

MULTISCALE MODELING OF COMPOSITE LAMINATES WITH FREE EDGE EFFECTS

By

Christopher R. Cater

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Mechanical Engineering – Doctor of Philosophy

2015

ABSTRACT

MULTISCALE MODELING OF COMPOSITE LAMINATES WITH FREE EDGE EFFECTS

By

Christopher R. Cater

Composite materials are complex structures comprised of several length scales. In composite laminates, the mechanical and thermal property mismatch between plies of varying orientations results in stress gradients at the free edges of the composites. These free edge stresses can cause initial micro-cracking during manufacture, and are a significant driver of delamination failure. While the phenomenon of free edge stresses have been studied extensively at the lamina level, less attention has been focused on the influence of the microstructure on initial cracking and development of progressive damage as a consequence of free edge stresses. This work aimed at revisiting the laminate free edge problem by developing a multiscale approach to investigate the effect of the interlaminar microstructure on free edge cracking.

First, a semi-concurrent multiscale modelling approach was developed within the commercial finite element software ABAQUS. An energetically consistent method for implementing free edge boundary conditions within a Computational Homogenization scheme was proposed to allow for micro-scale free edge analysis. The multiscale approach was demonstrated in 2D tests cases for randomly spaced representative volume elements of unidirectional lamina under tensile loading.

Second, a 3D multiscale analysis of a $[25_N/-25_N/90_N]_S$ composite laminate, known for its vulnerability to free edge cracking, was performed using a two-scale approach: the meso-scale model captured the lamina stacking sequence and laminate loading conditions (mechanical and thermal) and the micro-scale model predicted the local matrix level stresses at the free edge. A

one-way coupling between the meso- and micro-scales was enforced through a strain based localization rule, mapping meso-scale strains into displacement boundary conditions onto the micro-scale finite element model. The multiscale analysis procedure was used to investigate the local interlaminar microstructure. The results found that a matrix rich interlaminar interface exhibited the highest free edge stresses in the matrix constituent during thermal cooldown. The results from these investigations assisted in understanding the tendency for pre-cracks during manufacture to occur at ply boundaries at the free edge and the preferential orientation to the ply interfaces. Additionally, analysis of various 90/90 ply interfaces in the thicker $N=3$ laminate found that the free edge stresses were far more sensitive to the local interlaminar microstructure than the meso-scale stress/strain free edge gradients. The multiscale analysis helped explain the relative insensitivity of free edge pre-cracks to progressive damage during extensional loading observed in experiments.

Lastly, the multiscale analysis was extended to the interface between the -25 and 90 degree plies in the $[25_N/-25_N/90_N]_S$ laminate. A micro-model representing the dissimilar ply interface was developed, and the homogenized properties through linear perturbation steps were used to update the meso-scale analysis to model the interlaminar region as a unique material. The analysis of micro-scale free edge stresses found that significant matrix stresses only occurred at the fiber/matrix boundary at the 90 degree fibers. The highest stresses were located near the matrix rich interface for both thermal and mechanical loading conditions. The highest matrix stresses in the case of extensional loading of the laminate, however, were found at the interior of the micro-model dissimilar ply micro-model within the -25 degree fibers.

ACKNOWLEDGEMENTS

I would first like to thank my advisor and mentor, Dr. Xinran Xiao, who was instrumental in encouraging and supporting me through this entire research endeavor. She provided me with many opportunities to learn and grow, both in research and in life. I would also like to acknowledge the support and assistance from Dr. Robert K. Goldberg of the NASA Glenn Research Center in accepting a co-advisory role, mentoring me throughout my summer internships and through the course of my research. His involvement during my graduate studies has helped shape my perspectives and passion for research.

I would also like to thank the members of my advisory committee, Dr. Andre Benard, Dr. Rigoberto Burgueno, and Dr. Dashin Liu for their support and advice. Additionally, a special thanks is in order to the many graduate students and colleagues at the Composite Vehicle Research Center at MSU and colleagues at the NASA Glenn Research Center for providing their valuable feedback and input. I would also like to thank the financial support provided by the Cooperative Agreement No. W56HZV-07-2-0001 between U.S. Army TACOM LCMC and Michigan State University as well as the NASA NRA NNX12AL14A.

Lastly, I would like to thank my family from the bottom of my heart for their sacrifice, support and belief in my education, my wife Vida for her patience and faith, and my brother Charlie for his constant inspiration.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
1. Introduction	1
1.1 Composite Materials.....	1
1.2 Free Edge Effects in Composite Laminates.....	2
1.3 Free Edge and the Microstructure	4
1.4 Objectives & Scope of Work	7
2. Multiscale Modeling	8
2.1 Multiscale Modeling: State of the Art	8
2.2 Modeling the Microscale Stresses at the Free Edge.....	14
2.3 Summary	15
3. Meso-to-Microscale Multiscale Framework	16
3.1. Semi-concurrent Approach	16
3.2. Development of a Free Edge Multiscale Boundary Condition	18
3.3. An energetically consistent approach.....	19
4. Free Edge Boundary at the Microscale	26
4.1 ABAQUS Semi-concurrent Implementation	26
4.2 2D Model Verification	32
4.2.1 RVE Generation	32
4.2.2 Cohesive Zone Modeling	35
4.2.3 Results and Conclusions	38
4.3 Conclusions.....	47
5. Free Edge Analysis of the Interlaminar Microstructure	49
5.1 Overview.....	49
5.2 Scale Transitions	51
5.2.1 Kinematic Coupling	51
5.2.2 Validation of Strain Based Localization	54
5.2.3. Micro-scale Boundary Conditions	57
5.2 Meso-scale Model	60
5.3 Micro-scale Model	63
5.4 Model Parameters.....	64
5.5 Finite Element Results.....	67
5.5.1 Effect of the Interlaminar Thickness during Thermal Cooldown	67
5.5.2 Influence of Meso-Scale Strains on Free Edge Micro-stresses	74
5.5.3 Effect of Pure Mechanical Loading	79
5.6. Discussion.....	85

6. Free Edge Analysis of the Dissimilar Ply Microstructure	87
6.1 Model Development	87
6.1.1 Meso-scale Model.....	89
6.1.2. Micro-scale Model.....	90
6.1.3 Periodic Boundary Conditions Implementation	93
6.1.4 Homogenized Properties of the Dissimilar Ply Interface.....	95
6.2 Finite Element Results.....	99
6.2.1 Dissimilar Ply Interface (-25/90) during Thermal Cooldown.....	100
6.2.2 Dissimilar Ply Interface (-25/90) during Mechanical Loading	103
6.3 Discussion.....	106
7. Conclusions and Future Work.....	108
7.1 Free edges in a semi-concurrent scheme	108
7.2 Multiscale analysis of the 90/90 ply interface	108
7.3 Multiscale analysis of the -25/90 interface.....	109
7.4 Future work.....	110
APPENDICES.....	111
A.1 UMAT subroutine for semi-concurrent scheme	112
A.2 Python Script for RVE Generation.....	115
A.3 Python Script for standard computational homogenization.....	123
A.4 Python Script for z-scalar approach within ABAQUS	132
A.5 3D Periodic linear constraint equations for the micro-scale model	146
A.6 3D Free edge analysis linear constraint equations	148
BIBLIOGRAPHY	152

LIST OF TABLES

Table 4.1:	Computed elastic constants for periodic and non-periodic perturbation steps.....	39
Table 4.2:	Constituent properties for the multi-fiber RVE verification study.....	40
Table 5.1:	The four nodal displacements for the two prescribed test cases used in the single element localization test. The distance d is the length of edge AB of the square RVE shown in Figure 5.3. Nodes B and D in the single element were fully prescribed, node A was fixed and node C followed periodic constraints.....	55
Table 5.2:	Constituent and homogenized lamina mechanical and thermal properties. The constituent properties were obtained from Dustin and Pipes [76], while the lamina properties were obtained using the MAC/GMC micromechanics software.	65
Table 6.1:	The symmetric material stiffness matrix for the -25/90 micro-model in units of MPa	98
Table 6.2:	The anisotropic thermal expansion coefficients	99

LIST OF FIGURES

Figure 1.1:	Diagram of the various length scales associated with a composite structure.....	2
Figure 1.2:	Free edge interlaminar stresses in a [0/90] _s composite laminate subjected to uniaxial tension. The above example is a standard crossply laminate used for comparing various free edge analysis techniques.	3
Figure 1.3:	Microscopic images of the free edge of a [255/-255/905] _S with pre-cracks identified at the ply interfaces via color coding. Green regions indicated cracks at the 90/90 interface, red regions are cracks in the -25/25 interface, and the yellow regions are cracks at the -25/90 interface. The zoom inset shows an example of a micro-crack present in the 90° plies. Image reproduced from Dustin [31].	5
Figure 2.1:	Schematic of multiscale classifications.....	8
Figure 3.1:	Overview of the semi-concurrent workflow and passing of information between the macroscale (global) and microscale (RVE/local).	17
Figure 3.2:	Schematic of the RVE deformation as a superposition of macroscale and microscale influences.	21
Figure 3.3:	Illustration of the localization process, which has deformed the RVE to produce a unique amount of elastic strain energy defined by the box on the right.	22
Figure 3.4:	Stress-strain diagrams highlighting the elastic strain energy for two non-linear material responses where (a) the material is purely elastic and (b) the material has non-linearity due to elastic softening.	23
Figure 3.5:	The total amount of elastic strain energy at the macroscopic level for the 2D plane strain problem is a summation of the elastic strain energy resulting from the three degrees of freedom.....	23
Figure 4.1:	Workflow of the semi-concurrent computational homogenization scheme.....	26
Figure 4.2:	Schematic of (a) the test boundary conditions, which include the free edge, and (b) the standard periodic boundary conditions. Note: That P, F, S in the above diagram represent periodic, free and symmetric edge, respectively.	27
Figure 4.3:	Listing of the database post-processing steps performed in the custom Python script.	28
Figure 4.4:	The Jacobian matrix, J, is shown with the necessary uni-strain components for determining a particular column. The circled columns are computed via uni-strain perturbation steps where one of the three strain components are set to 1, and the remainder are zero. The condition for computing each column is labeled	

accordingly. The 33 component is determined externally. Symmetry of the Jacobian is assumed.	29
Figure 4.5: Boundary conditions used for the perturbation steps during non-periodic analysis. The right-hand-side BC, labeled U, represents uniform displacement as prescribed using Equation (4.1).....	30
Figure 4.6: Steps for post-processing the Jacobian.	30
Figure 4.7: Schematic of the overall semi-concurrent scheme as implemented into ABAQUS for a single increment at a given macroscopic integration point.....	31
Figure 4.8: Example of a randomly placed (a) 4-fiber RVE with fibers allowed to exit the boundary and (b) an RVE with fibers intersecting and exiting the boundary. (c) Definition of the intersection distance, d.	34
Figure 4.9: Workflow of the RVE generation script for high fiber volume fractions.....	35
Figure 4.10: Example traction-separation law (ABAQUS 6.11 Analysis User's Manual).....	36
Figure 4.11: Cohesive elements used in the (a) uniaxial tension simulation on a quarter fiber/matrix RVE and (b) on a Mode I delamination example.....	38
Figure 4.12: Energy history variable outputs for the 16 fiber RVE under periodic boundary conditions subject to uniaxial tensile loading.....	41
Figure 4.13: Percent difference in external work between the macro and micro scale work for the 16 fiber RVE for periodic and non-periodic boundary conditions.	42
Figure 4.14: Specific elastic strain energy vs macroscopic strain for the 4-fiber RVE. Dashed lines represent the periodic results, solid lines those of the free edge simulations, and the color coding corresponds to a given RVE iteration.....	44
Figure 4.15: Macroscopic (2-direction) stress vs strain for the 4-fiber RVE.	45
Figure 4.16: Macroscopic (2-direction) stress vs strain for the 16-fiber RVE.....	46
Figure 4.17: Macroscopic (22 direction) stress vs strain for the 36-fiber RVE.	47
Figure 5.1: Workflow of the multiscale analysis of a composite laminate showing relevant length scales. The current approach utilizes meso-scale and micro-scale finite element models which are one-way coupled as shown by the dashed arrow. Deformations from the meso-scale are localized into boundary conditions on the micro-scale finite element model.....	50
Figure 5.2: The definition of normal and shear displacements prescribed on a 2D RVE. The black dotted line represents tensile deformation prescribed by extensions δ_1 and δ_2 while the red dotted line shows a simple shear deformation according to δ_s	52

Figure 5.3:	Workflow of a 2D single element study to determine the efficacy of strain based localization methods.	54
Figure 5.4:	Comparison of resulting strain LEXX* for a variety of localization rules against the prescribed strain LEXX for (a) combined loading and (b) an applied shear.	56
Figure 5.5:	The applied boundary conditions for the free edge micro-scale analysis. The model is assumed periodic in the y and z directions. In the x-direction, the negative x face is assumed to be a free surface, while the positive x face is prescribed a unique Periodic* Dirichlet boundary conditions based on a periodic solution. The vertices of the micro-model are labeled A-D and O-R, which appear as superscripts in the displacement relations of Equations (8).	57
Figure 5.6:	Process flow for the application of the Periodic* BC.	58
Figure 5.7:	A portion of the meso-scale finite element model for the case of N=1 is shown. The meso-scale model is color coded to identify the unidirectional ply orientations. The meso-scale model is truncated in the figure. The micro-scale model associated with the midplane interface element is shown at the right.	61
Figure 5.8:	The meso-scale finite element model for the N=3 laminate. The meso-scale model is color coded to identify the unidirectional ply orientations. The meso-scale model is truncated in the figure.	62
Figure 5.9:	(a) 55% fiber volume fraction micro-model, (b) 48% fiber volume fraction micro-model and (b) 44% fiber volume fraction micro-model.	64
Figure 5.10:	The contour results (units of MPa) are shown for the residual free edge stresses in the y-direction for the N=1 laminate and 55% Vf micro-model. The logarithmic strain components extracted from the midplane 90/90 interface element are shown on the right for all three interlaminar thicknesses. Due to the thermal contraction, the strains are negative, however the stresses are positive in the laminate. The 1, 2 and 3 component directions correspond to the global x, y and z directions.	69
Figure 5.11:	The contour results are shown for the residual free edge maximum principal stresses in the matrix elements (fibers are hidden) for the 55% micro-model. Elements whose maximum principal stresses have exceeded 99MPa are highlighted in red.	70
Figure 5.12:	The contour results are shown for the residual free edge maximum principal stresses in the matrix elements (fibers are hidden) for all three interface micro-models. Elements whose maximum principal stresses have exceeded 99MPa are highlighted in red. The face of the micro-model shown is at the free edge of the laminate.	71
Figure 5.13:	Contour plots are shown for the residual free edge maximum principal stresses in the matrix elements (fibers are hidden) for all three interface micro-models. The contour legend is shown (units of Pa). The contour plots utilize an upper and lower limit, set at 99MPa and 0 MPa, respectively, to capture the variation of maximum principal stresses within the entire micro-model. Values above or below the	

specified limits are colored red or blue, respectively. The face of the micro-model shown is at the free edge of the laminate.	72
Figure 5.14: The contour results are shown for the residual free edge maximum principal stresses in the matrix elements (fibers are hidden) for the 48% micro-model. Elements whose maximum principal stresses have exceeded 182MPa are highlighted in red.	73
Figure 5.14: The logarithmic strain components at the meso-scale for the N=3 laminate at the three 90/90 ply interfaces. The 1, 2 and 3 component directions correspond to the global x, y and z directions.....	75
Figure 5.15: The contour results are shown for the residual free edge maximum principal stresses in the matrix elements (fibers are hidden) for all three ply interfaces for the 55% fiber volume fraction micro-model. Elements whose maximum principal stresses have exceeded 99MPa are highlighted in red. The face of the micro-model shown is at the free edge of the laminate.....	76
Figure 5.16: Contour plots are shown for the residual free edge maximum principal stresses in the matrix elements (fibers are hidden) for the 55% Vf micro-model at the three 90/90 pl interfaces. The contour legend is shown (units of Pa). The contour plots utilize an upper and lower limit, set at 99MPa and 0 MPa, respectively, to capture the variation of maximum principal stresses within the entire micro-model. Values above or below the specified limits are colored red or blue, respectively. The face of the micro-model shown is at the free edge of the laminate.....	77
Figure 5.17: The contour results are shown for the residual free edge maximum principal stresses in the matrix elements (fibers are hidden) for all three ply interfaces for the 48% fiber volume fraction micro-model. Elements whose maximum principal stresses have exceeded 99MPa are highlighted in red. The face of the micro-model shown is at the free edge of the laminate.....	78
Figure 5.18: The logarithmic strain components at the meso-scale for the N=3 laminate at the three 90/90 ply interfaces as a result of a mechanical extension of 0.1% strain. The 1, 2 and 3 component directions correspond to the global x, y and z directions.....	80
Figure 5.19: The contour results are shown for the free edge maximum principal stresses as a result of a 0.3% mechanical extension in the z-direction for the 55% interface micro-model. Elements whose maximum principal stresses have exceeded 99MPa are highlighted in red. The face of the micro-model shown is at the free edge of the laminate.....	81
Figure 5.20: Contour plots are shown for the free edge maximum principal stresses in the matrix elements (fibers are hidden) for the 55% Vf micro-model at the three 90/90 pl interfaces. The contour legend is shown (units of Pa). The contour plots utilize an upper and lower limit, set at 99MPa and 0 MPa, respectively, to capture the variation of maximum principal stresses within the entire micro-model. Values	

	above or below the specified limits are colored red or blue, respectively. The face of the micro-model shown is at the free edge of the laminate.....	82
Figure 5.21:	The contour results are shown for the free edge maximum principal stresses as a result of a 0.3% mechanical extension in the z-direction for the 48% interface micro-model. Elements whose maximum principal stresses have exceeded 99MPa are highlighted in red. The face of the micro-model shown is at the free edge of the laminate.....	83
Figure 5.22:	The contour results are shown for the free edge maximum principal stresses (MPa) as a result of a 0.3% mechanical extension in the z-direction for the 48% interface micro-model.	84
Figure 6.1	Micro-scale model for the dissimilar ply interface showing the fiber (green) and matrix (white) geometries.	88
Figure 6.2	Meso-scale model for the dissimilar ply interface analysis. The color coding represents the two different material definitions used for the standard unidirectional lamina (beige) and the interface (-25/90) material (green).	90
Figure 6.3:	Micro-models for the -25/90 interlaminar (a) Regular interface and (b) Thick interface. The micro-model is oriented so that the free edge is facing out of the page.	91
Figure 6.4:	The x and z faces of the Regular dissimilar interface micro-model, highlighting the periodic in the x and z direction. The opposing faces will appear as mirror images due to reflection, however, they are periodic when faced in the proper orientation.	92
Figure 6.5:	The Regular interface micro-model with the C3D4T tetrahedral elements. The free-form mesh resulted in irregular spacing of nodes on periodic faces (i.e. different node locations and number of nodes on corresponding periodic faces)	93
Figure 6.6:	(a) An example of non-periodic nodes on the negative and positive x faces. A surface is defined using the red nodes on the negative x face, shown in purple. (b) An example of periodic nodes on the negative and positive x face. A surface is defined using the green nodes on the negative x face, shown in yellow. The red nodes in (a) represent the actual nodes on the micro-model face, while the green nodes in (b) represent duplicate nodes generated to create a periodic set to blue nodes. The blue and green nodes are prescribed periodic constraints, while the two surfaces (yellow and purple) are tied using surface-to-surface constraints.	94
Figure 6.7:	Schematic of the six degrees of freedom shown as possible displacements by red dashed arrows. The displacements are prescribed for the three independent nodes in the micro-model shown in green.	96
Figure 6.8:	An example of the determination of the material stiffness matrix through unit, uni-strain perturbation steps. In the above example, unit-strain is applied in the 11 direction and all other strains are prescribed to be zero. The resulting volume	

averaged stresses of the micro-model, shown at the left hand side, are the necessary entries of a column of the stiffness matrix, highlighted above.....97

Figure 6.9: a) A stress contour plot (units of MPa) of the normal (y-stress) for the dissimilar ply interface model under thermal load of -155°C is shown, where a cut-out of the right-hand side (X-) of the laminate free edge is shown. b) The logarithmic strain components extracted from the dissimilar interface element are shown. Note that the anisotropic nature of the dissimilar ply interface causes strains in all three shearing material directions..... 100

Figure 6.10: Stress contour plot of maximum principal stress (Pa) for the two dissimilar ply interface micro-models. The location of the highest maximum principal stress is indicated by a black arrow. The highest maximum principal stresses occurred at the 90° fiber boundary facing the ply interface. It should be noted that only around the 90° fibers did the matrix maximum principal stress exceed the 99MPa tensile strength of the matrix. 102

Figure 6.11: a) A stress contour plot (units of MPa) of the normal (y-stress) for the dissimilar ply interface model under 0.3% strain tensile extension in z-direction, where a cut-out of the right-hand side (X-) of the laminate free edge is shown. b) The logarithmic strain components extracted from the dissimilar interface element are shown..... 103

Figure 6.12: Stress contour plot of maximum principal stress (Pa) for the two dissimilar ply interface micro-models. The location of the highest maximum principal stress is indicated by a black arrow. The highest reported values were found to be located at the interior of the micro-model between proximal -25° fibers..... 105

1. Introduction

1.1 Composite Materials

Beginning with its introduction to aerospace applications in the 1960's [1], composites, particularly Carbon-Fiber-Reinforced Polymer (CFRP) composites, have seen significant growth in recent decades in industrial and consumer transportation, wind energy applications and sporting goods. These composite materials have the benefit of high specific strength and stiffness, as well as increased fatigue resistance over traditional materials [2,3]. They are also hierarchical in nature, containing a variety of length scales which factor into the overall properties of the composite [4]. The various length scales associated with CFRP composites can be classified into the following:

- *Macro-scale*: The length scale associated with the overall structure which contains the loading and boundary conditions of the global test and/or analysis.
- *Meso-scale*: An intermediary scale associated with particular reinforcement architectures such as the layup configuration for laminated composites or the braid/weave pattern for textile based composites.
- *Micro-scale*: The length scale associated with the fiber and matrix level constituents of a unidirectional (UD) lamina or an individual fiber tow.

As can be seen in Figure 1.1, the micro-scale constituents, or their relative volume fractions, could be adjusted along with a chosen reinforcement architecture (laminated, woven, braided) to obtain a composite best fit for the intended application. A consequence of the hierarchical nature of CFRP composites, however, is an increased complexity in modeling due to

damage evolving at a variety of length scales [5,6]. Thus, a multiscale approach is required to address the problem. Additionally, the heterogeneity of CFRP composites, particularly laminated composites, introduces unique sources of failure associated with free edges, discussed in the section to follow for laminated composites.

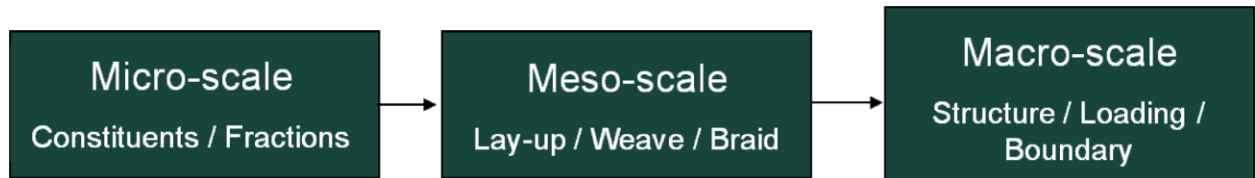


Figure 1.1: Diagram of the various length scales associated with a composite structure.

1.2 Free Edge Effects in Composite Laminates

The free edge problem in laminated composites refers to the stress gradients that develop at the intersection of the interface between unidirectional (UD) lamina of varying orientation and a free edge [7]. Figure 1.2 describes the state of interlaminar stresses which develop at the free edge of a composite laminate under uniaxial tension. The specific example shows a simple case of a $[0/90]_S$ laminate. The free edge stresses include an interlaminar shear stress shown in Detail A, as well as a through-thickness opening stress shown in Detail B. The source of these stresses is the mismatch in material properties at the interface and can arise during the application of mechanical and/or thermal loading [9]. These interlaminar stresses may play a crucial role in the determination of laminate strength and can be sources of delamination [7,10–13].

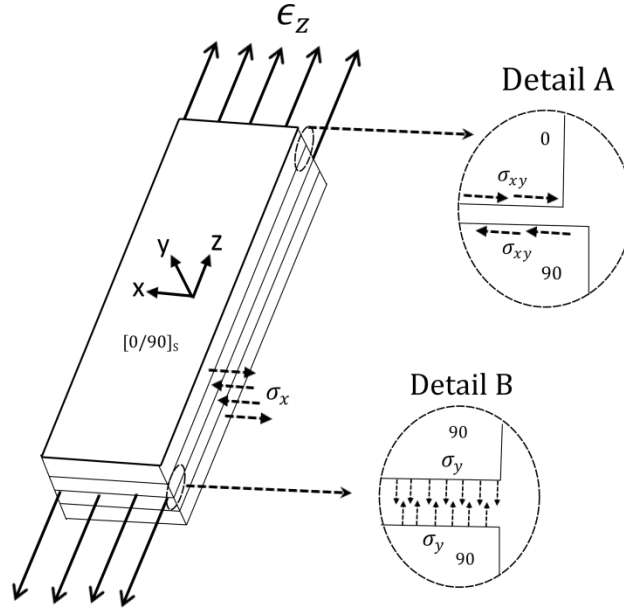


Figure 1.2: Free edge interlaminar stresses in a [0/90]_s composite laminate subjected to uniaxial tension. The above example is a standard crossply laminate used for comparing various free edge analysis techniques.

The free edge stresses have long been studied in available literature. These span from approximate closed-form solutions [7,14,15], to 2D generalized plane strain analysis using Finite Elements [9,16], and to advanced higher-order generalized laminate theories to capture the interlaminar stresses in a 2D domain space [17–25]. A more comprehensive review can be found by Mittelstedt and Becker [8]. Recent work has focused on developing three-dimensional models of the laminate free edge problem utilizing submodeling techniques [26] or new finite element approaches [27]. While the previous research has been successful in characterizing the nature of the free edge stresses with respect to changes in lamina orientation, geometry, and loading conditions, these analyses have been restricted to the meso-scale domain similar to the schematic of Figure 1.2. As previously mentioned, each lamina is considered to be a homogenous,

orthotropic structure where a discrete interface exists between the varying layers. The result of this material approximation is the singular nature of the through-thickness free edge stress fields.

1.3 Free Edge and the Microstructure

These free edge stresses are a phenomenon associated with the continuum, meso-scale based approach whereby a discrete interface exists at the lamina interface [8,16]. Although stress singularities are present when modeling at the meso-scale, the resolution of micro-scale features would reveal a very different stress state where singularities exist at the intersection of the free edge and the fiber-matrix interface [8]. Dustin and Pipes [29] compared the stress singularity associated with lamina interfaces at the free edge (meso-scale) to the stress singularity associated with fiber termination at the free edge (micro-scale), reporting that fiber termination may play a larger role in failure initiation. Furthermore, it was found that a crack-tip singularity at the free edge was roughly 1.5 times greater than that of the fiber termination, highlighting the importance of micro-cracks at the free edges. Fiber/matrix interface cracking were also found to be an initiating damage mechanism of inter-ply cracking in $[+15/-15]_s$ laminates [28]. Other micro-scale features that influence damage initiation were found to be local matrix distribution at the lamina interfaces [28,30]. For example, interface thickness [28] as well as the uneven interfaces in quasi-unidirectional plies [30] were found to effect the local distribution of free edge stress concentrations. Additionally, inelastic strains developed at the lamina interfaces prior to failure and were a result of observable micro-cracking [30].

Dustin performed a thorough experimental investigation of free edge micro-cracking in $[25_N/-25_N/90_N]_s$ IM7/8552 composite laminates which were highly susceptible to free edge stresses [31]. Numerous cracks were present after manufacturing and were measured and tracked during the application of extensional loading. For $N=1, 3$ and 5 , it was found that cracks tended

to occur primarily at the interface of the unidirectional plies. As shown in Figure 1.3, micro-cracks were observed between the 90/90 (green), -25/90 (yellow) and -25/+25 (red) interfaces in the N=5 laminate. From the experimental observations, it was concluded that the primary failure was typically delamination at the -25/90 interface initiated from transverse cracking in the 90 degree plies. It was also reported that the majority of micro-cracks present did not play a significant role in the delamination failure of the composite and only large micro-cracks (greater than 50 fiber diameters) influenced laminate failure. A clear relationship, however, was found between the presence of cracks and the location of ply interfaces. Around 90% of cracks in the 90-degree plies occurred within a distance of 30% of the laminate thickness from a ply interface. These 90/90 ply pre-cracks were generally larger than the cracks in the dissimilar ply boundary [31].

