and 4% for tension. The global external failure was consistent between the room temperature and cold temperature tests, discounting significant changes in failure mode as the reason for the observed strength reduction, Figures 6.10 and 6.11. The increase in temperature dependence for the flexure specimens may be the result of greater involvement of the foam in the failure process than in the pure tension or compression cases.

Table 6.3. Normalized shear strength for 3DFRFC; ambient Size 2 FWT=1.0.

Test Condition	# of samples	Average (Pa/Pa)	Standard Deviation (Pa/Pa)
Ambient	5	0.558	0.029
Cold	5	0.428	0.023
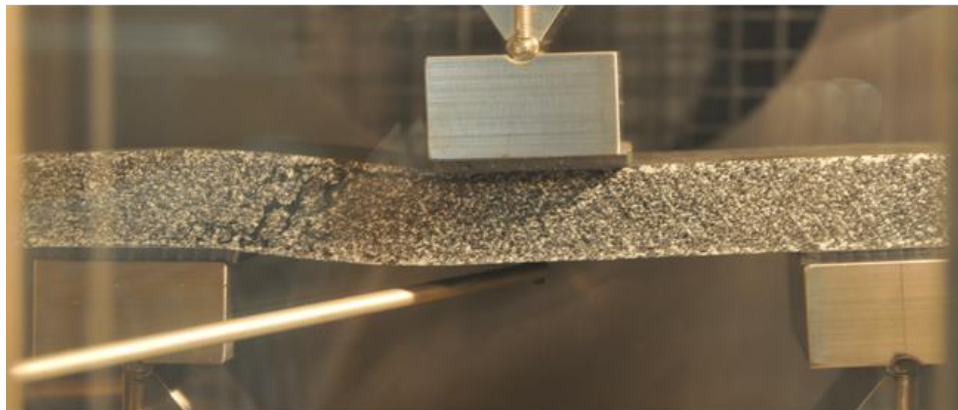
Figure 6.10. Failed three-point bending specimens under cold conditions.
(Not to scale. Image used with the permission of The Aerospace Corporation.)
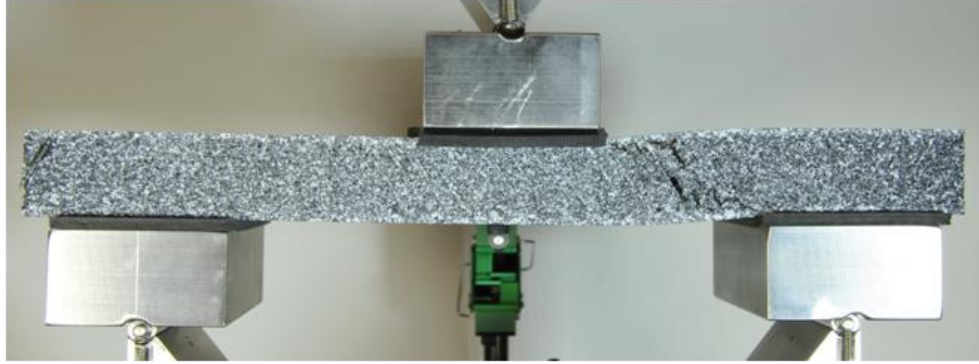
**Figure 6.11. Failed three-point bending specimens under ambient conditions.
(Not to scale. Image used with the permission of The Aerospace Corporation.)**

Embedded element modeling of the three-point bend configuration provided additional insight into the behavior of the specimens. First, the thermally induced stresses within the specimen were examined to determine the effect of the larger specimen size on the stress state within the foam and the truss. While the Size 2 flatwise specimens contained approximately 130 RUCs, the much larger three-point bending specimens contained over 700 RUCs.

Despite the larger size of the three-point bending specimens, the average residual stress within the foam was found to be consistent with the Size 1 and Size 2 flatwise specimens. A more significant increase of 30% was predicted in the local stresses within the truss members parallel to the long cut edge, Figure 6.12. Despite this relative increase, the peak stresses only represent approximately 10% of the available strength for the truss members. Since thermal loading accounted for a small percentage of the strength reduction, investigation into the mechanical loading behavior was needed.
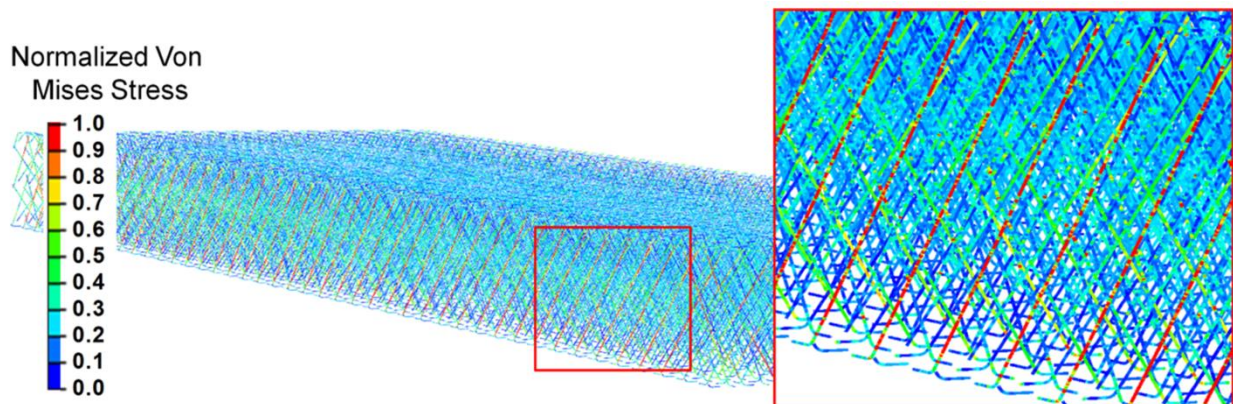
**Figure 6.12. Normalized Von Mises Stress in truss members due to cool down.
(Not to scale. Only core shown with foam removed.)**

112
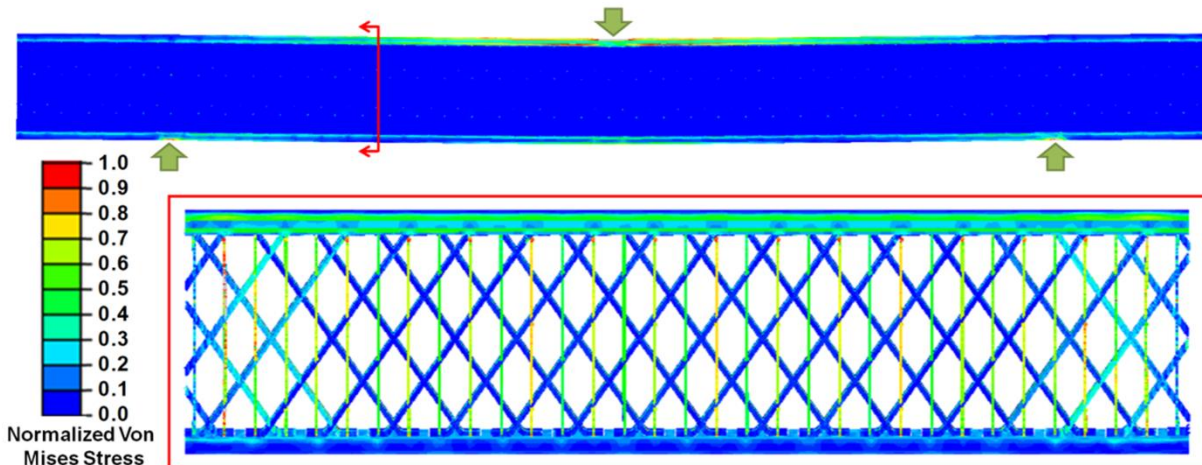
Figure 6.13. Normalized Von Mises Stress in cold flexure specimen at failure load.
(Not to scale. Section view shown with foam removed.)

Examination of the stress distribution through the cross-section of the specimen between the loading points highlights the key cause of the observed strength reduction for the three-point bending specimen. While the truss members within the pure tension and compression cases result in all of the fully bonded fibers being loaded in a similar manner, this is not the case with the bending specimen. There is relative inactivity of the reinforcement orthogonal to the span direction for the bending specimen, blue pins in Figure 6.13. The resultant effect is that the reinforcing pins along the span direction are relied on almost entirely for load transfer effectively reducing the number of reinforcing fibers available for load transfer by 50%. This explains the relative reduction in shear strength compared to the pure tension or compression cases.

It is interesting to note that the 23% strength reduction for the bending specimen at cold temperature falls within the trends seen for the two Size 1 through-thickness cases that had a lower percentage of fully bonded reinforcing fibers: 14% for compression and 38% for tension. While the larger three-point bending specimens actually have a higher percentage of fully bonded reinforcing fibers than the Size 2 specimens, the percentage of reinforcement available for load transfer is closer to that of the smaller Size 1 specimens. The reduced percentage of load bearing members within the 3DFRFC can allow for the highly temperature dependent foam properties to play a more significant role.

6.6 Summary

The development of thermo-mechanical modeling methods and results of experimental investigation into the failure of 3DFRFC sandwich composites at cold temperatures were presented. Key findings included:

- 3DFRFC specimens demonstrated considerable strength retention under cold conditions.
- 3DFRFC demonstrated a significant increase in through-thickness strength versus the similar density unreinforced core.
- The effect of increased strength with increased specimen size was confirmed by through-thickness tension and compression testing.
- Detailed finite element analysis highlighted the lower percentage of load bearing fibers as the primary cause for the reduced strength in the three-point-bend tests.

The investigation into the performance of 3DFRFC composite structures highlights the robust behavior of the structure to cold environments while underscoring the importance of loading direction on the structural response of these highly orthotropic composites. Future efforts will be focused on incorporating the detailed structural effects of the 3DFRFC microstructure into system level models, providing increased confidence in the design of structures with reinforced foam cores without requiring micromechanics-based detailed modeling.

CHAPTER 7

Conclusion

A brief summary is given highlighting the major points of the dissertation followed by some suggestions for related future work.

7.1 Summary

Chapter 2 discussed the details of the 3DFRFC microstructure through extensive use of microCT scans. The effort in this chapter formed the foundation for much of the modeling that took place in the remainder of the dissertation. The architecture measured directly from the microstructure was utilized to develop a parametric code for generating detailed embedded element models. These models were used for direct detailed modeling of test specimens in Chapters 3-6. The embedded element models were also used as the cornerstone of a new method of developing effective homogenized properties for 3DFRFCs based on the details of the microstructure. Part of this required the development of a generalized 6DoF periodic boundary condition code.

Chapter 3 went through the design, development, and initial failure of an interface fracture test for 3DFRFCs. The understanding gained by using Digital Image Correlation on the failed tests allowed for a different approach to be utilized in designing a new bonded double cantilever beam specimen for testing the Mode I fracture of a 3DFRFC sandwich structure. This method resulted in a successful interface fracture test. The bonded DCB specimens exhibited relatively smooth crack propagation and produced G_{Ic} values similar to honeycomb sandwich structures and significantly higher than comparable foam structures.

Chapter 4 detailed the predictive modeling capabilities of the methods presented in Chapter 2 applied to 3DFRFC sandwich structures with facesheet-to-core interface debonds. This included a full fabrication, testing, and evaluation of 3DFRFC specimens with differing sizes of defects. The analysis methods presented were able to predict the failure load and modes quite

well. The 3DFRFC proved to be tolerant to the presence of facesheet to core debonds with only the largest, 100 RUC debond demonstrating a significantly significant reduction of 22%.

Chapters 5 and 6 chronicled a detailed investigation of the through thickness behavior of a 3DFRFC composite under ambient and cold conditions. This included detailed microstructure modeling of the different loading configurations, modeling of thermal stresses, identification of failure modes and a thorough study of the effects of discrete specimen size and edge effects. MicroCT interrogation of tested specimens was then used to confirm the modes of failure in the tested specimens. The 3DFRFC specimens demonstrated better through thickness ambient performance than unreinforced cores of comparable density: >30% increase in tension, >100% increase in compression, and >5% increase in shear. The 3DFRFC's also demonstrated relatively small reductions in strength at cold temperatures: <2% reduction compression, <5% reduction tension, 23% reduction shear.

7.2 Future Work

7.2.1 Bending Periodic – Direct Shell Coupling

3DFRFCs are inherently structures and behave as such; however some of this behavior is lost when this structure is homogenized. In particular the resistance of a 3DFRFC structure to bending is often over predicted. This was demonstrated by the discrete model for the 3-point bending test where the transverse pins do not participate in transferring any load. In order to account for this effect it is suggested that the periodic modeling method presented in Chapter 2 be relaxed to allow for the introduction of global rotation at the boundaries. In this way it would be possible to directly derive an effective shell behavior that would account for the unique structural behavior of the 3DFRFC in a means that is much more conducive to modeling of large aerospace structures than modeling the discrete microstructure.

7.2.2 Development of Bulk and Edge Failure Envelope

It is worth investigating the ability to use a representative volumetric element approach coupled with progressive failure methodology to develop an effective material failure envelope. The failure envelope, analogous to a yield surface for 3DFRFCs, will aid in designing new aerospace structures. Part of the challenge is modeling a sufficiently large RVE that failure in

one part of the RVE is not directly interacting with itself. This is a hotly debated area, but in theory it should be possible to approach a "real world" bulk behavior; however, the size of the model required may still be computationally prohibitive.

7.2.3   Prediction of Component Level Failure

All work to this point has been at the coupon level. The next extension to this would be to directly predict failure of a representative aerospace (i.e. large) structure. This could be accomplished thorough the aforementioned failure envelope approach, concurrent multiscale modeling, or the global-local approach.

7.2.4   Optimization of 3DFRFC Structures

One of the most promising areas of untapped potential for 3DFRFCs is optimization. Unlike many other materials 3DFRFCs offer the potential not only for tailorability at the panel level, but at the local level. There is the potential to couple the analysis methods discussed in this dissertation at the design phase allowing for the 3DFRFC microstructure to be optimized based on the local stresses in the structure. This has potential to reduce mass, increase structural, or both.

APPENDICES

APPENDIX A

Parametric Script for Generating Reinforcing Geometry

A.1. Example

Included in this section is the full python script used for the development of all of the models given in the thesis. The code works by generating the reinforcing geometry beyond the desired size, radius the pin end (if desired) and then trimming the pins to the desired panel size. The code also offers some additional features not used in the thesis including the global panel rotation, Figure A.1, and manufacturing variability, Figure A.2. This code just generates the geometry. The user is free to mesh the output geometry using the preprocessor of their choice. An example model that was created using HyperMesh is given in Figure A.3. Note the variable "reveal" refers to the length of the foot of the pin, i.e. it is the part of the pin you would see on the surface of the foam.
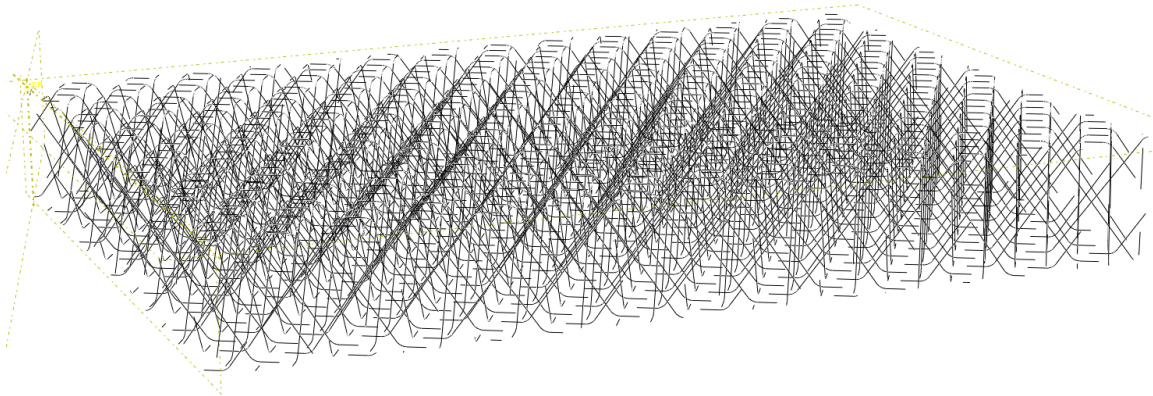
Figure A.1. Geometry output by parametric Abaqus script for the inputs as given below.
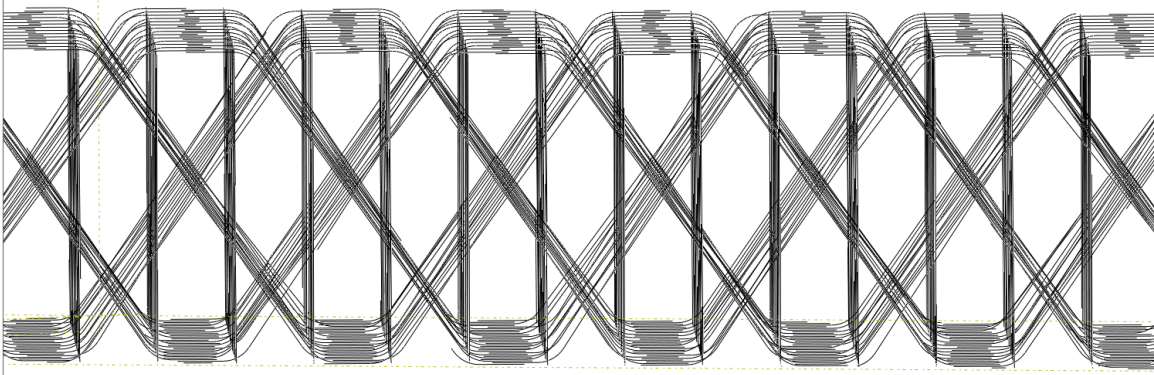
Figure A.2. Geometry output by parametric Abaqus script for the inputs as given below.
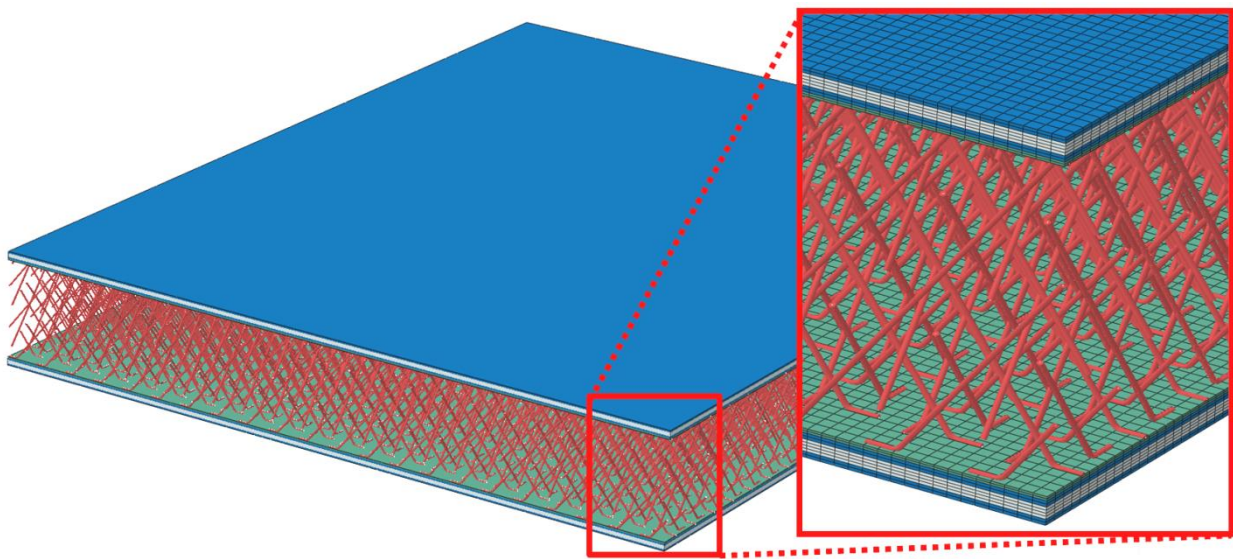
Figure A.3. 27 x 22 unit cell model of a 3DFRFC sandwich composite with foam removed. (Not to scale.)