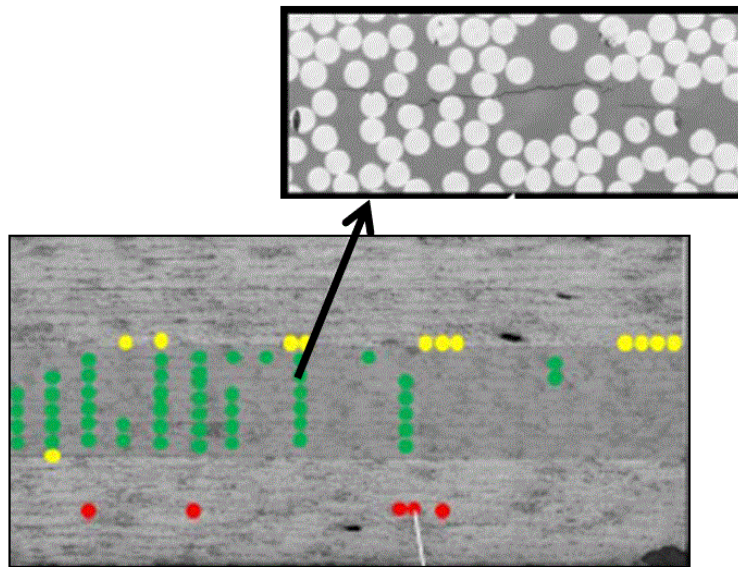


Figure 1.3: Microscopic images of the free edge of a $[25_s/-25_s/90_s]_s$ with pre-cracks identified at the ply interfaces via color coding. Green regions indicated cracks at the 90/90 interface, red regions are cracks in the -25/25 interface, and the yellow regions are cracks at the -25/90 interface. The zoom inset shows an example of a micro-crack present in the 90° plies. Image reproduced from Dustin [31].

Fiber/matrix interface cracking was also found to be an initiating damage mechanism in the inter-ply cracking of $[+15/-15]_S$ carbon fiber laminates [28]. Other micro-scale features that influence damage initiation were found to be local matrix distribution at the lamina interfaces [28,30]. For example, interface thickness was found to affect the local distribution of observed meso-scale displacement gradients at the free-edge [28,30]. Regions between plies with a smaller interface were also found to be more susceptible to micro-cracking during extensional loading. Additionally, inelastic strains developed at the lamina interfaces prior to failure and were a result of observable micro-cracking [30].

These largely experimental works have highlighted the presence of cracks at the free edge of composites and have suggested they play a role in free edge initiated damage. The observations of cracking at the free edge of the $[25_N/-25_N/90_N]_S$ laminates performed by Dustin [31] did not directly address questions regarding the nature of free edge cracking during manufacturing and subsequent extensional loading. This body of research is intended to reexamine the composite laminate free edge and develop a modeling methodology to answer the following questions:

- What is the influence of the microstructure (local fiber volume fraction and fiber spacing) on free edge matrix cracking?
- What causes the density of cracks to be found at interlaminar boundaries?
- Why is progressive failure in the $[25/-25/90]_S$ laminate under extensional loading not affected by the pre-cracks formed during manufacture?

1.4 Objectives & Scope of Work

This research revisits the problem of free edge effects in composite laminates using a combined multiscale modeling and computational micromechanics approach. The objective of this work is to develop analysis tools to further understand the influence of the local microstructure on free edge cracking and damage progression in composite laminates. First, a comprehensive review of multiscale modeling strategies and micro-scale free edge stress analysis is presented in Chapter 2. Second, an overview of semi-concurrent multiscale modeling and introduction to a proposed energetically consistent approach for implementing free edge boundary conditions at the microscale are discussed in Chapter 3. The implementation of a two-dimensional, semi-concurrent analysis using the proposed free edge multiscale analysis in the commercial finite element software ABAQUS is given in Chapter 4, along with a validation test case. In Chapter 5, a three-dimensional multiscale free edge analysis is developed and used to study the micro-scale stresses on $[25_N/-25_N/90_N]_S$ laminates under both thermal and mechanical loading. The analysis specifically explores the effect of the local interlaminar microstructure on the propensity for crack initiation. Chapter 6 extends the multiscale approach to the dissimilar ply interface between the -25° and 90° plies. Finally, a summary of this research and proposed future work are presented in Chapter 7.

2. Multiscale Modeling

The hierarchical nature of composite will require multiscale modeling to understand the influence of the microstructure at the laminate free edge. A review of literature on the state of the art for multiscale modeling strategies and a summary of previous works modeling the microstructure at the free edge will be presented. At the end of this chapter, the proposed multiscale framework and modeling objectives will be outlined.

2.1 Multiscale Modeling: State of the Art

There are three major classifications to multiscale modeling as outlined by Belytschko and Song [32]. The three approaches are categorized according to the means by which the various length scales are linked. They are sequential, concurrent and semi-concurrent strategies, and are schematically diagramed in Figure 2.1. In the figure, local scale is represented by an RVE which characterizes the local heterogeneity of a material point [33,34]. A review of the three multiscale classifications is discussed.

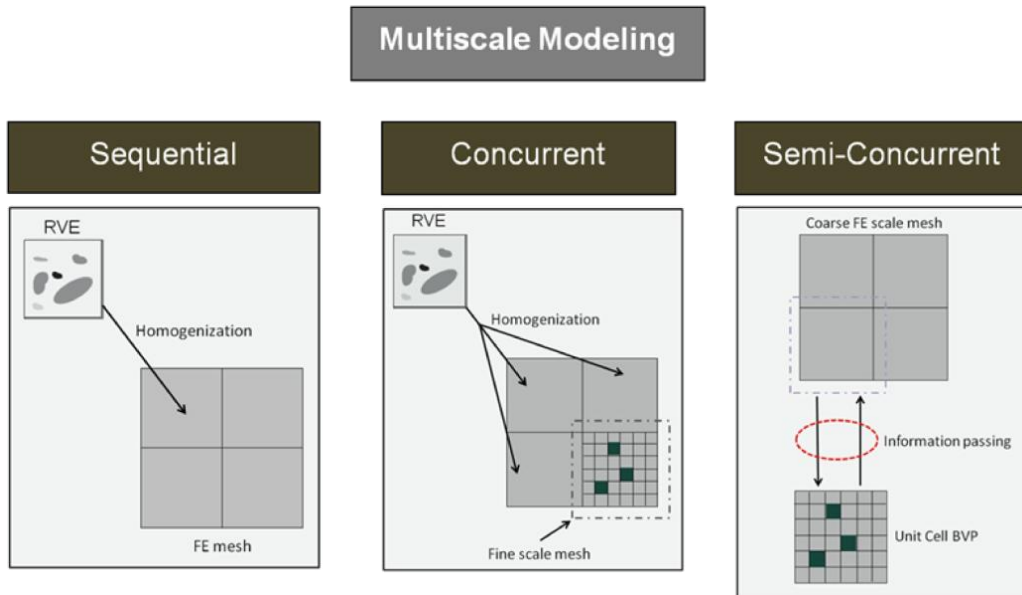


Figure 2.1: Schematic of multiscale classifications.

In a sequential approach, the micro-scale is modeled using an RVE, or unit cell, and is homogenized to determine the effective properties for use in a meso or macro-scale analysis. A large body of work exists for determining the effective properties (stiffness) of these composite structures through the use of analytical micromechanics [35–40] and other semi-analytical, numerical and finite element (FE) approaches [39,41–47]. Analytical methods of homogenization include the “rule of mixtures” mechanics of materials approach, the Self-Consistent Field Method [36], Bounding Methods developed by Voigt and Reuss, and Semi-empirical models employed by Halpin and Tsai[40]. A popular extension of the self-consistent methods is the well-known Mori-Tanaka homogenization [38]. These micromechanics models provide the necessary input to perform structural FE analysis based on the micro-scale properties using a “bottom-up” approach.

More recently, numerical methods have been employed such as virtual testing which can provide macro-scale properties using finite element (FE) simulations of an appropriate RVE [43,48,49]. In these methods, information from the lower, micro-scale is lost after the homogenization. Similar methods can be utilized to homogenize meso-scale features into effective properties at the macro-scale [45]. These models, however, do not preserve micro-structural information post-homogenization and only provide initial elastic properties. Although some methods can be used to recapture micro-stress fields [50,51], they do not provide a relation between evolving micro-scale fields and the macro-scale behavior.

It should also be mentioned that extrapolating the homogenized response to non-linear regions or determining composite failure will rely heavily on experimental tests to characterize material failure. While a variety of failure criteria exist at the homogenized scale of the lamina, or even the full laminate, the World Wide Failure Exercises (WWFE) have shown that no

available models successfully capture unidirectional (UD) ply failure under complex multi-axial loading states or for multi-ply laminates [52,53].

The second classification of multiscale frameworks is concurrent modeling approaches. Concurrent models, shown schematically in Figure 2.1, consist of a body of work where “bottom-up” homogenization is combined with Direct Numerical Simulation (DNS). In this way, the microstructure is directly inserted into the macro or meso-scale problem through the use of transitional elements or well-defined kinematic relations between regions of varying element sizes or types [54]. Ghosh had developed sophisticated concurrent schemes using the Voronoi Cell Finite Element Method (VCFEM) at the finest scale [55,56]. Due to the direct coupling of discrete FE RVE’s with globally homogenized elements, this methodology requires both adaptive re-meshing algorithms as well as modified transitional elements to deal with the coupling of the scale interfaces [57].

These methods present significant reduction in some computational cost with respect to DNS of transient analysis and/or component sized simulations but the increased complexity of the global stiffness matrices and strong scale coupling make it not suitable for large structural analyses [58]. Aside from being computationally expensive, concurrent models are similar to sequential approaches as they also do not provide a means of linking micro-scale behavior to macro-scale response. Thus, utilizing a concurrent approach beyond highly localized damage would require large regions of micro-structural refinement.

The last classification in multiscale frameworks is semi-concurrent modeling. Semi-concurrent modeling methods are schemes which rely on accessing information from the micro-structural domain at each increment within the FE analysis [32]. Thus, the kinematic and

constitutive relations between the scales act as a material model to provide the global homogenized response. In comparison to the “bottom-up” approach, these semi-concurrent schemes allow for the retention of micro-structural information throughout the analysis, and the two lengths scales are weakly coupled throughout the analysis. First attempts to include micromechanics in the global response were done by coupling the RVE scale response to the global response analytically. For example, mathematical derivations for homogenization were developed by Bensoussan et al. from the asymptotic analysis of periodic structures [59]. In this so-called asymptotic homogenization, influence functions – also referred to as elastic correctors or characterization functions in [41] - are formulated from a micro-structural boundary value problem (BVP) to provide the macroscopic constitutive tensor as well as the relations between macroscopic deformations to micro-structural stresses/strains.

In addition to the asymptotic analysis, mean-field approaches, originally derived from Eshelby’s work on spherical inclusions [37], is another popular form of homogenization which determines the effective properties through localization tensors [60]. Similar to the influence functions of asymptotic homogenization, these localization tensors provide coupling between the two length scales and can accommodate for non-linear constitutive models at the micro scale such as plasticity or damage [61]. Transformation Field Analysis (TFA), on the other hand, is a modeling scheme which discretizes the RVE into finite sub-volumes for which pre-computed stress-concentrations tensors and transformation influence factors provide the micro/macro coupling [62]. The determination of the necessary tensors and influence factors are derived from an assumption of piece-wise uniform strains in each of the sub-volumes.

Another semi-concurrent multiscale formulation is the Generalized Method of Cells (GMC) developed by Paley and Aboudi [39]. In the GMC, the repeating unit cell geometry of a

unique composite microstructure is simplified using rectangular sub-volumes. Traction and displacement continuity, in averaged sense, between these sub-cells along with periodic constraints are used to develop the kinematic and constitutive relation between the scales. The Generalized Method of Cells, like Transformation Field Analysis and Asymptotic Homogenization, is able to account for inelastic deformations of constituent materials and does so through the assumption of uniform eigenstrains, which represent inelastic strains in the subvolume [63]. These various semi-concurrent schemes have all been incorporated into structural FE simulations: the Generalized Method of Cells was incorporated into ABAQUS [5], a blend of TFA and Asymptotic Homogenization for finite element analysis was developed [64,65], TFA was used with the non-linear explicit finite element software LS-DYNA [62], and the Mori-Tanaka mean field homogenization has been implemented by the composite modeling software DIGIMAT-MF for use in various FE packages. This listing is not comprehensive and other FE implementations are contained in a multiscale modeling review [61].

Lastly, another semi-concurrent modeling is the nested FE approach [66] for which the RVE response is solved using the finite element method. This varies from the previous methods which solve the micro-scale unit cell problem through analytical and/or semi-analytical solutions. The popular extension of this scheme is the so-called Computational Homogenization (CH) advocated by Geers and Kouznetsova [42,67].

Based on the review of multiscale modeling strategies, a semi-concurrent computational homogenization approach was selected as a general framework from which the free edge analysis will be developed. The semi-concurrent, computational homogenization model is chosen due to the following disadvantages in other schemes:

- Sequential methods lack microstructural information post-homogenization
- Concurrent models are computationally expensive even in a 2D domain and the analysis of the composite laminate free edge at the microstructure requires a 3D approach
- Concurrent models do *not* provide a constitutive relationship between the lower scale response and a macroscopic, homogenized response
- Semi-analytic or asymptotic semi-concurrent models require strict periodic assumptions at the lower scales and will limit the applicability of results to the free edge problem
- Utilizing a method other than FEA at the microscale requires additional work to incorporate micro-scale damage such as explicit modeling through cohesive zone modeling [68,69] or through the extended-Finite Element Method (XFEM) [32,70,71]

As with any approach, even the semi-concurrent, computational homogenization is not free of limitations. There are restrictions which must be noted to the applicability of current computational homogenization schemes. The fundamental assumptions of computational homogenization are the separation of scales and the Hill-Mandel condition. Separation of scales requires, based on the “order” of homogenization, that the gradient of macroscopic values remain small over a unit cell. In first-order approximations, the macroscopic deformation gradient must remain virtually constant over the characteristic length scale of the RVE, while second order methods are capable of characterizing linear variances [72].

The multiscale modeling approaches discussed previously, however, have not addressed the analysis of the microstructure at the free edge of a laminate outside of a 2D domain space. The next section covers recent modeling efforts which focus on the micro-scale analysis of free edge stresses within a laminate.

2.2 Modeling the Microscale Stresses at the Free Edge

Recent modeling efforts on the microstructure and free edges of laminates have been found in the literature. They include numerical simulations of free edge stresses in single-fiber FE models under simplified transverse tensile loading and uniform temperature change [73] or a similar approach which investigated moisture absorption [74]. An inverse relationship between fiber volume fraction and the magnitude of fiber/matrix interface tractions approaching the free edge was found in [73]. These works [73,74] incorporated the micro-scale investigation as a standalone component, analyzing the stress fields around the fiber and matrix to a very specific and simplified set of loading and boundary conditions, and were not multiscale approaches by definition.

Multi-fiber models have also been investigated near a free-edge by utilizing domain decomposition [56], a superposition method [75], and de-homogenization methods [50,51]. All models (both single fiber and multi-fiber RVE's) were limited to the study of 90° lamina micro-stresses. The influence of cracks on stress distribution in the midplane of $[\pm 25/90]_s$ composites highlighted the influence of idealized micro-cracks [51] and 3D penny micro-cracks [76] on failure initiation. In [37], the computed critical edge crack size correlated well with the experimental measured values. It was found that cracks situated in regions closer to dissimilar ply interfaces exhibited the highest stress concentration factors at the crack tip.

All of the previously outlined modeling approaches were limited to the analysis of 90° plies at the micro-scale as a means to simplify the necessary boundary conditions. In the latter multi-RVE studies [51,76], crack front analysis was isolated to local regions within the RVE sufficiently far (1 fiber diameter) from the boundary to mitigate any errors in the de-homogenization procedure. Thus, the modeling approaches were unable to investigate the nature

of crack growth at the interlaminar interface (between dissimilar plies); nor do they address the effect of microstructure on crack growth in off-angle lamina. All material models (matrix/fiber) were assumed linear elastic in both the single fiber and multi-fiber RVE models (the multi-fiber models investigated brittle carbon/epoxy systems dominated by fracture failure modes).

2.3 Summary

While free edge stress analyses have been performed for single and multi-fiber models, they have yet to investigate irregular fiber spacing and the effect of the local microstructure at the interlaminar region. The currently available methods in literature do not provide explanations for the occurrence of observed free cracking reviewed in Section 1.3 and Section 1.4 at interlaminar regions nor the minimal effect these cracks seem to have in progressive damage development. The next section will utilize a semi-concurrent framework for developing a multiscale framework to address these research questions.

3. Meso-to-Microscale Multiscale Framework

The multiscale analysis of the free edge effects at the microscale in a composite laminate will require the establishment of a framework for linking the lamina, or meso-scale, with that of the individual fibers and matrix, or micro-scale. These next few sections will first cover the basics of the chosen semi-concurrent multiscale approach, discusses multiscale boundary condition issues with free edges at the micro-scale and proposes an algorithm for circumventing energy balance issues that arise.

3.1. Semi-concurrent Approach

The two main components in the semi-concurrent multiscale implementation are the localization and the homogenization rules. Localization refers to the passing of information from the homogenized global scale to local scale of the RVE. Conversely, homogenization refers to the determination of macroscopic quantities from the RVE, or the passing of information from RVE to the global integration point. In a deformation driven FE analysis, the localization rules provides the kinematic coupling from macroscopic to microscopic domains. Although the process can use the macroscopic strain at the selected integration point, the work that follows uses the macroscopic deformation gradient, F^M , as shown in Figure 3.1.

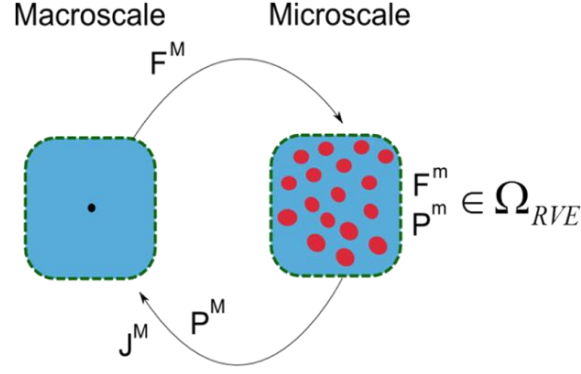


Figure 3.1: Overview of the semi-concurrent workflow and passing of information between the macroscale (global) and microscale (RVE/local).

The macroscopic deformation gradient is then used to specify necessary boundary conditions for the RVE's BVP. Standard procedures assume a volume average relationship between the deformation at the global scale and that of the RVE shown in Equation (3.1).

$$F_{ij}^M = \frac{1}{V_0} \int_{V_0} F_{ij}^m dV_0 \quad (3.1)$$

The superscripts M and m represent macroscopic and microscopic quantities, respectively, F is the deformation gradient tensor, and V_0 is the reference volume of the RVE. The way in which the macroscopic deformation gradient provides boundary conditions (displacements, or u) on the RVE are provided in Equation (3.2),

$$u_i^m = (F_{ij}^M - I_{ij})X_j^m \quad (3.2)$$

where X is the vector of reference configuration coordinates. The prescription of these displacements can be chosen for all nodes within the RVE, all nodes along the boundary, or restricted to the vertices in the case of periodicity. The homogenization procedure, which

provides the macroscopic stress as a function of microscopic stresses, also involves a volume averaging relationship and is provided in Equation (3.3).

$$P_{ij}^M = \frac{1}{V_0} \int_{V_0} P_{ij}^m dV_0 \quad (3.3)$$

The stress is written using the first Piola-Kirchhoff stress tensor, P , for convention, as it is the work conjugate to the deformation gradient. A second part of the homogenization process is determining the instantaneous material Jacobian, J , shown in Figure 3.1.

3.2. Development of a Free Edge Multiscale Boundary Condition

It has been noted in literature that the particular boundary conditions employed at the RVE level has a significant effect on both the effective properties of the computed homogenized response as well as the distribution of stresses/strains within the RVE [33,77]. This effect is especially important when non-linear constitutive models are utilized at the constituent level [78]. It was noted by van der Sluis that the three main boundary conditions employed in RVE analysis (uniform displacement, uniform traction and periodic) represented an upper bound, lower bound and mid-estimate, respectively, for the homogenized modulus [77]. A similar conclusion was obtained by Kaczmarczyk et al. even for second-order homogenization [79]. Inglis et al. found while including localization and damage in the RVE that the boundary conditions had little effect on the overall effective properties; the boundary conditions did in fact alter the distribution of stresses and localization within the RVE [78]. In addition, the work also investigated the Minimal Kinematic Boundary Conditions (MKBC) but found it to invoke a response identical to uniform traction boundary conditions [78].

Although multiscale schemes have been moving into the realm of understanding heterogeneous material fracture [80–82], serious limitations of periodic assumptions have been addressed for the case of localized damage within these semi-concurrent schemes [83]. These deficiencies in standard periodic boundary conditions have developed into very recent research attempts to develop various “forms” of periodic constraints, such as the recent work of Coenen et al. [84]. Another recent work in semi-concurrent boundary conditions was explored by Larsson et al. who proposed a weak form of micro-periodicity on the RVE domain [85]. Nevertheless, these developments still do not address the issues of free edges. The next section discusses a proposed approach on developing a methodology for implementing non-periodic boundary conditions in a semi-concurrent scheme while preserving energy between the scales. Although the intended application of free edge boundary conditions is in a 3D domain, this proposed approach is first presented for a 2D plane strain problem.

3.3. An energetically consistent approach

All multiscale schemes are required to preserve the Hill-Mandel relation which states that the variation of work at the global scale must equal the volume average of the variation of work at the local scale. This relation is shown in Equation (3.4),

$$\delta W^M = \int_{V_0} \delta W^m dV_0 \quad (3.4)$$

where δW is the variation of work and M and m are the macro and micro-scale specifiers, respectively. As a result of the Hill-Mandel relation, the kinematic and constitutive coupling between the macroscopic and microscopic domain are constrained to satisfy the averaging relations provided in equations (3.1) and (3.3). These two equations state the volume average of

the stress/strain field variables throughout the RVE are equal to the corresponding macroscopic variables. Given these averaging relations, it is a trivial calculation to show that the Hill-Mandel relation is satisfied, and these calculations are given for various types of RVE boundary conditions in the work by Kouznetsova [42]. Coenen et al. [84] enforced boundary conditions satisfying the strain averaging relations, then proved that the volume averaged microscopic stress tensor would in fact satisfy the work equivalence between the scales. Equation (3.5) below expresses the Hill-Mandel relation in terms of the individual deformation gradient and stress tensor.

$$P_{ij}^M \delta F_{ij}^M = \frac{1}{V_0} \int_{V_0} P_{ij}^m \delta F_{ij}^m dV_0, \quad \forall \delta \vec{x} \quad (3.5)$$

As was discussed in the work by Coenen et al. [84], the strain averaging relation (or alternatively, the deformation averaging relation) is satisfied when the contributions of the micro-fluctuation on the volume average are zero. A simplified form of this statement is shown in Equation (3.6),

$$F_{ij}^M = \frac{1}{V_0} \int_{V_0} x_{i,j}^m dV_0 = F_{ij}^M + \int_{\Omega_{RVE}} (\Delta w_{i,j}) d\Omega_{RVE} \quad (3.6)$$

which is derived from the right hand side of Equation (3.1) and using Equation (3.2). It should be noted that the strain averaging relation is being written with respect to the deformation gradient rather than the strain tensor, however the equations still hold. The following relation

$$x_i^m = F_{ij}^M X_j^m + w_i \quad (3.7)$$

describes the local, current coordinates \mathbf{x} within the RVE as a function of the macroscopic deformation gradient and a function of the microfluctuation field, \mathbf{w} , which is dependent on the microstructure. This superposition of a macroscale influence and the microfluctuation field is graphically represented in Figure 3.2.

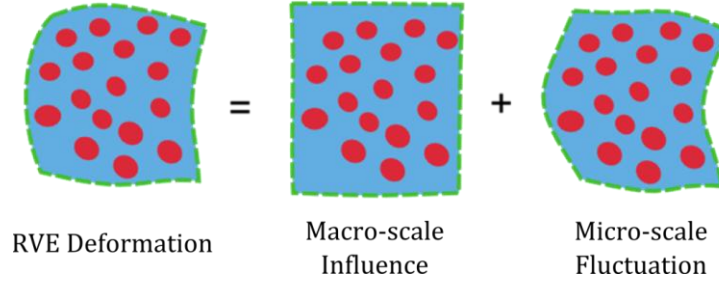


Figure 3.2: Schematic of the RVE deformation as a superposition of macroscale and microscale influences.

The strain, or deformation, averaging relation will only hold if the last integral term in Equation (3.6) is equal to zero. It has been shown in literature that periodic boundary conditions satisfy this requirement. Non-periodic, non-uniform boundary conditions, however, will contain a microscale influence field which will not set the integral term in Equation (3.6) to be zero. The microscale influence field will be an unknown solution, dependent on the microstructure, thus the form of the macroscopic stress tensor cannot be solved analytically *a-priori*. Instead, it is assumed in this work that the volume average of the RVE stress field is a sufficient approximation, although there is no guarantee (due to the nature of the microfluctuation field) that it is consistent with Hill's energetic conditions.

In the first iteration of developing an energetically consistent stress tensor, an equivalence of elastic strain energy between the scales, as introduced by Hill [35], is directly enforced. First, the localization process is performed as in Figure 3.3 according to a given macroscopic quantity of deformation. It is assumed that

- all constituents behave elastically
- the bond between the constituents may fail when certain criteria are met
- the localization provides the desired state of stress/strain (based on macroscopic deformation) within the RVE resulting in a computed microscopic, elastic strain energy, shown by the blue box in Figure 3.3.

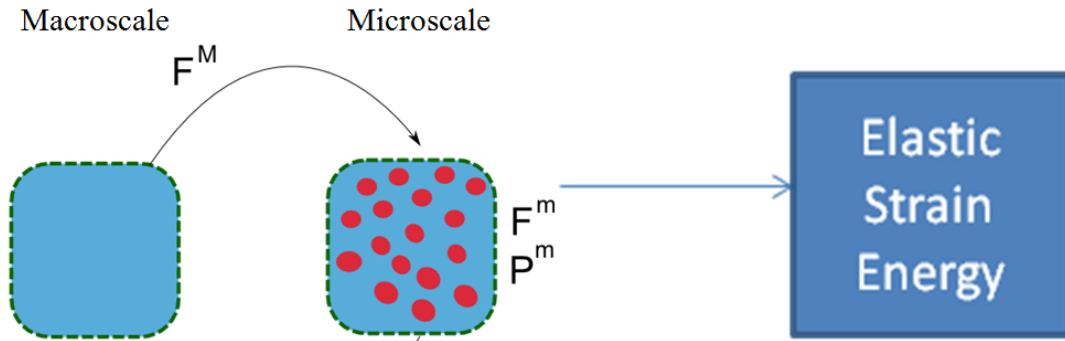


Figure 3.3: Illustration of the localization process, which has deformed the RVE to produce a unique amount of elastic strain energy defined by the box on the right.

It is desired to determine the macroscopic stress tensor required to ensure that the elastic strain energy at the global level matches that from the RVE. The formulation to follow assumes that the micro-constituents are linear elastic with cohesive interactions between the fibers and matrix (which may fail). Due to the presence of elastic softening from the failing/failed cohesive surfaces, the elastic strain energy at the macroscopic level will be computed according to the diagram in Figure 3.4(b), rather than the purely elastic case shown in Figure 3.4(a). The 2D plane strain problem, to which this work is currently focused, requires that the elastic strain energy is computed from the contributions of stress and strain in the three degrees of freedom in the two dimensional problem (1-1, 1-2, and 2-2 directions), shown in Figure 3.5. The condition of equivalent elastic strain energy between the scales enforces that the total energy shown in Figure

3.5 should be equal to that found during the localization process in Figure 3.3. Thus, this equality provides a constraint with which we can determine the form of the macroscopic stress update. Although there are an infinite number of stress tensors, given known values of deformation (strain) at the macroscopic level, that will satisfy the energy equivalence, it is hereby assumed that the macroscopic stress state can be approximated as being proportional to the volume average of the RVE stresses.

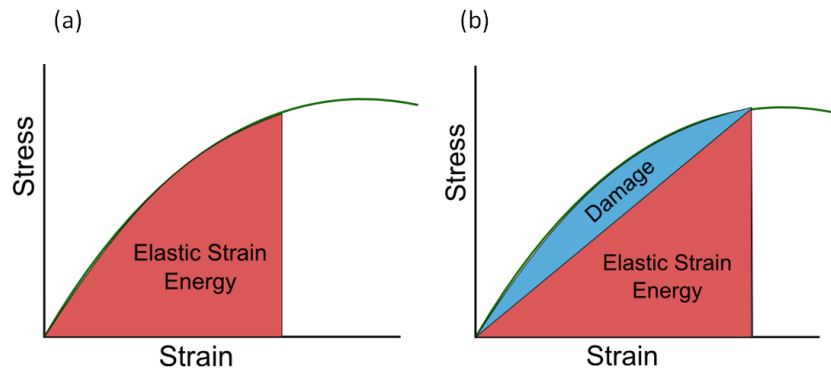


Figure 3.4: Stress-strain diagrams highlighting the elastic strain energy for two non-linear material responses where (a) the material is purely elastic and (b) the material has non-linearity due to elastic softening.