A.2. Python Script for Abaqus CAE

The entirety of the text in below can be directly copied and pasted into a text file and saved with the .py extension to be run by Abaqus CAE. This input file was shown to work. as-is, in Abaqus 6.11-1.

```
########################################
# Reinforcing Fiber Geometry Generator  #
# Z. T. Kier, University of Michigan    #
# ------------------------------------ #
# Parametric scrip for generating flat  #
# panel 3DFRFC geometry incorporating   #
# end bonding geometry and randomness   #
# ------------------------------------ #
#   SEE GENERATOR README FOR DETAILS    #
########################################

######### load python modules ###########
import math
import random
import gc
gc.disable()

############## User Inputs ###############

####### Global In-plane Rotation ########
PanelAngle = 30

########## Panel Dimensions #############
## Restriction: Y >= X if Angle not = 0 #
Xmax = 3.0
Ymax = 4.0

####### Reinforcement Parameters #########
PinSpacing = 0.25
CoreThickness = 0.5
PinInclination = 35

########## Pin End Parameters ###########
Reveal = 0.10
RoundRadius = 0.05

########################################
######### Parameters in Beta ###########
########################################
## Can affect pin round operation #######
## Will work for smaller values of ######
## STDEV or set RoundRadius = 0 #########
########################################

########## Random Parameters ###########
InclinationSTDEV = 1.1
SpacingSTDEV = 0.0005
LateralSTDEV = 0.0005
random.seed(4)

########################################
# No User Inputs Found Below This Line  #
########################################

#start of abaqus script
from abaqus import *
from abaqusConstants import *
#calculated parameters
tanPHI = math.tan(math.radians(PinInclination))
ProjectedPinLength = CoreThickness * tanPHI
```

121

```
session.Viewport(name='Viewport: 1', origin=(0.0, 0.0), width=309.233825683594,
    height=259.291656494141)
session.viewports['Viewport: 1'].makeCurrent()
session.viewports['Viewport: 1'].maximize()
from caeModules import *
from driverUtils import executeOnCaeStartup
executeOnCaeStartup()
session.viewports['Viewport: 1'].partDisplay.geometryOptions.setValues(
    referenceRepresentation=ON)
session.viewports['Viewport: 1'].setValues(displayedObject=None)
p = mdb.models['Model-1'].Part(name='Part-1', dimensionality=THREE_D,
    type=DEFORMABLE_BODY)
p.ReferencePoint(point=(0.0, 0.0, 0.0))
p = mdb.models['Model-1'].parts['Part-1']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
p = mdb.models['Model-1'].parts['Part-1']
p.DatumPlaneByPrincipalPlane(principalPlane=XYPLANE, offset=0.0)
p = mdb.models['Model-1'].parts['Part-1']
p.DatumAxisByPrincipalAxis(principalAxis=XAXIS)
p = mdb.models['Model-1'].parts['Part-1']
p = mdb.models['Model-1'].parts['Part-1']
p.DatumPlaneByPrincipalPlane(principalPlane=XZPLANE, offset=0.0)
p.DatumAxisByPrincipalAxis(principalAxis=ZAXIS)
d = p.datums
p.DatumPlaneByRotation(plane=d[4], axis=d[5], angle=(-PanelAngle/2))

TotalPinSets = 0
PanelCOS = math.cos(math.radians(PanelAngle))
PanelSIN = math.sin(math.radians(PanelAngle))
PanelTAN = math.tan(math.radians(PanelAngle))
CutCOS = math.cos(math.radians(-PanelAngle))
CutSIN = math.sin(math.radians(-PanelAngle))
CutTAN = math.tan(math.radians(-PanelAngle))
YmaxCOS = (Ymax) * PanelCOS
YmaxSIN = (Ymax) * PanelSIN
XmaxCOS = (Xmax) * PanelCOS
XmaxSIN = (Xmax) * PanelSIN

YmaxTEMP = ProjectedPinLength + YmaxCOS + XmaxSIN
XmaxTEMP = ProjectedPinLength + XmaxCOS

YminTEMP = - ProjectedPinLength
XminTEMP = - ProjectedPinLength - YmaxSIN

Yoffset = YminTEMP
Xoffset = XminTEMP
Ylist = []
Xlist = []

while Yoffset <= YmaxTEMP:
        Ylist.append(Yoffset)
        Yoffset = Yoffset + PinSpacing
while Xoffset <= XmaxTEMP:
        Xlist.append(Xoffset)
        Xoffset = Xoffset + PinSpacing

delta_A1 = 0
delta_B1 = 0
delta_C1 = 0
delta_D1 = 0
delta_A2 = 0
delta_B2 = 0
delta_C2 = 0
delta_D2 = 0

for Yoffset in Ylist:

        for Xoffset in Xlist:

                if PanelAngle > 0:
```

```
                              #Eliminating pins outside of sample for rotated assembly, not needed for 0
deg
                              if YminTEMP <= Yoffset <= YmaxCOS and Xoffset < PanelTAN * (((-
ProjectedPinLength) / (PanelCOS * PanelTAN)) - Yoffset):
                                      continue
                              elif YmaxCOS < Yoffset <= YmaxTEMP and Xoffset < (Yoffset - ((Ymax +
ProjectedPinLength) / PanelCOS)) / PanelTAN:
                                      continue
                              elif YminTEMP <= Yoffset <= XmaxSIN and Xoffset > (Yoffset - ((-
ProjectedPinLength) / PanelCOS)) / PanelTAN:
                                      continue
                              elif XmaxSIN < Yoffset <= YmaxTEMP and Xoffset > PanelTAN *
(((ProjectedPinLength + Ymax) / (PanelCOS * PanelTAN)) - Yoffset):
                                      continue


                      #Generate Random Variation Spacing
                      if SpacingSTDEV != 0 or InclinationSTDEV != 0 or LateralSTDEV != 0:
                              delta_AS = random.gauss(0,SpacingSTDEV)
                              delta_BS = random.gauss(0,SpacingSTDEV)
                              delta_CS = random.gauss(0,SpacingSTDEV)
                              delta_DS = random.gauss(0,SpacingSTDEV)
                              delta_AA = ProjectedPinLength - (CoreThickness *
math.tan(math.radians(random.gauss(PinInclination,InclinationSTDEV))))
                              delta_BA = ProjectedPinLength - (CoreThickness *
math.tan(math.radians(random.gauss(PinInclination,InclinationSTDEV))))
                              delta_CA = ProjectedPinLength - (CoreThickness *
math.tan(math.radians(random.gauss(PinInclination,InclinationSTDEV))))
                              delta_DA = ProjectedPinLength - (CoreThickness *
math.tan(math.radians(random.gauss(PinInclination,InclinationSTDEV))))
                              delta_AL = random.gauss(0,LateralSTDEV)
                              delta_BL = random.gauss(0,LateralSTDEV)
                              delta_CL = random.gauss(0,LateralSTDEV)
                              delta_DL = random.gauss(0,LateralSTDEV)
                              delta_A1 = delta_AS - 0.5 * delta_AA
                              delta_B1 = delta_BS + 0.5 * delta_BA
                              delta_C1 = delta_CS + 0.5 * delta_CA
                              delta_D1 = delta_DS - 0.5 * delta_DA
                              delta_A2 = delta_AS + 0.5 * delta_AA
                              delta_B2 = delta_BS - 0.5 * delta_BA
                              delta_C2 = delta_CS - 0.5 * delta_CA
                              delta_D2 = delta_DS + 0.5 * delta_DA


                  if Reveal != 0:
                          AR1 = (0.0 + Xoffset + delta_A1 + Reveal, 0.0 + Yoffset + delta_AL, 0.0),
\
                                (0.0 + Xoffset + delta_A1, 0.0 + Yoffset + delta_AL, 0.0)
                  A = (0.0 + Xoffset + delta_A1, 0.0 + Yoffset + delta_AL, 0.0), \
                      (0.0 - ProjectedPinLength + Xoffset + delta_A2, 0.0 + Yoffset + delta_AL, -
CoreThickness)
                  if Reveal != 0:
                          AR2 = (0.0 - ProjectedPinLength + Xoffset + delta_A2, 0.0 + Yoffset +
delta_AL, - CoreThickness), \
                                (0.0 - ProjectedPinLength + Xoffset + delta_A2 - Reveal, 0.0 + Yoffset +
delta_AL, - CoreThickness)

                  if Reveal != 0:
                          BR1 = (0.0 + Xoffset + delta_BL, 0.0 + 0.5 * PinSpacing + Yoffset - Reveal
+ delta_B1, 0.0), \
                                (0.0 + Xoffset + delta_BL, 0.0 + 0.5 * PinSpacing + Yoffset + delta_B1,
0.0)
                  B = (0.0 + Xoffset + delta_BL, 0.0 + 0.5 * PinSpacing + Yoffset + delta_B1, 0.0),
\
                      (0.0 + Xoffset + delta_BL, 0.0 + 0.5 * PinSpacing + ProjectedPinLength + Yoffset +
delta_B2, - CoreThickness)
                  if Reveal != 0:
                          BR2 = (0.0 + Xoffset + delta_BL, 0.0 + 0.5 * PinSpacing +
ProjectedPinLength + Yoffset + delta_B2, - CoreThickness), \
                                (0.0 + Xoffset + delta_BL, 0.0 + 0.5 * PinSpacing + ProjectedPinLength +
Yoffset + Reveal + delta_B2, - CoreThickness)
```

```
                    if Reveal != 0:
                            CR1 = (0.0 + 0.5 * PinSpacing + Xoffset - Reveal + delta_C1, 0.0 + 0.5 *
PinSpacing + Yoffset + delta_CL, 0.0), \
                            (0.0 + 0.5 * PinSpacing + Xoffset + delta_C1, 0.0 + 0.5 * PinSpacing +
Yoffset + delta_CL, 0.0)
                    C = (0.0 + 0.5 * PinSpacing + Xoffset + delta_C1, 0.0 + 0.5 * PinSpacing + Yoffset
+ delta_CL, 0.0), \
                            (0.0 + 0.5 * PinSpacing + ProjectedPinLength + Xoffset + delta_C2, 0.0 + 0.5 *
PinSpacing + Yoffset + delta_CL, - CoreThickness)
                    if Reveal != 0:
                            CR2 = (0.0 + 0.5 * PinSpacing + ProjectedPinLength + Xoffset + delta_C2,
0.0 + 0.5 * PinSpacing + Yoffset + delta_CL, - CoreThickness), \
                            (0.0 + 0.5 * PinSpacing + ProjectedPinLength + Xoffset + Reveal +
delta_C2, 0.0 + 0.5 * PinSpacing + Yoffset + delta_CL, - CoreThickness)

                    if Reveal != 0:
                            DR1 = (0.0 + 0.5 * PinSpacing + Xoffset + delta_DL, 0.0 + Yoffset + Reveal
+ delta_D1, 0.0), \
                            (0.0 + 0.5 * PinSpacing + Xoffset + delta_DL, 0.0 + Yoffset + delta_D1,
0.0)
                    D = (0.0 + 0.5 * PinSpacing + Xoffset + delta_DL, 0.0 + Yoffset + delta_D1, 0.0),
\
                            (0.0 + 0.5 * PinSpacing + Xoffset + delta_DL, 0.0 - ProjectedPinLength + Yoffset +
delta_D2, - CoreThickness)
                    if Reveal != 0:
                            DR2 = (0.0 + 0.5 * PinSpacing + Xoffset + delta_DL, 0.0 -
ProjectedPinLength + Yoffset + delta_D2, - CoreThickness), \
                            (0.0 + 0.5 * PinSpacing + Xoffset + delta_DL, 0.0 - ProjectedPinLength +
Yoffset - Reveal + delta_D2, - CoreThickness)

                    if Reveal != 0:
                            p = mdb.models['Model-1'].parts['Part-1']

        p.WirePolyLine(points=((AR1),(A),(AR2),(BR1),(B),(BR2),(CR1),(C),(CR2),(DR1),(D),(DR2)),
mergeWire=ON, meshable=ON)
                    else:
                            p = mdb.models['Model-1'].parts['Part-1']
                            p.WirePolyLine(points=((A),(B),(C),(D)), mergeWire=ON, meshable=ON)

                    TotalPinSets = TotalPinSets + 1

#Pin Joint Rounding
if Reveal > 0:
        if RoundRadius > 0:
                p = mdb.models['Model-1'].parts['Part-1']
                v = p.vertices
                RoundV = []
                for i in range(TotalPinSets):
                        RoundV.append(v[16*i+1])
                        RoundV.append(v[16*i+2])
                        RoundV.append(v[16*i+5])
                        RoundV.append(v[16*i+6])
                        RoundV.append(v[16*i+9])
                        RoundV.append(v[16*i+10])
                        RoundV.append(v[16*i+13])
                        RoundV.append(v[16*i+14])
                p.Round(radius=RoundRadius, vertexList=RoundV)

#Trim Panel
p = mdb.models['Model-1'].parts['Part-1']
e = p.edges
edges = e.getSequenceFromMask(mask=('[#3 ]', ), )
p.Set(edges=edges, name='Wire-1-Set-1')
p = mdb.models['Model-1'].parts['Part-1']
d1 = p.datums
t = p.MakeSketchTransform(sketchPlane=d1[2], sketchUpEdge=d1[3],
    sketchPlaneSide=SIDE1, sketchOrientation=BOTTOM, origin=(0.0, 0.0, 0.0))
s = mdb.models['Model-1'].ConstrainedSketch(name='__profile__',
    sheetSize=10.98, gridSpacing=0.27, transform=t)
g, v, d, c = s.geometry, s.vertices, s.dimensions, s.constraints
```

```
s.setPrimaryObject(option=SUPERIMPOSE)
p = mdb.models['Model-1'].parts['Part-1']
p.projectReferencesOntoSketch(sketch=s, filter=COPLANAR_EDGES)
s.Line(point1=(0, 0), point2=(Xmax * CutCOS, -Xmax * CutSIN))
s.Line(point1=(Xmax * CutCOS, - Xmax * CutSIN), point2=((Xmax * CutCOS) + Ymax * CutSIN), (Ymax
* CutCOS) - (Xmax * CutSIN)))
s.Line(point1=((Xmax * CutCOS) + (Ymax * CutSIN), (Ymax * CutCOS) - (Xmax * CutSIN)),
point2=(Ymax * CutSIN , Ymax * CutCOS))
s.Line(point1=(Ymax * CutSIN , Ymax * CutCOS), point2=(0, 0))
s.rectangle(point1=(2 * (XminTEMP - ProjectedPinLength), 2 * (YminTEMP - ProjectedPinLength)),
point2=(2 * (XmaxTEMP + ProjectedPinLength), 2 * (YmaxTEMP + ProjectedPinLength)))
p = mdb.models['Model-1'].parts['Part-1']
d2 = p.datums
p.CutExtrude(sketchPlane=d2[2], sketchUpEdge=d2[3], sketchPlaneSide=SIDE1,
    sketchOrientation=BOTTOM, sketch=s, flipExtrudeDirection=OFF)
s.unsetPrimaryObject()
del mdb.models['Model-1'].sketches['__profile__']

#K-Cor Global Rotation
if PanelAngle > 0:
        p = mdb.models['Model-1'].parts['Part-1']
        d = p.datums
        p.Mirror(mirrorPlane=d[4], keepOriginal=OFF)
        p.Mirror(mirrorPlane=d[6], keepOriginal=OFF)

session.viewports['Viewport: 1'].setValues(displayedObject=p)
session.viewports['Viewport: 1'].view.setProjection(projection=PARALLEL)
session.viewports['Viewport: 1'].view.setValues(session.views['Front'])
session.viewports['Viewport: 1'].view.fitView()
```

APPENDIX B

Matlab Script for Generating Periodic Boundary Conditions for Abaqus Input with 3 or 6 DoF Nodes

B.1. Overview

Automated generation of 6 degree-of-freedom periodic boundary conditions (PBCs) is conducted using a script written in Matlab. This script relies on 4 primary components:

1) GPBC6DOF.m, primary Matlab code for generating PBCs
2) BoundedVoronoiArea.m, Matlab function called by GPBC6DOF used for calculating area of influence for nodes belonging to non-continuum elements (beams, shells, etc.)
3) NodalAreaInfluence.m, Matlab function called by GPBC6DOF used for calculating are of influence for nodes belonging to linear and serendipity three dimensional continuum elements
   a. C3D4, C3D6, C3D8, C3D8R, & C3D8I
   b. C3D15 C3D10, & C3D20
4) User supplied Abaqus input file with named node sets for the surfaces to which the boundary conditions are to be applied (not all sets need be present):
   - setx03dof – 3 degree-of-freedom nodes on the -x plane
   - setx13dof – 3 degree-of-freedom nodes on the +x plane
   - sety03dof – 3 degree-of-freedom nodes on the -y plane
   - sety13dof – 3 degree-of-freedom nodes on the +y plane
   - setz03dof – 3 degree-of-freedom nodes on the -z plane
   - setz13dof – 3 degree-of-freedom nodes on the +z plane
   - setx06dof – 6 degree-of-freedom nodes on the -x plane
   - setx16dof – 6 degree-of-freedom nodes on the +x plane
   - sety06dof – 6 degree-of-freedom nodes on the -y plane
   - sety16dof – 6 degree-of-freedom nodes on the +y plane
   - setz06dof – 6 degree-of-freedom nodes on the -z plane
   - setz16dof – 6 degree-of-freedom nodes on the +z plane

The code works by reading the Abaqus input file to obtain the nodal and elemental information from the model and writing a separate equation file that can be incorporated into the original abaqus input file through the use of the *include command. By default the equation file

126

that is saved has the same name as the original Abaqus file with the addition of "_EQN" added to the end of the filename.

B.2. BoundedVoronoiArea.m MATLAB Function

The entirety of the text in below can be directly copied and pasted into a text file and saved with the file name BoundedVoronoiArea.m to be run by the Matlab code GPBC6DOF.m. This function was shown to work. as-is, in Matlab R2010b.

```
function [Area] = BoundedVoronoiArea(Xlist,Ylist,BVAtitle,PlotYN)
X=[Xlist,Ylist];
[V,C]=voronoin(X);
xmin=min(Xlist);
xmax=max(Xlist);
ymin=min(Ylist);
ymax=max(Ylist);
BoundingBox=[xmin,ymin;xmin,ymax;xmax,ymax;xmax,ymin];
A=[];
cmin=[0,0];
if PlotYN==1
    f1=figure('units','normalized');
    title(['Nodal Area of Influence for ',BVAtitle],'FontSize',10,'FontUnits','normalized')
    colormap(hot)
    hcb=colorbar;
    set(get(hcb,'Title'),'String',{'Nodal';'Influence';'(%/Unit^2)'},...
        'FontSize',7,'FontUnits','normalized','FontWeight','light',...
        'HorizontalAlignment','left','VerticalAlignment','baseline',...
        'Units','normalized','Position',[0,1.09,0]);
    set(hcb,'Position',[0.915,0.11,0.03,0.74]);
end
[vx, vy]=voronoi(X(:,1),X(:,2));
plotV=0;
% h1=subplot(1,2,1);
% h2=subplot(1,2,2);
for j=1:length(X);
    Vtemp=[];
    Vcut=[];
    C0=[];
    Cp1=[];
    Cn1=[];
    k=[];
    xInf=[];
    yinf=[];
    C0=C{j};
    Cp1=circshift(C0,[0,1]);
    Cn1=circshift(C0,[0,-1]);
    k=find(any(V(C0,:)==Inf,2));%k coresponds to row index with Inf
    if isempty(k)
        [Vtemp(:,1),Vtemp(:,2)]=poly2cw(V(C0(:),1),V(C0(:),2));
    else %infinite vertex at edge
        %determine location of point
        if X(j,1)==xmin && X(j,2)==ymin;xInf=xmin-(1000*abs(xmax-xmin));yInf=ymin-(1000*abs(ymax-ymin));
        elseif X(j,1)==xmin && X(j,2)==ymax;xInf=xmin-(1000*abs(xmax-xmin));yInf=ymax+(1000*abs(ymax-ymin));
        elseif X(j,1)==xmax && X(j,2)==ymin;xInf=xmax+(1000*abs(xmax-xmin));yInf=ymin-(1000*abs(ymax-ymin));
        elseif X(j,1)==xmax && X(j,2)==ymax;xInf=xmax+(1000*abs(xmax-xmin));yInf=ymax+(1000*abs(ymax-ymin));
        elseif X(j,1)==xmin;xInf=xmin-(1000*abs(xmax-xmin));yInf=X(j,2);
        elseif X(j,1)==xmax;xInf=xmax+(1000*abs(xmax-xmin));yInf=X(j,2);
```

127

```
          elseif X(j,2)==ymin;xInf=X(j,1);yInf=ymin-(1000*abs(ymax-ymin));
          elseif X(j,2)==ymax;xInf=X(j,1);yInf=ymax+(1000*abs(ymax-ymin));
          end
          Vtemp=[V(C0(:),1),V(C0(:),2)];
          Vtemp(k,:)=[xInf,yInf];
          [Vtemp(:,1),Vtemp(:,2)]=poly2cw(Vtemp(:,1),Vtemp(:,2));
          Vtempnew=[];
          Vtempn1=circshift(Vtemp,[-1,0]);
          Vtempp1=circshift(Vtemp,[1,0]);
          for i=1:length(Vtemp);
              if Vtemp(i,:)==[xInf,yInf];
                  if X(j,1)==xmin && X(j,2)==ymin;
Vtempnew=[Vtempnew;Vtempn1(i,1),yInf;xInf,yInf;xInf,Vtempp1(i,2)];
                  elseif X(j,1)==xmin && X(j,2)==ymax;
Vtempnew=[Vtempnew;xInf,Vtempp1(i,2);xInf,yInf;Vtempn1(i,1),yInf];
                  elseif X(j,1)==xmax && X(j,2)==ymin;
Vtempnew=[Vtempnew;xInf,Vtempp1(i,2);xInf,yInf;Vtempn1(i,1),yInf];
                  elseif X(j,1)==xmax && X(j,2)==ymax;
Vtempnew=[Vtempnew;Vtempp1(i,1),yInf;xInf,yInf;xInf,Vtempn1(i,2)];
                  elseif X(j,1)==xmin;Vtempnew=[Vtempnew;xInf,Vtempp1(i,2);xInf,Vtempn1(i,2)];
                  elseif X(j,1)==xmax;Vtempnew=[Vtempnew;xInf,Vtempp1(i,2);xInf,Vtempn1(i,2)];
                  elseif X(j,2)==ymin;Vtempnew=[Vtempnew;Vtempp1(i,1),yInf;Vtempn1(i,1),yInf];
                  elseif X(j,2)==ymax;Vtempnew=[Vtempnew;Vtempp1(i,1),yInf;Vtempn1(i,1),yInf];
                  end
              else Vtempnew=[Vtempnew;Vtemp(i,:)];
              end
          end
          Vtemp=[Vtempnew(:,1),Vtempnew(:,2)];
      end
      if ~isempty(Vtemp);

[Vcut(:,1),Vcut(:,2)]=polybool('intersection',BoundingBox(:,1),BoundingBox(:,2),Vtemp(:,1),Vtemp(
:,2));
          %           Subplot(h2)
          A=[A;polyarea(Vcut(:,1),Vcut(:,2))];
          if PlotYN==1
              patch(Vcut(:,1),Vcut(:,2),A(end)*100*ones(1,length(Vcut)));
          end
          [warnmsg, msgid] = lastwarn;
          warning('off','map:vectorsToGPC:noExternalContours');
          if strcmp(msgid,'map:vectorsToGPC:noExternalContours')
              disp('POLYBOOL Warning detected!');
              j
              Vtemp
              Vcut
              A(end)
              warning('');
              plotV=1;
          end
      else
          A=[A;0.01];
      end
end
SC=0.01;
% subplot(h1)
% hold on;scatter(X(:,1),X(:,2),20,'filled');xlim([xmin-SC*(xmax-xmin) xmax+SC*(xmax-
xmin)]);ylim([ymin-SC*(ymax-ymin) ymax+SC*(ymax-ymin)]);
% subplot(h2)
if PlotYN==1
    hold on;
    % scatter(X(:,1),X(:,2),AC*A);
    scp=scatter(X(:,1),X(:,2),21,'.k');
    xlim([xmin-SC*(xmax-xmin) xmax+SC*(xmax-xmin)]);ylim([ymin-SC*(ymax-ymin) ymax+SC*(ymax-
ymin)]);
    % set(scp,'Position',[0,0,1,1]);
    if plotV==1;plot(Xlist,Ylist,'r+',vx,vy,'b-');end;
    cmax=125*max(A,[],1);
    caxis([cmin(1) cmax(1)])
end
Area=A;
```

B.3. NodalAreaInfluence.m MATLAB Function

The entirety of the text in below can be directly copied and pasted into a text file and saved with the file name NodalAreaInfluence.m to be run by the Matlab code GPBC6DOF.m. This function was shown to work. as-is, in Matlab R2010b.

```
function [Area] = NodalAreaInfluence(nset,ELEMENTS,BVAtitle,PlotYN)
% nset, list of node numbers with coordinates belonging to surface of interest
% [N#, x1, y1, z1 ; N#, x2, y2, z2 ; ...]
% ELEMENTS cell array of the form: ELEMENTS={eC3D4;eC3D6;eC3D8;eC3D10;eC3D15;eC3D20};
% each element of the form [El#, node 1, node 2,..., node N; El#, node 1, node 2, ..., node N]
% Area output in the form [A1:A2] where A1 corresponds to area for a corner
% node and A2 corresponds to area for midpoint node. I.e. if all linear
% elements then all A2=0. this allows for easy decoupling of corner and
% mid-point nodes.
tic
eC3D4=ELEMENTS{1};
eC3D6=ELEMENTS{2};
eC3D8=ELEMENTS{3};
eC3D10=ELEMENTS{4};
eC3D15=ELEMENTS{5};
eC3D20=ELEMENTS{6};

xmin=min(nset(:,2));
xmax=max(nset(:,2));
ymin=min(nset(:,3));
ymax=max(nset(:,3));
zmin=min(nset(:,4));
zmax=max(nset(:,4));

cmin=[0,0];

[~,D] = min([abs(xmax-xmin),abs(ymax-ymin),abs(zmax-zmin)]);
% D corresponds with plane normal direction [1,2,3] [x,y,z]

if D==1 %(n#,y,z)
    planarnset=[nset(:,1),nset(:,3),nset(:,4)];
elseif D==2 %(n#,z,x)
    planarnset=[nset(:,1),nset(:,4),nset(:,2)];
elseif D==3 %(n#,x,y)
    planarnset=[nset(:,1),nset(:,2),nset(:,3)];
end

A=zeros(length(nset),2);
if PlotYN
    f1=figure('units','normalized');
    title(['Nodal Area of Influence for Corner Nodes on
',BVAtitle],'FontSize',10,'FontUnits','normalized')
    colormap(hot)
    hcb=colorbar;
    set(get(hcb,'Title'),'String',{'Nodal';'Influence';'(%/Unit^2)'},...
        'FontSize',7,'FontUnits','normalized','FontWeight','light',...
        'HorizontalAlignment','left','VerticalAlignment','baseline',...
        'Units','normalized','Position',[0,1.09,0]);
    set(hcb,'Position',[0.915,0.11,0.03,0.74]);
    set(f1,'Position',[0.005,0.045,0.49,.875]);
    SC=0.01;
    xmin=min(planarnset(:,2));
    xmax=max(planarnset(:,2));
    ymin=min(planarnset(:,3));
    ymax=max(planarnset(:,3));
    xlim([xmin-SC*(xmax-xmin) xmax+SC*(xmax-xmin)]);ylim([ymin-SC*(ymax-ymin) ymax+SC*(ymax-
ymin)]);
end
```

```
cornernodes=[];

if PlotYN && any([~isempty(eC3D10),~isempty(eC3D15),~isempty(eC3D20)])
    f2=figure('units','normalized');
    title(['Nodal Area of Influence for Midpoint Nodes on
',BVAtitle],'FontSize',10,'FontUnits','normalized')
    colormap(hot)
    hcb2=colorbar;
    set(get(hcb2,'Title'),'String',{'Nodal';'Influence';'(%/Unit^2)'},...
        'FontSize',7,'FontUnits','normalized','FontWeight','light',...
        'HorizontalAlignment','left','VerticalAlignment','baseline',...
        'Units','normalized','Position',[0,1.09,0]);
    set(hcb2,'Position',[0.915,0.11,0.03,0.74]);
    set(f2,'Position',[0.505,0.045,0.49,.875]);
%       hold on
%       scatter(planarnset(:,2),planarnset(:,3),23,'.k');
%       SC=0.01;
    xmin=min(planarnset(:,2));
    xmax=max(planarnset(:,2));
    ymin=min(planarnset(:,3));
    ymax=max(planarnset(:,3));
    xlim([xmin-SC*(xmax-xmin) xmax+SC*(xmax-xmin)]);ylim([ymin-SC*(ymax-ymin) ymax+SC*(ymax-
ymin)]);
    midpointnodes=[];
end

ntemp=0;
midpointTF=0;
Vels={};

for j=1:length(planarnset);
    Vprint=[];
    Vall=[];
    ntemp=planarnset(j,1);
    ncoordtemp=planarnset(j,2:3);
    for elset={'eC3D4','eC3D6','eC3D8','eC3D10','eC3D15','eC3D20'}

HigherOrderTF=any([strncmpi(elset{1},'eC3D10',10),strncmpi(elset{1},'eC3D15',10),strncmpi(elset{1
},'eC3D20',10)]);%Higer order element TF==1
        if eval(['~isempty(',elset{1},')'])
            r=[];c=[];
            %[r,c]=find(elset{1}(:,2:end)==ntemp);
            [r,c]=eval(['find(',elset{1},'(:,2:end)==ntemp);']);
            c=c+1;
            if any(r)
                for k=1:length(r) %iterates throught each element that contains node
                    if HigherOrderTF
                        midpointTF =
any([strncmpi(elset{1},'eC3D10',10)&&c(k)>5,strncmpi(elset{1},'eC3D15',10)&&c(k)>7,strncmpi(elset
{1},'eC3D20',10)&&c(k)>9]);%midpoint node TF==1
                    else midpointTF=0;
                    end
                    etemp=eval([elset{1},'(r(k),:);']);
                    V=[];
                    Vnew=[];
                    Vp1=[];
                    Vn1=[];
                    Centroid=[];
                    Atemp=0;
                    %Only use corner nodes
                    if any([strncmpi(elset{1},'eC3D4',10),strncmpi(elset{1},'eC3D10',10)])
                        mmax=4;
                    elseif any([strncmpi(elset{1},'eC3D6',10),strncmpi(elset{1},'eC3D15',10)])
                        mmax=6;
                    elseif any([strncmpi(elset{1},'eC3D8',10),strncmpi(elset{1},'eC3D20',10)])
                        mmax=8;
                    end
                    %                       Vtemp=zeros(mmax,2);
                    Vtemp=[];
                    for m=2:mmax+1 %first term is element number
```

130

```
                [V,~]=find(planarnset(:,1)==etemp(m));
                if ~isempty(V)
                    Vtemp=[Vtemp;planarnset(V,2:3)];
                end
            end
            if ~isempty(Vtemp)
                % determin if collinear
                if length(Vtemp)==3
                    mat = [Vtemp(1,1)-Vtemp(3,1), Vtemp(1,2)-Vtemp(3,2); Vtemp(2,1)-
Vtemp(3,1), Vtemp(2,2)-Vtemp(3,2)];
                    tf = det(mat) == 0;
                elseif length(Vtemp)<3
                    tf=1;
                else tf=0;
                end
                if tf==0;
                    [Ktemp,~]=convhull(Vtemp(:,1),Vtemp(:,2));
                    Vnew=zeros(size(Vtemp));
                    for n=1:length(Ktemp)-1
                        Vnew(n,:)=Vtemp(Ktemp(n),:);
                    end
                    %Vels={Vels{:},Vnew};
                    Vels=[Vels(:)' {Vnew}];
                    Centroid=sum(Vtemp,1)/size(Vtemp,1);
                    if midpointTF==1 %midpoint node
                        Vp1=circshift(Vnew,[1,0]);

[~,r2]=min(polyarea([Vnew(:,1),ncoordtemp(1)*ones(size(Vnew,1),1),Vp1(:,1)],[Vnew(:,2),ncoordtemp
(2)*ones(size(Vnew,1),1),Vp1(:,2)],2));
                        Vtemp=[Vp1(r2,:);ncoordtemp;Vnew(r2,:);Centroid];
                    else %corner node
                        Vp1=circshift(Vnew,[1,0]);
                        Vn1=circshift(Vnew,[-1,0]);
                        r2=[];
                        for m=1:size(Vnew,1)
                            if Vnew(m,:)==ncoordtemp
                                r2=m;
                            end
                        end

Vtemp=[mean([Vn1(r2,:);Vnew(r2,:)],1);Vnew(r2,:);mean([Vnew(r2,:);Vp1(r2,:)],1);Centroid];
                    end
                    if isempty(Vall)
                        [Vall(:,1),Vall(:,2)]=poly2cw(Vtemp(:,1),Vtemp(:,2));
                    else
                        Vbtemp=[];
                        VbX=[];
                        VbY=[];
                        [Vbtemp(:,1),Vbtemp(:,2)]=poly2cw(Vtemp(:,1),Vtemp(:,2));

[VbX,VbY]=polybool('union',Vbtemp(:,1),Vbtemp(:,2),Vall(:,1),Vall(:,2));
                        Vall=[VbX,VbY];
                    end
                end
            end
        end
        if ~isempty(Vall)
            Atemp=polyarea(Vall(:,1),Vall(:,2));
            if isequal(elset{1},'eC3D10')&&c(k)>5 %midpoint node (of element))
                A(j,2)=A(j,2)+Atemp;
            elseif isequal(elset{1},'eC3D15')&&c(k)>7
                A(j,2)=A(j,2)+Atemp;
            elseif isequal(elset{1},'eC3D20')&&c(k)>9
                A(j,2)=A(j,2)+Atemp;
            else %corner node (of element)
                A(j,1)=A(j,1)+Atemp;
            end
        end
    end
end
end
```

```
    Atemp=0;
    Ktemp=[];
    Vprint=[];
    if ~isempty(Vall) && PlotYN
        Vprint=Vall;
        cmax=125*max(A,[],1);
        if A(j,1)~=0 %corner node weighting
            figure(f1)
            patch(Vprint(:,1),Vprint(:,2),A(j,1)*100*ones(size(Vprint,1),1),'linestyle','--
','edgecolor',[0 0.5 1]);
            caxis([cmin(1) cmax(1)])
            cornernodes=[cornernodes;planarnset(j,2),planarnset(j,3)];
        elseif A(j,2)~=0 %midpoint node weighting
            figure(f2)
            patch(Vprint(:,1),Vprint(:,2),A(j,2)*100*ones(size(Vprint,1),1),'linestyle','--
','edgecolor',[0 0.5 1]);
            caxis([cmin(2) cmax(2)])
            midpointnodes=[midpointnodes;planarnset(j,2),planarnset(j,3)];
        end
    end
end
if PlotYN
    SC=0.01;
    xmin=min(planarnset(:,2));
    xmax=max(planarnset(:,2));
    ymin=min(planarnset(:,3));
    ymax=max(planarnset(:,3));
    cmax=125*max(A,[],1);
    figure(f1)
    xlim([xmin-SC*(xmax-xmin) xmax+SC*(xmax-xmin)]);ylim([ymin-SC*(ymax-ymin) ymax+SC*(ymax-
ymin)]);
    caxis([cmin(1) cmax(1)])
    for j=1:size(Vels,2)

patch(Vels{j}(:,1),Vels{j}(:,2),ones(size(Vels{j},1),1)','facecolor','none','LineWidth',2);
    end
    if any([~isempty(eC3D10),~isempty(eC3D15),~isempty(eC3D20)])
        figure(f2)
        xlim([xmin-SC*(xmax-xmin) xmax+SC*(xmax-xmin)]);ylim([ymin-SC*(ymax-ymin) ymax+SC*(ymax-
ymin)]);
        caxis([cmin(2) cmax(2)])
        for j=1:size(Vels,2)

patch(Vels{j}(:,1),Vels{j}(:,2),ones(size(Vels{j},1),1)','facecolor','none','LineWidth',2);
        end
    end
    figure(f1)
    hold on
    scatter(cornernodes(:,1),cornernodes(:,2),60,'ob');
    if any([~isempty(eC3D10),~isempty(eC3D15),~isempty(eC3D20)])
        figure(f1)
        scatter(midpointnodes(:,1),midpointnodes(:,2),50,'*k');
        figure(f2)
        hold on
        scatter(cornernodes(:,1),cornernodes(:,2),60,'ob');
        figure(f2)
        scatter(midpointnodes(:,1),midpointnodes(:,2),50,'*k')
    end
end
Area=A;
% p = patch(xdata,ydata,cdata,'Marker','o','MarkerFaceColor','flat','FaceColor','none')
end
```