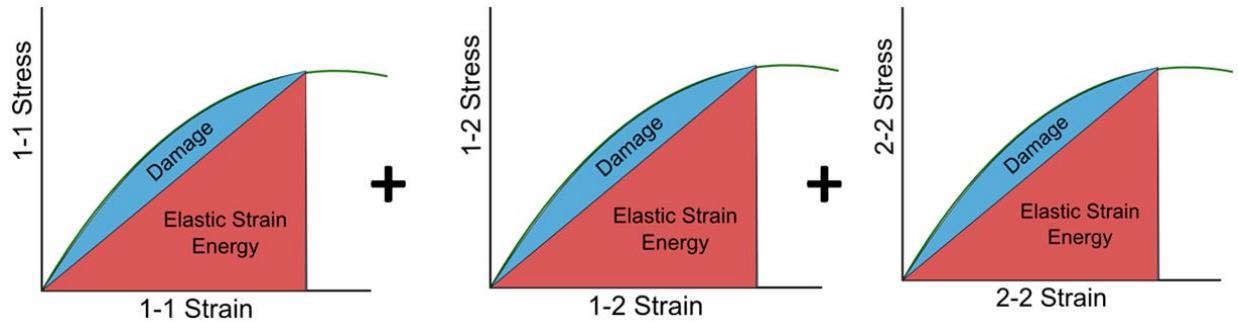


Figure 3.5: The total amount of elastic strain energy at the macroscopic level for the 2D plane strain problem is a summation of the elastic strain energy resulting from the three degrees of freedom.

This assumption of proportionality introduces a scaling parameter, z , which will be found by enforcing energy equivalence between the global and local scale. The computation of the macroscopic stress tensor will now be found using

$$\sigma_{ij}^M = z \sigma_{ij}^{*M} = z \frac{1}{V_0} \int_{V_0} \sigma_{ij}^m dV_0 \quad (3.8)$$

where σ^{*M} is the macroscopic Cauchy stress tensor computed from volume averaging the microscale Cauchy stress tensors, σ^m , in the RVE. The energy balance with respect to the elastic strain energy at the macro and micro scales can be written

$$\begin{aligned} E_{strain}^m &= \left(\frac{1}{2} \epsilon_{11}^M \sigma_{11}^M + \frac{1}{2} \epsilon_{12}^M \sigma_{12}^M + \frac{1}{2} \epsilon_{22}^M \sigma_{22}^M \right) \\ &= z \left(\frac{1}{2} \epsilon_{11}^M \sigma_{11}^{*M} + \frac{1}{2} \epsilon_{12}^M \sigma_{12}^{*M} + \frac{1}{2} \epsilon_{22}^M \sigma_{22}^{*M} \right) \end{aligned} \quad (3.9)$$

where left hand side represents the elastic strain energy at the microlevel, E_{strain}^m , from the RVE localization, while the right hand side represents the elastic strain energy computed using the volume averaged stresses. The z scalar parameter is then computed using

$$z = \frac{2E_{strain}^m}{(\epsilon_{11}^M \sigma_{11}^{*M} + \epsilon_{12}^M \sigma_{12}^{*M} + \epsilon_{22}^M \sigma_{22}^{*M})} \quad (3.10)$$

as a function of the microscopic elastic strain energy, macroscopic strains, and volume averaged RVE stresses. This scalar parameter represents the energy based constitutive coupling between the microscopic and macroscopic scales. Although the current, elastic strain energy based

formulation restricts the fiber/matrix constituents to be linear elastic, a similar approach could be developed using the assumption in Equation (3.8) utilizing the external work.

4. Free Edge Boundary at the Microscale

4.1 ABAQUS Semi-concurrent Implementation

The semi-concurrent scheme as well as the proposed z-scalar methodology was implemented numerically utilizing Python scripting to invoke the nested FE solution within the commercial FE software ABAQUS. To reduce initial overhead in implementing the multiscale framework the current model was restricted to 2D, plane strain analysis. The iterative algorithm is presented below in Figure 4.1. At each incremental step in the analysis, the user defined material subroutine (UMAT) was utilized to perform the communication between Python and the ABAQUS solver. The left-hand side of Figure 4.1 highlights the localization process which involves the passing of the macroscopic deformation gradient from the UMAT to the custom Python script. The Python code modifies the boundary conditions to a unit-cell, or representative volume element, ABAQUS input file. The unit-cell job is submitted and is post-processed upon completion according to the right-hand side of Figure 4.1.

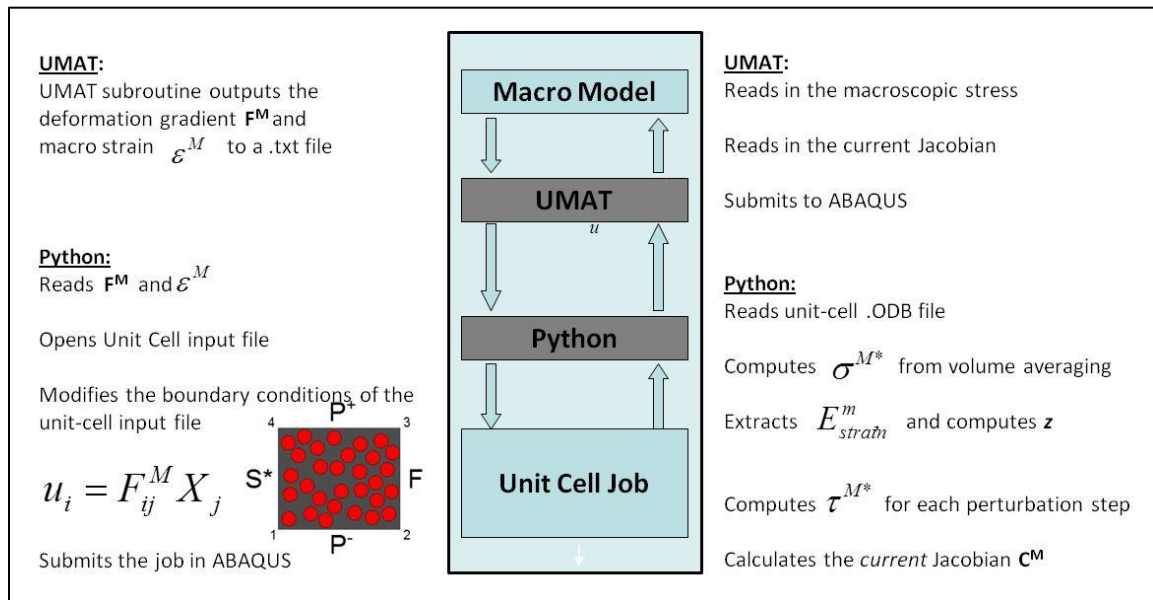


Figure 4.1: Workflow of the semi-concurrent computational homogenization scheme.

As a test case for implementing non-periodic boundary conditions, a choice was made to implement boundary conditions equivalent to that shown in Figure 4.2(a). These test boundary conditions contains a free edge at one boundary, corresponding to a macro scale with similar boundaries. The symmetric edge at the left of the RVE was an approximation chosen, due to the lack of 1-2 shear deformations in the loading cases explored, to simplify the implementation within the semi-concurrent scheme.

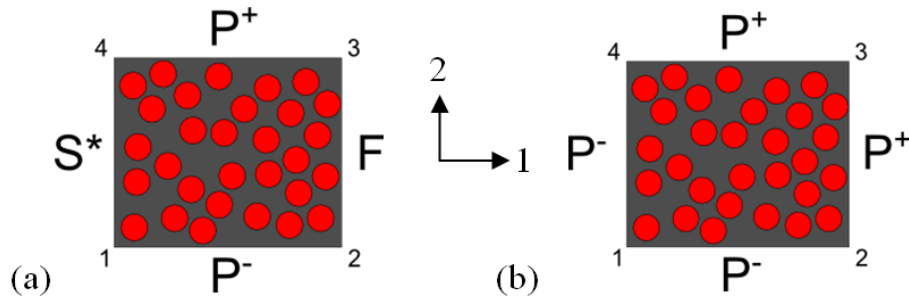


Figure 4.2: Schematic of (a) the test boundary conditions, which include the free edge, and (b) the standard periodic boundary conditions. Note: That P, F, S in the above diagram represent periodic, free and symmetric edge, respectively.

The implementation of the free-edge boundary conditions results in a different set of localization rules than those used for periodic boundary conditions given as

$$u_i^v = (F_{ij}^M - I_{ij})X_j^v \quad (4.1)$$

The localization in equation (4.1) is identical to that in (3.2) except that the microscale displacements are only prescribed on the vertices, v , of the RVE. Elsewhere, non-vertex nodes are constrained to obey periodicity according to the formulation of Van der Sluis et al. [77] and using *Equation keywords in ABAQUS. In the case of the boundary conditions shown in Figure

4.2 (a), the localization shown in Equation (4.1) is prescribed for vertices 1 and 4. For vertices 2 and 3, displacements are only prescribed in the 2-direction to account for the free-edge requirement which must remain traction free in the 1-direction. Additionally, all nodes lying on the symmetric edge (S^*) are prescribed displacements in the 1-direction and left undefined in the 2-direction.

Once the unit-cell analysis was completed, the Python script would post-process the database according to the following steps listed in Figure 4.3. It should be noted that when periodic boundary conditions are employed, step 4 in Figure 4.3 is omitted, since it is unique to the z-scalar approach.

1. Reads the EVOL, or current element volume for all elements
2. Extracts the element stresses within the unit-cell RVE
3. Computes the volume average of the stresses based on the EVOL values
4. Using the volume averaged stresses and using Equation (10), z is calculated
5. The macroscopic stress tensor is exported to an external “.csv” file

Figure 4.3: Listing of the database post-processing steps performed in the custom Python script.

Aside from the macroscopic stress tensor, the material Jacobian needs to be determined for the subsequent increment in the global FE analysis and is a required output of the UMAT within ABAQUS Standard. The construction of the Jacobian from the perturbation steps is summarized in Figure 4.4. For a given perturbation step, a component of strain is set to unity, while all others are set to zero. Thus, the resulting stresses computed as a result provide a given column in the Jacobian stiffness matrix. The stresses are computed using the same procedure as was done for

the macroscopic stress tensor, utilizing the scaling parameter when necessary (e.g. when non-periodic localization is employed).

$$\begin{array}{c}
 \text{Set } \epsilon_{11}=1 \qquad \qquad \text{Set } \epsilon_{12}=1 \\
 \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} \\ J_{21} & J_{22} & J_{23} & J_{24} \\ J_{31} & J_{32} & J_{33} & J_{34} \\ J_{41} & J_{42} & J_{43} & J_{44} \end{bmatrix} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ 0 \\ \epsilon_{12} \end{bmatrix} \\
 \text{Set } \epsilon_{22}=1
 \end{array}$$

Figure 4.4: The Jacobian matrix, J , is shown with the necessary uni-strain components for determining a particular column. The circled columns are computed via uni-strain perturbation steps where one of the three strain components are set to 1, and the remainder are zero. The condition for computing each column is labeled accordingly. The 33 component is determined externally. Symmetry of the Jacobian is assumed.

The J_{33} component must be found externally using standard micromechanics, or basic homogenization techniques. Assuming the current 2D plane strain analysis of the RVE, minimal damage will accumulate in the fiber direction, thus this component could be approximated as constant through the entire 2D analysis. The remaining components in the J_{X3} column are obtained by assuming symmetry of the Jacobian. Once the Jacobian is computed and exported to a “.csv” file, the Python script is completed.

The UMAT will then read in the macroscopic stress tensor and Jacobian which is input back into the ABAQUS simulation at the global scale. When periodic boundary conditions are employed, the perturbation steps are executed immediately after the stress analysis within the

same ABAQUS job. For the case of the test boundary conditions in Figure 4.2(a), the operation must be performed as a separate unit-cell job after the completion of the stress analysis. Additionally, the boundary conditions are modified to remove the free-edge and are shown in Figure 4.5. The right-hand side of the RVE is perturbed using Equation (4.1) for all nodes in both the 1- and 2-directions. The process for determining the Jacobian during the post-processing steps is listed in Figure 4.6 An overall summary of the discussed workflow with the Fortran subroutine and Python script are highlighted in Figure 4.7.

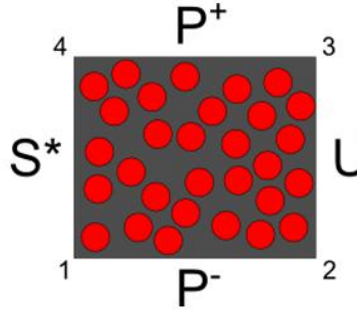


Figure 4.5: Boundary conditions used for the perturbation steps during non-periodic analysis. The right-hand-side BC, labeled U, represents uniform displacement as prescribed using Equation (4.1).

- | | |
|-----|---|
| 6. | Extract the nodal coordinates at the end of the localization step for all perturbed nodes |
| 7. | Use Equation (14) to determine BC's for the three perturbation steps
<div style="text-align: center;">Note: These steps are defined as linear perturbation steps in ABAQUS</div> |
| 8. | Generate the necessary input file job and submit (Perturb.inp) |
| 9. | Compute the differential macroscopic stress tensor associated with each perturbation |
| 10. | Build the Jacobian and export the resultant to an external ".csv" file |

Figure 4.6: Steps for post-processing the Jacobian.

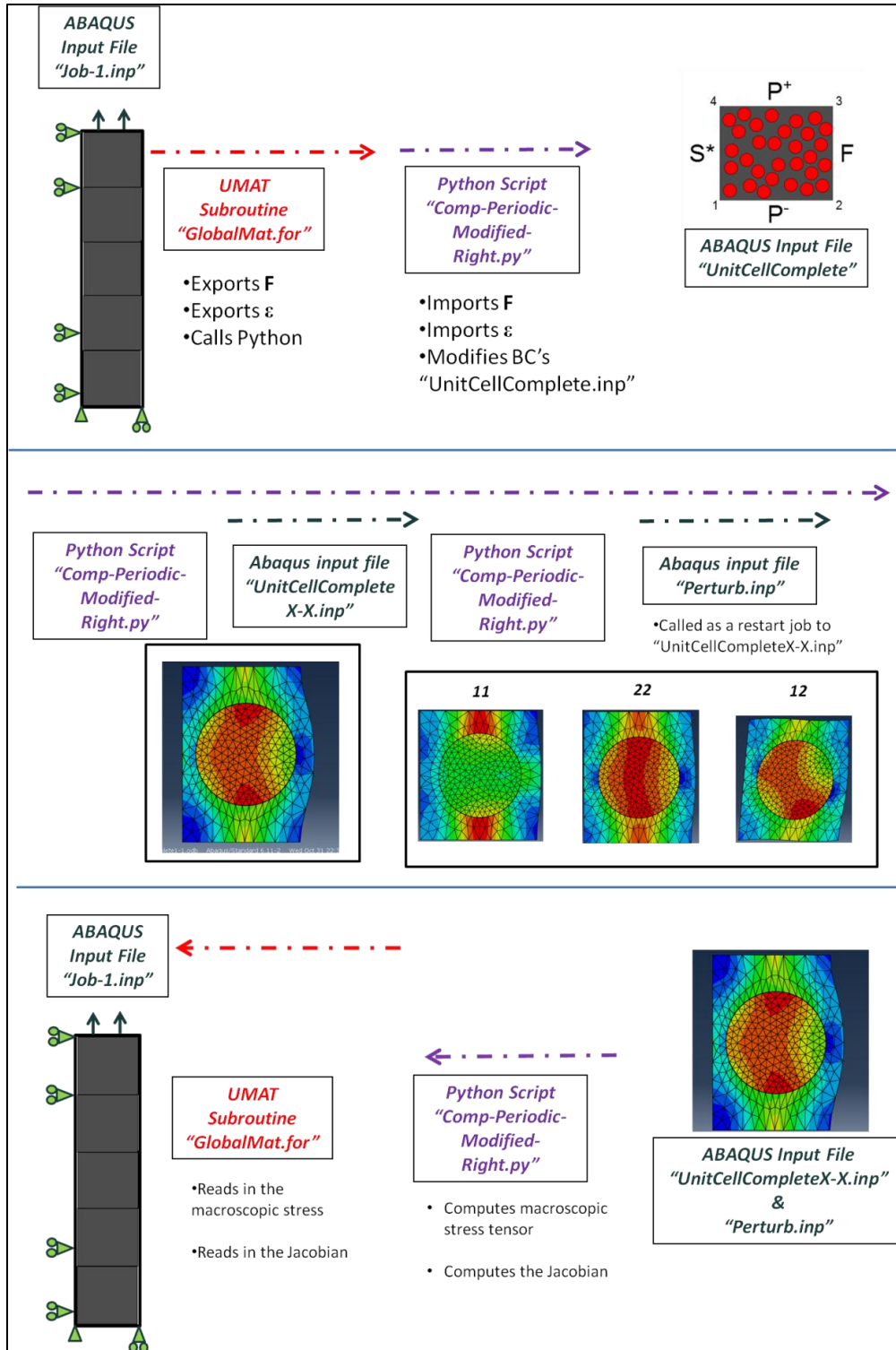


Figure 4.7: Schematic of the overall semi-concurrent scheme as implemented into ABAQUS for a single increment at a given macroscopic integration point.

4.2 2D Model Verification

Verifying the implementation of the z-scalar approach required the determination of the efficacy of both the proposed Jacobian update method and energy preservation. For both test cases, multi-fiber RVE's were required using both random placement and the implementation of cohesive surface definition between the fibers and matrix constituents. A tool for the generation of multi-fiber RVE generation is discussed, the implementation of cohesive interactions between the fiber/matrix within ABAQUS is presented, and finally the initial results of the verification study are offered.

4.2.1 RVE Generation

It has been found in literature that a fiber-matrix RVE must include a sufficient number of fibers for the macroscopic response (including micro-constituent plasticity and damage) to be independent of the random-fiber placement [48,49,86]. It was found that a fiber quantity greater than ~25-30 fibers was sufficient for obtaining constituent macroscopic responses for arbitrary fiber placement. Due to the significant pre-processing required to define fiber/matrix surfaces to be used for cohesive interfaces and complicated node-set numbering required to employ periodic boundary conditions, a custom Python script was developed to allow for future parametric studies. The custom Python script takes the following inputs:

1. Number of fibers –or- an input file with fiber center locations
2. Desired volume fraction
3. Fiber/Matrix constitutive parameters
4. On/Off flag for cohesive surfaces
5. Cohesive surface parameters
6. Fiber & matrix seed size

The scripting not only defines all necessary fiber/matrix surfaces for cohesive interactions, but also defines necessary node sets for applying periodic boundary conditions for use in the semi-concurrent CH modeling. The periodic boundary conditions are enforced in ABAQUS using the *Equation constraint keyword, which allows for the specification of node-to-node linear constraint equations. By creating node sets for the periodic faces and sorting them according to their (x,y,z) coordinates, the number of specified *Equation keywords in the ABAQUS input file can be greatly reduced by specifying corresponding sets rather than individual nodes.

Two methods for determining the random fiber placement were utilized: (1) traditional random fiber placement and (2) close-packed perturbation approach. Random fiber placement iteratively places fibers, checking for intersection, and prescribing a set number of attempts until failure. Requiring fibers to lie completely within the domain of the RVE, this method was found to only provide efficient results for fiber volume fractions under 50%. The initial result shown later in Section 4.2.3 utilizes this simple random placement of fibers. For the laminate studies of the intended [25/-25/90]_s composite, higher fiber volume fractions were required to represent traditional laminate VF's (55-62%).

To overcome the limitations of random placement, the following close-packed perturbation algorithm was proposed which also allowed fibers to exit the RVE edge as in Figure 4.8(a). First, fibers were allowed to initially penetrate each other. Second, an iterative algorithm was developed to “push” the fibers apart using standard kinematic equations and a pseudo-force based on fiber proximity and amount of intersection. An example of a periodic RVE with intersecting fibers is shown in Figure 4.8(b). The pseudo-force, F , shown in Equation (4.1), is a function of the intersection distance between two fibers, shown in Figure 4.8(c) as d , and a scalar

parameter f_{scale} , which modifies the amount of force per distance squared to achieve an efficient solution scheme. Both variables are adjusted accordingly until the desired RVE is achieved within a reasonable run time.

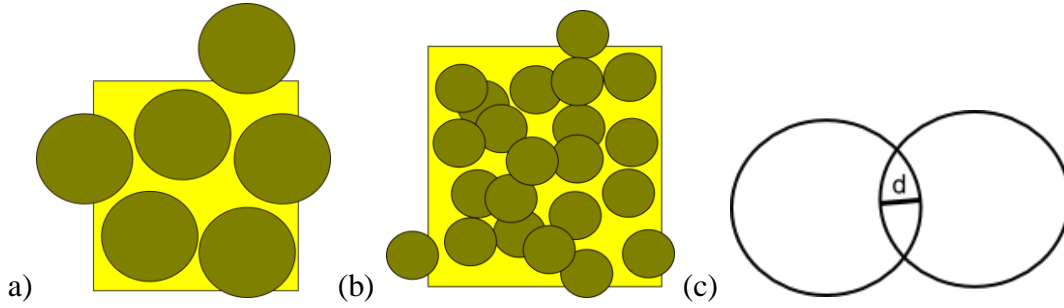


Figure 4.8:Example of a randomly placed (a) 4-fiber RVE with fibers allowed to exit the boundary and (b) an RVE with fibers intersecting and exiting the boundary. (c) Definition of the intersection distance, d .

$$F = d^2 f_{scale} \quad (4.1)$$

Standard constant acceleration kinematics are employed, assuming no initial velocities for all fibers, and the solution procedure marches forward according to the $tscale$ time increment as shown in Equation (4.2). S and S_0 are the current and initial displacements, v is the velocity and a is the acceleration assumed for duration of the time increment. The code, written in Python, tracks fiber centers and will end iterations when zero forces are achieved inside the RVE (thus signaling no penetration between fibers). The workflow of the generator is shown in Figure 4.9.

$$S = S_0 + v \, tscale + \frac{1}{2} a \, tscale^2 \quad (4.2)$$

- Randomly place fibers with intersection
- Compute forces
- Increment in time (tscale)
- Update displacements (using tscale , force and velocity vectors)
- Update velocity vectors
- Check if a fiber exists “outside RVE”
 - Delete it
- Check if a fiber completely inside RVE now exits an edge
 - Create instances
- Check if a fiber which was once a group of instances, is now the only instance
- Check if forces are zero in system
- If not, repeat from step 2

Figure 4.9: Workflow of the RVE generation script for high fiber volume fractions

4.2.2 Cohesive Zone Modeling

Cohesive zone modeling is an attractive numerical tool for fracture analyses when there exists a pre-determined crack path and no initial crack tip is present, such as in the case for fiber/matrix interface debonding. The development of cohesive zone modeling originated from the introduction of the Barenblatt–Dugdale Model which accounted for the plastic fracture process zone ahead of the crack tip. In CZM however, the stresses ahead of the crack tip (and consequently in the cohesive / fracture process zone) are modeled through a constitutive traction-separation law [87]. Accordingly, two crack tips exist in the CZM model: (1) the physical crack tip for which the preceding crack faces are traction-free and (2) the mathematical crack tip where the cohesive zone begins. A sample, arbitrary traction separation law is highlighted in Figure 4.10. Although a number of laws describing this constitutive cohesive behavior have been

developed, it was found by Alfano [68] and Volokh [69] to have little effect on the overall macroscopic response of the model.

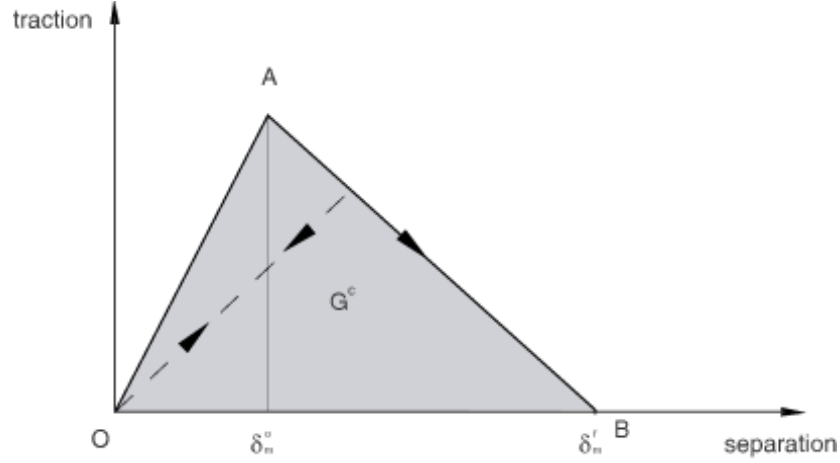


Figure 4.10: Example traction-separation law (ABAQUS 6.11 Analysis User's Manual)

Alfano concluded that bi-linear traction separation laws obtain accurate results for little computational cost and are therefore utilized in this work. ABAQUS v6.11 inherently has two methods for incorporating CZM into the FE model. The first is through cohesive elements (available as both 2D and 3D elements). Using cohesive elements is useful when the cohesive interface exists physically as a finite thickness layer, as with the modeling of adhesives. The cohesive elements are then constrained to the adjacent material using either the *Tie constraint or through coincident nodes on a merged mesh. The second method for CZM deployment is through the use of the *Cohesive Surface Behavior option in the Interaction Module of ABAQUS. This feature allows for essentially “zero-thickness” cohesive interfaces between contact surface pairs through the *Contact Pair option.

In a bi-linear traction separation law like the one used in this study, the main parameters are the cohesive strength (σ^c) and the separation work (G_c) –synonymous to the fracture energy.

The first determines the peak stress for which the cohesive tip is defined, and separation occurs, while the separation work determines the area under the curve of the associated law. The simplest criterion, implemented in this work, assumes a maximum stress failure criterion, defined as

$$\max \left\{ \frac{\langle t_n \rangle}{t_n^0}, \frac{t_s}{t_s^0} \right\} = 1 \quad (4.3)$$

where t_n and t_s are the normal and shear tractions along the cohesive front and t_n^0 and t_s^0 are the corresponding cohesive strengths for the particular directions.

As is the case with fiber/matrix debonding within a UD ply, the mode-mixity at the crack tip is undefined and changing as the crack propagates. One must then define the cohesive behavior's initiation criterion as either uncoupled between Mode I and Mode II or coupled through a standard quadratic relationship. The mode-mixity of damage evolution must also be defined. Linear degradation evolution is assumed for the multi-fiber RVE used in the validation study, where the damage variable D is defined as

$$D = \frac{\delta_m^f (\delta_m^{\max} - \delta_m^0)}{\delta_m^{\max} (\delta_m^f - \delta_m^0)} \quad (4.4)$$

where δ_m^{\max} corresponds to the maximum separation achieved in the simulation at the cohesive interface.

A number of field variables are available for post-processing simulation results containing cohesive surfaces/elements. For example, the DMICRT represents the damage initiation criterion field variable and achieves a value of 1 when the element/surface has failed. The SDEG field variable represents the element or surface's relative state of damage. A value of

1 represents a completely failed element and separated surface. Implementations of the cohesive elements within ABAQUS are shown in Figure 4.11(a) and Figure 4.11(b).

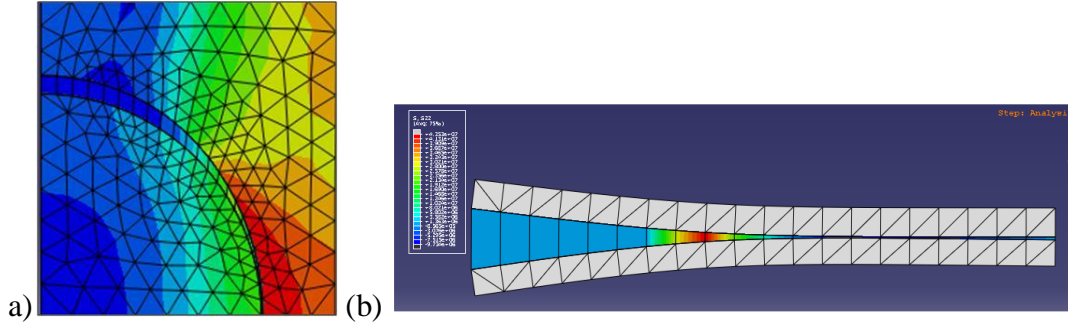


Figure 4.11: Cohesive elements used in the (a) uniaxial tension simulation on a quarter fiber/matrix RVE and (b) on a Mode I delamination example

4.2.3 Results and Conclusions

The first step in verifying the modifications coded into the semi-concurrent scheme was to check the validity and robustness of the Jacobian updating procedure using the boundary conditions discussed in the previous section. This test was carried out by comparing computed elastic moduli (using Mathematica and the exported Jacobian matrices from Python) to those predicted from purely periodic boundary conditions. A number of test cases were examined including a homogeneous RVE, a single fiber RVE, and multi-fiber RVE. For the fiber/matrix RVE's the fiber volume fraction was 45%. Table 4.1 presents a comparison of the reference periodic values with those computed using the proposed perturbation boundary conditions for the z-scalar approach. For all three test cases, the non-periodic perturbations provided reasonable estimations of the material constants. It should be noted, however, that the uniform displacement boundary conditions on the right-hand side of the RVE caused the elastic moduli to be slightly over-predictive in shear.

Table 4.1: Computed elastic constants for periodic and non-periodic perturbation steps.