B.4. PBC6DOF Matlab Code

The entirety of the text in below can be directly copied and pasted into a text file and saved with the file extension .m. This code is the primary code for creating the generalized 6-degree-of-freedom boundry conditions and was shown to work. as-is, in Matlab R2010b.

```
%Zachary T. Kier
%University of Michigan

%This function creates Equation contraints on the sides of an FEM mesh in
%order to apply periodic BCs on 3DoF and 6DoF nodes in any or all of the
%Cartesian directions (x,y,z)

%Note code currently assumes the input file is ordered as follows:
%Nodes (1 block of data)
%...
%Elements (can be multiple types and sets)
%...
%Node sets (can contain additional node sets)

% function[]=Periodic_BCs_FEM()
format compact
clear
close all
%% User variables
%Plot surface maping for strain-averaged/non periodic surfaces? PlotNPBC=1
%will plot
PlotNPBC=0;

%tolerance for matching nodes, should to be less than element size
tol=1e-6;

%Periodicity direction boolean [X,Y,Z],
%i.e [1,1,0] periodic in X and Y, but not Z
PBD=[1,1,0];

%6dof elements embedded elements? if so = 1. Will ignore translational dof
%on 6dof sets
embedded=1;

%non-periodic boundrys 6dof elements clampped? if so each dof = 1.
%Only applies to non-periodic boundry conditions [4,5,6], 6 = drilling mode
clampped=[0,0,0];

%non-periodic boundrys free? if so = 1.
%if = 0, average displacements between non-periodic surfaces linked to
% reference points, i.e. control global displacement/strain

% Note average displacement caluclated by nodal points. 3dof nodes are
% weighted by area of influence on surface. For higher order elements the
% midpoint and corner nodes are decoupled (1 equation for corners another
% for midpoints). Non-periodic 6-dof nodes are weighted by Voronoi Cell
% area. (non-periodic 6dof nodes are not affected if they are embedded)
% Freeedge direction boolean [X,Y,Z], ignored for direction if PBD = 1
% i.e [1,1,0] free in X and Y, but not Z (z strain controlled)

freeedges=[0,0,0];

%If there are less than 2 free edges (sum freeedges <2) and at least 1
%non-free direction is not periodic (i.e. average strain controlled) then
%at least 2x2 elements are required on each strain controlled surface).
%Otherwise abaqus will give a DOF eliminated error.

%No User defined inputs found below this line
```

```matlab
[meshinp, pathname, filterindex] = uigetfile('*.inp','Select Input File to create
PBCs','*.inp','MultiSelect','off');

%% Initialize
clc;
tic
tStart=tic;
tPause=0;
error_flag=0;
warning_flag=0;

%Display user inputs
fprintf('USER INPUTS:\n');
fprintf('PBD [X,Y,Z] = [%i,%i,%i]\n',PBD);
fprintf('6 DOF elements embedded: ');
if embedded==1; fprintf('yes\n'); else fprintf('no\n'); end
fprintf('Clampped DOF Boolean [4,5,6] = [%i,%i,%i]\n',clampped);
fprintf('Free DOF Boolean [X,Y,Z] = [');
if PBD(1)==1; fprintf('NA,'); else fprintf('%i,',freeedges(1)); end
if PBD(2)==1; fprintf('NA,'); else fprintf('%i,',freeedges(2)); end
if PBD(3)==1; fprintf('NA'); else fprintf('%i',freeedges(3)); end
fprintf(']\n');
fprintf('Plot non-periodic boundary surfaces: ');
if PlotNPBC==1; fprintf('yes\n'); else fprintf('no\n'); end

%Open mesh
finp=fopen([pathname meshinp],'r');
fprintf('Reading ABAQUS Input File: %s...\n',meshinp);
%% Get nodal information
%first line of input file
line=fgetl(finp);

%Iterate through lines until it finds begining of node section
while ~strncmpi(line,'*node',5)%case insensitive
    line=fgetl(finp);
end

fprintf('Gathering information about nodal coordinates...\n');
fprintf([line,'\n']);
NC=textscan(finp,'%f,%f,%f,%f','CollectOutput',true);
NODES=NC{1};
%% Element Information
element_position=ftell(finp);
line=fgetl(finp);
%does not include 6dof, voronoi used for 6dof
%3DofElements
NeC3D4=zeros(1,'uint32');
NeC3D6=zeros(1,'uint32');
NeC3D8=zeros(1,'uint32');
NeC3D10=zeros(1,'uint32');
NeC3D15=zeros(1,'uint32');
NeC3D20=zeros(1,'uint32');
fprintf('Gathering information about elements...\n');

while ~feof(finp)
    %Iterate through lines to find element information
    if strncmpi(line,'*nset',5) %case insensitive
        break %skips to next section which handles the nsets
    elseif strncmpi(line,'*element',8) %case insensitive
        fprintf([line,'\n']);
        etemp=[];
        if any(regexpi(line, 'C3D4')) %case insensitive
            etemp=textscan(finp,'%u','delimiter', ',');
            NeC3D4=NeC3D4+(size(etemp{1},1)/5);
        elseif any(regexpi(line, 'C3D6')) %case insensitive
            etemp=textscan(finp,'%u','delimiter', ',');
            NeC3D6=NeC3D6+(size(etemp{1},1)/7);
        elseif any(regexpi(line, 'C3D8')) %case insensitive
            etemp=textscan(finp,'%u','delimiter', ',');
            NeC3D8=NeC3D8+(size(etemp{1},1)/9);
        elseif any(regexpi(line, 'C3D10')) %case insensitive
```

134

```
            etemp=textscan(finp,'%u','delimiter', ',');
            NeC3D10=NeC3D10+(size(etemp{1},1)/11);
        elseif any(regexpi(line, 'C3D15')) %case insensitive
            etemp=textscan(finp,'%u','delimiter', ',');
            NeC3D15=NeC3D15+(size(etemp{1},1)/16);
        elseif any(regexpi(line, 'C3D20')) %case insensitive
            etemp=textscan(finp,'%u','delimiter', ',');
            NeC3D20=NeC3D20+(size(etemp{1},1)/21);
        else
        end
        line=fgetl(finp);
    else line=fgetl(finp);
    end
end

%initialize element variables
eC3D4=zeros(NeC3D4,5,'uint32');
eC3D6=zeros(NeC3D6,7,'uint32');
eC3D8=zeros(NeC3D8,9,'uint32');
eC3D10=zeros(NeC3D10,11,'uint32');
eC3D15=zeros(NeC3D15,16,'uint32');
eC3D20=zeros(NeC3D20,21,'uint32');

%initialize element counters
CeC3D4=zeros(1,'uint32');
CeC3D6=zeros(1,'uint32');
CeC3D8=zeros(1,'uint32');
CeC3D10=zeros(1,'uint32');
CeC3D15=zeros(1,'uint32');
CeC3D20=zeros(1,'uint32');

%move back to line after nodes
fseek(finp, element_position, 'bof');
line=fgetl(finp);

while ~feof(finp)
    %Iterate through lines to find element information
    if strncmpi(line,'*nset',5) %case insensitive
        break %skips to next section which handles the nsets
    elseif strncmpi(line,'*element',8) %case insensitive
        etemp=[];
        if any(regexpi(line, 'C3D4')) %case insensitive
            etemp=textscan(finp,'%u','delimiter', ',');
            for j=1:(size(etemp{1},1)/5)
                eC3D4(CeC3D4+j,:)=etemp{1}(5*j-4:5*j);
            end
            CeC3D4=CeC3D4+(size(etemp{1},1)/5);
        elseif any(regexpi(line, 'C3D6')) %case insensitive
            etemp=textscan(finp,'%u','delimiter', ',');
            for j=1:(size(etemp{1},1)/7)
                eC3D6(CeC3D6+j,:)=etemp{1}(7*j-6:7*j);
            end
            CeC3D6=CeC3D6+(size(etemp{1},1)/7);
        elseif any(regexpi(line, 'C3D8')) %case insensitive
            etemp=textscan(finp,'%u','delimiter', ',');
            for j=1:(size(etemp{1},1)/9)
                eC3D8(CeC3D8+j,:)=etemp{1}(9*j-8:9*j);
            end
            CeC3D8=CeC3D8+(size(etemp{1},1)/9);
        elseif any(regexpi(line, 'C3D10')) %case insensitive
            etemp=textscan(finp,'%u','delimiter', ',');
            for j=1:(size(etemp{1},1)/11)
                eC3D10(CeC3D10+j,:)=etemp{1}(11*j-10:11*j);
            end
            CeC3D10=CeC3D10+(size(etemp{1},1)/11);
        elseif any(regexpi(line, 'C3D15')) %case insensitive
            etemp=textscan(finp,'%u','delimiter', ',');
            for j=1:(size(etemp{1},1)/16)
                eC3D15(CeC3D15+j,:)=etemp{1}(16*j-15:16*j);
            end
            CeC3D15=CeC3D15+(size(etemp{1},1)/16);
```

135

```
        elseif any(regexpi(line, 'C3D20')) %case insensitive
            etemp=textscan(finp,'%u','delimiter', ',');
            for j=1:(size(etemp{1},1)/21)
                eC3D20(CeC3D20+j,:)=etemp{1}(21*j-20:21*j);
            end
            CeC3D20=CeC3D20+(size(etemp{1},1)/21);
        else
        end
        line=fgetl(finp);
    else line=fgetl(finp);
    end
end
ELEMENTS={eC3D4;eC3D6;eC3D8;eC3D10;eC3D15;eC3D20};
%% Create node set information
%3DofNodes
setx03dof=[];
setx13dof=[];
sety03dof=[];
sety13dof=[];
setz03dof=[];
setz13dof=[];

%6DofNodes
setx06dof=[];
setx16dof=[];
sety06dof=[];
sety16dof=[];
setz06dof=[];
setz16dof=[];
fprintf('Gathering information about node sets...\n');
while ~feof(finp)
    %Iterate through lines to find node set
    while ~feof(finp)
        if strncmpi(line,'*nset',5) %case insensitive
            break
        else line=fgetl(finp);
        end
    end

    if feof(finp)
        break %break while loop if at end of file
    end

    if strncmpi(line,'*nset',5) %case insensitive
        fprintf([line,'\n']);
        %3Dof Node Sets
        if strcmpi(line,'*NSET, NSET=setx03dof') %case insensitive
            set=setx03dof;
            setstr='setx03dof';
        elseif strcmpi(line,'*NSET, NSET=setx13dof') %case insensitive
            set=setx13dof;
            setstr='setx13dof';
        elseif strcmpi(line,'*NSET, NSET=sety03dof') %case insensitive
            set=sety03dof;
            setstr='sety03dof';
        elseif strcmpi(line,'*NSET, NSET=sety13dof') %case insensitive
            set=sety13dof;
            setstr='sety13dof';
        elseif strcmpi(line,'*NSET, NSET=setz03dof') %case insensitive
            set=setz03dof;
            setstr='setz03dof';
        elseif strcmpi(line,'*NSET, NSET=setz13dof') %case insensitive
            set=setz13dof;
            setstr='setz13dof';

            %6Dof Node Sets
        elseif strcmpi(line,'*NSET, NSET=setx06dof') %case insensitive
            set=setx06dof;
            setstr='setx06dof';
        elseif strcmpi(line,'*NSET, NSET=setx16dof') %case insensitive
            set=setx16dof;
```

```
            setstr='setx16dof';
        elseif strcmpi(line,'*NSET, NSET=sety06dof') %case insensitive
            set=sety06dof;
            setstr='sety06dof';
        elseif strcmpi(line,'*NSET, NSET=sety16dof') %case insensitive
            set=sety16dof;
            setstr='sety16dof';
        elseif strcmpi(line,'*NSET, NSET=setz06dof') %case insensitive
            set=setz06dof;
            setstr='setz06dof';
        elseif strcmpi(line,'*NSET, NSET=setz16dof') %case insensitive
            set=setz16dof;
            setstr='setz16dof';
        else %Set not Found
            %             fprintf('Warning! Set not used for PCBs:  ');
            %             fprintf('%s;\n', line);
            %             warning_flag=warning_flag+1;
            line=fgetl(finp); %Next line, i.e. ignore this set
            continue %go to start of next while iteration
        end
    end

    gen=isempty(strfind(line,'generate'));

    if gen % i.e. nodes are NOT being generated
        if ~feof(finp)
            set=textscan(finp,'%u','delimiter', ',');
            str2=sprintf('%s=set{1};',setstr);
            eval(str2);
        end
    else line=fgetl(finp);
        if ~feof(finp)
        data=str2num(line);
        set=[data(1):data(3):data(2)];
        str2=sprintf('%s=set;',setstr);
        eval(str2);
        end
    end
    line=fgetl(finp);
end

%Close mesh
fclose(finp);
fprintf('Reading of ABAQUS Input File Completed in %6.4g seconds.\n',toc);
tic

if PBD(1) && PBD(2) && PBD(3);
    fprintf('Periodic Boundry Conditions applied in the X, Y, & Z directions.\n\n');
else fprintf('Periodic Boundry Conditions ');
    if sum(PBD)>0; fprintf('only applied in the ');
        if PBD(1) && PBD(2) && ~PBD(3); fprintf('X & Y directions.\n');end
        if PBD(1) && ~PBD(2) && PBD(3); fprintf('X & Z directions.\n');end
        if ~PBD(1) && PBD(2) && PBD(3); fprintf('Y & Z directions.\n');end
        if PBD(1) && ~PBD(2) && ~PBD(3); fprintf('X direction.\n');end
        if ~PBD(1) && PBD(2) && ~PBD(3); fprintf('Y direction.\n');end
        if ~PBD(1) && ~PBD(2) && PBD(3); fprintf('Z direction.\n');end
    else fprintf('not applied in any direction\n');
    end
end
%% Create tie sets to apply equation constraint

%[DUM,I]=sort(NSET(:,1));
%NSET=NSET(I,:);

NSET_OLD=NODES;
NODES=zeros(max(NODES(:,1)),length(NODES(1,:)));
for i=1:length(NSET_OLD(:,1))
    n=NSET_OLD(i,1);
    NODES(n,:)=NSET_OLD(i,:);
end
```

```matlab
%Display Node Sets Found in Input
fprintf('**Node Sets Found in Input File: \n');
fprintf('\t\t\t3 DOF\t\t\t\t\t6 DOF\n');
if ~isempty(setx03dof);fprintf('\tsetx03dof\t');else fprintf('\t\t\t\t');end
if ~isempty(setx13dof);fprintf('setx13dof\t\t');else fprintf('\t\t\t\t');end
if ~isempty(setx06dof);fprintf('setx06dof\t');else fprintf('\t\t\t');end
if ~isempty(setx16dof);fprintf('setx16dof\t');else fprintf('\t\t\t');end
fprintf('\n');
if ~isempty(sety03dof);fprintf('\tsety03dof\t');else fprintf('\t\t\t\t');end
if ~isempty(sety13dof);fprintf('sety13dof\t\t');else fprintf('\t\t\t\t');end
if ~isempty(sety06dof);fprintf('sety06dof\t');else fprintf('\t\t\t');end
if ~isempty(sety16dof);fprintf('sety16dof\t');else fprintf('\t\t\t');end
fprintf('\n');
if ~isempty(setz03dof);fprintf('\tsetz03dof\t');else fprintf('\t\t\t\t');end
if ~isempty(setz13dof);fprintf('setz13dof\t\t');else fprintf('\t\t\t\t');end
if ~isempty(setz06dof);fprintf('setz06dof\t');else fprintf('\t\t\t');end
if ~isempty(setz16dof);fprintf('setz16dof\t');else fprintf('\t\t\t');end
fprintf('\n');

%Display Node Sets Missing from Input
if isempty(setx03dof)||isempty(setx13dof)||isempty(setx06dof)||...
        isempty(setx16dof)||isempty(sety03dof)||isempty(sety13dof)||...
        isempty(sety06dof)||isempty(sety16dof)||isempty(setz03dof)||...
        isempty(setz13dof)||isempty(setz06dof)||isempty(setz16dof)
    fprintf('Node Sets Missing From Input File: \n');
    if isempty(setx03dof)||isempty(setx13dof)||isempty(setx06dof)||isempty(setx16dof)
        if isempty(setx03dof);fprintf('setx03dof\t');else fprintf('\t\t\t');end
        if isempty(setx13dof);fprintf('setx13dof\t');else fprintf('\t\t\t');end
        if isempty(setx06dof);fprintf('setx06dof\t');else fprintf('\t\t\t');end
        if isempty(setx16dof);fprintf('setx16dof\t');else fprintf('\t\t\t');end
        fprintf('\n');
    end
    if isempty(sety03dof)||isempty(sety13dof)||isempty(sety06dof)||isempty(sety16dof)
        if isempty(sety03dof);fprintf('sety03dof\t');else fprintf('\t\t\t');end
        if isempty(sety13dof);fprintf('sety13dof\t');else fprintf('\t\t\t');end
        if isempty(sety06dof);fprintf('sety06dof\t');else fprintf('\t\t\t');end
        if isempty(sety16dof);fprintf('sety16dof\t');else fprintf('\t\t\t');end
        fprintf('\n');
    end
    if isempty(setz03dof)||isempty(setz13dof)||isempty(setz06dof)||isempty(setz16dof)
        if isempty(setz03dof);fprintf('setz03dof\t');else fprintf('\t\t\t');end
        if isempty(setz13dof);fprintf('setz13dof\t');else fprintf('\t\t\t');end
        if isempty(setz06dof);fprintf('setz06dof\t');else fprintf('\t\t\t');end
        if isempty(setz16dof);fprintf('setz16dof\t');else fprintf('\t\t\t');end
        fprintf('\n');
    end
end

if PBD(1) && length(setx03dof)~=length(setx13dof)
    fprintf('ERROR! x0 and x1 do not have the same number of 3dof nodes!\n');
    error_flag=error_flag+1;
end
if PBD(2) && length(sety03dof)~=length(sety13dof)
    fprintf('ERROR! y0 and y1 do not have the same number of 3dof nodes!\n');
    error_flag=error_flag+1;
end
if PBD(3) && length(setz03dof)~=length(setz13dof)
    fprintf('ERROR! z0 and z1 do not have the same number of 3dof nodes!\n');
    error_flag=error_flag+1;
end
if PBD(1) && length(setx06dof)~=length(setx16dof)
    fprintf('ERROR! x0 and x1 do not have the same number of 6dof nodes!\n');
    error_flag=error_flag+1;
end
if PBD(2) && length(sety06dof)~=length(sety16dof)
    fprintf('ERROR! y0 and y1 do not have the same number of 6dof nodes!\n');
    error_flag=error_flag+1;
end
if PBD(3) && length(setz06dof)~=length(setz16dof)
    fprintf('ERROR! z0 and z1 do not have the same number of 6dof nodes!\n');
    error_flag=error_flag+1;
```

```
end
%% write sets with coordinates
coordx03dof = zeros(length(setx03dof),size(NODES,2));
coordx13dof = zeros(length(setx13dof),size(NODES,2));
coordy03dof = zeros(length(sety03dof),size(NODES,2));
coordy13dof = zeros(length(sety13dof),size(NODES,2));
coordz03dof = zeros(length(setz03dof),size(NODES,2));
coordz13dof = zeros(length(setz13dof),size(NODES,2));
coordx06dof = zeros(length(setx06dof),size(NODES,2));
coordx16dof = zeros(length(setx16dof),size(NODES,2));
coordy06dof = zeros(length(sety06dof),size(NODES,2));
coordy16dof = zeros(length(sety16dof),size(NODES,2));
coordz06dof = zeros(length(setz06dof),size(NODES,2));
coordz16dof = zeros(length(setz16dof),size(NODES,2));

for i=1:length(setx03dof)
    coordx03dof(i,:)=NODES(setx03dof(i),:);
end
for i=1:length(setx13dof)
    coordx13dof(i,:)=NODES(setx13dof(i),:);
end
for i=1:length(sety03dof)
    coordy03dof(i,:)=NODES(sety03dof(i),:);
end
for i=1:length(sety13dof)
    coordy13dof(i,:)=NODES(sety13dof(i),:);
end
for i=1:length(setz03dof)
    coordz03dof(i,:)=NODES(setz03dof(i),:);
end
for i=1:length(setz13dof)
    coordz13dof(i,:)=NODES(setz13dof(i),:);
end
for i=1:length(setx06dof)
    coordx06dof(i,:)=NODES(setx06dof(i),:);
end
for i=1:length(setx16dof)
    coordx16dof(i,:)=NODES(setx16dof(i),:);
end
for i=1:length(sety06dof)
    coordy06dof(i,:)=NODES(sety06dof(i),:);
end
for i=1:length(sety16dof)
    coordy16dof(i,:)=NODES(sety16dof(i),:);
end
for i=1:length(setz06dof)
    coordz06dof(i,:)=NODES(setz06dof(i),:);
end
for i=1:length(setz16dof)
    coordz16dof(i,:)=NODES(setz16dof(i),:);
end
%% determine planar coordinates
x0 = (tol/2)*(round((2/tol)*(mean([coordx03dof(:,2);coordx06dof(:,2)]))));
x1 = (tol/2)*(round((2/tol)*(mean([coordx13dof(:,2);coordx16dof(:,2)]))));
y0 = (tol/2)*(round((2/tol)*(mean([coordy03dof(:,3);coordy06dof(:,3)]))));
y1 = (tol/2)*(round((2/tol)*(mean([coordy13dof(:,3);coordy16dof(:,3)]))));
z0 = (tol/2)*(round((2/tol)*(mean([coordz03dof(:,4);coordz06dof(:,4)]))));
z1 = (tol/2)*(round((2/tol)*(mean([coordz13dof(:,4);coordz16dof(:,4)]))));
xmean = x1-x0;
ymean = y1-y0;
zmean = z1-z0;
%% find corner, edge, and surface nodes%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf('Finding corner, edge, and surface nodes...');tic

%initialize all variables

%3dof corners
x0y0z03dof=[];
x0y0z13dof=[];
x0y1z03dof=[];
```