Material Constant	Homogeneous Solid		Single Fiber RVE ($E_m = 3.5$ GPa, $E_f = 22$ GPa)		25-fiber RVE ($E_m = 3.5$ GPa, $E_f = 22$ GPa)	
	Periodic	Non-Periodic	Periodic	Non-Periodic	Periodic	Non-Periodic
E_x (GPa)	100.00	100.000	20.897	21.257	20.247	20.455
E_y (GPa)	100.00	100.000	20.897	21.137	20.378	20.419
ν_{xy}	0.300	0.300	0.294	0.295	0.330641	0.328911
G_{xy} (GPa)	38.462	38.462	6.438	6.901	7.207075	7.363027

The next step in the verification process was to confirm that the z-scalar method preserved energy between the global and local scales even when using non-periodic boundary conditions. A single, reduced-order quadrilateral element (CPE4R) at the global scale was placed under uniaxial tensile loading. The element's integration point was coupled to a 25 multi-fiber RVE with fibers placed randomly within the boundary of the RVE. Both periodic as well as the non-periodic free edge boundary conditions were employed in two separate iterations and the global elements were stretched to 1% strain. As was assumed in the formulation of the z-scalar method, all constituents are linear-elastic, however, a cohesive interface following a bilinear traction separation exists between the fiber and matrix. The necessary parameters for the cohesive interface are the cohesive strength, σ_{cohe} , and the critical displacement, d_{crit} . The cohesive strength defines the peak traction value in the traction-separation law, and the critical displacement specifies the traction-free separation distance. The cohesive parameters are chosen

to be representative of a relatively weak interface. The constituent and interface properties are provided in Table 4.2.

Table 4.2: Constituent properties for the multi-fiber RVE verification study

	E (GPa)	ν	σ_{cohe} (MPa)	d_{crit} (m)
Fiber	22	0.3	--	--
Matrix	3.5	0.4	--	--
Interface	10E7	--	50	0.005

It was found during the verification studies that the ETOTAL, or total energy history variable, of the system was non-zero, a phenomenon that was indicative of incorrect energy computations. Figure 4.12 shows all the energy history variables output by ABAQUS. Further investigations revealed that the negative ETOTAL, highlighted by the dashed box in Figure 4.12, was a result of the contributions of the cohesive surface to the internal strain energy of the system. When using cohesive interactions rather than cohesive elements, the strain energy present in the separation of the fiber/matrix interface was not getting added to the total strain energy of the RVE system. Although the contribution of the cohesive interface to the strain energy of the system can be reduced by increasing the cohesive stiffness, even a sufficiently large value of 10E7 GPa had a measurable contribution as shown in Figure 4.12. Equation (4.5) shows the energy balance equations of relevant energy sources in the FE simulations. Due to the neglecting of cohesive strain energy, the E_{strain} , and consequently the E_{internal} , energy terms were being under reported resulting in a negative ETOTAL result.

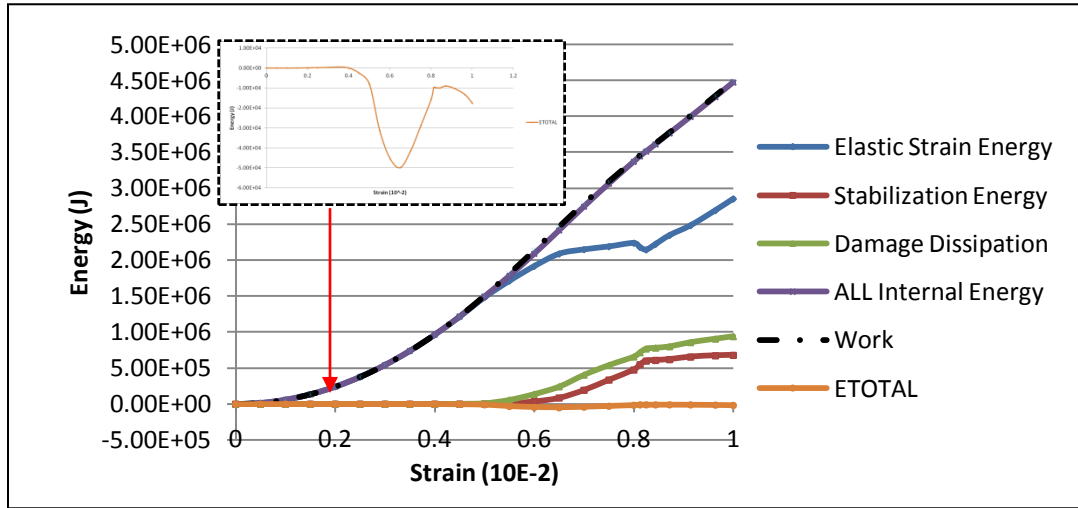


Figure 4.12:Energy history variable outputs for the 16 fiber RVE under periodic boundary conditions subject to uniaxial tensile loading.

$$E_{total} = E_{work} - E_{internal} = E_{work} - E_{stable} - E_{damage} - E_{Strain} \quad (4.5)$$

As a consequence of the negative ETOTAL values, the computation of the z-scalar parameter in Equation (3.10) from the micro-structural strain energy had to be modified as in

$$E_{strain}^m = E_{strain} - E_{total} \quad (4.6)$$

The values E_{strain} and E_{total} in Equation (4.6) were extracted from ABAQUS from the ALLE and ETOTAL history variables.

The results of the z-scalar verification are presented in Figure 4.13, plotting the percent difference in external work computed by ABAQUS between the global and local scales for the two boundary conditions cases (periodic and non-periodic with a free edge). Both the global and local external work energies were found to be nearly identical for the two test cases, as seen from the relatively small values (<0.4%) in Figure 4.13. Coincidentally, the two test cases had similar

external work values regardless of the differences in applied boundary conditions on the level of the RVE. For the case of the non-periodic boundary conditions, the z-scalar was computed to be a value slightly less than unity. For example, at the last increment of the macroscale analysis (~1% strain) the z-scalar parameter was computed as 0.984. It was hypothesized that this small variation from unity was a function of the test boundary conditions and a lack of significant differences in the RVE response for the periodic and non-periodic BC's. Since periodic boundary conditions have been proven to be energetically consistent, the close agreement between the macroscale periodic and non-periodic external work in Figure 4.13 would suggest that the implemented z-scalar parameter ensures similar consistency. Also, the small variations seen even with the periodic boundary conditions were hypothesized to be numerical construct within ABAQUS, rather than an imbalance of work between the two scales.

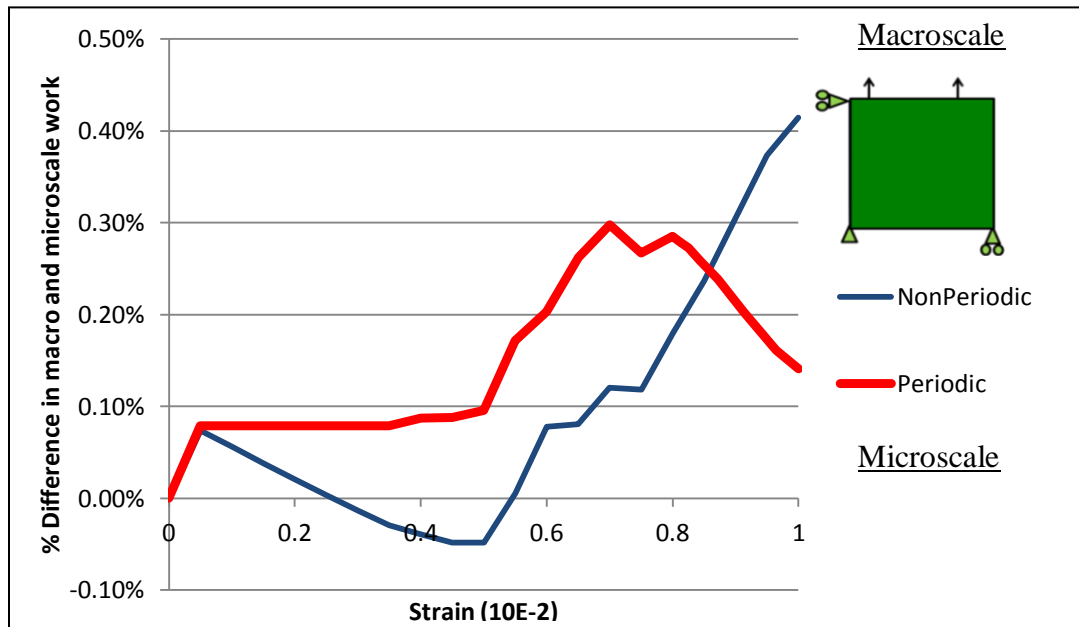


Figure 4.13: Percent difference in external work between the macro and micro scale work for the 16 fiber RVE for periodic and non-periodic boundary conditions.

A case study was undertaken to show an application of the z-scalar energy based method. The 2D RVE used for the energy verifications and discussed in the previous sections was used in the following case study. The objective was to investigate the effects of a free edge boundary at the macroscopic level on the constitutive response provided by the RVE. The boundary conditions for the RVE were the test, free-edge boundary conditions proposed in Figure 4.2 (a). The RVE's constituent properties were those reported in Table II. Again, damage was restricted to fiber/matrix debonding. Fibers were placed into the RVE using an iterative random placement script written in Python, for which the highest achievable fiber volume fraction was 45%. Macroscopic loading was uniaxial tension as in the diagram on Figure 4.13. Differences between periodic and non-periodic RVE boundary conditions were evaluated according to the macroscopic response (evaluated by plotting the elastic strain energy vs strain) as well as the elastic moduli (E_x , E_y). Previous research works [86,88] have shown that RVE size (number of fibers) affects the macroscopic response, and the appropriate RVE size is dependent on the constituent properties of the RVE. Other works have plotted the variance in macroscopic (stress-strain) response over a variety of boundary conditions, none of which included a free-edge [84,85,89]. In this case study, the RVE size is varied from 4 fibers to 36 fibers with multiple iterations (3-4) of random fiber placement.

Figure 4.14 plots the elastic strain energy versus macroscopic strain for the 4 fiber RVE, across several random placement iterations, as well as the comparisons between periodic and non-periodic (free) boundary conditions. The curves are labeled according to XPerY or XFreeY, where X is the RVE size and Y is the iteration number. Dotted points on the curves represent the periodic simulations, and solid curves represent those with the non-periodic boundary. The sudden plateaus seen at ~0.6% strain, or dip in the case of iterations 2 & 3, are a result of the

cohesive failure. The curves were as expected from trend seen in the elastic energy curve shown in Figure 15.

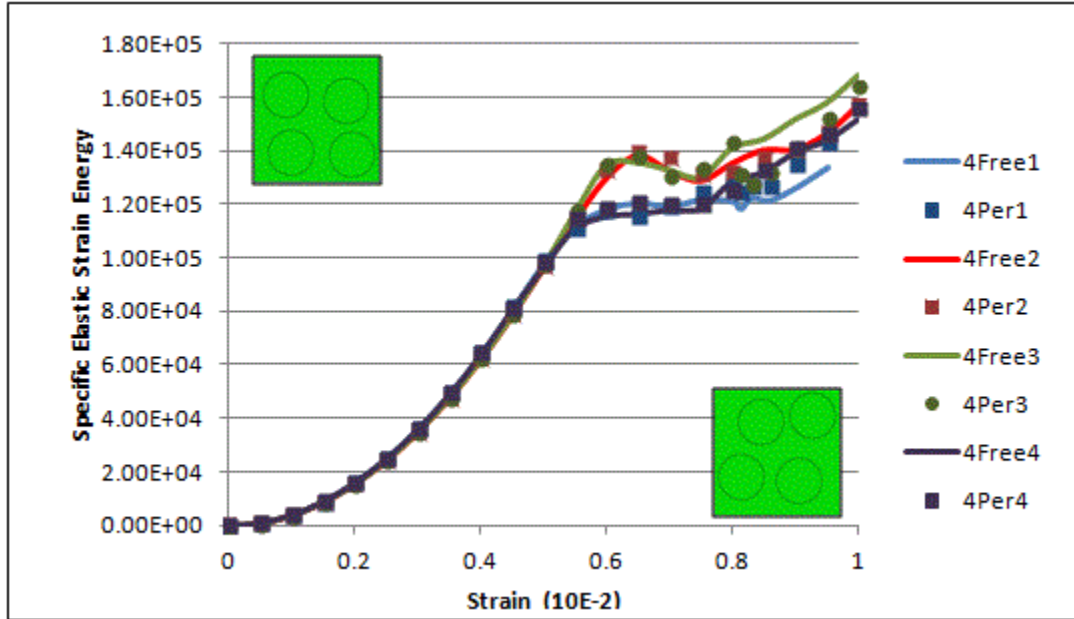


Figure 4.14: Specific elastic strain energy vs macroscopic strain for the 4-fiber RVE. Dashed lines represent the periodic results, solid lines those of the free edge simulations, and the color coding corresponds to a given RVE iteration.

Figures 4.15-4.17 plot the macroscopic stress in the loading (22) direction versus macroscopic strain. These plots provide the RVE's macroscopic material response under the uniaxial tensile loading. For the four 4-fiber RVE test cases shown in Figure 4.14, only one iteration (case 3) had a significantly different response between the two BC's. A similar observation can be made from Figure 4.14. Figure 4.15 also highlighted the significant variation in material behavior (cohesive failure) between the 4 different RVE iterations as a function of the random placement. Iterations 1 & 4 had significantly lower peak stress values (pre-cohesive failure) and a more drastic drop in stress before reloading.

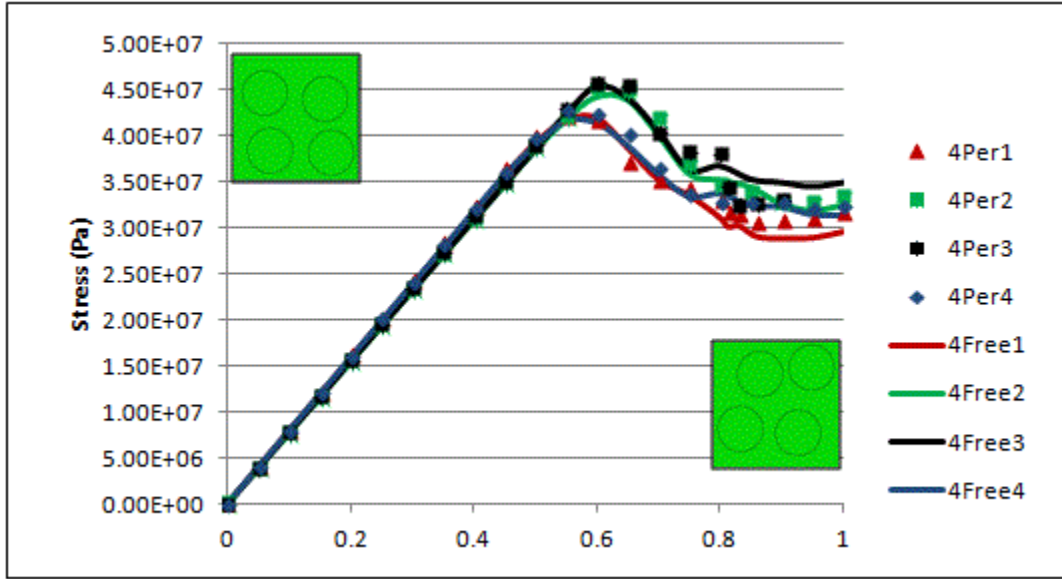


Figure 4.15: Macroscopic (2-direction) stress vs strain for the 4-fiber RVE.

Figure 4.16 shows the results from the 16 fiber test. The simulation for the 16Per3 test case terminated early due to convergence issues present after cohesive failure and is indicated on the figure. The first 16 fiber RVE iteration had a nearly identical macroscopic response between periodic and non-periodic, while the second iteration had significant softer post-peak response in the periodic BC case. For both of these iterations, the periodic response was “softer”, indicative of increased cohesive damage as a result of the constraint on the right-hand side of the RVE. The significant differences in iterations 1 and 2 between the periodic and non-periodic counterparts could be attributed to the RVE fiber placement with respect to the free-edge. The RVE for iteration 1, shown at the top left of Figure 14.6, had fewer fibers in the proximity of the right-hand edge versus the RVE for iteration 2, shown at the bottom left of the figure. The periodic BC caused cohesive failure and separation at the fiber highlighted by the black arrow on Figure 4.16, whereas the free edge did not cause failure/separation.

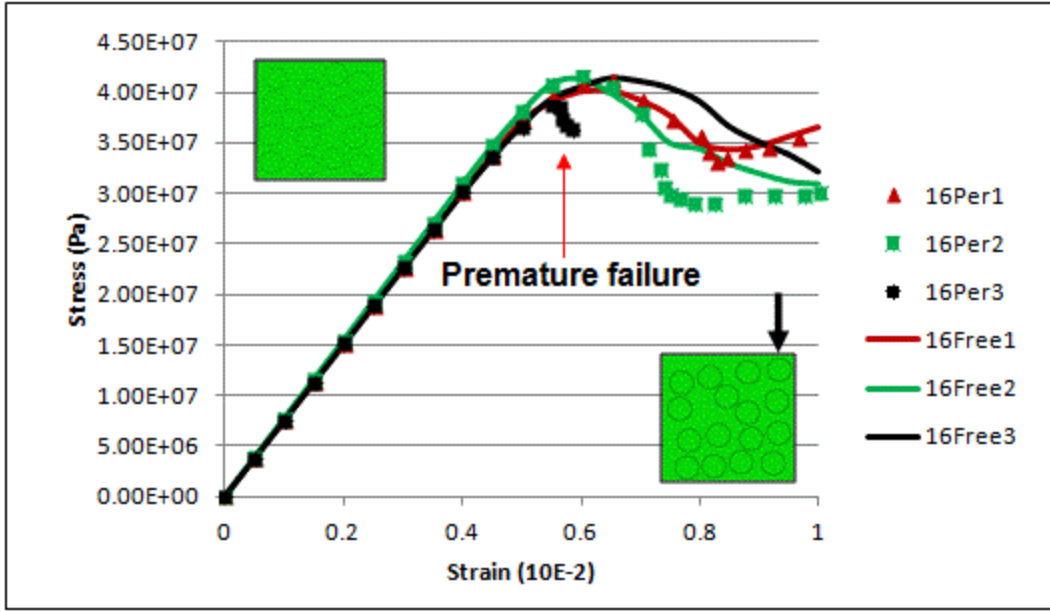


Figure 4.16: Macroscopic (2-direction) stress vs strain for the 16-fiber RVE.

The 36 fiber RVE results presented in Figure 17 highlighted two main points. First, as expected from literature, the variation in material macroscopic response decreased across the random fiber iterations. Second, the periodic boundary conditions suffered from numerical convergence issues resulting from premature failure in the cohesive surfaces, highlighted on the figure. This happened for two of the three periodic cases; however, the premature failure did not affect the material's tensile strength as was with the premature failure of the 16Per3 case. The free-edge simulations, however, were able to reach the final macroscopic loading of 1% strain with no issues. Iteration 1, which completed to 1% for both BC's, showed little differences in the macroscopic response. The similarity of periodic and non-periodic response for iteration 1 can again be attributed to the proximity of fibers to the right edge of the RVE.

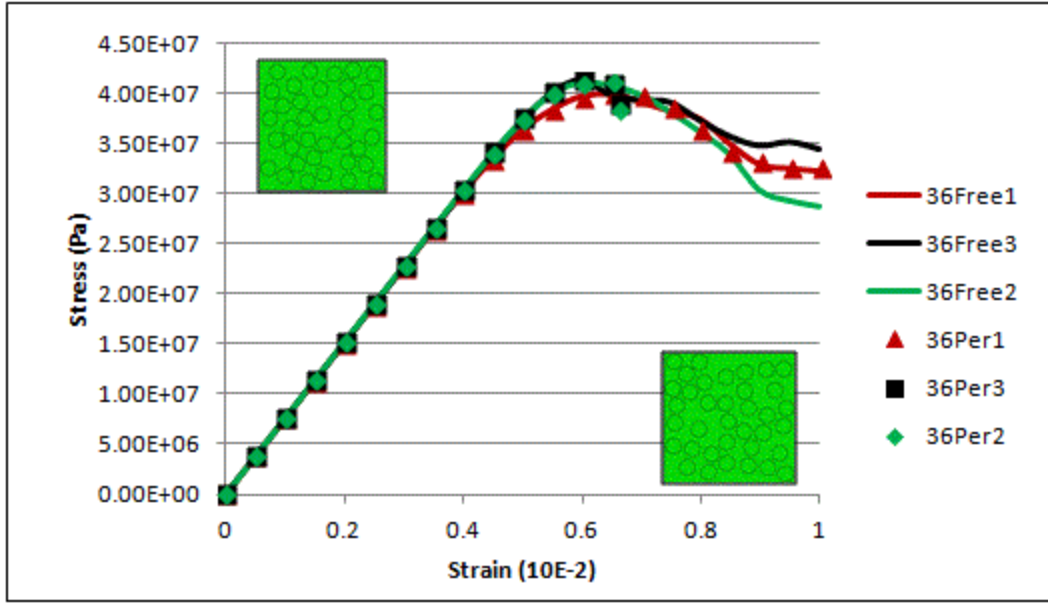


Figure 4.17: Macroscopic (22 direction) stress vs strain for the 36-fiber RVE.

Computation of the initial modulus (slope) from the periodic and non-periodic simulations in Figures 4.15-4.17 was performed for all cases to compare the influence of RVE size. There was an observed decrease in variation of the initial modulus with increasing RVE size. The initial modulus converged for the 36 fiber case was 6.7 GPa, and was slightly softer than the average modulus computed for the 4 and 16 fiber cases which ranged between 7.1 and 6.8 GPa.

4.3 Conclusions

The results from this implementation of the free edge boundary condition within the semi-concurrent scheme pointed out several useful applications for the 3D study of the composite laminate. First, the influence of the free edge boundary condition on the overall strain energy preservation across the scales was minimal (i.e. the z-scalar value was small across the simulations, particularly prior to any damage development). Thus for a linearly elastic 3D multiscale analysis, the z-scalar method may not need to be implemented. Additionally, the RVE

size study showed that a sufficient number of fibers should be chosen to best represent the macroscopic, homogenized response since random microstructures will exhibit variability in their response when the RVE size is small. For future work in Chapters 5 and 6, the micro-scale analysis will be defined as the use of a “micro-model” rather than an RVE, since the finite size of the micro-model may not necessarily represent the global macroscopic behavior.

5. Free Edge Analysis of the Interlaminar Microstructure

In order to understand the influence of the local microstructure on free edge cracking observed experimentally, a multiscale analysis was performed on the IM7/8552 [25/-25/90]_S composite laminate. It employed a computational homogenization framework, as outlined in the previous chapter. To reduce the computational complexity associated with implementing the fully coupled, semi-concurrent approach in 3D, the laminate investigation was performed using only one-way localization. Boundary conditions similar to those employed in Chapter 4 were utilized to allow for a free edge at the micro-scale. A one-way, multiscale approach was sufficient in determining important characteristics of the nature of the fiber/matrix stresses prior to failure, which will assume the composite and its constituents are linear elastic.

The multiscale analysis was used to address specific questions regarding the cause and influence of matrix cracks. Of particular interest was the influence of the local interlaminar microstructure on cracking during the manufacturing process, which was observed experimentally to occur at ply interfaces, the most susceptible being the 90°/90° ply interfaces. In addition, the multiscale modeling will be used to understand why initial pre-cracks did not influence damage development upon the application of mechanical loading.

5.1 Overview

In this work, the IM7/8552 laminated [25_N/-25_N/90_N]_S composite is modeled under both thermal and mechanical loading for N=1 and N=3 with similar dimensions to the specimens tested by Dustin (2012). The overall workflow of the multiscale approach is presented in Figure 5.1 showing the relevant length scales of the composite laminate problem as discussed in the Introduction. The components of the multiscale approach are outlined next.

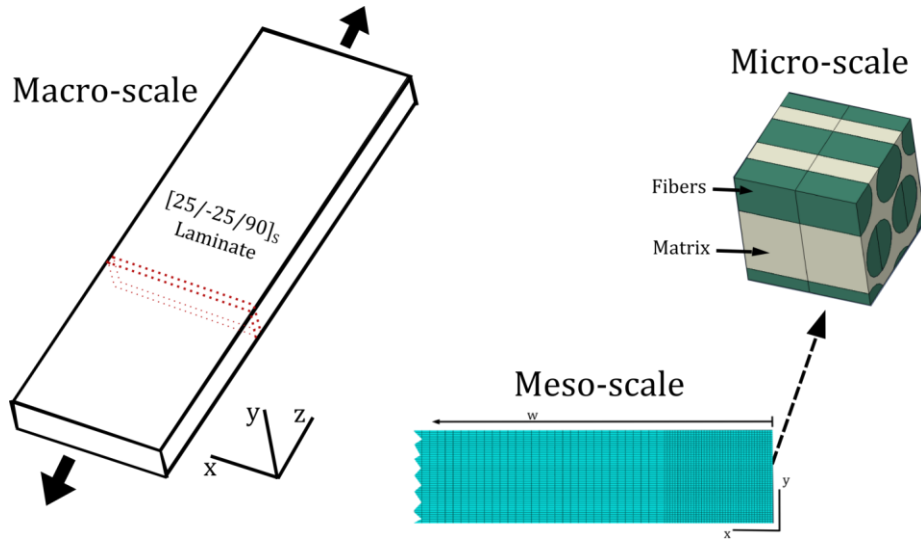


Figure 5.1: Workflow of the multiscale analysis of a composite laminate showing relevant length scales. The current approach utilizes meso-scale and micro-scale finite element models which are one-way coupled as shown by the dashed arrow. Deformations from the meso-scale are localized into boundary conditions on the micro-scale finite element model.

The macro-scale of the laminate problem assumed to be one of a homogeneous, yet orthotropic material based on the effective properties of the laminated composite which undergoes both uniaxial tensile loads and a uniform temperature change to account for residual stresses. As a result of the homogeneous material definition and simple loading at the macro-scale, the analysis would yield an expected uniform displacement gradient through the entirety of the macro-scale coupon. Due to the uniformity of displacements at the macro-scale, the finite element analysis begins with a sub-section slice model at the meso-scale as shown in Figure 5.1

At the meso-scale, the RVE consists of individual unidirectional (UD) lamina assumed to be homogeneous and orthotropic. It is used to capture the stacking sequence of the IM7/8552 composite laminate and the nature of the macroscopic loading, as shown in Figure 5.1. The meso-scale model would predict the deformation gradients present at the free edge at the homogeneous lamina level. The deformation results from the meso-scale model would then be

coupled to a micro-scale finite element model in order to determine the fiber/matrix level stresses.

The micro-scale finite element model explicitly represents the carbon fibers in a random, periodic array in the 90° plies. Various micro-models will be utilized to simulate a change in microstructure at the ply interfaces and will be discussed in detail later. The displacement boundary conditions of the micro-scale model are determined via a coupling to the deformation of a given integration point in the meso-scale model. The direction of the coupling is shown by the dashed arrow in Figure 5.1.

5.2 Scale Transitions

5.2.1 Kinematic Coupling

The one-way coupling is achieved through a kinematic relation between the meso- and micro- scale models. This kinematic relation was developed based on strain localization methods used for Computational Homogenization multiscale frameworks [42,90]. In this deformation driven finite element analysis, the strains at an integration point in the meso-scale model were used to determine the appropriate boundary conditions to apply onto the micro-scale finite element model.

A localization rule to provide the kinematic, one-way coupling was developed to relate the strain components at the meso-scale to nodal displacements for the micro-scale model. A simple 2D example of this strain-displacement relation is shown in Figure 5.2. A relation in the form of Equation (1) was proposed, where the strain components are related to prescribe displacements onto the RVE. In this way, the meso-scale strains obtained from an element integration point would provide the necessary displacements for the micro-scale model.

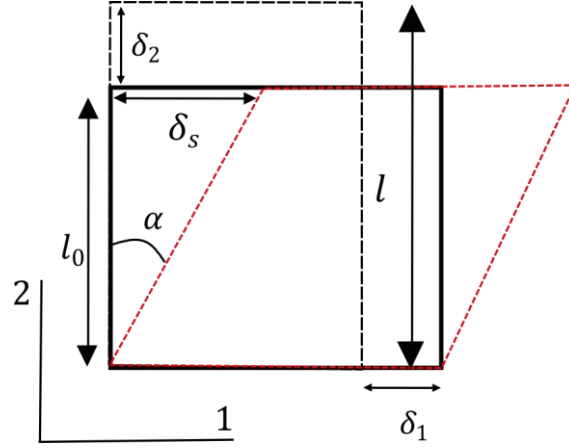


Figure 5.2: The definition of normal and shear displacements prescribed on a 2D RVE. The black dotted line represents tensile deformation prescribed by extensions δ_1 and δ_2 while the red dotted line shows a simple shear deformation according to δ_s .

$$\begin{aligned}\delta_t &= f(\epsilon_t), \text{ for } t=1, 2 \\ \delta_s &= f(\epsilon_s)\end{aligned}\tag{5.1}$$

Since the analysis of the laminate free-edge stress is a geometrically nonlinear problem, the large-displacement formulation was selected in ABAQUS for the static stress analysis. As a result, the default strain outputs from ABAQUS were the logarithmic strain tensor components. Thus, the localization rules, relating the meso-scale strains to micro-scale nodal displacements had to be adjusted according to the definition of logarithmic strain in one dimension,

$$\epsilon^{LOG} = \ln\left(\frac{l}{l_0}\right)\tag{5.2}$$

where ϵ^{LOG} is the logarithmic strain, and l and l_0 are the current and original length respectively. (Note: In the remainder of this paper, strain will always be reported using the logarithmic strain,

unless otherwise noted, and the superscript LOG will be dropped). The relation between extensional strains and applied displacement in the 2D example of Figure 4 is given by

$$\begin{aligned}\epsilon_1 &= \ln\left(\frac{l}{l_0}\right) = \ln\left(\frac{l_0 + \delta_1}{l_0}\right) \\ \epsilon_2 &= \ln\left(\frac{l}{l_0}\right) = \ln\left(\frac{l_0 + \delta_2}{l_0}\right)\end{aligned}\tag{5.3}$$

The relation between the shear displacement, δ_s , and logarithmic shear strain prescribes a logarithmic relation according to

$$\epsilon_s = \ln\left(\frac{l_0 + \delta_s}{l_0}\right)\tag{5.4}$$

where ϵ_s is equal to twice the tensorial component of logarithmic shear strain outputted from ABAQUS. The form of Equation (5.4) can be interpreted as a modification of standard relations between shear strain and RVE displacement as presented by Sun and Vaidya [33]. For the simple shear deformation in Figure 5.2, the total length extending in the 2-direction is simply δ_s .

In summary, the strain-displacement relations shown in Equations (5.3) and (5.4) represent the kinematic relations which couple the meso- and micro-scale models. The strains were obtained from integration points in the meso-scale model and then the corresponding displacements were defined for the micro-scale model. The exact form of the kinematic relations used in three dimensions for the micro-scale model will be presented later. In the next section, a validation of the strain-based localization rules will be addressed.

5.2.2 Validation of Strain Based Localization

First, a simple 2D, one element study was performed in order to determine the efficacy of using a strain-based approach in the localization between the meso and micro-scales. The workflow of this strain localization study is shown schematically in Figure 5.3. The variable LEXX represents the logarithmic strain output from ABAQUS in the XX direction (note: shear strain is outputted as double the tensorial component). Two test cases were run with the nodal displacements for each simulation shown in Table 5.1. The prescribed strains LEXX, outputted from an initial single element test, were compared to the resulting strains LEXX* after three separate localization rules were applied using either the deformation gradient in Equation (3.2) or the strain-based methods using Equations (5.3) and (5.4).

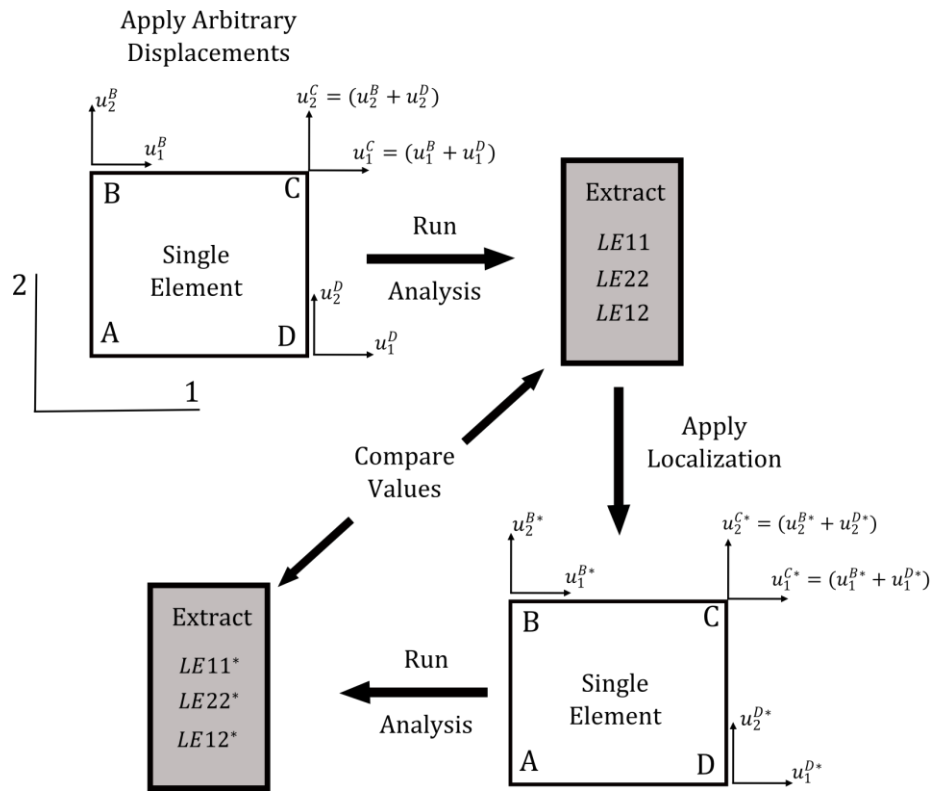


Figure 5.3: Workflow of a 2D single element study to determine the efficacy of strain based localization methods.

Table 5.1: The four nodal displacements for the two prescribed test cases used in the single element localization test. The distance d is the length of edge AB of the square RVE shown in Figure 5.3. Nodes B and D in the single element were fully prescribed, node A was fixed and node C followed periodic constraints.

Nodal Displacement	Loading Case	
	Combined Shear & Tensile Loading	Applied Shear
u_1^B	$0.05*d$	$0.05*d$
u_2^B	$0.1*d$	--
u_1^D	--	--
u_2^D	0	0

Figure 5.4 (a) and (b) compares the results of the three different localization methods against the originally prescribed strain. The results show that the deformation gradient method, as expected, was the most consistent in returning an element deformation where the computed strains were closest to the originally prescribed strain. The two alternative methods, however, were relatively close in both loading conditions. The strain based localization method varied significantly only in the case of the tensile direction loading case, where the normal components were over-predicted by 50%. It should be noted that these two components were relatively small in comparison to the larger shear component. Thus, the strain based localization methods proved to be a reliable tool in prescribing displacements where the deformation gradient is not readily available (in the case of using a built-in material model within ABAQUS).

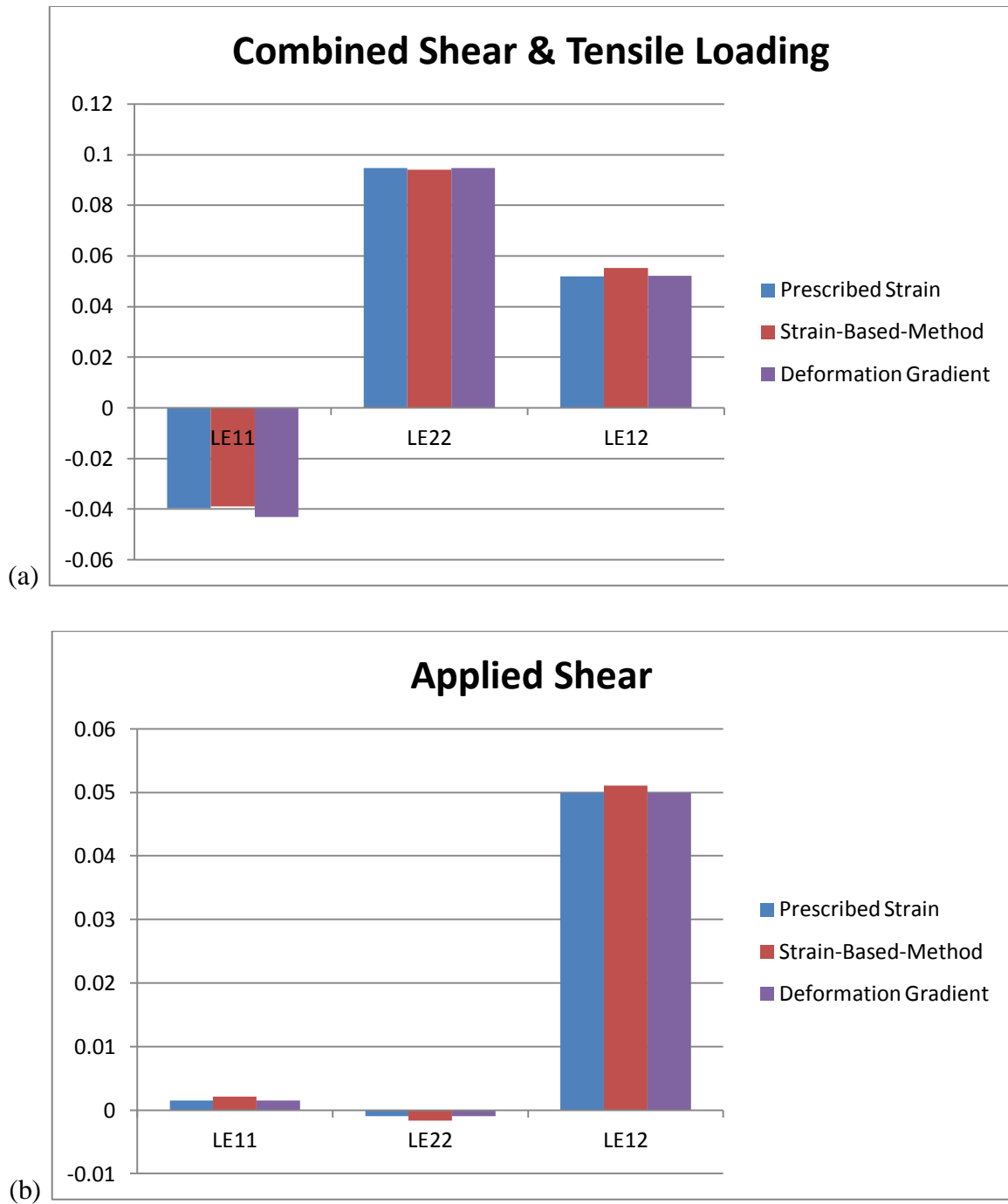


Figure 5.4: Comparison of resulting strain LEXX* for a variety of localization rules against the prescribed strain LEXX for (a) combined loading and (b) an applied shear.

5.2.3. Micro-scale Boundary Conditions

The boundary conditions enforced at the micro-scale are shown in the schematic of Figure 5.5, showing the assumed periodicity in the y and z directions and free edge in the negative x face. The left hand side of the micro-model, oriented to the interior (the positive x face), is prescribed a unique Periodic* boundary condition. On this face all nodes are prescribed a fixed displacement according to a periodic solution which is run prior to the free-edge analysis. The workflow of this process is shown in Figure 5.6.

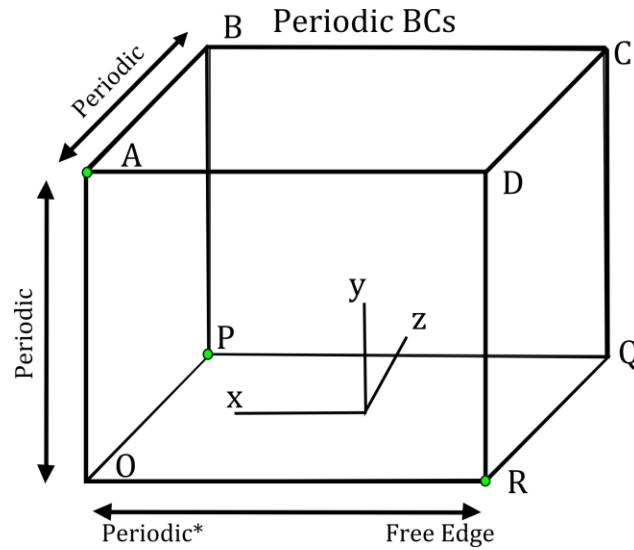


Figure 5.5: The applied boundary conditions for the free edge micro-scale analysis. The model is assumed periodic in the y and z directions. In the x -direction, the negative x face is assumed to be a free surface, while the positive x face is prescribed a unique Periodic* Dirichlet boundary conditions based on a periodic solution. The vertices of the micro-model are labeled A-D and O-R, which appear as superscripts in the displacement relations of Equations (8).

In step 1 of Figure 5.6, a purely periodic micro-model analysis is first run based on the displacements determined from the meso-scale solution. Upon completion of the periodic analysis, the nodal displacements on the entire positive x face of the micro-model are extracted.

The positive x face is the interior facing portion of the cubic micro-model. These nodal displacements are then set as a Dirichlet boundary condition on the positive x face of a free edge micro-model analysis, as shown in Figure 5.5.

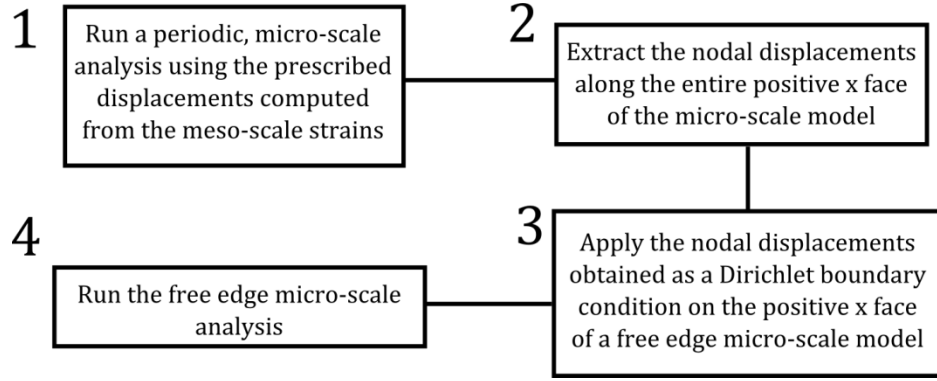


Figure 5.6: Process flow for the application of the Periodic* BC.

The nodal constraints used to prescribed periodic BC's initially, as in step 1 of 7, are shown in Equations (5.5),

$$\begin{aligned}
 u_i^O &= 0 \text{ for } i = x, y, z \\
 u_i^B &= u_i^A + u_i^P \text{ for } i = x, y, z \\
 u_i^D &= u_i^R + u_i^A, \text{ for } i = x, y, z \\
 u_i^Q &= u_i^R + u_i^P, \text{ for } i = x, y, z \\
 u_i^C &= u_i^R + u_i^A + u_i^P, \text{ for } i = x, y, z \\
 u_i^{Z+} - u_i^{Z-} &= u_i^P, \text{ for } i = x, y, z \\
 u_i^{X-} - u_i^{X+} &= u_i^R, \text{ for } i = x, y, z \\
 u_i^{Y+} - u_i^{Y-} &= u_i^A, \text{ for } i = x, y, z
 \end{aligned} \tag{5.5}$$

where the superscripts A through D and O through Q above correspond to the nodes specified in Figure 8. The localization rules, based on the 3D extension of Equations (5.3) and (5.4) are

$$\begin{aligned}
u_x^P &= (e^{\epsilon_{xz}} - 1) * l_0 \\
u_y^P &= (e^{\epsilon_{yz}} - 1) * l_0 \\
u_z^P &= (e^{\epsilon_{zz}} - 1) * l_0 \\
u_x^R &= (e^{\epsilon_{xx}} - 1) * l_0 \\
u_y^R &= (e^{\epsilon_{xy}} - 1) * l_0 \\
u_y^A &= (e^{\epsilon_{yy}} - 1) * l_0 \\
u_z^R &= u_x^A = u_z^A = 0
\end{aligned} \tag{5.6}$$

where ϵ_{ij} corresponds to the logarithmic strain component of the corresponding meso-scale integration point. Nodes at the vertices of the micro-scale RVE are fully constrained, while the nodes on the faces are constrained to relative displacements based on periodicity. Once the periodic solution is completed, the displacements of all nodes on the positive x surface are obtained. The boundary conditions and prescribed displacement on the free edge micro-scale problem are reduced to

$$\begin{aligned}
u_i^{X+} &= u_i^{X+(Periodic)} \text{ for } i = x, y, z \\
u_i^D &= u_i^R + u_i^A, \text{ for } i = x, y, z \\
u_i^Q &= u_i^R + u_i^P, \text{ for } i = x, y, z \\
u_i^C &= u_i^R + u_i^A + u_i^P, \text{ for } i = x, y, z \\
u_i^{Z+} - u_i^{Z-} &= u_i^P, \text{ for } i = x, y, z \\
u_i^{Y+} - u_i^{Y-} &= u_i^A, \text{ for } i = x, y, z \\
u_y^R &= (e^{\epsilon_{xy}} - 1) * l_0 \\
u_z^R &= 0
\end{aligned} \tag{5.7}$$

where the x direction periodicity is removed and the displacement of node R, shown in Figure 5.5, is left unspecified in the x-direction (creating a traction free boundary at the X^- surface).

5.2 Meso-scale Model

The meso-scale model of the $[25_N/-25_N/90_N]_S$ laminate is shown in Figure 5.7 for the case of $N=1$. The width of all laminates investigated in this work was set to 25 mm and the thickness of each lamina was 0.1616 mm. The model is assumed to be periodic in the z-direction to drastically reduce the size of the meso-scale model to a “slice”. The depth of the model in the z-direction was set to 0.018 mm. The periodic assumption is valid for regions sufficiently far from the gripped ends of the coupon. It should be noted that the entire 25 mm width of the meso-scale model for $N=1$ is not shown in Figure 5.7.

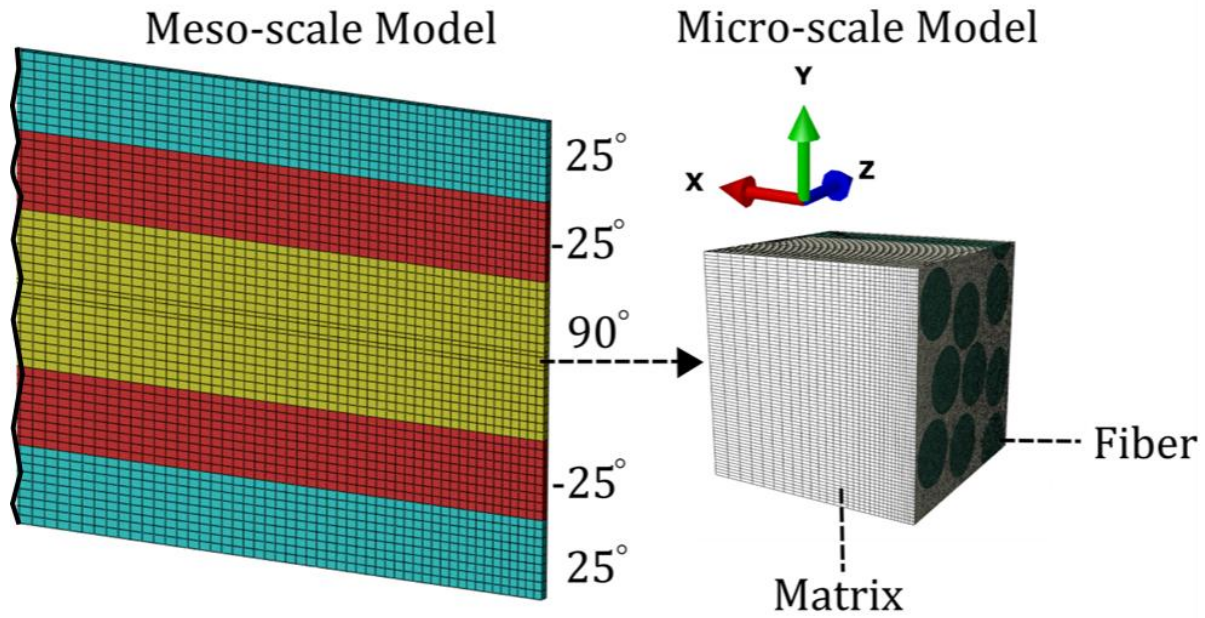


Figure 5.7: A portion of the meso-scale finite element model for the case of $N=1$ is shown. The meso-scale model is color coded to identify the unidirectional ply orientations. The meso-scale model is truncated in the figure. The micro-scale model associated with the midplane interface element is shown at the right.

The meso-scale model consists of homogeneous, orthotropic unidirectional plies modeled using C8DRT brick elements. The lamina orientations are shown in Figure 5.7 by color coding, and the finite element mesh shown for the $N=1$ case. The density of the finite element mesh was chosen such that elements near the free edge were close in dimensions to that of the entire micro-scale finite element model (0.018 mm x 0.018 mm x 0.018 mm). This requirement was strictly enforced at the areas of interest (e.g., the lamina boundaries) and was prescribed user-defined edge seeding throughout.

The purpose of enforcing the meso-scale mesh density to be equal to the size of the micro-scale was to ensure that the kinematic coupling provided reasonable strain measures to be coupled to the micro-scale model. Due to the singular nature of the meso-scale stress/strain fields at the free edge, continued mesh refinement would result in higher strains in elements at the free

edge and near the lamina interfaces. Since these strains are then coupled into the displacements of a finite size micro-scale model, limiting the mesh density to the size of the micro-model ensures that unrealistically large meso-scale strains are not coupled down to the micro-scale finite element model.

The same mesh density, laminate width and thickness were used for the N=3 meso-scale FE mesh as shown in Figure 5.7. The total laminate height is 4.848 mm. In the case of the N=3 laminate, three different 90/90 ply interfaces are present. The three 90/90 ply interfaces are here on referred to as the midplane, second and third interface, beginning with the laminate midplane and proceeding in the positive y direction as shown in Figure 5.8. Again, it is enforced that an element exists exactly at the interface location, such that the micro-model results can be obtained from localizing the meso-scale strains at these interface elements.

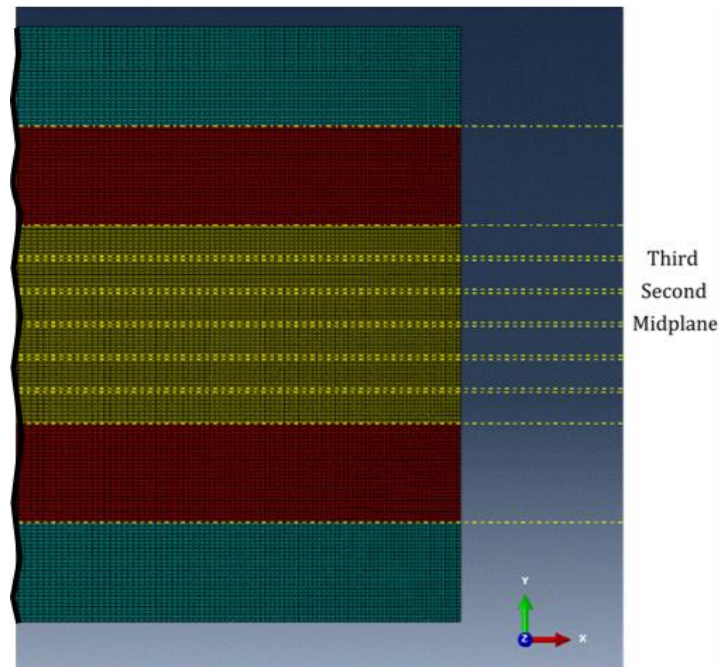


Figure 5.8: The meso-scale finite element model for the N=3 laminate. The meso-scale model is color coded to identify the unidirectional ply orientations. The meso-scale model is truncated in the figure.

5.3 Micro-scale Model

In order to investigate the effect of the microstructure at the interface of 90/90 ply interfaces, a set of modified micro-models are developed with varying amounts of pure resin rich regions along the center line of the model. It was shown by Andrew and Garnich that there is an inverse relationship between the free edge fiber/matrix stresses (located at the dissimilar material interface and free edge) and local fiber volume fraction [73]. Thus, the original micro-model, shown previously in Figure 5.6, was modified by adding a layer of resin thereby increasing the micro-model's size in the y-direction. This pure resin represents a finite thickness interlaminar region between neighboring plies, which can be identified in optical micrographs of the laminated composites [31].

Three different interlaminar micro-models are shown in Figure 5.9 (a)-(c) which have total fiber volume fractions of 55%, 48% and 44% respectively. They all consisted of an explicit finite element representation of 9 IM7 carbon fibers in the 8552 epoxy matrix. The fibers were randomly distributed, and the micro-model was assumed to be periodic in the y and z directions. The entire model consisted of C3D6T wedge elements with enhanced refinement approaching the model free edge. The two micro-models differ only by the spacing between the top 5 and lower 5 fiber instances, used to represent a realistic matrix-rich ply interface. The three different micro-models will be used to understand the effect of the local interlaminar microstructure on free edge micro-stresses. The same boundary conditions are used for all three micro-models.

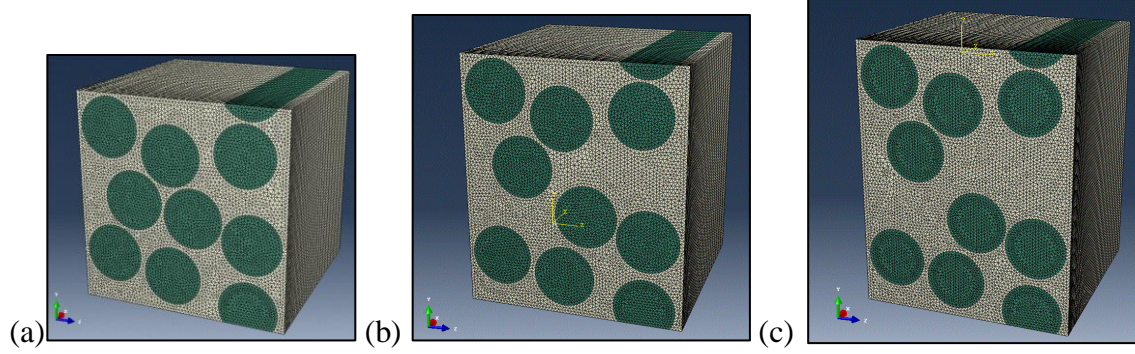


Figure 5.9: (a) 55% fiber volume fraction micro-model, (b) 48% fiber volume fraction micro-model and (c) 44% fiber volume fraction micro-model.

5.4 Model Parameters

At the meso-scale, the lamina properties are obtained by homogenizing the fiber and matrix mechanical and thermal properties with the respective fiber volume fraction and utilizing the MAC/GMC micromechanics software [91]. This homogenization procedure was repeated for the 55% fiber volume fraction of the overall composite, and also re-computed for the varying interlaminar elements which will have a locally reduced fiber volume fraction of 48% or 44% depending on the investigated interlaminar interface thickness. The properties utilized in this work are presented in Table 5.2.

Table 5.2: Constituent and homogenized lamina mechanical and thermal properties. The constituent properties were obtained from Dustin and Pipes [76], while the lamina properties were obtained using the MAC/GMC micromechanics software.

		E_L , GPa	E_T , GPa	ν_{LT}	G_{LT} , GPa	G_T , GPa	α_L (10- 6/°C)	α_T (10- 6/°C)
Fiber	IM7	276	19.5	0.27	70	5.74	-0.4	5.6
Matrix	8552	4.67	--	0.34	1.74	--	50	50
Matrix (Relaxed)	8552	3.51	--	0.38	1.27	--	50	50
Lamina	Vf = 55%	153.9	9.71	0.305	7.37	5.87	0.3	29.3
Lamina (Relaxed)	Vf = 55%	153.4	8.882	0.309	5.91	2.58	0.3	29.3
Lamina	Vf = 47%	134.4	9	0.303	5.87	2.85	0.3	29.3
Lamina (Relaxed)	Vf = 47%	133.8	7.67	0.32	4.6	2.3	0.3	29.3
Lamina	Vf = 44%	124.1	8.56	0.305	4.5	2.7	0.3	29.3
Lamina (Relaxed)	Vf=44%	123.4	7.24	0.325	4.12	2.1	0.3	29.3

Since free-edge cracking was present prior the application of extensional loading, it was important to properly account for the residual stresses which could be present from thermal cool down during manufacturing. As with any thermoset composite, the neglect of viscoelastic effects has been reported to overestimate the residual stresses in the laminated composite by 20% [92]. It is typically assumed that there is a relaxation in the shear response of the thermoset resin, provided by shear relaxation tests [93]. Due to the unavailability of an orthotropic, viscoelastic material model in ABAQUS, a simpler method to account for the shear relaxation of the matrix is considered and is utilized to modify both the micro-scale matrix, and meso-scale homogenized lamina properties.

To account for relaxation of the matrix shear modulus, an “effective” matrix modulus is considered whereby a long-term relaxation of the matrix shear modulus is assumed. The long-term relaxation of the matrix shear modulus is assumed to be 27% based on storage modulus plots presented by Miyano for the 8552 resin [94]. The resultant relaxed elastic properties for the matrix are obtained by reducing the shear modulus of the matrix, preserving the bulk modulus of the material, and computing the consistent Young’s modulus and Poisson’s ratio for isotropic elasticity. The relations for computing the relaxed Poisson’s ratio and Young’s modulus are shown in Equations (5.8) and (5.9), where K is the bulk modulus, G is the shear modulus, E is the Young’s modulus and μ is the Poisson’s ratio. The superscript R represents the relaxed property.

$$\mu^R = \frac{3K - 2G^R}{2 * (3K + G^R)} \quad (5.8)$$

$$E^R = 2 * G^R * (1 + \mu^R) \quad (5.9)$$

In the simulation of the composite laminates in this work, the residual stresses are computed at both the lamina and micro-scale utilizing the relaxed lamina properties. Accounting for the matrix relaxation will ensure that the local matrix stresses at the free edge during the manufacturing phase are not overestimated in the current analysis. When mechanical extension of the composite laminate is considered, however, the instantaneous (unrelaxed) properties for both the lamina and matrix are utilized.