139

```
x0y1z13dof=[];
x1y0z03dof=[];
x1y0z13dof=[];
x1y1z03dof=[];
x1y1z13dof=[];

%3dof edges
x0y03dof=[];
x0y13dof=[];
x0z03dof=[];
x0z13dof=[];
x1y03dof=[];
x1y13dof=[];
x1z03dof=[];
x1z13dof=[];
y0z03dof=[];
y0z13dof=[];
y1z03dof=[];
y1z13dof=[];

%3dof surfaces
x03dof=[];
x13dof=[];
y03dof=[];
y13dof=[];
z03dof=[];
z13dof=[];

%6dof corners
x0y0z06dof=[];
x0y0z16dof=[];
x0y1z06dof=[];
x0y1z16dof=[];
x1y0z06dof=[];
x1y0z16dof=[];
x1y1z06dof=[];
x1y1z16dof=[];

%6dof edges
x0y06dof=[];
x0y16dof=[];
x0z06dof=[];
x0z16dof=[];
x1y06dof=[];
x1y16dof=[];
x1z06dof=[];
x1z16dof=[];
y0z06dof=[];
y0z16dof=[];
y1z06dof=[];
y1z16dof=[];

%6dof surfaces
x06dof=[];
x16dof=[];
y06dof=[];
y16dof=[];
z06dof=[];
z16dof=[];

coordtemp=zeros(1,size(NODES,2));
%%% Note sets to be grouped into a list with each line:
%%% (Node number, X coordinate, Y coordinate, Z coordinate)
%%3dof%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Organize 3dof nodes on corner, edges, and surface of X0
for i = 1 : size(coordx03dof,1)
    coordtemp=coordx03dof(i,:);
    % Find corner nodes on X0
    if ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        x0y0z03dof=coordtemp;
    elseif ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(4)-z1) <= tol))
```

140

```
        x0y0z13dof=coordtemp;
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        x0y1z03dof=coordtemp;
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        x0y1z13dof=coordtemp;
        % Find edge nodes on X0
    elseif ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        x0y03dof=[x0y03dof;coordtemp];
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        x0y13dof=[x0y13dof;coordtemp];
    elseif ((abs(coordtemp(4)-z0) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        x0z03dof=[x0z03dof;coordtemp];
    elseif ((abs(coordtemp(4)-z1) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        x0z13dof=[x0z13dof;coordtemp];
        % Surface nodes on X0
    else x03dof=[x03dof;coordtemp];
    end
end
% Organize 3dof nodes on corner, edges, and surface of X1
for i = 1 : size(coordx13dof,1)
    coordtemp=coordx13dof(i,:);
    % Find corner nodes on X1
    if ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        x1y0z03dof=coordtemp;
    elseif ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        x1y0z13dof=coordtemp;
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        x1y1z03dof=coordtemp;
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        x1y1z13dof=coordtemp;
        % Find edge nodes on X1
    elseif ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        x1y03dof=[x1y03dof;coordtemp];
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        x1y13dof=[x1y13dof;coordtemp];
    elseif ((abs(coordtemp(4)-z0) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        x1z03dof=[x1z03dof;coordtemp];
    elseif ((abs(coordtemp(4)-z1) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        x1z13dof=[x1z13dof;coordtemp];
        % Surface nodes on X1
    else x13dof=[x13dof;coordtemp];
    end
end
% Organize 3dof nodes on corner, edges, and surface of Y0
for i = 1 : size(coordy03dof,1)
    coordtemp=coordy03dof(i,:);
    % Find corner nodes on Y0: already found on X0 & X1
    if ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        dummy=1;
        % Find edge nodes on Y0
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        dummy=1; %Found on X0
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        dummy=1; %Found on X1
```

141

```
    elseif ((abs(coordtemp(4)-z0) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        y0z03dof=[y0z03dof;coordtemp];
    elseif ((abs(coordtemp(4)-z1) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        y0z13dof=[y0z13dof;coordtemp];
        % Surface nodes on Y0
    else y03dof=[y03dof;coordtemp];
    end
end
% Organize 3dof nodes on corner, edges, and surface of Y1
for i = 1 : size(coordy13dof,1)
    coordtemp=coordy13dof(i,:);
    % Find corner nodes on Y1: already found on X0 & X1
    if ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        dummy=1;
        % Find edge nodes on Y1
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        dummy=1; %Found on X0
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        dummy=1; %Found on X1
    elseif ((abs(coordtemp(4)-z0) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        y1z03dof=[y1z03dof;coordtemp];
    elseif ((abs(coordtemp(4)-z1) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        y1z13dof=[y1z13dof;coordtemp];
        % Surface nodes on Y1
    else y13dof=[y13dof;coordtemp];
    end
end
% Organize 3dof nodes on corner, edges, and surface of Z0
for i = 1 : size(coordz03dof,1)
    coordtemp=coordz03dof(i,:);
    % Find corner nodes on Z0: already found on X0 & X1
    if ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(3)-y0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(3)-y1) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(3)-y0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(3)-y1) <= tol))
        dummy=1;
        % Find edge nodes on Z0
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        dummy=1; %Found on X0
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        dummy=1; %Found on X1
    elseif ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        dummy=1; %Found on Y0
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        dummy=1; %Found on Y1
        % Surface nodes on Z0
    else z03dof=[z03dof;coordtemp];
    end
end
% Organize 3dof nodes on corner, edges, and surface of Z1
for i = 1 : size(coordz13dof,1)
    coordtemp=coordz13dof(i,:);
```

142

```
    % Find corner nodes on Z1: already found on X0 & X1
    if ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(3)-y0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(3)-y1) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(3)-y0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(3)-y1) <= tol))
        dummy=1;
        % Find edge nodes on Z1
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        dummy=1; %Found on X0
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        dummy=1; %Found on X1
    elseif ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        dummy=1; %Found on Y0
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        dummy=1; %Found on Y1
        % Surface nodes on Z1
    else z13dof=[z13dof;coordtemp];
    end
end
%% 6dof%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Organize 6dof nodes on corner, edges, and surface of X0
for i = 1 : size(coordx06dof,1)
    coordtemp=coordx06dof(i,:);
    % Find corner nodes on X0
    if ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        x0y0z06dof=coordtemp;
    elseif ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        x0y0z16dof=coordtemp;
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        x0y1z06dof=coordtemp;
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        x0y1z16dof=coordtemp;
        % Find edge nodes on X0
    elseif ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        x0y06dof=[x0y06dof;coordtemp];
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        x0y16dof=[x0y16dof;coordtemp];
    elseif ((abs(coordtemp(4)-z0) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        x0z06dof=[x0z06dof;coordtemp];
    elseif ((abs(coordtemp(4)-z1) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        x0z16dof=[x0z16dof;coordtemp];
        % Surface nodes on X0
    else x06dof=[x06dof;coordtemp];
    end
end
% Organize 6dof nodes on corner, edges, and surface of X1
for i = 1 : size(coordx16dof,1)
    coordtemp=coordx16dof(i,:);
    % Find corner nodes on X1
    if ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        x1y0z06dof=coordtemp;
    elseif ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        x1y0z16dof=coordtemp;
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        x1y1z06dof=coordtemp;
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        x1y1z16dof=coordtemp;
        % Find edge nodes on X1
    elseif ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
```

```
        x1y06dof=[x1y06dof;coordtemp];
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        x1y16dof=[x1y16dof;coordtemp];
    elseif ((abs(coordtemp(4)-z0) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        x1z06dof=[x1z06dof;coordtemp];
    elseif ((abs(coordtemp(4)-z1) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        x1z16dof=[x1z16dof;coordtemp];
        % Surface nodes on X1
    else x16dof=[x16dof;coordtemp];
    end
end
% Organize 6dof nodes on corner, edges, and surface of Y0
for i = 1 : size(coordy06dof,1)
    coordtemp=coordy06dof(i,:);
    % Find corner nodes on Y0: already found on X0 & X1
    if ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        dummy=1;
        % Find edge nodes on Y0
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        dummy=1; %Found on X0
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        dummy=1; %Found on X1
    elseif ((abs(coordtemp(4)-z0) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        y0z06dof=[y0z06dof;coordtemp];
    elseif ((abs(coordtemp(4)-z1) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        y0z16dof=[y0z16dof;coordtemp];
        % Surface nodes on Y0
    else y06dof=[y06dof;coordtemp];
    end
end
% Organize 6dof nodes on corner, edges, and surface of Y1
for i = 1 : size(coordy16dof,1)
    coordtemp=coordy16dof(i,:);
    % Find corner nodes on Y1: already found on X0 & X1
    if ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(4)-z0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(4)-z1) <= tol))
        dummy=1;
        % Find edge nodes on Y1
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        dummy=1; %Found on X0
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(4)-z0) > tol) && (abs(coordtemp(4)-
z1) > tol))
        dummy=1; %Found on X1
    elseif ((abs(coordtemp(4)-z0) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        y1z06dof=[y1z06dof;coordtemp];
    elseif ((abs(coordtemp(4)-z1) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        y1z16dof=[y1z16dof;coordtemp];
        % Surface nodes on Y1
    else y16dof=[y16dof;coordtemp];
    end
```

```
end
% Organize 6dof nodes on corner, edges, and surface of Z0
for i = 1 : size(coordz06dof,1)
    coordtemp=coordz06dof(i,:);
    % Find corner nodes on Z0: already found on X0 & X1
    if ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(3)-y0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(3)-y1) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(3)-y0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(3)-y1) <= tol))
        dummy=1;
        % Find edge nodes on Z0
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        dummy=1; %Found on X0
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        dummy=1; %Found on X1
    elseif ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        dummy=1; %Found on Y0
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        dummy=1; %Found on Y1
        % Surface nodes on Z0
    else z06dof=[z06dof;coordtemp];
    end
end
% Organize 6dof nodes on corner, edges, and surface of Z1
for i = 1 : size(coordz16dof,1)
    coordtemp=coordz16dof(i,:);
    % Find corner nodes on Z1: already found on X0 & X1
    if ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(3)-y0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(3)-y1) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(3)-y0) <= tol))
        dummy=1;
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(3)-y1) <= tol))
        dummy=1;
        % Find edge nodes on Z1
    elseif ((abs(coordtemp(2)-x0) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        dummy=1; %Found on X0
    elseif ((abs(coordtemp(2)-x1) <= tol) && (abs(coordtemp(3)-y0) > tol) && (abs(coordtemp(3)-
y1) > tol))
        dummy=1; %Found on X1
    elseif ((abs(coordtemp(3)-y0) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        dummy=1; %Found on Y0
    elseif ((abs(coordtemp(3)-y1) <= tol) && (abs(coordtemp(2)-x0) > tol) && (abs(coordtemp(2)-
x1) > tol))
        dummy=1; %Found on Y1
        % Surface nodes on Z1
    else z16dof=[z16dof;coordtemp];
    end
end
fprintf(' completed in %6.4g seconds.\n',toc);
%% Find matching node pairs
% all X1,Y1,Z1 will be sorted to match X0,Y0,Z0 respectively
fprintf('Finding matching node pairs...');tic
coordtempi=zeros(length(x13dof(:,1)),size(NODES,2));
coorddif=zeros(length(x13dof(:,1)),size(NODES,2));
% sort X13dof to match X03dof
if PBD(1)&&~isempty(x03dof)
    x13dof_sorted=zeros(length(x13dof(:,1)),length(x13dof(1,:)));
    for i = 1 : size(x03dof,1)
        coordtempi=ones(length(x13dof(:,1)),1)*x03dof(i,:);
        coorddif=abs(coordtempi-x13dof);
```

```matlab
            %iterate through Y(3) and Z(4) of X13dof to find matching node
            x13dof_sorted(i,:)=x13dof(all((coorddif(:,[3,4])<= tol),2),:);
            if x13dof_sorted(i,1)==0
                fprintf('ERROR! Matching node for node %i not found!\n',x03dof(i,1));
                error_flag=error_flag+1;
            end
        end
end
% sort Y13dof to match Y03dof
if PBD(2)&&~isempty(y03dof)
    y13dof_sorted=zeros(length(y13dof(:,1)),length(y13dof(1,:)));
    for i = 1 : size(y03dof,1)
        coordtempi=ones(length(y13dof(:,1)),1)*y03dof(i,:);
        coorddif=abs(coordtempi-y13dof);
        %iterate through X(2) and Z(4) of Y13dof to find matching node
        y13dof_sorted(i,:)=y13dof(all((coorddif(:,[2,4])<= tol),2),:);
        if y13dof_sorted(i,1)==0
            fprintf('ERROR! Matching node for node %i not found!\n',x03dof(i,1));
            error_flag=error_flag+1;
        end
    end
end
% sort Z13dof to match Z03dof
if PBD(3)&&~isempty(z03dof)
    z13dof_sorted=zeros(length(z13dof(:,1)),length(z13dof(1,:)));
    for i = 1 : size(z03dof,1)
        coordtempi=ones(length(z13dof(:,1)),1)*z03dof(i,:);
        coorddif=abs(coordtempi-z13dof);
        %iterate through X(2) and Y(3) of Z13dof to find matching node
        z13dof_sorted(i,:)=z13dof(all((coorddif(:,[2,3])<= tol),2),:);
        if z13dof_sorted(i,1)==0
            fprintf('ERROR! Matching node for node %i not found!\n',z03dof(i,1));
            error_flag=error_flag+1;
        end
    end
end
% sort X16dof to match X06dof
if PBD(1)&&~isempty(x06dof)
    x16dof_sorted=zeros(length(x16dof(:,1)),length(x16dof(1,:)));
    for i = 1 : size(x06dof,1)
        coordtempi=ones(length(x16dof(:,1)),1)*x06dof(i,:);
        coorddif=abs(coordtempi-x16dof);
        %iterate through Y(3) and Z(4) of X13dof to find matching node
        x16dof_sorted(i,:)=x16dof(all((coorddif(:,[3,4])<= tol),2),:);
        if x16dof_sorted(i,1)==0
            fprintf('ERROR! Matching node for node %i not found!\n',x06dof(i,1));
            error_flag=error_flag+1;
        end
    end
end
% sort Y16dof to match Y06dof
if PBD(2)&&~isempty(y06dof)
    y16dof_sorted=zeros(length(y16dof(:,1)),length(y16dof(1,:)));
    for i = 1 : size(y06dof,1)
        coordtempi=ones(length(y16dof(:,1)),1)*y06dof(i,:);
        coorddif=abs(coordtempi-y16dof);
        %iterate through X(2) and Z(4) of Y13dof to find matching node
        y16dof_sorted(i,:)=y16dof(all((coorddif(:,[2,4])<= tol),2),:);
        if y16dof_sorted(i,1)==0
            fprintf('ERROR! Matching node for node %i not found!\n',y06dof(i,1));
            error_flag=error_flag+1;
        end
    end
end
% sort Z16dof to match Z06dof
if PBD(3)&&~isempty(z06dof)
    z16dof_sorted=zeros(length(z16dof(:,1)),length(z16dof(1,:)));
    for i = 1 : size(z06dof,1)
        coordtempi=ones(length(z16dof(:,1)),1)*z06dof(i,:);
        coorddif=abs(coordtempi-z16dof);
        %iterate through X(2) and Y(3) of Z13dof to find matching node
```

```
            z16dof_sorted(i,:)=z16dof(all((coorddif(:,[2,3])<= tol),2),:);
            if z16dof_sorted(i,1)==0
                fprintf('ERROR! Matching node for node %i not found!\n',z06dof(i,1));
                error_flag=error_flag+1;
            end
        end
    end
end
%% Break for errors
if error_flag~=0;fprintf('\n');end
if warning_flag>1 || warning_flag==0
    fprintf('File processed with %i warnings',warning_flag);
elseif warning_flag==1
    fprintf('File processed with %i warning',warning_flag);
end
if error_flag>1 || error_flag==0
    fprintf(' and %i errors',error_flag);
elseif error_flag==1
    fprintf(' and %i error',error_flag);
end
if error_flag==0
    fprintf('.\n');
else fprintf('!\n');
end
%Stop code if errors are present
if error_flag~=0;fprintf('Creation of boundry conditions failed!\n');return;end
fprintf(' completed in %6.4g seconds.\n',toc);
%% Open file to write equation constraints in
mesh_PBC=sprintf('%s_EQN.inp',meshinp(1:end-4));
fprintf('Opening ABAQUS Equation File: %s...\n',mesh_PBC);
tic
if exist([pathname mesh_PBC], 'file')
    fprintf('Equation file already exists. \n');
    button = questdlg('Equation File Already Exists. Overwrite?','','Yes','No','No');
%     waitfor(button);
    switch button
        case 'Yes'
            fprintf('User choose to overwrite...\n');
        case 'No'
            fprintf('User choose not to overwrite... program terminated\n');
            return
    end
    tPause=toc;
    tic
end
fpbc=fopen([pathname mesh_PBC],'w');
%Write user inputs
fprintf(fpbc,'**USER INPUTS:\n');
fprintf(fpbc,'**ABAQUS Input File: %s\n',meshinp);
fprintf(fpbc,'**PBD [X,Y,Z] = [%i,%i,%i]\n',PBD);
fprintf(fpbc,'**6 DOF elements embedded: ');
if embedded==1; fprintf(fpbc,'yes\n'); else fprintf(fpbc,'no\n'); end
fprintf(fpbc,'**Clampped DOF Boolean [4,5,6] = [%i,%i,%i]\n',clampped);
fprintf(fpbc,'**Free DOF Boolean [X,Y,Z] = [');
if PBD(1)==1; fprintf(fpbc,'NA,'); else fprintf(fpbc,'%i,',freeedges(1)); end
if PBD(2)==1; fprintf(fpbc,'NA,'); else fprintf(fpbc,'%i,',freeedges(2)); end
if PBD(3)==1; fprintf(fpbc,'NA'); else fprintf(fpbc,'%i',freeedges(3)); end
fprintf(fpbc,']\n');
%% Add reference nodes
RefNodeX=max(NODES(:,1))*10;
RefNodeY=RefNodeX+1;
RefNodeZ=RefNodeX+2;
fprintf(fpbc,'**********************\n');
fprintf(fpbc,'**Reference Nodes to apply displacements on Periodic BCs\n');
fprintf(fpbc,'*NODE, NSET=RefNodeX\n');
fprintf(fpbc,'%d, 1., 0., 0.\n',RefNodeX);
fprintf(fpbc,'*NODE, NSET=RefNodeY\n');
fprintf(fpbc,'%d, 0., 1., 0.\n',RefNodeY);
fprintf(fpbc,'*NODE, NSET=RefNodeZ\n');
fprintf(fpbc,'%d, 0., 0., 1.\n',RefNodeZ);
%% Write equation data for periodic BCs
lcdof=['x','y','z'];
```

147

```
UCDOF=['X','Y','Z'];
edgelc=['yz';'xz';'xy'];
EDGEUC=['YZ';'XZ';'XY'];
edgedof=[2,3;1,3;1,2];
%% Corners
fprintf('Writing equations for corners...');tic
%Pin Origin
if sum(PBD)~=3
    if ~isempty(x0y0z03dof)
        fprintf(fpbc,'**\n**pin x0y0z03dof BCs\n');
        for DOF = 1 : 3
            fprintf(fpbc,'*BOUNDARY\n');
            fprintf(fpbc,'%d,%d,,%g\n',x0y0z03dof(1),DOF,0.0);
        end
    end
    if ~isempty(x0y0z06dof)
        fprintf(fpbc,'**\n**pin x0y0z06dof BCs\n');
        if embedded==0
            for DOF = 1 : 3
                fprintf(fpbc,'*BOUNDARY\n');
                fprintf(fpbc,'%d,%d,,%g\n',x0y0z06dof(1),DOF,0.0);
            end
        end
    end
end
if sum(PBD)==3 %i.e. periodic in all directions
    for k = 0 : 1
        for j = 0 : 1
            for i = 0 : 1
                if ~isempty(eval(sprintf('x%dy%dz%d3dof',i,j,k)))
                    fprintf(fpbc,'**\n**x%dy%dz%d3dof BCs\n',i,j,k);
                    for DOF = 1 : 3
                        fprintf(fpbc,'*EQUATION\n');
                        fprintf(fpbc,'%d\n',4);
                        fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('x%dy%dz%d3dof(1)',i,j,k)),DOF,1);
                        fprintf(fpbc,'RefNodeX,%d,%g\n',DOF,-i);
                        fprintf(fpbc,'RefNodeY,%d,%g\n',DOF,-j);
                        fprintf(fpbc,'RefNodeZ,%d,%g\n',DOF,-k);
                    end
                end
            end
        end
    end
    if embedded==0
        for k = 0 : 1
            for j = 0 : 1
                for i = 0 : 1
                    if ~isempty(eval(sprintf('x%dy%dz%d6dof',i,j,k)))
                        fprintf(fpbc,'**\n**x%dy%dz%d6dof BCs\n',i,j,k);
                        for DOF = 1 : 3
                            fprintf(fpbc,'*EQUATION\n');
                            fprintf(fpbc,'%d\n',4);
                            fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('x%dy%dz%d6dof(1)',i,j,k)),DOF,1);
                            fprintf(fpbc,'RefNodeX,%d,%g\n',DOF,-i);
                            fprintf(fpbc,'RefNodeY,%d,%g\n',DOF,-j);
                            fprintf(fpbc,'RefNodeZ,%d,%g\n',DOF,-k);
                        end
                    end
                end
            end
        end
    end
elseif sum(PBD)==2 %i.e. periodic in only 2 directions
    for k = 1 : 3
        if PBD(k)==0 %not periodic in k
            if ~isempty(eval(sprintf('%c%d%c%d%c%d3dof','x',0,'y',0,'z',0)))
                for m = 0 : 1
                    for j = 0 : 1
                        for i = 0 : 1
```

```
                if (i+j)~=0
                    for DOF = 1 : 3
                        fprintf(fpbc,'*EQUATION\n');
                        fprintf(fpbc,'%d\n',4);
                        if k==1 %not periodic in X
                            fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d3dof(1)','x',m,'y',i,'z',j)),DOF,1);
                            fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d3dof(1)','x',m,'y',0,'z',0)),DOF,-1);
                        elseif k==2 %not periodic in Y
                            fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d3dof(1)','x',i,'y',m,'z',j)),DOF,1);
                            fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d3dof(1)','x',0,'y',m,'z',0)),DOF,-1);
                        elseif k==3 %not periodic in Z
                            fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d3dof(1)','x',i,'y',j,'z',m)),DOF,1);
                            fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d3dof(1)','x',0,'y',0,'z',m)),DOF,-1);
                        end
                        fprintf(fpbc,'RefNode%c,%d,%g\n',EDGEUC(k,1),DOF,-i);
                        fprintf(fpbc,'RefNode%c,%d,%g\n',EDGEUC(k,2),DOF,-j);
                    end
                end
            end
        end
    end
    if ~isempty(eval(sprintf('%c%d%c%d%c%d6dof','x',0,'y',0,'z',0)))
        for m = 0 : 1
            for j = 0 : 1
                for i = 0 : 1
                    if (i+j)~=0
                        if embedded==0
                            for DOF = 1 : 3
                                fprintf(fpbc,'*EQUATION\n');
                                fprintf(fpbc,'%d\n',4);
                                if k==1 %not periodic in X
                                    fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',m,'y',i,'z',j)),DOF,1);
                                    fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',m,'y',0,'z',0)),DOF,-1);
                                elseif k==2 %not periodic in Y
                                    fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',i,'y',m,'z',j)),DOF,1);
                                    fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',0,'y',m,'z',0)),DOF,-1);
                                elseif k==3 %not periodic in Z
                                    fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',i,'y',j,'z',m)),DOF,1);
                                    fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',0,'y',0,'z',m)),DOF,-1);
                                end
                                fprintf(fpbc,'RefNode%c,%d,%g\n',EDGEUC(k,1),DOF,-i);
                                fprintf(fpbc,'RefNode%c,%d,%g\n',EDGEUC(k,2),DOF,-j);
                            end
                        end
                        for DOF = 4 : 6
                            fprintf(fpbc,'*EQUATION\n');
                            fprintf(fpbc,'%d\n',2);
                            if k==1 %not periodic in X
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',m,'y',i,'z',j)),DOF,1);
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',m,'y',0,'z',0)),DOF,-1);
                            elseif k==2 %not periodic in Y
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',i,'y',m,'z',j)),DOF,1);
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',0,'y',m,'z',0)),DOF,-1);
                            elseif k==3 %not periodic in Z
```

149

```
                                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',i,'y',j,'z',m)),DOF,1);
                                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',0,'y',0,'z',m)),DOF,-1);
                                            end
                                        end
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    elseif sum(PBD)==1 %i.e. periodic in only 1 direction
        for k = 1 : 3
            if PBD(k)==1 %periodic in k
                if ~isempty(eval(sprintf('%c%d%c%d%c%d3dof','x',0,'y',0,'z',0)))
                    for j = 0 : 1
                        for i = 0 : 1
                            for DOF = 1 : 3
                                fprintf(fpbc,'*EQUATION\n');
                                fprintf(fpbc,'%d\n',3);
                                if k==1 %periodic only in X
                                    fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d3dof(1)','x',1,'y',i,'z',j)),DOF,1);
                                    fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d3dof(1)','x',0,'y',i,'z',j)),DOF,-1);
                                    fprintf(fpbc,'RefNodeX,%d,%g\n',DOF,-1);
                                elseif k==2 %periodic only in Y
                                    fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d3dof(1)','x',i,'y',1,'z',j)),DOF,1);
                                    fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d3dof(1)','x',i,'y',0,'z',j)),DOF,-1);
                                    fprintf(fpbc,'RefNodeY,%d,%g\n',DOF,-1);
                                elseif k==3 %periodic only in Z
                                    fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d3dof(1)','x',i,'y',j,'z',1)),DOF,1);
                                    fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d3dof(1)','x',i,'y',j,'z',0)),DOF,-1);
                                    fprintf(fpbc,'RefNodeZ,%d,%g\n',DOF,-1);
                                end
                            end
                        end
                    end
                end
            end
        end
        if ~isempty(eval(sprintf('%c%d%c%d%c%d6dof','x',0,'y',0,'z',0)))
            for j = 0 : 1
                for i = 0 : 1
                    if embedded==0
                        for DOF = 1 : 3
                            fprintf(fpbc,'*EQUATION\n');
                            fprintf(fpbc,'%d\n',3);
                            if k==1 %periodic only in X
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',1,'y',i,'z',j)),DOF,1);
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',0,'y',i,'z',j)),DOF,-1);
                                fprintf(fpbc,'RefNodeX,%d,%g\n',DOF,-1);
                            elseif k==2 %periodic only in Y
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',i,'y',1,'z',j)),DOF,1);
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',i,'y',0,'z',j)),DOF,-1);
                                fprintf(fpbc,'RefNodeY,%d,%g\n',DOF,-1);
                            elseif k==3 %periodic only in Z
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',i,'y',j,'z',1)),DOF,1);
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',i,'y',j,'z',0)),DOF,-1);
```

150

```
                             fprintf(fpbc,'RefNodeZ,%d,%g\n',DOF,-1);
                         end
                     end
                 end
                 for DOF = 4 : 6
                     fprintf(fpbc,'*EQUATION\n');
                     fprintf(fpbc,'%d\n',2);
                     if k==1 %periodic only in X
                         fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',1,'y',i,'z',j)),DOF,1);
                         fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',0,'y',i,'z',j)),DOF,-1);
                     elseif k==2 %periodic only in Y
                         fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',i,'y',1,'z',j)),DOF,1);
                         fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',i,'y',0,'z',j)),DOF,-1);
                     elseif k==3 %periodic only in Z
                         fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',i,'y',j,'z',1)),DOF,1);
                         fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d%c%d6dof(1)','x',i,'y',j,'z',0)),DOF,-1);
                     end
                 end
             end
         end
     end
end
fprintf(' completed in %6.4g seconds.\n',toc);
%% Edges
fprintf('Writing equations for edges...');tic
% sort edges
for k = 1 : 3
    if ~isempty(eval(sprintf('%c%d%c%d3dof',edgelc(k,1),0,edgelc(k,2),0)))
        fprintf(fpbc,'**\n**%c%d%c%d3dof\n',edgelc(k,1),0,edgelc(k,2),0);
        for j = 0 : 1
            for i = 0 : 1
                eval(sprintf('%c%d%c%d3dof_sorted=sortrows(%c%d%c%d3dof,k+1);'...
                    ,edgelc(k,1),i,edgelc(k,2),j,edgelc(k,1),i,edgelc(k,2),j));
            end
        end
        if PBD(edgedof(k,1))==1 && PBD(edgedof(k,2))==1 %periodic in both directions
            for m = 1 : eval(sprintf('size(%c%d%c%d3dof_sorted,1)',edgelc(k,1),0,edgelc(k,2),0))
                for j = 0 : 1
                    for i = 0 : 1
                        if (i+j)~=0
                            for DOF = 1 : 3
                                fprintf(fpbc,'*EQUATION\n');
                                fprintf(fpbc,'%d\n',4);
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d3dof_sorted(m,1)',edgelc(k,1),i,edgelc(k,2),j)),DOF,1);
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d3dof_sorted(m,1)',edgelc(k,1),0,edgelc(k,2),0)),DOF,-1);
                                fprintf(fpbc,'RefNode%c,%d,%g\n',EDGEUC(k,1),DOF,-i);
                                fprintf(fpbc,'RefNode%c,%d,%g\n',EDGEUC(k,2),DOF,-j);
                            end
                        end
                    end
                end
            end
        elseif PBD(edgedof(k,1))==1
            for m = 1 : eval(sprintf('size(%c%d%c%d3dof_sorted,1)',edgelc(k,1),0,edgelc(k,2),0))
                for i = 0 : 1
                    for DOF = 1 : 3
                        fprintf(fpbc,'*EQUATION\n');
                        fprintf(fpbc,'%d\n',3);
                        fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d3dof_sorted(m,1)',edgelc(k,1),1,edgelc(k,2),i)),DOF,1);
                        fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d3dof_sorted(m,1)',edgelc(k,1),0,edgelc(k,2),i)),DOF,-1);
                        fprintf(fpbc,'RefNode%c,%d,%g\n',EDGEUC(k,1),DOF,-1);
```

151

```
                    end
                end
            end
        elseif PBD(edgedof(k,2))==1
            for m = 1 : eval(sprintf('size(%c%d%c%d3dof_sorted,1)',edgelc(k,1),0,edgelc(k,2),0))
                for j = 0 : 1
                    for DOF = 1 : 3
                        fprintf(fpbc,'*EQUATION\n');
                        fprintf(fpbc,'%d\n',3);
                        fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d3dof_sorted(m,1)',edgelc(k,1),j,edgelc(k,2),1)),DOF,1);
                        fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d3dof_sorted(m,1)',edgelc(k,1),j,edgelc(k,2),0)),DOF,-1);
                        fprintf(fpbc,'RefNode%c,%d,%g\n',EDGEUC(k,2),DOF,-1);
                    end
                end
            end
        end
    end
end
for k = 1 : 3
    if ~isempty(eval(sprintf('%c%d%c%d6dof',edgelc(k,1),0,edgelc(k,2),0)))
        fprintf(fpbc,'**\n**%c%d%c%d6dof\n',edgelc(k,1),0,edgelc(k,2),0);
        for j = 0 : 1
            for i = 0 : 1
                eval(sprintf('%c%d%c%d6dof_sorted=sortrows(%c%d%c%d6dof,k+1);'...
                    ,edgelc(k,1),i,edgelc(k,2),j,edgelc(k,1),i,edgelc(k,2),j));
            end
        end
        if PBD(edgedof(k,1))==1 && PBD(edgedof(k,2))==1 %periodic in both directions
            for m = 1 : eval(sprintf('size(%c%d%c%d6dof_sorted,1)',edgelc(k,1),0,edgelc(k,2),0))
                for j = 0 : 1
                    for i = 0 : 1
                        if (i+j)~=0
                            if embedded==0
                                for DOF = 1 : 3
                                    fprintf(fpbc,'*EQUATION\n');
                                    fprintf(fpbc,'%d\n',4);
                                    fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d6dof_sorted(m,1)',edgelc(k,1),i,edgelc(k,2),j)),DOF,1);
                                    fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d6dof_sorted(m,1)',edgelc(k,1),0,edgelc(k,2),0)),DOF,-1);
                                    fprintf(fpbc,'RefNode%c,%d,%g\n',EDGEUC(k,1),DOF,-i);
                                    fprintf(fpbc,'RefNode%c,%d,%g\n',EDGEUC(k,2),DOF,-j);
                                end
                            end
                            for DOF = 4 : 6
                                fprintf(fpbc,'*EQUATION\n');
                                fprintf(fpbc,'%d\n',2);
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d6dof_sorted(m,1)',edgelc(k,1),i,edgelc(k,2),j)),DOF,1);
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d6dof_sorted(m,1)',edgelc(k,1),0,edgelc(k,2),0)),DOF,-1);
                            end
                        end
                    end
                end
            end
        elseif PBD(edgedof(k,1))==1
            for m = 1 : eval(sprintf('size(%c%d%c%d6dof_sorted,1)',edgelc(k,1),0,edgelc(k,2),0))
                for i = 0 : 1
                    if embedded==0
                        for DOF = 1 : 3
                            fprintf(fpbc,'*EQUATION\n');
                            fprintf(fpbc,'%d\n',3);
                            fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d6dof_sorted(m,1)',edgelc(k,1),1,edgelc(k,2),i)),DOF,1);
                            fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d6dof_sorted(m,1)',edgelc(k,1),0,edgelc(k,2),i)),DOF,-1);
                            fprintf(fpbc,'RefNode%c,%d,%g\n',EDGEUC(k,1),DOF,-1);
                        end
```

152

```
                        end
                        for DOF = 4 : 6
                            fprintf(fpbc,'*EQUATION\n');
                            fprintf(fpbc,'%d\n',2);
                            fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d6dof_sorted(m,1)',edgelc(k,1),1,edgelc(k,2),i)),DOF,1);
                            fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d6dof_sorted(m,1)',edgelc(k,1),0,edgelc(k,2),i)),DOF,-1);
                        end
                    end
                end
            elseif PBD(edgedof(k,2))==1
                for m = 1 : eval(sprintf('size(%c%d%c%d6dof_sorted,1)',edgelc(k,1),0,edgelc(k,2),0))
                    for j = 0 : 1
                        if embedded==0
                            for DOF = 1 : 3
                                fprintf(fpbc,'*EQUATION\n');
                                fprintf(fpbc,'%d\n',3);
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d6dof_sorted(m,1)',edgelc(k,1),j,edgelc(k,2),1)),DOF,1);
                                fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d6dof_sorted(m,1)',edgelc(k,1),j,edgelc(k,2),0)),DOF,-1);
                                fprintf(fpbc,'RefNode%c,%d,%g\n',EDGEUC(k,2),DOF,-1);
                            end
                        end
                        for DOF = 4 : 6
                            fprintf(fpbc,'*EQUATION\n');
                            fprintf(fpbc,'%d\n',2);
                            fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d6dof_sorted(m,1)',edgelc(k,1),j,edgelc(k,2),1)),DOF,1);
                            fprintf(fpbc,'%d,%d,%g\n',
eval(sprintf('%c%d%c%d6dof_sorted(m,1)',edgelc(k,1),j,edgelc(k,2),0)),DOF,-1);
                            fprintf(fpbc,'RefNode%c,%d,%g\n',EDGEUC(k,2),DOF,-1);
                        end
                    end
                end
            end
        end
    end
end
fprintf(' completed in %6.4g seconds.\n',toc);
%% initiate size variables
nterms=0;
%% non-periodic faces
fprintf('Writing equations for non-periodic faces...');tic
for j = 1 : 3
    if PBD(j)==1
    elseif freeedges(j)==1 && sum(clampped)==0
    else %non periodic boundry condition (average strain and/or controlled rotation)
        fprintf(fpbc,['**\n**\n*********************\n****',UCDOF(j),'0-',UCDOF(j),'1
equations****\n*********************\n']);
        if j==1 %X direction
            if freeedges(j)==0
                %X0 list
                X03dofList=[x03dof;...
                    x0y03dof;x0y13dof;x0z03dof;x0z13dof;...
                    x0y0z03dof;x0y0z13dof;x0y1z03dof;x0y1z13dof];
                %X1 list
                X13dofList=[x13dof;...
                    x1y03dof;x1y13dof;x1z03dof;x1z13dof;...
                    x1y0z03dof;x1y0z13dof;x1y1z03dof;x1y1z13dof];
                if ~isempty(X03dofList)&&~isempty(X13dofList)
                    AreaX03dof=NodalAreaInfluence(X03dofList,ELEMENTS,'Surface X0
3dof',PlotNPBC);
                    AreaX13dof=NodalAreaInfluence(X13dofList,ELEMENTS,'Surface X1
3dof',PlotNPBC);
                end
            end
            if embedded==0||sum(clampped)>0
                %X0 list
                X06dofList=[x06dof;...
                    x0y06dof;x0y16dof;x0z06dof;x0z16dof;...
```