When the size of the interlaminar thickness is considered in the finite element simulations, the local element at the meso-scale which represents the 90/90 ply interface will use the appropriate lamina level properties. For example, when the interface is assumed to have a local fiber volume fraction of 44%, the properties listed in Table 1 for the given fiber volume

fraction are assigned to the row of elements at the ply interface. Thus, the influence of increased resin at the interlaminar interface is both accounted for by utilizing the appropriate micro-model and adjusting the meso-scale lamina properties to reflect the reduced elastic properties of the interface.

5.5 Finite Element Results

Three separate investigations were performed in this work and the results are presented and discussed for each one separately. The first investigation studied the effect of the interlaminar thickness by modeling the N=1 laminate whereby the 90/90 ply interfaces is examined using the three different micro-models which vary in the size of the pure resin rich interface. In this first study, the laminate experiences only the thermal loading from manufacture. The second investigation examines the effect of the meso-scale free edge deformations on the local micro-scale stresses and compares that effect to the influence of the interlaminar thickness. In this second study, the N=3 composite laminate will be examined under thermal loading at all three unique 90/90 ply interfaces. The 55% and 48% micro-models will be used to compare the influence of meso-scale strains versus interlaminar thickness. The third investigation will re-examine the N=3 composite laminate as in the second investigation; however, the loading on the laminate will be purely mechanical extension in the z-direction. In all cases, the simulation results will be used to understand to Dustin's experimental observations on the laminate [31].

5.5.1 Effect of the Interlaminar Thickness during Thermal Cooldown

Investigation I looked at the effect of interlaminar thickness on the development of the residual free edge stresses at the 90/90 ply interface of the N=1 composite laminate. A thermal

loading of $\Delta T = -150^\circ\text{C}$ was applied to the IM7-8552 laminate, and the resulting meso-scale strains were localized onto the micro-model. The three interface micro-models were considered, which varied in the proportion of pure resin regions located at the center of the micro-model. The elements at the 90/90 ply interface at the meso-scale were updated to account for the reduced fiber volume fraction in the case of the 48% and 44% fiber volume fraction interface micro-models.

Figure 5.10 plots the free edge stresses at the lamina level in the y-direction, which represents the peeling stresses in the laminate. As expected, the highest concentration of free edge peeling stresses are located at the dissimilar ply interface, however, in this study the midplane 90/90 ply interface will be examined. Figure 5.10 also shows the logarithmic strains of the meso-scale element at the 90/90 ply interface for all three micro-models. Note that the strains change due to the differing mechanical properties across the three different interlaminar thicknesses examined. These strains will be used to localize the micro-model for the 55%, 48% and 44% fiber volume fraction models. At the micro-scale, not only are the applied displacements given by the meso-scale strains prescribed, but also the thermal cooldown boundary condition where the temperature is assumed to change uniformly at both the meso- and micro-scales.

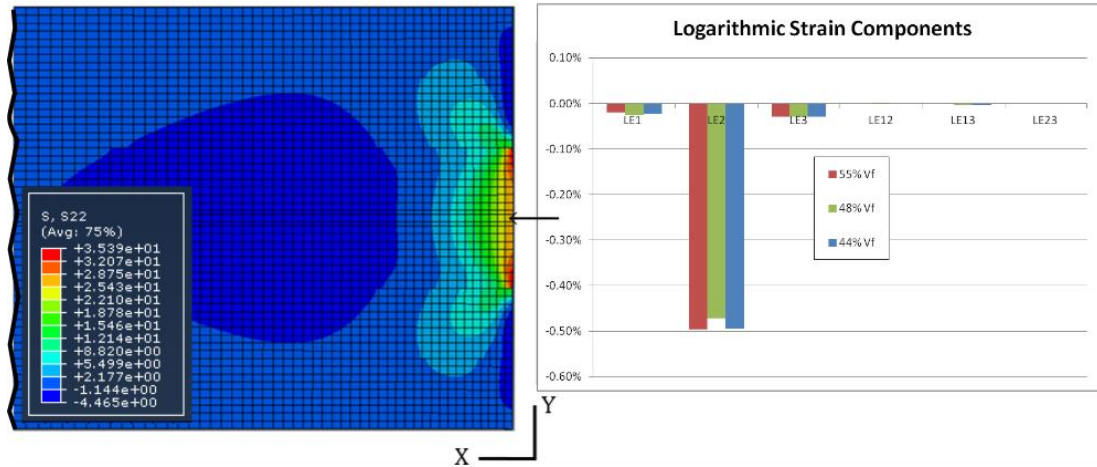


Figure 5.10: The contour results (units of MPa) are shown for the residual free edge stresses in the y-direction for the N=1 laminate and 55% Vf micro-model. The logarithmic strain components extracted from the midplane 90/90 interface element are shown on the right for all three interlaminar thicknesses. Due to the thermal contraction, the strains are negative, however the stresses are positive in the laminate. The 1, 2 and 3 component directions correspond to the global x, y and z directions.

With the aim of comparing the three individual interface micro-models, contour plots were generated which highlighted in red the elements whose maximum principal stress had exceeded the reported matrix tensile strength of 99 MPa. Figure 5.11 shows an example of this contour plot of maximum principal stress for the 55% fiber volume fraction micro-model. The face shown in the negative x direction is at the laminate free edge and fibers were removed for clarity. Figure 5.11 demonstrates the localized nature of the free edge matrix stresses, which takes on high values above the matrix tensile strength at the interface of the fiber and matrix at the free edge.

55% Vf Micro-model

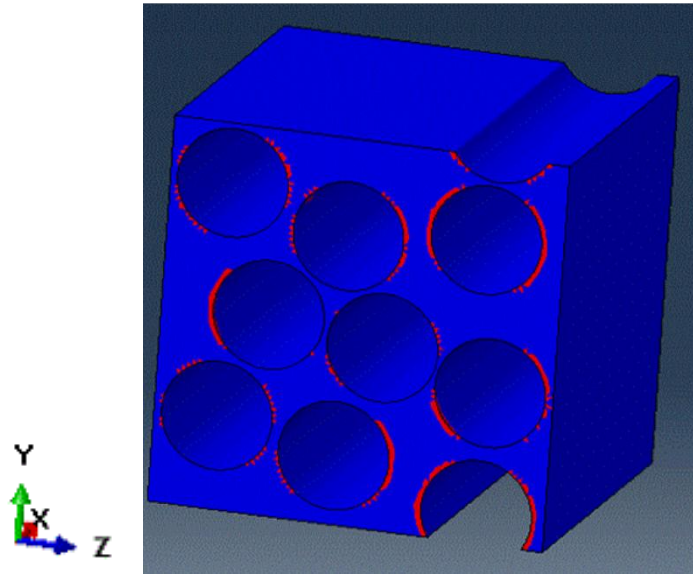


Figure 5.11: The contour results are shown for the residual free edge maximum principal stresses in the matrix elements (fibers are hidden) for the 55% micro-model. Elements whose maximum principal stresses have exceeded 99MPa are highlighted in red.

These localized free edge matrix stresses seen in Figure 5.11 tend to be located at the edges of the fiber the fiber matrix interface. The localized nature of the highest matrix stresses is a result of the singularity present at the free edge due to the material discontinuity [29,73]. In addition, the location of the highest maximum principal stresses at the free edge occurred near matrix rich regions. For example, regions with proximal fibers tended to exhibit fewer red elements versus regions with greater fiber spacing. The location of the highest matrix stresses were on the edges of the fibers facing the z-direction.

Figure 5.12 compares the results for all three interface micro-models with the free edge face shown. The increase in interlaminar thickness due to the additional resin in the 48% micro-model resulted in a 45% increase in the highest reported maximum principal stress value. In addition to the increased maximum value, the number of elements which have exceeded the

threshold stress of 99 MPa increased significantly between the 55% and 48% micro-models. In the 48% micro-model, the critical matrix elements nearly encompass the entire boundary of the fiber at the free edge. A similar contour plot to the 48% was predicted for the 44% micro-model case; however, the highest maximum principal stress was reported slightly lower at 252 MPa. The results in Figure 5.13 highlight the impact of additional resin in the interlaminar micro-model to simulate increased interface thickness. There is a clear increase in the highest free edge stresses in the matrix that is localized at the fiber/matrix boundary. It is not clear from the three test cases examined in this work if additional resin would result in continued increases in matrix free edge stresses. Additional micro-models with varying interlaminar thicknesses would be required to determine whether a threshold exists whereby increasing matrix would continue to cause increased matrix free edge stresses.

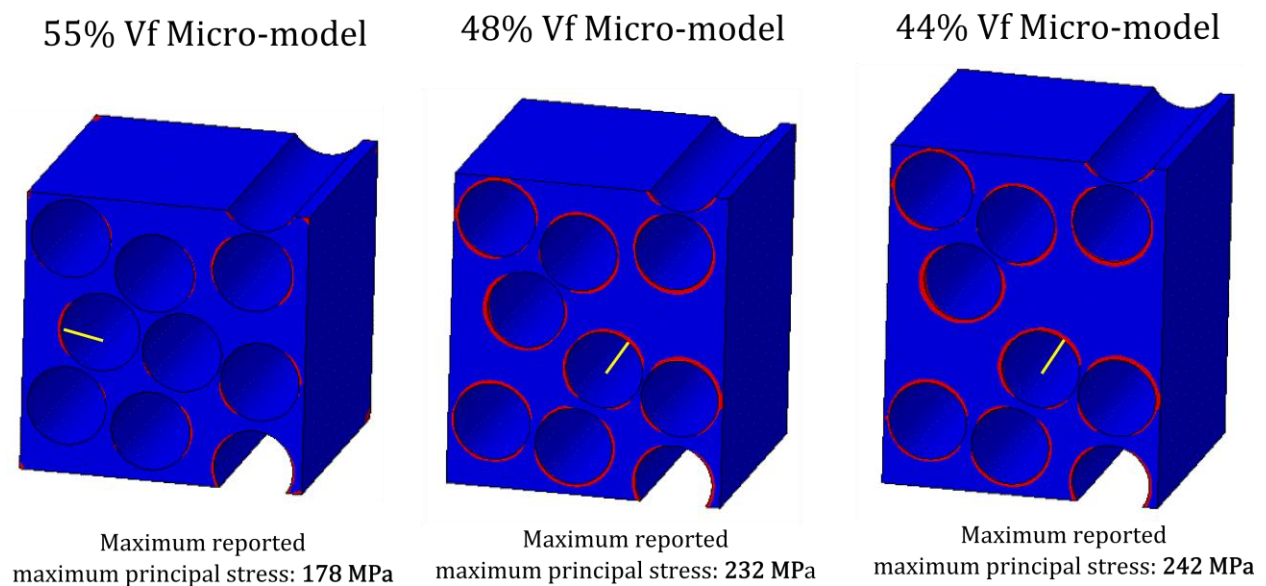


Figure 5.12: The contour results are shown for the residual free edge maximum principal stresses in the matrix elements (fibers are hidden) for all three interface micro-models. Elements whose maximum principal stresses have exceeded 99MPa are highlighted in red. The face of the micro-model shown is at the free edge of the laminate.

Full contour plots of the maximum principal stress for all three micro-models are shown in Figure 5.13. These contour plots emphasize the general stress relaxation that occurs in the matrix at the free edge away from fibers, shown by the concentration of blue regions at the free edge face. As expected, it is only in localized regions near the fiber/matrix boundary that the highest matrix stresses are concentrated. There was also an observed general increase in the maximum principal stresses of the matrix comparing the contours of the 55% Vf micro-model and the 48% micro-model. This was likely due to the difference in meso-scale strains localized for the two interlaminar micro-models (seen in Figure 5.10).

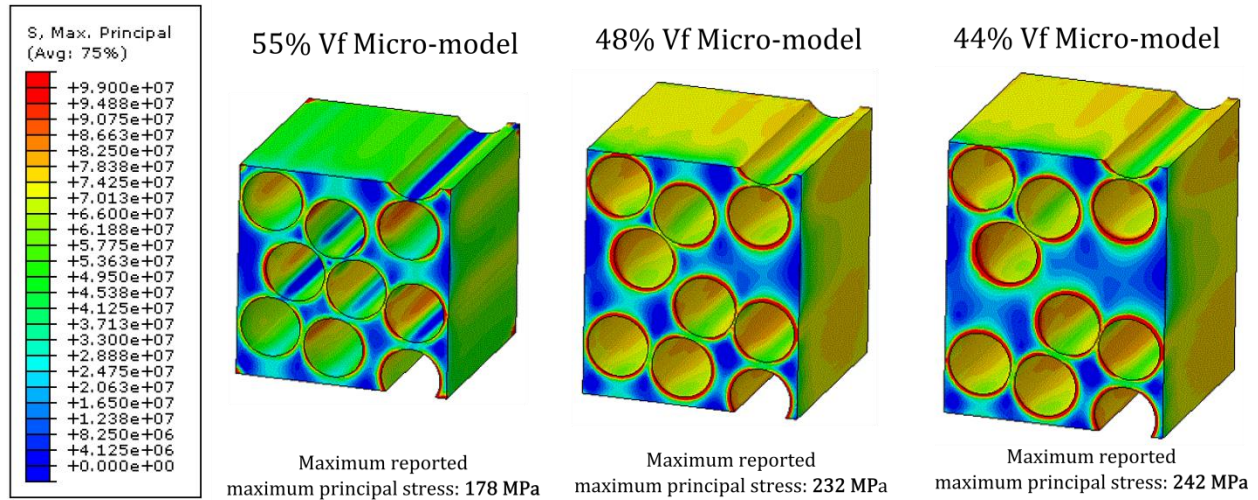


Figure 5.13: Contour plots are shown for the residual free edge maximum principal stresses in the matrix elements (fibers are hidden) for all three interface micro-models. The contour legend is shown (units of Pa).

The contour plots utilize an upper and lower limit, set at 99MPa and 0 MPa, respectively, to capture the variation of maximum principal stresses within the entire micro-model. Values above or below the specified limits are colored red or blue, respectively. The face of the micro-model shown is at the free edge of the laminate.

Figure 5.14 shows the contour plot for the 48% fiber volume fraction micro-model with an increase in the threshold stress for which matrix elements are highlighted in red. The

threshold stress is set to 182 MPa, to identify the elements whose maximum principal stresses have exceed the highest value reported for the 55% micro-model. The concentrations of red elements in Figure 5.14 show a preference for fiber boundaries facing the resin-rich interlaminar region at the center of the micro-model. These locations of highest maximum principal stresses are in contrast with the orientation of the highest regions in the 55% micro-model in Figure 5.12.

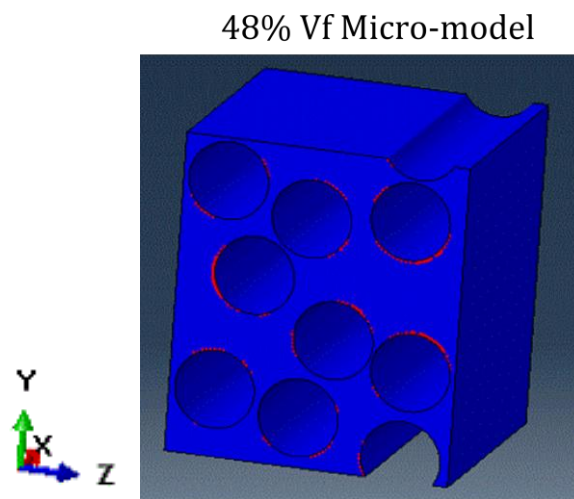


Figure 5.14: The contour results are shown for the residual free edge maximum principal stresses in the matrix elements (fibers are hidden) for the 48% micro-model. Elements whose maximum principal stresses have exceeded 182MPa are highlighted in red.

From the results in Investigation 1, three important observations were made. First, it was found that the highest free edge matrix stresses were observed at the fiber-matrix boundary in resin rich regions. Second, the introduction of pure resin into the interlaminar interface micro-models resulted in an increase in the free edge matrix stresses. Third, the highest matrix stresses in the 48% and 44% micro-models were oriented facing the resin-rich sections of the micro-models.

These three observations imply that free-edge cracking is likely to occur at the ply interfaces where the composite laminate is likely to see increased resin rich regions. This result was observed by Dustin for all cases of $N=1, 3$ and 5 at the $90/90$ ply interfaces [31]. The simulation confirms that there is a higher propensity for cracking near ply interfaces with a distinguishable matrix interface, than at the interior of a lamina.

Additionally, the orientation of the highest matrix stresses which face the resin-rich interface is supported by the experimental observations of crack orientation. The majority of cracks observed by Dustin at the $90/90$ ply interfaces were predominantly oriented in the z -direction. The contour plots in Figure 14 imply that the greatest propensity for crack initiation is likely to occur along the fiber/matrix boundary in-line with the ply interface.

5.5.2 Influence of Meso-Scale Strains on Free Edge Micro-stresses

Investigation II explores the sensitivity of the local free edge micro-scale stresses to the meso-scale strains at various $90/90$ ply interfaces in the $N=3$ composite laminate. The three interlaminar $90/90$ ply interfaces, labeled midplane, second and third interface are explored at the micro-scale. In this investigation, the residual stresses during thermal cool down are again considered to be the meso-scale loading.

Both the 55% and 48% fiber volume fraction micro-models are compared at all three locations in order to compare the influence of the interlaminar thickness versus that of the meso-scale strains which vary as the ply interface location changes. Figure 5.14 graphs the logarithmic strain components at all three interfaces, labeled as *Midplane*, *Second* and *Third*, and corresponding to the interfaces identified on 5.8. The identifier *Thick* is appended to the lamina

models whereby the interface has the increased matrix regions, corresponding to the 48% fiber volume fraction micro-model.

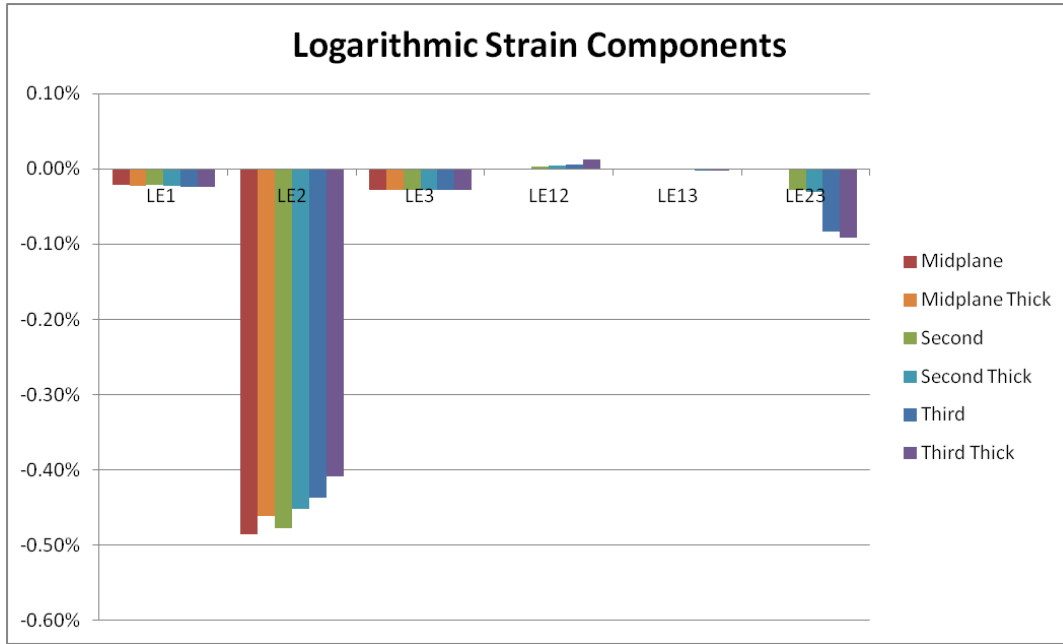


Figure 5.14: The logarithmic strain components at the meso-scale for the N=3 laminate at the three 90/90 ply interfaces. The 1, 2 and 3 component directions correspond to the global x, y and z directions.

The strain components in Figure 5.14 indicate an increase in the through-thickness peeling strain, labeled LE2, as one moves from the midplane interface to the interface closest to the dissimilar -25/90 interface. There is also an increase in the interlaminar shear strains (LE23) as one proceeds closer to the dissimilar ply interface. Additionally, the reduced elastic properties of the 48% Vf lamina at the interface causes an increase in the meso-scale strains, as shown by comparing the interface elements to their respective *Thick* counterparts.

All 6 of these micro-models, representing the three different 90/90 ply interfaces and two different interlaminar thicknesses are compared similar to the procedure done for investigation one. Thus, the three different ply interface strains from the meso-scale are localized into

boundary conditions onto the corresponding interlaminar micro-model. The thermal cooldown, along with the boundary displacements are prescribed onto the finite element micro-model.

The contour plots for all three ply interface locations for the 55% fiber volume fraction micro-model interface are shown in Figure 5.15. All three locations ply interface locations exhibit minimal change between the three locations, despite the increase in meso-scale strains as the interfaces approach the -25/90 dissimilar ply interfaces. For example, the increase in highest reported maximum principal stress between the *Midplane* interface, and that of the *Third* is only a fraction of a percent. In addition, the introduction of interlaminar shear strains at the meso-scale has little influence to the orientation of the highest matrix stresses, which are located at the fiber boundary facing the z-direction. The full color contour plots shown in Figure 5.16 show a similar result to those of the 55% Vf micro-model from Investigation I (Figure 5.13).

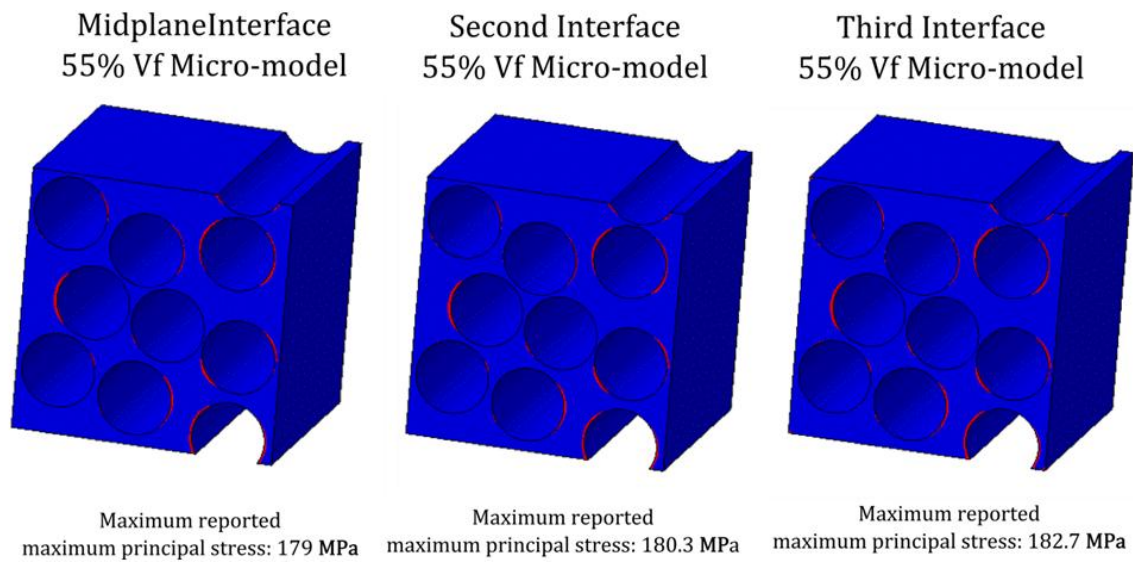


Figure 5.15: The contour results are shown for the residual free edge maximum principal stresses in the matrix elements (fibers are hidden) for all three ply interfaces for the 55% fiber volume fraction micro-model. Elements whose maximum principal stresses have exceeded 99MPa are highlighted in red. The face of the micro-model shown is at the free edge of the laminate.

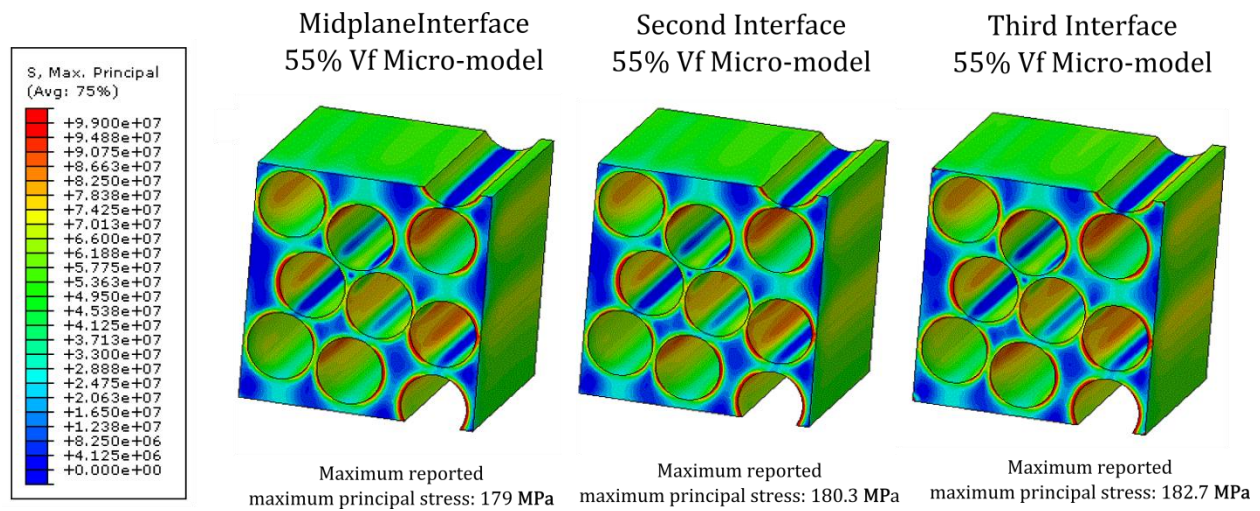


Figure 5.16: Contour plots are shown for the residual free edge maximum principal stresses in the matrix elements (fibers are hidden) for the 55% Vf micro-model at the three 90/90 pl interfaces. The contour legend is shown (units of Pa). The contour plots utilize an upper and lower limit, set at 99MPa and 0 MPa, respectively, to capture the variation of maximum principal stresses within the entire micro-model. Values above or below the specified limits are colored red or blue, respectively. The face of the micro-model shown is at the free edge of the laminate.

The threshold stress contour plots for the 48% micro-model are shown in Figure 5.17 for the three interface locations. Even for the thick interlaminar interface micro-model, there was negligible change in the contours and highest maximum principal stress values between the three 90/90 ply interface locations. The locations of the highest matrix stresses in these micro-models are oriented facing the resin rich region of the micro-model.

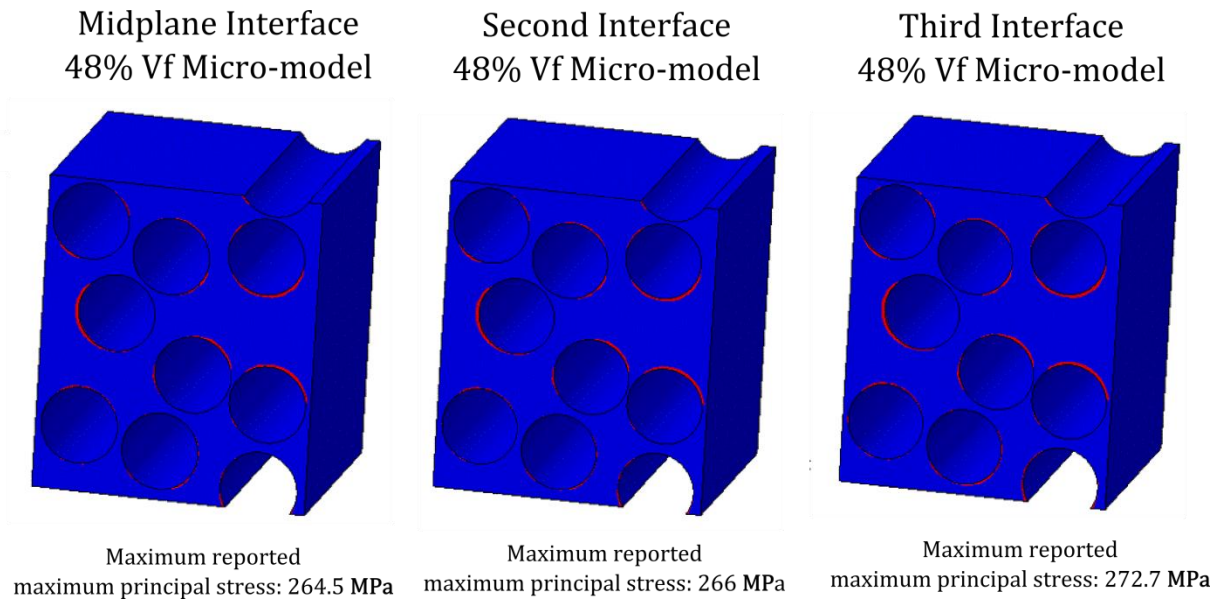


Figure 5.17: The contour results are shown for the residual free edge maximum principal stresses in the matrix elements (fibers are hidden) for all three ply interfaces for the 48% fiber volume fraction micro-model. Elements whose maximum principal stresses have exceeded 99MPa are highlighted in red. The face of the micro-model shown is at the free edge of the laminate.

The results from this second investigation highlight the importance of local microstructural variation in the micro-models. The insignificant changes in free edge stresses between the varying ply interface locations suggests that the variation in meso-scale free edge strain play only a small role in causing free edge cracking. The large difference in matrix free edge stresses between the 55% and 48% micro-models suggest that the local resin rich layers between lamina are a stronger driver for cracking due to residual stresses.

These results support the experimental observations seen by Dustin in the N=3 and N=5 laminates where 90/90 ply cracking was seen dispersed through the thickness of the laminate [31]. For example, even the N=5 laminate had observed free edge cracking at the midplane interface. The numerical analysis from this investigation suggest that cracking, as a result of high free edge matrix stresses, is likely to occur at the location of resin rich interfaces. These cracks

will tend to oriented with the z-axis and are equally likely at the *Midplane* or the *Third* interface closest to the dissimilar ply interface.

5.5.3 Effect of Pure Mechanical Loading

Investigation III repeats the study of the three ply interfaces of the N=3 laminate, however, the applied loading to the laminate is a mechanical extension in the z-direction with no thermal cool down. The purpose for isolating only the mechanical contribution to free edge micro-stresses to those obtained from the thermal cool down is to distinguish the influence of the laminate extension on further cracking of the free edge cracks observed during manufacture. A mechanical extension of 0.1% to 0.3% strain is applied to the meso-scale finite element model in the z-direction. Subsequently, the resulting meso-scale strains at the various 90/90 ply interface elements are localized onto their respective micro-models.

Figure 5.18 graphs the logarithmic strain components for the three different locations and two different interface micro-models. As with Investigation 2, the Thick identifier specifies the use of the 48% micro-model. One can observe the expected increase in free edge meso-scale strains approaching the dissimilar -25/90 ply interface. As with the thermal loading, the mechanical loading also exhibits an increase in meso-scale strains between the 55% fiber volume fraction interface and the 48% fiber volume fraction interface.

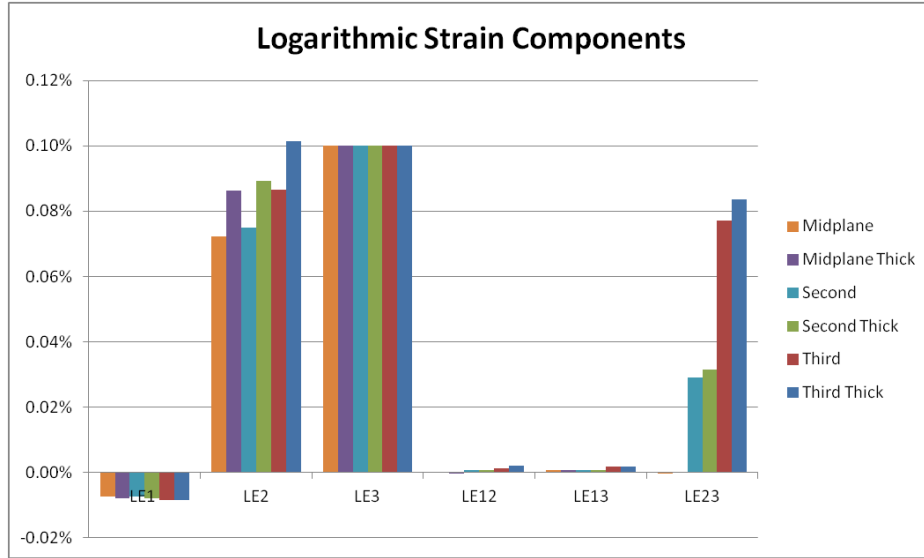


Figure 5.18: The logarithmic strain components at the meso-scale for the N=3 laminate at the three 90/90 ply interfaces as a result of a mechanical extension of 0.1% strain. The 1, 2 and 3 component directions correspond to the global x, y and z directions.

Figure 5.19 plots the maximum principal stress of the matrix elements for the 55% micro-model at the three various 90°/90° ply interfaces at an applied extensional strain of 0.3%. Again, red elements indicate integration points whose maximum principal stress has exceed 99 MPa, the tensile strength of the matrix. There are two interesting results in the contour plots of Figure 5.19. First, there is also a noticeable decrease in the highest reported maximum principal stress across the three micro-models moving from the midplane to the third interface. This maximum value is reported at the same location in all three micro-models: at the interior, away from the free-edge between two proximic fibers. Second, although the number of highlighted matrix elements at the free edge increase slightly approaching the dissimilar ply interface (reading the plots left to right), the overall number goes down, due to decreasing elements at the interior of the micro-model.

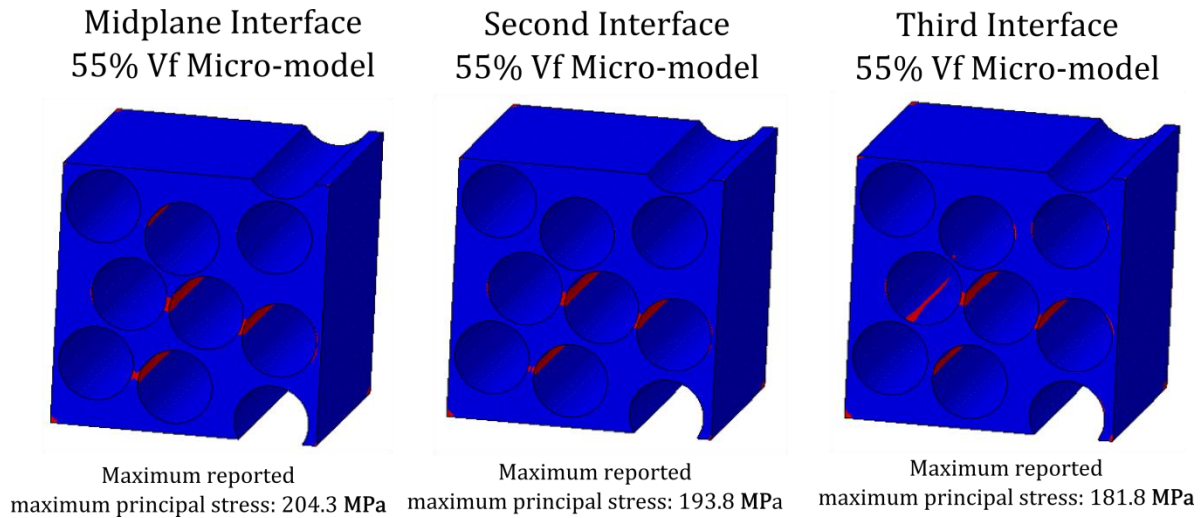


Figure 5.19: The contour results are shown for the free edge maximum principal stresses as a result of a 0.3% mechanical extension in the z-direction for the 55% interface micro-model. Elements whose maximum principal stresses have exceeded 99MPa are highlighted in red. The face of the micro-model shown is at the free edge of the laminate

Although the contour plots in Figure 5.19 show some elements at the free edge which have exceed the threshold value, regions of significantly higher matrix stresses are found between proximic fibers. These regions extend through the entire x-axis of micro-model, indicating matrix failure due to these localized stresses would likely results in significant failure in the micro-model. These regions of high matrix stresses between proximic fibers are orientated perpendicular to the z-direction, indicating a propensity for transverse ply cracking. This contrasts with the highly localized stresses seen only at the free-edge in the case of thermal loading, which were preferentially aligned with the ply interface.

Figure 5.20 shows a contour plot for the maximum principal stress for all three interfaces and the 55% Vf micro-model. Although the maximum principal stress at the free edges are decreasing when moving from the Midplane to the Third interface, there are regions at the

interior of the micro-model between proximal fibers where the matrix stresses are increasing. An example of this region is shown by the black arrow on the Third interface micro-model. The increase in matrix stresses in this region is likely due to the increasing interlaminar shear strains at the meso-scale.

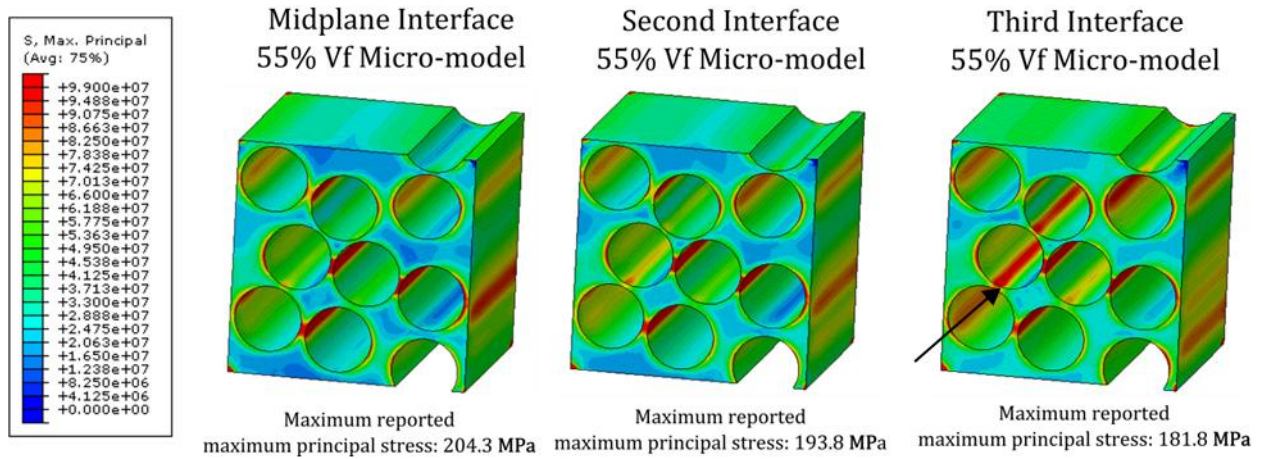


Figure 5.20: Contour plots are shown for the free edge maximum principal stresses in the matrix elements (fibers are hidden) for the 55% Vf micro-model at the three 90/90 pl interfaces. The contour legend is shown (units of Pa). The contour plots utilize an upper and lower limit, set at 99MPa and 0 MPa, respectively, to capture the variation of maximum principal stresses within the entire micro-model. Values above or below the specified limits are colored red or blue, respectively. The face of the micro-model shown is at the free edge of the laminate.

Figure 5.21 shows the maximum principal stress contours for the thicker interlaminar interface micro-models (48% Vf) at the three 90/90 ply interfaces. For all three interface locations, there is only a slight increase in the number of elements whose maximum principal stresses have exceeded the matrix tensile strength as one progresses from the midplane to the third 90/90 ply interface. In addition, there is also an increase in the highest reported maximum principal stress.

There are two noticeable differences when comparing these contour results to those in Figure 19 for the 55% micro-model. First, the highest maximum principal stress values are much lower than their counterparts in Figure 5.19. The free edge matrix stresses at the fiber boundaries also increase slightly as one approaches the dissimilar ply interface. Additionally, the thick interface micro-models indicate that there is no longer a strong concentration of matrix stresses likely to indicate transverse matrix cracking as there was in the 55% interface micro-model. The concentration of the pure resin in the thick interface micro-model seems to have reduced the stress even between the proximal fibers.

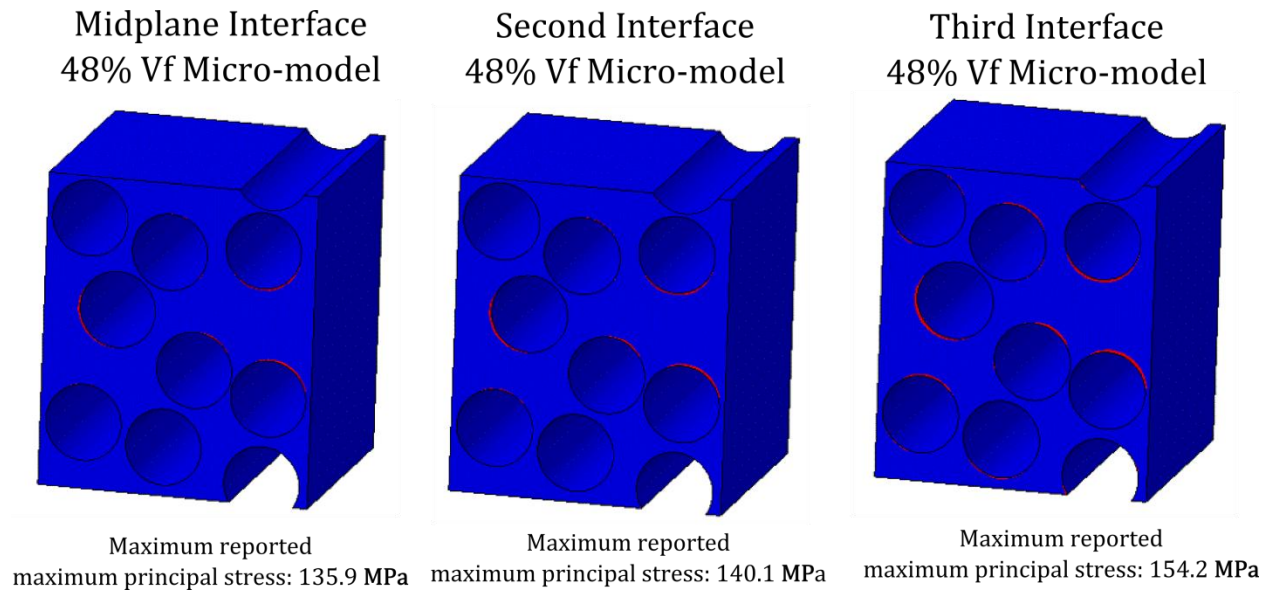


Figure 5.21: The contour results are shown for the free edge maximum principal stresses as a result of a 0.3% mechanical extension in the z-direction for the 48% interface micro-model. Elements whose maximum principal stresses have exceeded 99MPa are highlighted in red. The face of the micro-model shown is at the free edge of the laminate

Based on the results from this investigation, a number of conclusions can be inferred on the influence of extensional loading of the composite laminate on the local free edge micro-stresses in the 90° ply interfaces. First, the results from the 55% micro-model suggest that the

global extensional loading will drive primarily transverse ply cracking, rather than the localized edge cracks seen in the case of the thermal loading. The maximum values of these high stress regions between proximic fibers tended to decrease as the interlaminar shear stresses increased approaching the dissimilar ply interface. Second, the results from the 48% micro-model showed an overall decrease in the matrix maximum principal stresses, particularly in the zone between proximic fibers which was critical in the 55% model. Additionally, at all three ply interface locations the highest matrix maximum principal stresses at the free edge were about 25% lower than their counterpart in the 55% micro-model. The contour plots in Figure 5.22 for the 48% Vf micro-model show that the increasing interlaminar shear stresses at the Third interface does cause increased matrix stresses between proximic fibers, although there is minimal increase in the free edge matrix stresses.

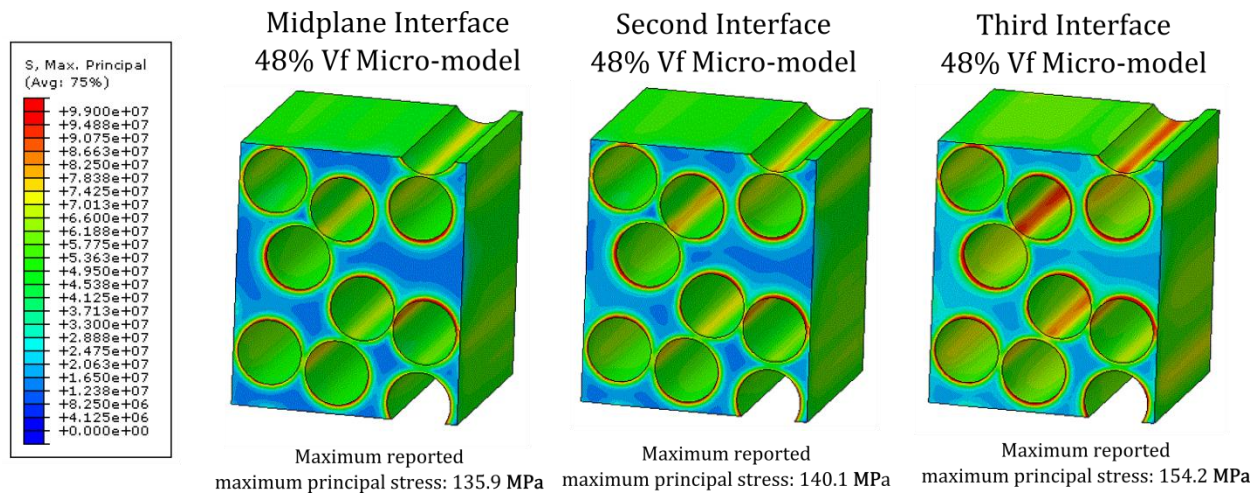


Figure 5.22: The contour results are shown for the free edge maximum principal stresses (MPa) as a result of a 0.3% mechanical extension in the z-direction for the 48% interface micro-model.

Based on these results, the numerical analysis supports the notion that the free edge cracking caused by the thermal cool down may not be a critical damage source in this particular composite laminate. This conclusion is supported by Dustin's experimental observations which

saw nearly all of the free edge micro-cracks present after manufacture play no role in damage initiation and evolution of the transverse ply cracks. The initial conclusion based on the experimental evidence pointed to a critical crack size which needed to be met. The simulations, on the other hand, suggest the initial free edge micro-cracks would not be susceptible to continued growth for the following reasons: (1) the thermal loading which drove the initial cracks was much more sensitive to the local free edge microstructure and created potentially shallow cracking, and (2) the increased interlaminar thickness between plies, which played a role in initiating free edge micro-cracks, actually reduced the propensity for transverse cracks. Thus, the location of damage initiation during extensional loading for transverse ply cracking is likely to originate in a region which did not experience initial free edge micro-cracking.

5.6. Discussion

The results from investigation one indicated that matrix rich interfaces caused significantly higher free edge matrix stresses localized at the fiber matrix boundary and located adjacent to the matrix rich regions. This observation explains the prevalence of micro-cracking during manufacturing at the interlaminar interface. In addition, the orientations of the highest stresses along the fiber/matrix boundary indicate a propensity for matrix cracking aligned with the interface. While there was a correlation between increased matrix rich regions and higher matrix stresses, there were minimal differences between the 48% and 44% micro-models indicating a possible threshold by which increasing matrix at the interface no longer increases the free edge stresses.

Investigation two, which looked at the thicker $N=3$ laminate, found that the micro-scale stresses were relatively unchanged even when the meso-scale strains at each of the three

investigated 90/90 ply interfaces were different. The increasing meso-scale interlaminar shear strains observed closer to the -25/90 interface changed the maximum reported principal stress values by only a few percent. This trend was true for both the 55% and 48% micro-models. The results of this analysis help to understand the occurrence of matrix cracking at all 90/90 interlaminar interfaces regardless of location in the experiments. It was found that the interlaminar micro-structure (i.e., local matrix density) played a more important role than the level of meso-scale strain variation with respect to interface location.

The results from the third investigation determined that the influence of the interlaminar micro-structure differed significantly in the case of a purely mechanical loading. The matrix elements which exceeded the threshold maximum principal stress in the 58% micro-model were located between proximate fibers and indicated a propensity for transverse cracking. The introduction of a thick matrix interface between the 90 plies in the 48% micro-model caused significant relaxation in the maximum principal stresses in the micro-model, resulting in only limited regions at the free edge and fiber/matrix boundary exceeding the threshold stress. This investigation suggests an explanation as to why the micro-cracks observed after manufacturing did not play a critical role in failure and damage development. The cracking caused by thermal cooldown were located at different micro-structural regions (e.g. the ply interfaces) which were not critical to the development of transverse matrix cracks. In addition, the orientations of the cracks from the thermal cooldown were along the interface, in contrast to the progressive development of transverse matrix cracks observed during the mechanical loading.

6. Free Edge Analysis of the Dissimilar Ply Microstructure

Based on the results from Chapter 5, the next step in understanding the influence of the micro-cracks at the free edge during the manufacturing process and subsequent mechanical extension is to explore the interlaminar $-25^\circ/90^\circ$ ply interface. As mentioned in Chapter 1, this interlaminar interface is the location of the critical delamination which results in composite failure. This chapter builds on the framework utilized in Chapter 5 and extends the investigation into the $-25^\circ/90^\circ$ interface. The effect of interlaminar thickness at the ply boundary will be explored under both thermal and mechanical loading.

6.1 Model Development

Two assumptions were made for the interlaminar micro-models at the dissimilar interface. First, the model was assumed to continue to be periodic in the x and z directions. Second, since the interlaminar micro-models exist at the dissimilar ply boundary, they represent a unique material whose properties are a blend of the two ply orientations. These two assumptions will be outlined in greater detail in what follows.

First, the maintenance of periodicity in the x and z directions required the determination of a suitable size for the micro-model to ensure periodic fibers despite the off-angle orientation of the -25° fibers with respect to the orientation of the micro-model, which is still fixed in the standard x-y-z coordinate system established in Chapter 5 for this laminate problem. It was assumed that the size of the $-25^\circ/90^\circ$ micro-model will be based on the size used in the previous analysis for the $90^\circ/90^\circ$ interface (0.018 mm). The -25° fibers are rotated 65° about the y axis when compared to the original fiber direction of the 90° fibers. This rotation causes a change in the required size in the z and x directions, referred to as the length and width, respectively.

Equations (6.1) and (6.2) outline the simple relations between the original length (L), original width (W), revised dimensions L^* and W^* , and the angle of rotation (θ) about the y axis.

$$L^* = \frac{L}{\cos(\theta)} \quad (6.1)$$

$$W^* = \frac{W}{\sin(\theta)} \quad (6.2)$$

The final dimensions of the micro-model are a length of 0.0426 mm and width of 0.0198 mm, based on the original dimensions of a cubic 0.018 mm micro-model. Since the rotation of the fibers is only about the y-axis, there are no changes necessary to the y axis. The height of the micro-models are fixed to double the original dimensions to account for the two layers of lamina represented (-25° and 90°). An example of the micro-model is shown in Figure 6.1.

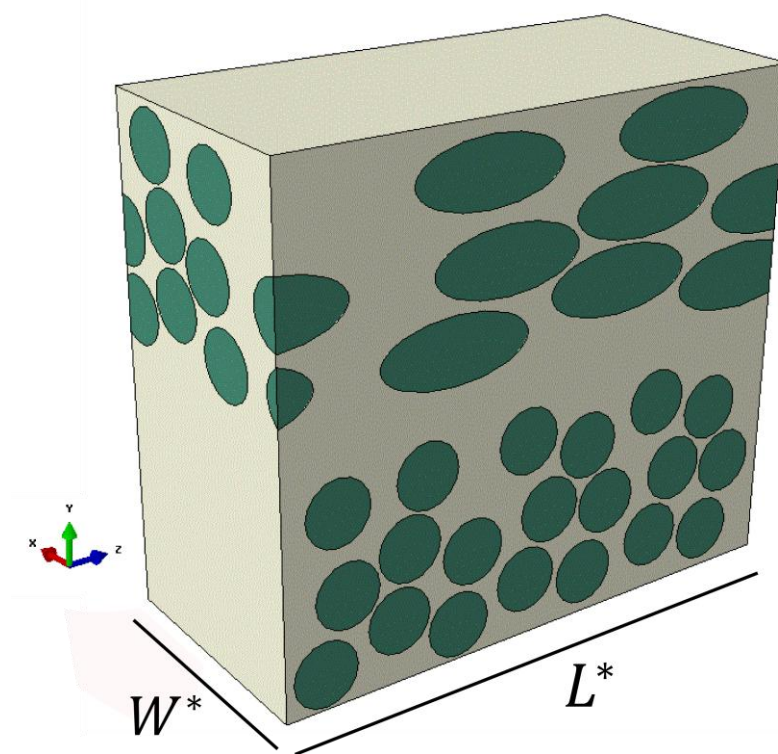


Figure 6.1 Micro-scale model for the dissimilar ply interface showing the fiber (green) and matrix (white) geometries.

Second, the interlaminar micro-models representing the dissimilar ply interface are homogenized to provide material properties for the meso-scale analysis of the laminate. Thus, the continuum elements at the meso-scale representing the dissimilar interface will be assigned material properties different from those of the remainder of the laminate. These meso-scale material properties will be determined through homogenization of the dissimilar ply micro-models discussed later in this section. It should be noted that a similar approach was taken in Chapter 5.

6.1.1 Meso-scale Model

The $[25_3/-25_3/90_3]_S$ laminate was used due to the higher mesh refinement for the fixed element size. Figure 6.2 presents the a cut-out of the meso-scale model, emphasizing the unique region (green) between the -25° and 90° lamina which was assigned as a second material. The remainder of the laminate was assumed to maintain the IM7/8552 composite lamina properties at the fiber volume fraction of 55% used in the previous Chapter. The properties of the interlaminar material were determined through a homogenization process and are discussed in Section 6.1.4. The dimensions of the interlaminar region were fixed to the micro-model size. The boundary conditions employed at the meso-scale are identical to those enforced during the $90^\circ/90^\circ$ interface studies in the previous chapter.

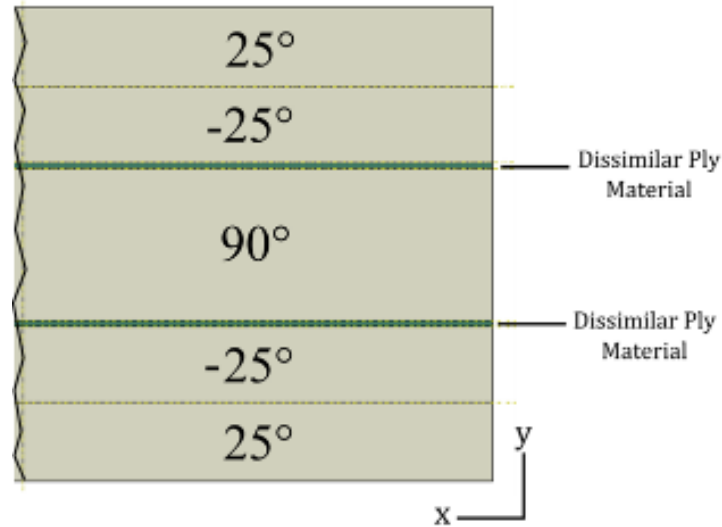


Figure 6.2 Meso-scale model for the dissimilar ply interface analysis. The color coding represents the two different material definitions used for the standard unidirectional lamina (beige) and the interface (-25/90) material (green).

6.1.2. Micro-scale Model

Two different micro-models were used to examine the influence of the interlaminar matrix region between the -25° and 90° fibers on free edge microscale stresses. The first model, hereon referenced as the Regular interface, had an interlaminar matrix region of $2.7\ \mu\text{m}$ inserted between the -25° and 90° fibers. This represented a 15% increase in the total micro-model height. The second micro-model, referred to as the Thick interface, had an interlaminar region of $4.5\ \mu\text{m}$ inserted between the ply fibers, resulting in a 25% increase in the overall micro-model height. Figure 6.3 shows the two micro-models of varying interlaminar thicknesses.

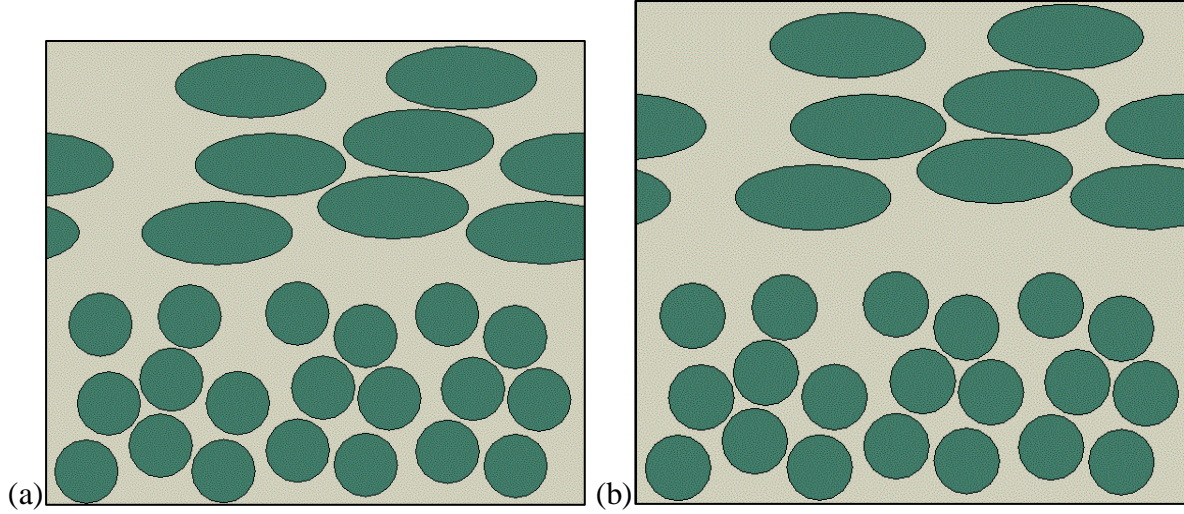


Figure 6.3: Micro-models for the -25/90 interlaminar (a) Regular interface and (b) Thick interface. The micro-model is oriented so that the free edge is facing out of the page.