153

```
                    x0y0z06dof;x0y0z16dof;x0y1z06dof;x0y1z16dof];
                %X1 list
                X16dofList=[x16dof;...
                    x1y06dof;x1y16dof;x1z06dof;x1z16dof;...
                    x1y0z06dof;x1y0z16dof;x1y1z06dof;x1y1z16dof];
                if embedded==0
                    if ~isempty(X06dofList)&&~isempty(X16dofList)
                        AreaX06dof=BoundedVoronoiArea(X06dofList(:,3),X06dofList(:,4),'Surface X0
6dof',PlotNPBC);
                        AreaX16dof=BoundedVoronoiArea(X16dofList(:,3),X16dofList(:,4),'Surface X1
6dof',PlotNPBC);
                    end
                end
            end
        elseif j==2 %Y direction
            %Create list of all points on surface
            if freeedges(j)==0
                %Y0 list
                Y03dofList=[y03dof;...
                    x0y03dof;x1y03dof;y0z03dof;y0z13dof;...
                    x0y0z03dof;x0y0z13dof;x1y0z03dof;x1y0z13dof];
                %Y1 list
                Y13dofList=[y13dof;...
                    x0y13dof;x1y13dof;y1z03dof;y1z13dof;...
                    x0y1z03dof;x0y1z13dof;x1y1z03dof;x1y1z13dof];
                if ~isempty(Y03dofList)&&~isempty(Y13dofList)
                    AreaY03dof=NodalAreaInfluence(Y03dofList,ELEMENTS,'Surface Y0
3dof',PlotNPBC);
                    AreaY13dof=NodalAreaInfluence(Y13dofList,ELEMENTS,'Surface Y1
3dof',PlotNPBC);
                end
            end
            if embedded==0||sum(clampped)>0
                %Y0 list
                Y03dofList=[y06dof;...
                    x0y06dof;x1y06dof;y0z06dof;y0z16dof;...
                    x0y0z06dof;x0y0z16dof;x1y0z06dof;x1y0z16dof];
                %Y1 list
                Y13dofList=[y16dof;...
                    x0y16dof;x1y16dof;y1z06dof;y1z16dof;...
                    x0y1z06dof;x0y1z16dof;x1y1z06dof;x1y1z16dof];
                if embedded==0
                    if ~isempty(Y06dofList)&&~isempty(Y16dofList)
                        AreaY06dof=BoundedVoronoiArea(Y06dofList(:,2),Y06dofList(:,4),'Surface Y0
6dof',PlotNPBC);
                        AreaY16dof=BoundedVoronoiArea(Y16dofList(:,2),Y16dofList(:,4),'Surface Y1
6dof',PlotNPBC);
                    end
                end
            end
        elseif j==3 %Z direction
            %Create list of all points on surface
            if freeedges(j)==0
                %Z0 list
                Z03dofList=[z03dof;...
                    x0z03dof;x1z03dof;y0z03dof;y1z03dof;...
                    x0y0z03dof;x0y1z03dof;x1y0z03dof;x1y1z03dof];
                %Z1 list
                Z13dofList=[z13dof;...
                    x0z13dof;x1z13dof;y0z13dof;y1z13dof;...
                    x0y0z13dof;x0y1z13dof;x1y0z13dof;x1y1z13dof];
                if ~isempty(Z03dofList)&&~isempty(Z13dofList)
                    AreaZ03dof=NodalAreaInfluence(Z03dofList,ELEMENTS,'Surface Z0
3dof',PlotNPBC);
                    AreaZ13dof=NodalAreaInfluence(Z13dofList,ELEMENTS,'Surface Z1
3dof',PlotNPBC);
                end
            end
            if embedded==0||sum(clampped)>0
                %Z0 list
                Z06dofList=[z06dof;...
```

```
                        x0z06dof;x1z06dof;y0z06dof;y1z06dof;...
                        x0y0z06dof;x0y1z06dof;x1y0z06dof;x1y1z06dof];
                    %Z1 list
                    Z16dofList=[z16dof;...
                        x0z16dof;x1z16dof;y0z16dof;y1z16dof;...
                        x0y0z16dof;x0y1z16dof;x1y0z16dof;x1y1z16dof];
                    if embedded==0
                        if ~isempty(Z06dofList)&&~isempty(Z16dofList)
                            AreaZ06dof=BoundedVoronoiArea(Z06dofList(:,2),Z06dofList(:,3),'Surface Z0
6dof',PlotNPBC);
                            AreaZ16dof=BoundedVoronoiArea(Z16dofList(:,2),Z16dofList(:,3),'Surface Z1
6dof',PlotNPBC);
                        end
                    end
                end
            end
        if freeedges(j)==0
            if j==1 %X direction
                if exist('AreaX13dof','var') && exist('AreaX03dof','var')
                    %equation for corner nodes of elements
                    if sum(any(AreaX13dof(:,1),2))~=0 && sum(any(AreaX03dof(:,1),2))~=0
                        nterms=sum(any(AreaX03dof(:,1),2))+sum(any(AreaX13dof(:,1),2))+1;
                        for DOF = 1 : 3
                            fprintf(fpbc,'**Non-Periodic Boundry Condition for element corner
3DOF nodes X0-X1, dof=%c\n',DOF);
                            fprintf(fpbc,'*EQUATION\n');
                            fprintf(fpbc,'%d\n',nterms);
                            %X1 corner terms
                            for i = 1 : size(AreaX13dof,1)
                                if AreaX13dof(i,1)~=0

fprintf(fpbc,'%d,%d,%9.9g\n',X13dofList(i,1),DOF,(1.0*nterms*AreaX13dof(i,1)/sum(AreaX13dof(:,1))
));
                                end
                            end
                            %X0 corner terms
                            for i = 1 : size(AreaX03dof,1)
                                if AreaX03dof(i,1)~=0
                                    fprintf(fpbc,'%d,%d,%9.9g\n',X03dofList(i,1),DOF,(-
1.0*nterms*AreaX03dof(i,1)/sum(AreaX03dof(:,1))));
                                end
                            end
                            fprintf(fpbc,'RefNodeX,%d,%9.9g\n',DOF,(-1.0*nterms));
                        end
                    end
                    %equation for midpoint nodes of elements
                    if sum(any(AreaX13dof(:,2),2))~=0 && sum(any(AreaX03dof(:,2),2))~=0
                        nterms=sum(any(AreaX03dof(:,2),2))+sum(any(AreaX13dof(:,2),2))+1;
                        for DOF = 1 : 3
                            fprintf(fpbc,'**Non-Periodic Boundry Condition for element midpoint
3DOF nodes X0-X1, dof=%c\n',DOF);
                            fprintf(fpbc,'*EQUATION\n');
                            fprintf(fpbc,'%d\n',nterms);
                            if sum(any(AreaX13dof(:,2),2))~=0 && sum(any(AreaX03dof(:,2),2))~=0
                                %X1 midpoint terms
                                for i = 1 : size(AreaX13dof,1)
                                    if AreaX13dof(i,2)~=0

fprintf(fpbc,'%d,%d,%9.9g\n',X13dofList(i,1),DOF,(1.0*nterms*AreaX13dof(i,2)/sum(AreaX13dof(:,2))
));
                                    end
                                end
                                %X0 midpoint terms
                                for i = 1 : size(AreaX03dof,1)
                                    if AreaX03dof(i,2)~=0
                                        fprintf(fpbc,'%d,%d,%9.9g\n',X03dofList(i,1),DOF,(-
1.0*nterms*AreaX03dof(i,2)/sum(AreaX03dof(:,2))));
                                    end
                                end
                                fprintf(fpbc,'RefNodeX,%d,%9.9g\n',DOF,(-1.0*nterms));
                            end
```

155

```matlab
                        end
                    end
                end
                if embedded==0 %6DOF translational boundry conditions
                    if exist('AreaX16dof','var') && exist('AreaX06dof','var')
                        %equation for corner nodes of elements
                        if sum(any(AreaX16dof(:,1),1))~=0 && sum(any(AreaX06dof(:,1),1))~=0
                            nterms=sum(any(AreaX06dof(:,1),1))+sum(any(AreaX16dof(:,1),1))+1;
                            for DOF = 1 : 3
                                fprintf(fpbc,'**Non-Periodic Boundry Condition for 6DOF X0-X1,
dof=%c\n',DOF);
                                fprintf(fpbc,'*EQUATION\n');
                                fprintf(fpbc,'%d\n',nterms);
                                if sum(any(AreaX16dof(:,2),2))~=0 &&
sum(any(AreaX06dof(:,2),2))~=0
                                    %X1 terms
                                    for i = 1 : size(AreaX16dof,1)
                                        if AreaX16dof(i,1)~=0

fprintf(fpbc,'%d,%d,%9.9g\n',X16dofList(i,1),DOF,(1.0*nterms*AreaX16dof(i,1)/sum(AreaX16dof)));
                                        end
                                    end
                                    %X0 terms
                                    for i = 1 : size(AreaX06dof,1)
                                        if AreaX06dof(i,1)~=0
                                            fprintf(fpbc,'%d,%d,%9.9g\n',X06dofList(i,1),DOF,(-
1.0*nterms*AreaX06dof(i,1)/sum(AreaX06dof)));
                                        end
                                    end
                                    fprintf(fpbc,'RefNodeX,%d,%9.9g\n',DOF,(-1.0*nterms));
                                end
                            end
                        end
                    end
                end
            elseif j==2 %Y direction
                if exist('AreaY13dof','var') && exist('AreaY03dof','var')
                    %equation for corner nodes of elements
                    if sum(any(AreaY13dof(:,1),2))~=0 && sum(any(AreaY03dof(:,1),2))~=0
                        nterms=sum(any(AreaY03dof(:,1),2))+sum(any(AreaY13dof(:,1),2))+1;
                        for DOF = 1 : 3
                            fprintf(fpbc,'**Non-Periodic Boundry Condition for element corner
3DOF nodes Y0-Y1, dof=%c\n',DOF);
                            fprintf(fpbc,'*EQUATION\n');
                            fprintf(fpbc,'%d\n',nterms);
                            %Y1 corner terms
                            for i = 1 : size(AreaY13dof,1)
                                if AreaY13dof(i,1)~=0

fprintf(fpbc,'%d,%d,%9.9g\n',Y13dofList(i,1),DOF,(1.0*nterms*AreaY13dof(i,1)/sum(AreaY13dof(:,1))
));
                                end
                            end
                            %Y0 corner terms
                            for i = 1 : size(AreaY03dof,1)
                                if AreaY03dof(i,1)~=0
                                    fprintf(fpbc,'%d,%d,%9.9g\n',Y03dofList(i,1),DOF,(-
1.0*nterms*AreaY03dof(i,1)/sum(AreaY03dof(:,1))));
                                end
                            end
                            fprintf(fpbc,'RefNodeY,%d,%9.9g\n',DOF,(-1.0*nterms));
                        end
                    end
                    %equation for midpoint nodes of elements
                    if sum(any(AreaY13dof(:,2),2))~=0 && sum(any(AreaY03dof(:,2),2))~=0
                        nterms=sum(any(AreaY03dof(:,2),2))+sum(any(AreaY13dof(:,2),2))+1;
                        for DOF = 1 : 3
                            fprintf(fpbc,'**Non-Periodic Boundry Condition for element midpoint
3DOF nodes Y0-Y1, dof=%c\n',DOF);
                            fprintf(fpbc,'*EQUATION\n');
                            fprintf(fpbc,'%d\n',nterms);
```

```
                if sum(any(AreaY13dof(:,2),2))~=0 && sum(any(AreaY03dof(:,2),2))~=0
                    %Y1 midpoint terms
                    for i = 1 : size(AreaY13dof,1)
                        if AreaY13dof(i,2)~=0

fprintf(fpbc,'%d,%d,%9.9g\n',Y13dofList(i,1),DOF,(1.0*nterms*AreaY13dof(i,2)/sum(AreaY13dof(:,2))
));
                        end
                    end
                    %Y0 midpoint terms
                    for i = 1 : size(AreaY03dof,1)
                        if AreaY03dof(i,2)~=0
                            fprintf(fpbc,'%d,%d,%9.9g\n',Y03dofList(i,1),DOF,(-
1.0*nterms*AreaY03dof(i,2)/sum(AreaY03dof(:,2))));
                        end
                    end
                    fprintf(fpbc,'RefNodeY,%d,%9.9g\n',DOF,(-1.0*nterms));
                end
            end
        end
    end
    if embedded==0 %6DOF translational boundry conditions
        if exist('AreaY16dof','var') && exist('AreaY06dof','var')
            %equation for corner nodes of elements
            if sum(any(AreaY16dof(:,1),1))~=0 && sum(any(AreaY06dof(:,1),1))~=0
                nterms=sum(any(AreaY06dof(:,1),1))+sum(any(AreaY16dof(:,1),1))+1;
                for DOF = 1 : 3
                    fprintf(fpbc,'**Non-Periodic Boundry Condition for 6DOF Y0-Y1,
dof=%c\n',DOF);
                    fprintf(fpbc,'*EQUATION\n');
                    fprintf(fpbc,'%d\n',nterms);
                    if sum(any(AreaY16dof(:,2),2))~=0 &&
sum(any(AreaY06dof(:,2),2))~=0
                        %Y1 terms
                        for i = 1 : size(AreaY16dof,1)
                            if AreaY16dof(i,1)~=0

fprintf(fpbc,'%d,%d,%9.9g\n',Y16dofList(i,1),DOF,(1.0*nterms*AreaY16dof(i,1)/sum(AreaY16dof)));
                            end
                        end
                        %Y0 terms
                        for i = 1 : size(AreaY06dof,1)
                            if AreaY06dof(i,1)~=0
                                fprintf(fpbc,'%d,%d,%9.9g\n',Y06dofList(i,1),DOF,(-
1.0*nterms*AreaY06dof(i,1)/sum(AreaY06dof)));
                            end
                        end
                        fprintf(fpbc,'RefNodeY,%d,%9.9g\n',DOF,(-1.0*nterms));
                    end
                end
            end
        end
    end
elseif j==3 %Z direction
    if exist('AreaZ13dof','var') && exist('AreaZ03dof','var')
        %equation for corner nodes of elements
        if sum(any(AreaZ13dof(:,1),2))~=0 && sum(any(AreaZ03dof(:,1),2))~=0
            nterms=sum(any(AreaZ03dof(:,1),2))+sum(any(AreaZ13dof(:,1),2))+1;
            for DOF = 1 : 3
                fprintf(fpbc,'**Non-Periodic Boundry Condition for element corner
3DOF nodes Z0-Z1, dof=%c\n',DOF);
                fprintf(fpbc,'*EQUATION\n');
                fprintf(fpbc,'%d\n',nterms);
                %Z1 corner terms
                for i = 1 : size(AreaZ13dof,1)
                    if AreaZ13dof(i,1)~=0

fprintf(fpbc,'%d,%d,%9.9g\n',Z13dofList(i,1),DOF,(1.0*nterms*AreaZ13dof(i,1)/sum(AreaZ13dof(:,1))
));
                    end
                end
```

```
                    %Z0 corner terms
                    for i = 1 : size(AreaZ03dof,1)
                        if AreaZ03dof(i,1)~=0
                            fprintf(fpbc,'%d,%d,%9.9g\n',Z03dofList(i,1),DOF,(-
1.0*nterms*AreaZ03dof(i,1)/sum(AreaZ03dof(:,1))));
                        end
                    end
                    fprintf(fpbc,'RefNodeZ,%d,%9.9g\n',DOF,(-1.0*nterms));
                end
            end
            %equation for midpoint nodes of elements
            if sum(any(AreaZ13dof(:,2),2))~=0 && sum(any(AreaZ03dof(:,2),2))~=0
                nterms=sum(any(AreaZ03dof(:,2),2))+sum(any(AreaZ13dof(:,2),2))+1;
                for DOF = 1 : 3
                    fprintf(fpbc,'**Non-Periodic Boundry Condition for element midpoint
3DOF nodes Z0-Z1, dof=%c\n',DOF);
                    fprintf(fpbc,'*EQUATION\n');
                    fprintf(fpbc,'%d\n',nterms);
                    if sum(any(AreaZ13dof(:,2),2))~=0 && sum(any(AreaZ03dof(:,2),2))~=0
                        %Z1 midpoint terms
                        for i = 1 : size(AreaZ13dof,1)
                            if AreaZ13dof(i,2)~=0

fprintf(fpbc,'%d,%d,%9.9g\n',Z13dofList(i,1),DOF,(1.0*nterms*AreaZ13dof(i,2)/sum(AreaZ13dof(:,2))
));
                            end
                        end
                        %Z0 midpoint terms
                        for i = 1 : size(AreaZ03dof,1)
                            if AreaZ03dof(i,2)~=0
                                fprintf(fpbc,'%d,%d,%9.9g\n',Z03dofList(i,1),DOF,(-
1.0*nterms*AreaZ03dof(i,2)/sum(AreaZ03dof(:,2))));
                            end
                        end
                        fprintf(fpbc,'RefNodeZ,%d,%9.9g\n',DOF,(-1.0*nterms));
                    end
                end
            end
        end
        if embedded==0 %6DOF translational boundry conditions
            if exist('AreaZ16dof','var') && exist('AreaZ06dof','var')
                %equation for corner nodes of elements
                if sum(any(AreaZ16dof(:,1),1))~=0 && sum(any(AreaZ06dof(:,1),1))~=0
                    nterms=sum(any(AreaZ06dof(:,1),1))+sum(any(AreaZ16dof(:,1),1))+1;
                    for DOF = 1 : 3
                        fprintf(fpbc,'**Non-Periodic Boundry Condition for 6DOF Z0-Z1,
dof=%c\n',DOF);
                        fprintf(fpbc,'*EQUATION\n');
                        fprintf(fpbc,'%d\n',nterms);
                        if sum(any(AreaZ16dof(:,2),2))~=0 &&
sum(any(AreaZ06dof(:,2),2))~=0
                            %Z1 terms
                            for i = 1 : size(AreaZ16dof,1)
                                if AreaZ16dof(i,1)~=0

fprintf(fpbc,'%d,%d,%9.9g\n',Z16dofList(i,1),DOF,(1.0*nterms*AreaZ16dof(i,1)/sum(AreaZ16dof)));
                                end
                            end
                            %Z0 terms
                            for i = 1 : size(AreaZ06dof,1)
                                if AreaZ06dof(i,1)~=0
                                    fprintf(fpbc,'%d,%d,%9.9g\n',Z06dofList(i,1),DOF,(-
1.0*nterms*AreaZ06dof(i,1)/sum(AreaZ06dof)));
                                end
                            end
                            fprintf(fpbc,'RefNodeZ,%d,%9.9g\n',DOF,(-1.0*nterms));
                        end
                    end
                end
            end
        end
    end
```

```
                end
            end
        for k = 1 : 3
            if clampped(k)==1
                if j==1&&~isempty(X16dofList)&&~isempty(X06dofList) %X direction
                    fprintf(fpbc,'**Rotation Clamped for X0-X1\n');
                    %X1 terms
                    for i = 1 : size(X16dofList,1)
                        fprintf(fpbc,'*BOUNDARY\n');
                        fprintf(fpbc,'%d,%d,,%g\n',X16dofList(i,1),3+k,0.0);
                    end
                    %X0 terms
                    for i = 1 : size(X06dofList,1)
                        fprintf(fpbc,'*BOUNDARY\n');
                        fprintf(fpbc,'%d,%d,,%g\n',X06dofList(i,1),3+k,0.0);
                    end
                elseif j==2&&~isempty(Y16dofList)&&~isempty(Y06dofList) %Y direction
                    fprintf(fpbc,'**Rotation Clamped for Y0-Y1\n');
                    %Y1 terms
                    for i = 1 : size(Y16dofList,1)
                        fprintf(fpbc,'*BOUNDARY\n');
                        fprintf(fpbc,'%d,%d,,%g\n',Y16dofList(i,1),3+k,0.0);
                    end
                    %Y0 terms
                    for i = 1 : size(Y06dofList,1)
                        fprintf(fpbc,'*BOUNDARY\n');
                        fprintf(fpbc,'%d,%d,,%g\n',Y06dofList(i,1),3+k,0.0);
                    end
                elseif j==3&&~isempty(Z16dofList)&&~isempty(Z06dofList) %Z direction
                    fprintf(fpbc,'**Rotation Clamped for Z0-Z1\n');
                    %Z1 terms
                    for i = 1 : size(Z16dofList,1)
                        fprintf(fpbc,'*BOUNDARY\n');
                        fprintf(fpbc,'%d,%d,,%g\n',Z16dofList(i,1),3+k,0.0);
                    end
                    %Z0 terms
                    for i = 1 : size(Z06dofList,1)
                        fprintf(fpbc,'*BOUNDARY\n');
                        fprintf(fpbc,'%d,%d,,%g\n',Z06dofList(i,1),3+k,0.0);
                    end
                end
            end
        end
    end
end
fprintf(' completed in %6.4g seconds.\n',toc);
%% periodic faces
fprintf('Writing equations for periodic faces...');tic
for j = 1 : 3
    if PBD(j)==1
        fprintf(fpbc,['**\n**\n**********************\n****',UCDOF(j),'0-',UCDOF(j),'1
equations****\n**********************\n']);
        for i = 1 : eval(sprintf('size(%s03dof,1)',lcdof(j)))
            for DOF = 1 : 3
                fprintf(fpbc,'*EQUATION\n');
                fprintf(fpbc,'%d\n',3);
                fprintf(fpbc,'%d,%d,1\n',eval([lcdof(j),'13dof_sorted(i,1)']),DOF);
                fprintf(fpbc,'%d,%d,-1\n',eval([lcdof(j),'03dof(i,1)']),DOF);
                fprintf(fpbc,'RefNode%c,%d,-1\n',UCDOF(j),DOF);
            end
        end
        for i = 1 : eval(sprintf('size(%s06dof,1)',lcdof(j)))
            for DOF = 1 : 6
                if DOF<=3 && embedded==1;continue;end
                fprintf(fpbc,'*EQUATION\n');
                if DOF<=3;fprintf(fpbc,'%d\n',3);else fprintf(fpbc,'%d\n',2);end
                fprintf(fpbc,'%d,%d,1\n',eval([lcdof(j),'16dof_sorted(i,1)']),DOF);
                fprintf(fpbc,'%d,%d,-1\n',eval([lcdof(j),'06dof(i,1)']),DOF);
                if DOF<=3;fprintf(fpbc,'RefNode%c,%d,-1\n',UCDOF(j),DOF);end
            end
        end
```

```matlab
        elseif freeedges(j)==1 && sum(clampped)==0
            fprintf(fpbc,['**\n**\n**********************\n****',UCDOF(j),'0-',UCDOF(j),'1
equations****\n**********************\n']);
            fprintf(fpbc,'**Boundry Free, no constraint equations\n');
        end
end
fprintf(' completed in %6.4g seconds.\n',toc);
%% Close files

% %Close mesh
% fclose(finp);

%Close Equation constrain file
fclose(fpbc);
fprintf('Equation file written successfully. \n');
fprintf('Total elapsed time is %6.4g seconds.\n',(toc(tStart)-tPause));
```

BIBLIOGRAPHY

[1] Nida-Core Corporation, "NidaFusion SXO/SXF," *2006-2008* Available: http://home.nordnet.fr/~jmaquet/french/fusiprod_sxosxf.htm.

[2] Nida-Core Corporation, "NidaFusion STO" Available: http://home.nordnet.fr/~jmaquet/french/fusiprod_sto.htm.

[3] Milliken & Company, "TYCOR | Foam Core Fiberglass Reinforced Composite Material" Available: http://tycor.milliken.com/Pages/home.aspx.

[4] Albany International Corp., "Sandwich Structures" Available: http://www.albint.com/businesses/aec/ProductsAndTechnologies/Pages/Sandwich-Structures.aspx.

[5] Marasco, A. I., Cartié, D. D. R., Partridge, I. K., and Rezai, A., "Mechanical properties balance in novel Z-pinned sandwich panels: Out-of-plane properties," *Composites Part A: Applied Science and Manufacturing*, vol. 37, 2006, pp. 295–302.

[6] Baral, N., Cartié, D. D. R., Partridge, I. K., Baley, C., and Davies, P., "Improved impact performance of marine sandwich panels using through-thickness reinforcement: Experimental results," *Composites Part B: Engineering*, vol. 41, 2010, pp. 117–123.

[7] Casari, P., Cartié, D., and Davies, P., "Characterisation of novel K-Cor sandwich structures," *Sandwich Structures 7: Advancing with Sandwich Structures and Materials*, Springer, 2005, pp. 865–874.

[8] Mouritz, a. P., "Compression properties of z-pinned sandwich composites," *Journal of Material Science*, vol. 41, 2006, pp. 5771–5774.

[9] Lascoup, B., Aboura, Z., Khellil, K., and Benzeggagh, M., "On the mechanical effect of stitch addition in sandwich panel," *Composites Science and Technology*, vol. 66, 2006, pp. 1385–1398.

[10] Potluri, P., Kusak, E., and Reddy, T. Y., "Novel stitch-bonded sandwich composite structures," *Composite Structures*, vol. 59, 2003, pp. 251–259.

[11] Gardiner, G., "Tooling up for larger launch vehicles : CompositesWorld," *High-Performance Composites* Available: http://www.compositesworld.com/articles/tooling-up-for-larger-launch-vehicles.

[12] Aboudi, J., Arnold, S. M., and Bednarcyk, B. A., *Micromechanics of composite materials: a generalized multiscale analysis approach*, Butterworth-Heinemann, 2012.

[13] Hashin, Z., "The Elastic Moduli of Heterogeneous Materials," *Journal of Applied Mechanics*, 1960, pp. 143–150.

[14] Christensen, R. M., and Waals, F. M., "Effective Stiffness of Randomly Oriented Fibre Composites," *Journal of Composite Materials*, vol. 6, 1972, pp. 518–535.

[15] Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J.-Y., White, D. J., Hartenstein, V., Eliceiri, K., Tomancak, P., and Cardona, A., "Fiji: an open-source platform for biological-image analysis," *Nature Methods*, vol. 9, 2012, pp. 676–682.

[16] "Fiji Is Just ImageJ" Available: http://fiji.sc/Fiji.

[17] "GIMP - The GNU Image Manipulation Program" Available: http://www.gimp.org/.

[18] Kylander, O. S., and Kylander, K., *Gimp the Official Handbook with Cdrom*, Coriolis Value, 1999.

[19] Totry, E., González, C., and LLorca, J., "Influence of the loading path on the strength of fiber-reinforced composites subjected to transverse compression and shear," *International Journal of Solids and Structures*, vol. 45, Mar. 2008, pp. 1663–1675.

[20] Heinrich, C., Aldridge, M., Wineman, A. S., Kieffer, J., Waas, A. M., and Shahwan, K., "The influence of the representative volume element (RVE) size on the homogenized response of cured fiber composites," *Modelling and Simulation in Materials Science and Engineering*, vol. 20, Oct. 2012, p. 075007.

[21] Chen, A., and Davalos, J. F., "A solution including skin effect for stiffness and stress field of sandwich honeycomb core," *International Journal of Solids and Structures*, vol. 42, 2005, pp. 2711–2739.

[22] Frank Xu, X., and Qiao, P., "Homogenized elastic properties of honeycomb sandwich with skin effect," *International Journal of Solids and Structures*, vol. 39, 2002, pp. 2153–2188.

[23] Chen, A., and Davalos, J. F., "Transverse Shear Including Skin Effect for Composite Sandwich with Honeycomb Sinusoidal Core," *Journal of Engineering Mechanics*, vol. 133, 2007, pp. 247–256.

[24] Qiao, P., and Wang, J., "Transverse Shear Stiffness of Composite Honeycomb Cores and Efficiency of Material," *Mechanics of Advanced Materials and Structures*, vol. 12, 2005, pp. 159–172.

[25] Skiena, S. S., *The algorithm design manual*, Springer Science & Business Media, 1998.

[26] Dirichlet, G. L., "Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen," *J. reine angew. Math.*, vol. 40, 1850, pp. 209–227.

[27] Voronoi, G., "Nouvelles applications des paramètres continus à la théorie des formes quadratiques.," *J. reine angew. Math*, vol. 133, 1907, pp. 97–178.

[28] Achenbach, J. D., *A theory of elasticity with microstructure for directionally reinforced composites*, Springer, 1975.

[29] Timoshenko, S., *History of strength of materials: with a brief account of the history of theory of elasticity and theory of structures*, Courier Corporation, 1983.

[30] Duhamel, J. M. C., *Mémoire sur le calcul des actions moléculaires développées par les changements de température dans les corps solides*, 1836.

[31] Timoshenko, S., and Goodier, J. N., *Theory of Elasticity*, York, PA: The Maple Press Company, 1951.

[32] Herakovich, C. T., *Mechanics of fibrous composites*, Wiley New York, 1998.

[33] Liu, T., Deng, Z. C., and Lu, T. J., "Analytical modeling and finite element simulation of the plastic collapse of sandwich beams with pin-reinforced foam cores," *International Journal of Solids and Structures*, vol. 45, Sep. 2008, pp. 5127–5151.

[34] Mohr, D., "Mechanism-based multi-surface plasticity model for ideal truss lattice materials," *International Journal of Solids and Structures*, vol. 42, 2005, pp. 3235–3260.

[35] Dow, N. F., *Study of stresses near a discontinuity in a filament-reinforced composite metal*, 1963.

[36] Rosen, B. W., Dow, N. F., and Hashin, Z., *Mechanical Properties of Fibrous Composites.*, 1964.

[37] Amirbayat, J., and Hearle, J. W. S., "Properties of unit composites as determined by the properties of the interface. Part I: Mechanism of matrix-fibre load transfer," *Fibre Science and Technology*, vol. 2, 1969, pp. 123–141.

[38] Amirbayat, J., and Hearle, J. W. S., "Factors affecting the bond failure in unit composites," *Fibre Science and Technology*, vol. 3, 1970, pp. 147–155.

[39] Chon, C. T., and Sun, C. T., "Stress distributions along a short fibre in fibre reinforced plastics," *Journal of Materials Science*, vol. 15, 1980, pp. 931–938.

[40] Prasad, S., and Carlsson, L. A., "Debonding and crack kinking in foam core sandwich beams—II. Experimental investigation," *Engineering Fracture Mechanics*, vol. 47, 1994, pp. 825–841.

[41] Cantwell, W. J., and Davies, P., "A test technique for assessing core-skin adhesion in composite sandwich structures," *Journal of Materials Science Letters*, vol. 13, 1994, pp. 203–205.

[42] Cantwell, W. J., and Davies, P., "A study of skin-core adhesion in glass fibre reinforced sandwich materials," *Applied Composite Materials*, vol. 3, 1996, pp. 407–420.

[43] Cantwell, W. J., Scudamore, R., Ratcliffe, J., and Davies, P., "Interfacial fracture in sandwich laminates," *Composites Science and Technology*, vol. 59, Nov. 1999, pp. 2079–2085.

[44] Carlsson, L. A., "On the Design of the Cracked Sandwich Beam (CSB) Specimen," *Journal of Reinforced Plastics and Composites*, vol. 10, Jul. 1991, pp. 434–444.

[45] Zenkert, D., "Strength of sandwich beams with interface debondings," *Composite Structures*, vol. 17, Jan. 1991, pp. 331–350.

[46] Zenkert, D., "Poly(vinyl chloride) sandwich core materials: Fracture behaviour under mode II loading and mixed-mode conditions," *Materials Science and Engineering: A*, vol. 108, Feb. 1989, pp. 233–240.

[47] Zenkert, D., "Strength of sandwich beams with mid-plane debondings in the core," *Composite Structures*, vol. 15, Jan. 1990, pp. 279–299.

[48] Sundararaman, V., and Davidson, B. D., "An unsymmetric double cantilever beam test for interfacial fracture toughness determination," *International Journal of Solids and Structures*, vol. 34, Mar. 1997, pp. 799–817.

[49] Sundararaman, V., and Davidson, B. D., "An Unsymmetric End-Notched Flexure Test for Interfacial Fracture Toughness Determination," *Engineering Fracture Mechanics*, vol. 60, Jun. 1998, pp. 361–377.

[50] Davidson, P., Waas, A. M., and Yerramalli, C. S., "Experimental determination of validated, critical interfacial modes I and II energy release rates in a composite sandwich panel," *Composite Structures*, vol. 94, Jan. 2012, pp. 477–483.

[51] Zenkert, D., Shipsha, A., and Burman, M., "Fatigue of Closed Cell Foams," *Journal of Sandwich Structures and Materials*, vol. 8, Nov. 2006, pp. 517–538.

[52] Goyal, V., Rome, J., Schubel, P., and Tuck-Lee, J., "Strength Evaluation of Foam Core Sandwich Structures with Butt Joints," *25th Annual Technical Conference*, 2010.

[53]   Cytec, "AEROSPACE MATERIALS FM ® 300 Epoxy Film Adhesive," 2013.

[54]   Alfano, G., and Crisfield, M. A., "Finite element interface models for the delamination analysis of laminated composites: mechanical and computational issues," *International Journal for Numerical Methods in Engineering*, vol. 50, Mar. 2001, pp. 1701–1736.

[55]   Goyal, V. K., Johnson, E. R., and Dávila, C. G., "Irreversible constitutive law for modeling the delamination process using interfacial surface discontinuities," *Composite Structures*, vol. 65, Sep. 2004, pp. 289–305.

[56]   Xie, D., and Waas, A. M., "Discrete cohesive zone model for mixed-mode fracture using finite element analysis," *Engineering Fracture Mechanics*, vol. 73, Sep. 2006, pp. 1783–1796.

[57]   Wang, H., and Vu-Khanh, T., "Use of end-loaded-split (ELS) test to study stable fracture behaviour of composites under mode II loading," *Composite Structures*, vol. 36, Sep. 1996, pp. 71–79.

[58]   Heinrich, C., and Waas, A. M., "Investigation of progressive damage and fracture in laminated composites using the smeared crack approach," *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, vol. 35, 2012, pp. 155–181.

[59]   Kier, Z. T., Waas, A. M., Rome, J. I., Goyal, V. K., Schubel, P., Steckel, G., Patel, D., and Kim, Y., "Modeling Failure of 3D Fiber Reinforced Foam Core Sandwich Structures with Defects," *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2012, pp. 1–11.

[60]   Jones, R. M., *Mechanics of composite materials*, CRC press, 1998.

[61]   "D5528-01 2001. Standard Test Method for Mode I Interlaminar Fracture Toughness of Unidirectional Fiber-Reinforced Polymer Matrix Composites," *American Society for Testing and Materials*, 2014, pp. 1–13.

[62]   Martin Rinker, James G. Ratcliffe, Daniel O. Adams, R. K., "Characterizing Facesheet/Core Disbonding in Honeycomb Core Sandwich Structure," 2013, p. 39.

[63]   Kier, Z. T., Rome, J. I., Goyal, V. K., Schubel, P., Steckel, G., Patel, D., and Kim, Y., "Predicting Strength Reduction of Sandwich Structures with Interfacial Debonds (Presentation Only)," *25th American Society for Composites Conference*, Dayton, OH: 2010.

[64]   ASTM International, "ASTM C364 - Standard Test Method for Edgewise Compressive Strength of Sandwich Constructions," *Annual Book of ASTM Standards*, vol. 15.03, 1999, pp. 1–2.

[65]   3M Corporation, "3M Scotch-Weld Structural Adhesive Film AF 191," 2009, pp. 1–12.

[66]   Comiez, J. M., Waas, A. M., and Shahwan, K. W., "Delamination buckling; Experiment and analysis," *International Journal of Solids and Structures*, vol. 32, Mar. 1995, pp. 767–782.

[67]   Shahwan, K. W., and Waas, A. M., "A mechanical model for the buckling of unilaterally constrained rectangular plates," *International Journal of Solids and Structures*, vol. 31, Jan. 1994, pp. 75–87.

[68]   Student (William Sealy Gosset), "The Probable Error of a Mean," *Biometrika*, vol. 6, 1908, pp. 1–25.

[69]   Nuzzo, R., "Statistical errors," *Nature*, vol. 506, 2014, pp. 150–152.

[70]   ASTM International, "ASTM C297 Standard Test Method for Flatwise Tensile Strength of Sandwich Constructions," *ASTM International*, vol. 04, 2013, pp. 1–6.

[71]   Bažant, Z. P., and Oh, B. H., "Crack band theory for fracture of concrete," *Matériaux et Constructions*, vol. 16, 1983, pp. 155–177.

[72]   Rots, J. G., Nauta, P., Kusters, G. M. A., and Blaauwendraad, J., "Smeared crack approach and fracture localization in concrete," *Heron*, vol. 30, 1985.

[73]   Melcher, R. J., and Johnson, W. S., "Mode I fracture toughness of an adhesively bonded composite-composite joint in a cryogenic environment," *Composites Science and Technology*, vol. 67, 2007, pp. 501–506.

[74]   ASTM International, "ASTM C365-03: Standard test method for flatwise compressive properties of sandwich cores," *Annual book of ASTM standards*, 2011, pp. 1–8.

[75]   Kier, Z. T., Waas, A. M., Rome, J. I., and Goyal, V. K., "Specimen Size and Effective Compressive Stiffness of 3D Fiber Reinforced Foam Core Sandwich Structures," *28th Annual Technical Conference of the American Society for Composites*, Boston, MA: 2013.

[76]   Nanayakkara, a., Feih, S., and Mouritz, a. P., "Experimental analysis of the through-thickness compression properties of z-pinned sandwich composites," *Composites Part A: Applied Science and Manufacturing*, vol. 42, 2011, pp. 1673–1680.

[77]   ASTM International, "ASTM C393/C 393 M - 06 Standard Test Method for Core Shear Properties of Sandwich Constructions by Beam," *Annual Book of ASTM Standards*, vol. I, 2009, pp. 1–8.

[78]   Evonik, "Rohacell WF Techical Data Sheet," 2012.

[79]   Kier, Z. T., and Waas, A. M., "Determining Effective Interface Fracture Properties of 3D Fiber Reinforced Foam Core Sandwich Structures," *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2013, pp. 1–9.

[80]   Kier, Z. T., Waas, A. M., Rome, J. I., Goyal, V. K., Patel, D., and Steckel, G., "Through-thickness Failure of 3D Fiber Reinforced Foam Core Sandwich Structures," *SAMPE Journal*, vol. 50, 2014, pp. 32–38.

[81]   Daniel, I. M., Ishai, O., Daniel, I. M., and Daniel, I., *Engineering mechanics of composite materials*, Oxford university press New York, 1994.

[82]   Arezoo, S., Tagarielli, V. L., Siviour, C. R., and Petrinic, N., "Compressive deformation of Rohacell foams: Effects of strain rate and temperature," *International Journal of Impact Engineering*, vol. 51, Jan. 2013, pp. 50–57.

[83]   Gibson, L. J., and Ashby, M. F., *Cellular solids: structure and properties*, Cambridge university press, 1999.

[84]   Richeton, J., Ahzi, S., Vecchio, K. S., Jiang, F. C., and Adharapurapu, R. R., "Influence of temperature and strain rate on the mechanical behavior of three amorphous polymers: Characterization and modeling of the compressive yield stress," *International Journal of Solids and Structures*, vol. 43, Apr. 2006, pp. 2318–2335.

[85]   Rinde, J. A., and Hoge, K. G., "Time and temperature dependence of the mechanical properties of polystyrene bead foam," *Journal of Applied Polymer Science*, vol. 15, Jun. 1971, pp. 1377–1395.