The periodicity of the micro-model was enforced by restricting fibers from intersecting the top and bottom boundaries. Similar fiber spacing was assumed between the -25° fibers as was used in the 90° fiber micro-models from Chapter 5. By utilizing a similar fiber distribution, the influence of orientation of fibers between the lamina will have a larger role in differences in local micro-scale stresses between the 90° fibers and those of the -25° fibers. With the updated micro-model dimensions computed from Equations (6.1) and (6.2), the dissimilar ply micro-model remains periodic in the x and z directions as shown in Figure 6.4. The images of the periodic faces appear as mirrored reflections, however, they are periodic when properly oriented (facing away from each other) with their respective x or z axis. An advantage of utilizing periodic boundary conditions in the z direction is that there is no need for sacrificial domains within the micro-model in the z direction. This is in comparison to de-homogenization methods [50] which required the extraction of micro-scale information at a minimum distance of 1 fiber diameter from the entire micro-model boundary due to the improper application of micro-scale boundary conditions. In this work, however, the same care has to be taken in the y-direction.

Since the top and bottom faces are modeled ideally as periodic, micro-scale stresses at these boundaries within 1 fiber diameter of the top/bottom of the micro-model may not be valid.

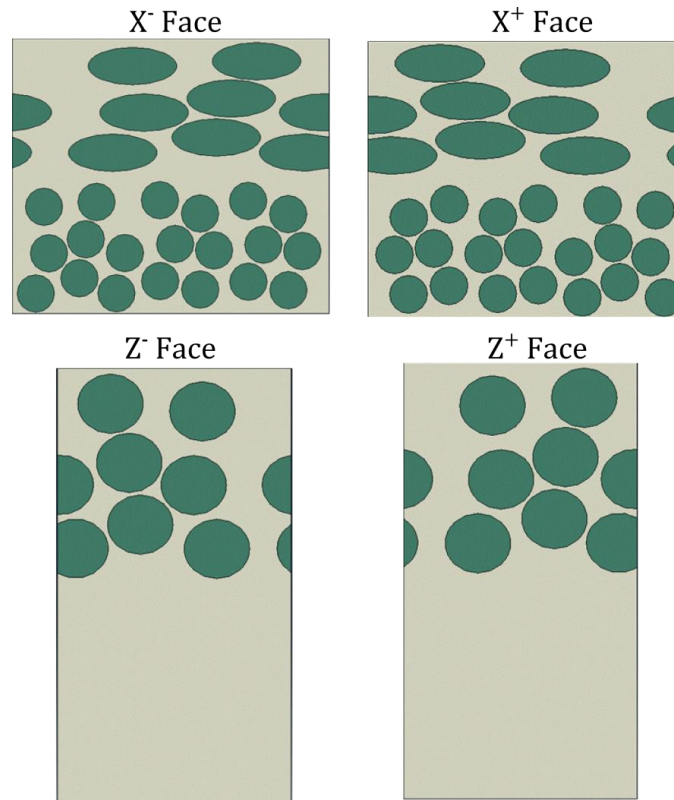


Figure 6.4: The x and z faces of the Regular dissimilar interface micro-model, highlighting the periodic in the x and z direction. The opposing faces will appear as mirror images due to reflection, however, they are periodic when faced in the proper orientation.

An unintended consequence of the $-25^{\circ}/90^{\circ}$ finite element micro-model was the complexity in preserving mesh congruence with respect to nodes on periodic faces. Rather than using wedge elements as in the $90^{\circ}/90^{\circ}$ interlaminar micro-models, successful meshing of the complex fiber matrix shapes in the dissimilar ply micro-model required the use of C3D4T tetrahedral elements in ABAQUS. The finite element mesh is shown in Figure 6.5. As a result of

the free-form mesh generation, there was no guarantee that opposing faces, although having periodic geometries, would have periodic meshes (e.g. one-to-one node periodic node correspondence). The next section outlines modifications to the implementation of periodic boundary conditions used to circumvent the issue of non-periodic node sets.

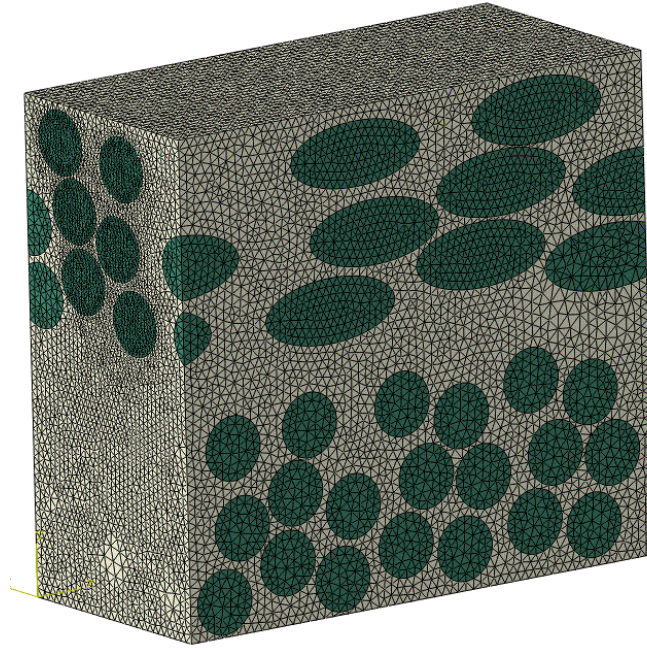


Figure 6.5: The Regular interface micro-model with the C3D4T tetrahedral elements. The free-form mesh resulted in irregular spacing of nodes on periodic faces (i.e. different node locations and number of nodes on corresponding periodic faces)

6.1.3 Periodic Boundary Conditions Implementation

Rather than using direct linear constraint equations to enforce periodicity directly between the nodes on periodic faces, a surface-to-surface constraint was employed to overcome the non-congruence of node sets. Figure 6.6 diagrams the surface-to-surface constraints used to

enforce periodicity. Figure 6.6 (a) provides an example where nodes on the positive x face (shown in blue) do not have one-to-one correspondence to the nodes on the negative x face (shown in red). As such, linear constraint equations prescribed directly to these node sets would fail due to the mismatch in location and number of nodes. Ideally, one would prefer to have a periodic set of nodes, as shown in Figure 6.6 (b) in green.

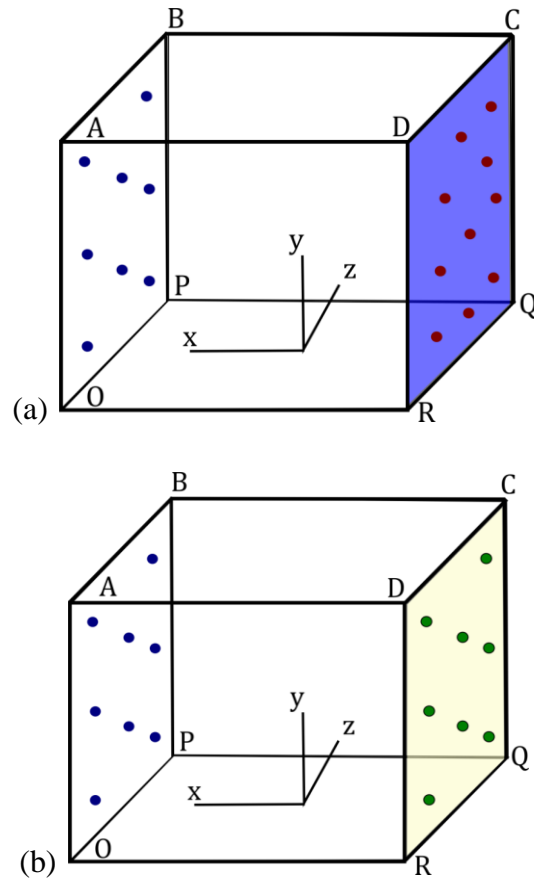


Figure 6.6: (a) An example of non-periodic nodes on the negative and positive x faces. A surface is defined using the red nodes on the negative x face, shown in purple. (b) An example of periodic nodes on the negative and positive x face. A surface is defined using the green nodes on the negative x face, shown in yellow. The red nodes in (a) represent the actual nodes on the micro-model face, while the green nodes in (b) represent duplicate nodes generated to create a periodic set to blue nodes. The blue and green nodes are prescribed periodic constraints, while the two surfaces (yellow and purple) are tied using surface-to-surface constraints.

To overcome the non-periodic node sets shown in Figure 6.6 (a), the following steps were performed. First, a surface was defined using the nodes on the negative x face as shown in purple in Figure 6.6 (a). Next, a duplicate set of nodes are defined on the negative x face by copying the blue nodes, shown in Figure 6.6 (b), and translating them to the negative x face as seen by the green nodes. Then, another surface is defined based on the green nodes as shown by the yellow surface in Figure 6.6 (b). The two surfaces are then constrained via a *Tie surface-to-surface constraint within ABAQUS. Lastly, the blue and green nodes, which are periodic, are constrained using the same linear constraint equations prescribed in previous analyses to enforce periodic boundary conditions. The resulting effect is a surface based periodic constraint between the blue nodes and red nodes both shown in Figure 6.6 (a). A similar process was performed to apply periodicity in the y and z directions of the micro-model.

6.1.4 Homogenized Properties of the Dissimilar Ply Interface

The meso-scale material properties of the $-25^{\circ}/90^{\circ}$ interlaminar interface to be used in the laminate analysis were obtained through homogenization via perturbation steps. Standard periodic boundary conditions were employed in the x, y and z directions via the method discussed in the previous section. The model was perturbed in the 6 material directions as outlined in Figure 6.7. The three independent nodes are used to define the perturbation steps. The assigned nodes and prescribed displacements are outlined and labeled in the figure. Uniaxial strain loading was enforced by prescribing a displacement in one of the six listed directions (corresponding to a material direction) and fixing all other DOF's on the remaining independent nodes.

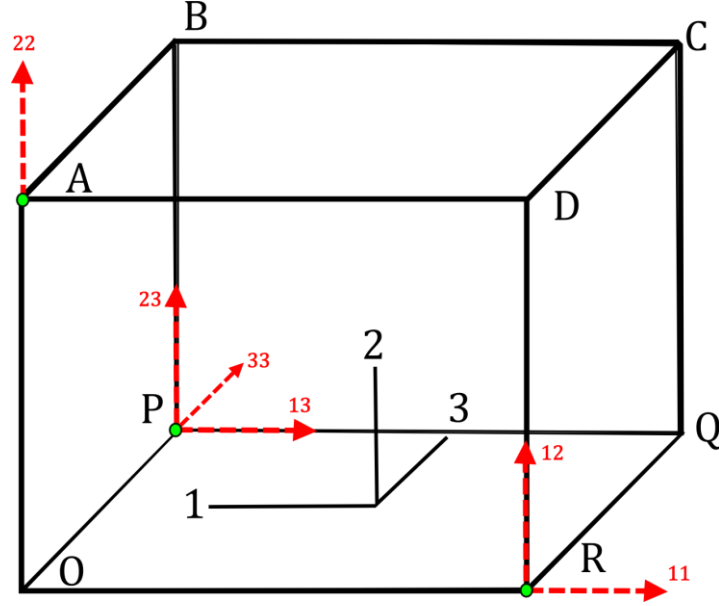


Figure 6.7: Schematic of the six degrees of freedom shown as possible displacements by red dashed arrows.

The displacements are prescribed for the three independent nodes in the micro-model shown in green.

A similar methodology was employed by Yuan and Fish [95] to determine the material Jacobian. In this work, the material stiffness matrix was obtained directly through six linear perturbation steps. The perturbation in the 11 direction is shown in Figure 6.8 through the application of unit strain in the 11 direction and holding all other strains to zero. The localization rules from Equations (5.6) were used to determine the appropriate displacements. During this perturbation step, a column of the material stiffness matrix (highlighted in Figure 6.8) would be determined. Each entry in the column corresponds to a component of the resultant volume averaged stresses of the micro-model, shown in the left hand side of the equation.

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{bmatrix} = \begin{bmatrix} - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Figure 6.8: An example of the determination of the material stiffness matrix through unit, uni-strain perturbation steps. In the above example, unit-strain is applied in the 11 direction and all other strains are prescribed to be zero. The resulting volume averaged stresses of the micro-model, shown at the left hand side, are the necessary entries of a column of the stiffness matrix, highlighted above.

The computation of the macroscopic stress of the micro-model was done for each perturbation step. Rather than computing the volume averaged stress directly through all elements within the micro-model, the surface integral form using Green's Theorem was used as in Equation (6.1), where \mathbf{t}^m is the microscopic tractions along the outer boundary S of the micro-model and \mathbf{x} is the current position vector. The application of periodic boundary conditions reduces the calculation of the surface integral in Equation (6.1) to the discrete summation shown in Equation (6.2), where \mathbf{f}^n is the nodal force vector on the independent nodes of the micro-model.

$$\sigma_{ij}^M = \frac{1}{V_0} \int \sigma_{ij}^m dV = \frac{1}{V_0} \int t_i^m x_j dS \quad (6.1)$$

$$\sigma^M = \frac{1}{V_0} \sum_{n=A,O,P,R} f_i^n x_j^n \quad (6.2)$$

The resultant force at each node of the four nodes (A,O, P and R) was extracted for each perturbation step. After using the vector calculations in Equation (6.2), corresponding off-diagonal terms representing the tension-shear coupling of the elastic stiffness matrix were averaged together to enforce symmetry (e.g. the 11-23 entry in the 6x6 matrix would be averaged with the 23-11 entry). The computed Jacobian, or stiffness matrix, in units of MPa is shown in Table 6.1 in its final symmetric form. This stiffness matrix was inputted directly into ABAQUS using the *Elastic material keyword in conjunction with the Type=Anisotropic parameter.

Table 6.1: The symmetric material stiffness matrix for the -25/90 micro-model in units of MPa

	11	22	33	12	23	13
11	9.68E+04	7.34E+03	1.67E+04	2.83E+01	4.65E+02	8.30E+00
22	7.34E+03	1.26E+04	6.86E+03	3.34E+01	1.73E+02	2.50E+01
33	1.67E+04	6.86E+03	5.48E+04	6.65E+00	1.95E+04	1.08E+01
12	2.83E+01	3.34E+01	6.65E+00	3.17E+03	9.72E+00	3.44E+02
13	4.65E+02	1.73E+02	1.95E+04	9.72E+00	1.28E+04	8.61E+00
23	8.30E+00	2.50E+01	1.08E+01	3.44E+02	8.61E+00	2.97E+03

In addition to the material stiffness matrix, the anisotropic thermal expansion coefficients had to be determined for the micro-model. A single linear perturbation steps was utilized for the determination of the six thermal expansion coefficients. A temperature loading of 1° C was applied to the micro-model with assumed fully periodic boundary conditions. The nodes A, P and R were left unconstrained in the 6 DOF's highlighted in Figure 6.7. The resulting nodal displacements from nodes A, P and R were extracted from the thermal perturbation step, and the

localization rules of Equations (5.6) were used to determine the thermal expansion coefficients through the determination of the resulting six strain components. The thermal expansion coefficients for the six strain components are shown in Table 6.2.

Table 6.2: The anisotropic thermal expansion coefficients

Thermal Expansion Coefficient	$10^{-6}/^{\circ}\text{C}$
α_{11}	1.983
α_{22}	50.1
α_{33}	4.0
α_{12}	0.457
α_{13}	21.5
α_{23}	-0.457

6.2 Finite Element Results

Two separate investigations were conducted on the -25/90 dissimilar ply micro-models. The first looked at the effect of thermal loading on the local micro-scale stresses, similar to that performed in Chapter 5 for the 90/90 interlaminar micro-models. The second investigation looked specifically at the effect of mechanical loading due to the tensile loading of the laminate in the global z direction. Although the mechanical extension in the real composite would be applied in combination to the residual thermal loads from manufacturing, the two loads were again separated intentionally. It was seen in Chapter 5, that the local interlaminar micro-structure had different influences on free edge stresses for the two loading scenarios. The goal here was to investigate: 1) the effect of the thermal cooldown on the generation of pre-cracks observed experimentally, and 2) the propensity for crack initiation (progressive damage) during the extensional loading of the laminate.

6.2.1 Dissimilar Ply Interface (-25/90) during Thermal Cooldown

A thermal loading of -155°C was applied to the N=3 laminate (discussed in Section 6.11) with no mechanical restrictions on the laminate contraction in the x, y or z directions. A stress contour plot of the normal stresses (y-direction stresses) from the meso-scale analysis is shown in Figure 6.9 (a). The contour plot matches well with that from the N=3 laminate used in Section 5.5.2, with a variation of only 7% for the maximum reported normal stress value. The difference in reported stress was due to the different discretization at the dissimilar interface in the current analysis. The similar contour values between the two meso-scale analyses demonstrated that the current method for using a second material for elements at the interface did not drastically change the overall laminate solution.

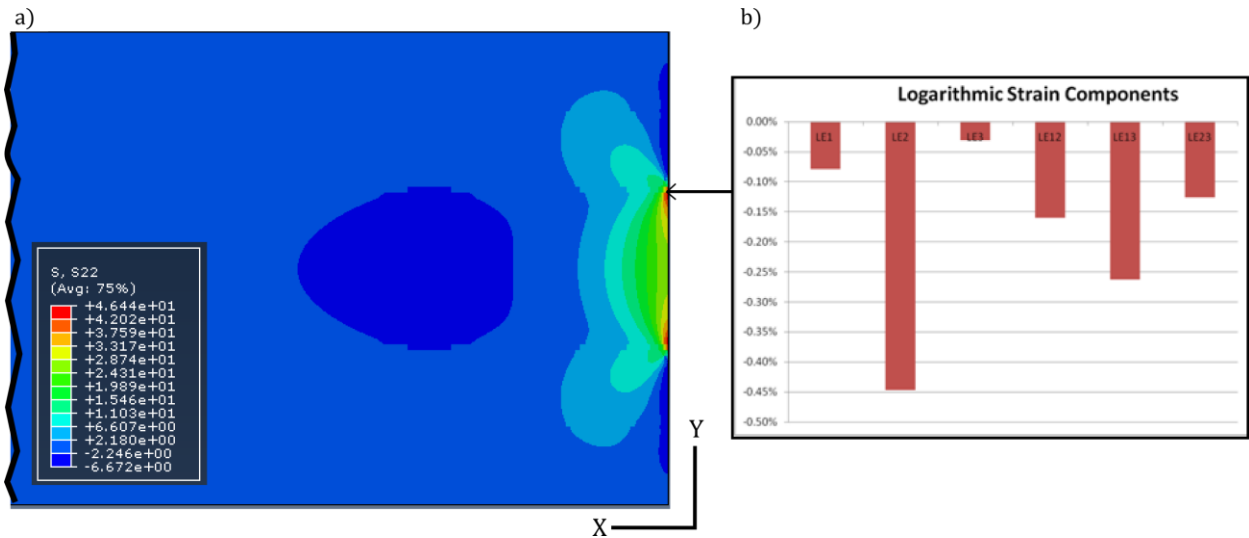


Figure 6.9: a) A stress contour plot (units of MPa) of the normal (y-stress) for the dissimilar ply interface model under thermal load of -155°C is shown, where a cut-out of the right-hand side (X) of the laminate free edge is shown. b) The logarithmic strain components extracted from the dissimilar interface element are shown. Note that the anisotropic nature of the dissimilar ply interface causes strains in all three shearing material directions.

The meso-scale strains of the dissimilar ply element, shown in Figure 6.9 (b), contained significant interlaminar shear strains. This differed from what was observed at the 90/90 ply interfaces, where only LE23 shear strains were present. Not only are the LE23 strains over 25% greater at this dissimilar ply interface than they were at the closest 90/90 ply interface to the -25/90 interface, but the LE13 strains are nearly double those of the LE23 strains.

The results from the micro-scale analysis are shown in Figure 6.10 (a) and (b), which plot the maximum principal stress contours for the two different ply interface thicknesses. The highest reported value is indicated on both the Regular interface and Thick interface micro-models and coincides with similar locations on a 90° fiber at the dissimilar ply interface. The highest maximum principal stress for the Regular interface micro-model was 184 MPa, while the Thick interface micro-model's maximum value was 195 MPa. Thus, similarly to the 90/90 ply interface, the dissimilar ply interface also exhibits increasing free edge matrix stresses with increasing matrix rich regions between lamina.

It was also observed in Figure 6.10, for both interface micro-models, that the maximum principal stresses in the matrix only exceeded 99 MPa at the 90° degree fibers. The lack of significant free edge stresses at the -25° fibers indicate that the orientation of the fiber relative to the free edge has a strong effect on the micro-scale free edge stresses. The -25° orientation of the fibers would change the material property mismatch between the anisotropic fiber and isotropic matrix at the free edge. For this particular composite laminate, the 90° fibers were the most critical for the development of high matrix stresses at the free edge to cause pre-cracking. In the experimental literature, pre-cracks were only observed for these laminates on the 90° fibers as well. In addition, the cracks originating at this interlaminar interface were much shorter than those that developed at the 90/90 interface. This difference in average crack length between the

two types of ply interfaces was likely due to the lack of high free edge stresses near -25° fibers which may have inhibited the development of longer cracks at the $-25/90$ interface.

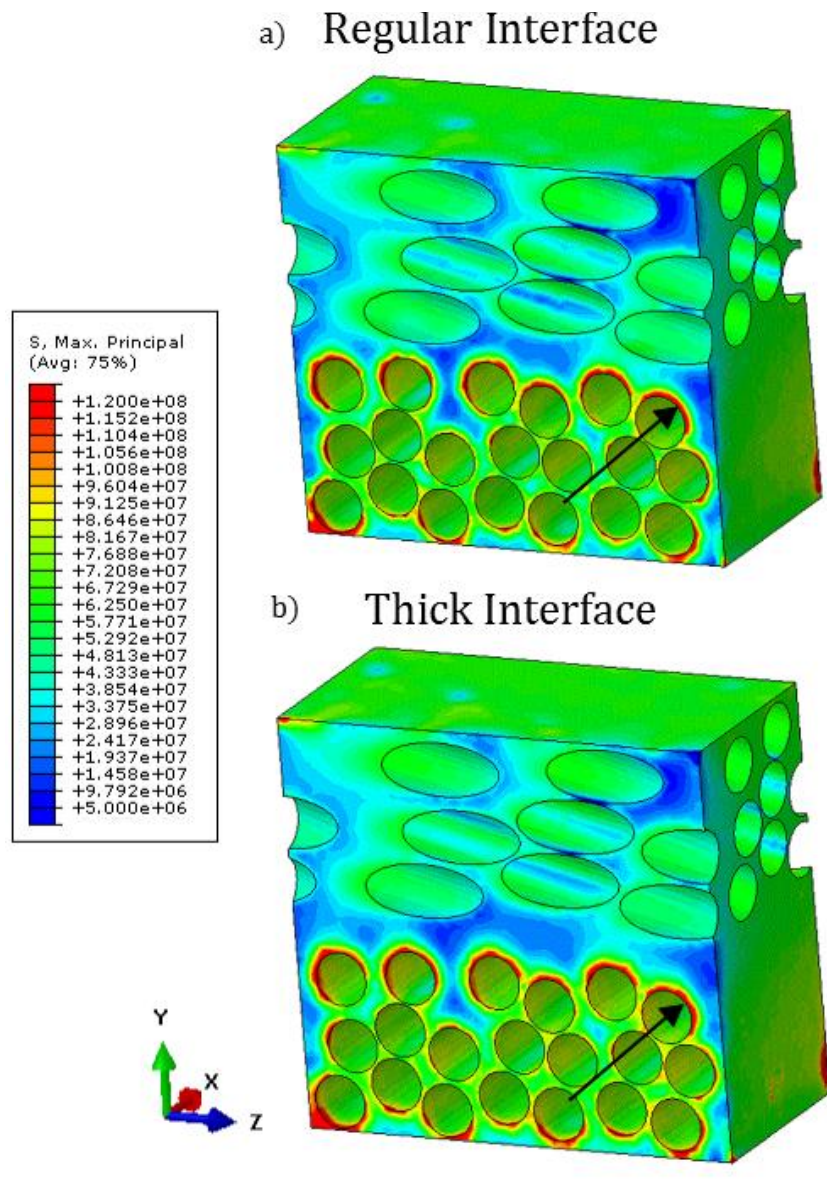


Figure 6.10: Stress contour plot of maximum principal stress (Pa) for the two dissimilar ply interface micro-models. The location of the highest maximum principal stress is indicated by a black arrow. The highest maximum principal stresses occurred at the 90° fiber boundary facing the ply interface. It should be noted that only around the 90° fibers did the matrix maximum principal stress exceed the 99MPa tensile strength of the matrix.

6.2.2 Dissimilar Ply Interface (-25/90) during Mechanical Loading

The composite laminate was prescribed an extensional load in tension in the z-direction up to 0.3% strain. It should be noted that the actual composite failure strain was 0.6% strain, and progressive damage (matrix cracking) was observed in the experiments prior to 0.3% strain [31]. No thermal load was prescribed in this investigation.

The meso-scale results are shown in Figure 6.11, displaying both the contour plots of the normal peeling stresses (in the y-direction) as well as the individual strain components at the dissimilar ply interface element. Comparing these strain components to those at the 90/90 ply interfaces (Figure 5.18), there is a noticeable increase in the strains in both the y-direction and interlaminar shear. The LE23 shear strain, for example, was over three times that of the applied extensional strain in the z-direction. In addition, the strain in the y-direction (LE2) was nearly 50% higher than the applied z-direction strain.

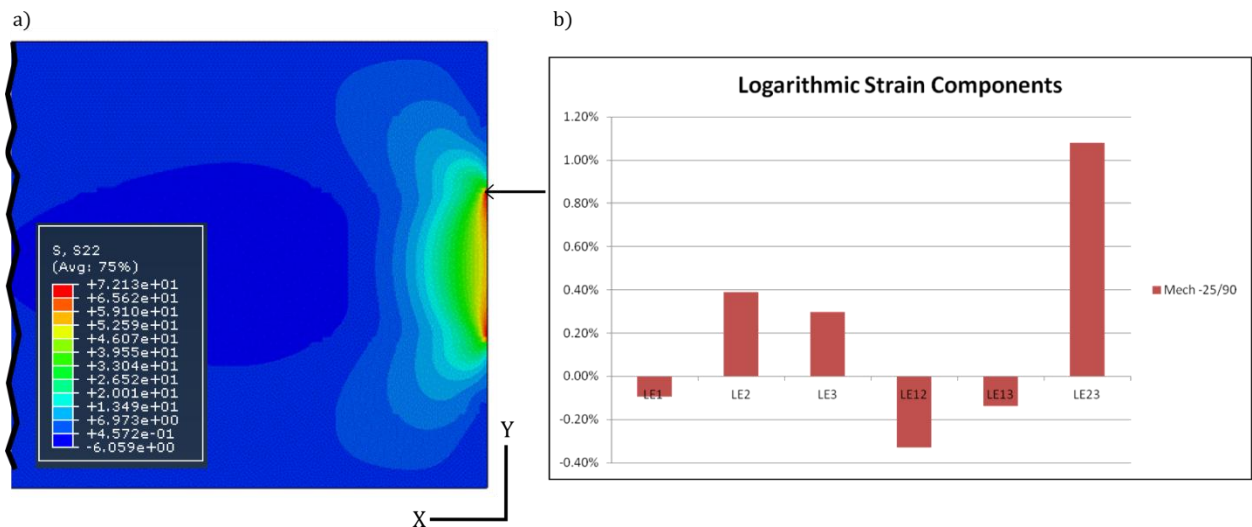


Figure 6.11: a) A stress contour plot (units of MPa) of the normal (y-stress) for the dissimilar ply interface model under 0.3% strain tensile extension in z-direction, where a cut-out of the right-hand side (X) of the laminate free edge is shown. b) The logarithmic strain components extracted from the dissimilar interface element are shown.

The results of the micro-scale analysis are shown in Figures 6.12 (a) and (b) as contour plots of the maximum principal stress. The location of the highest reported maximum principal stress is highlighted via a black arrow. The highest maximum principal stress was found to be located at the interior of the micro-model, not at the free edge. For both interface micro-models the highest matrix maximum principal stresses were found between proximal -25° fibers. In the case of the Regular interface, the highest maximum principal stress was 291 MPa, whereas the highest reported maximum principal stress at the free edge was significantly lower at 184 MPa. A similar gap was found between the two similar values in the Thick interface model, whose highest reported value of 284 MPa was higher than the largest free edge maximum principal stress at 182 MPa.

Similar trends can be observed in Figures 6.12 (a) and (b) to those seen in the mechanical loading case of the 90/90 ply interfaces. The free edge matrix stress concentrations are relatively localized to a small region at the free edge, whereas high matrix stresses have developed between proximal fibers through the width of the micro-model (x-direction). These stresses are likely to cause progressive matrix cracking, however, it should be noted that the stresses between the 90° fibers are not as severe as those between the -25° fibers. Thus, although the pre-cracks during manufacture tended to occur at the 90° fiber boundary (as seen in Section 6.2.1), progressive cracking at the dissimilar ply interface during the extensional loading is much more likely to occur within the -25° ply.

Another similar trend found at the dissimilar ply interface was a reduction of matrix stresses with the addition of a thicker matrix-rich region. The Thick interface micro-model was found to have lower maximum reported values than that of the Regular interface model. In the

stress contour plots, the free edge stress concentrations along the boundary of the 90° fibers are more pronounced in the Regular interface model.

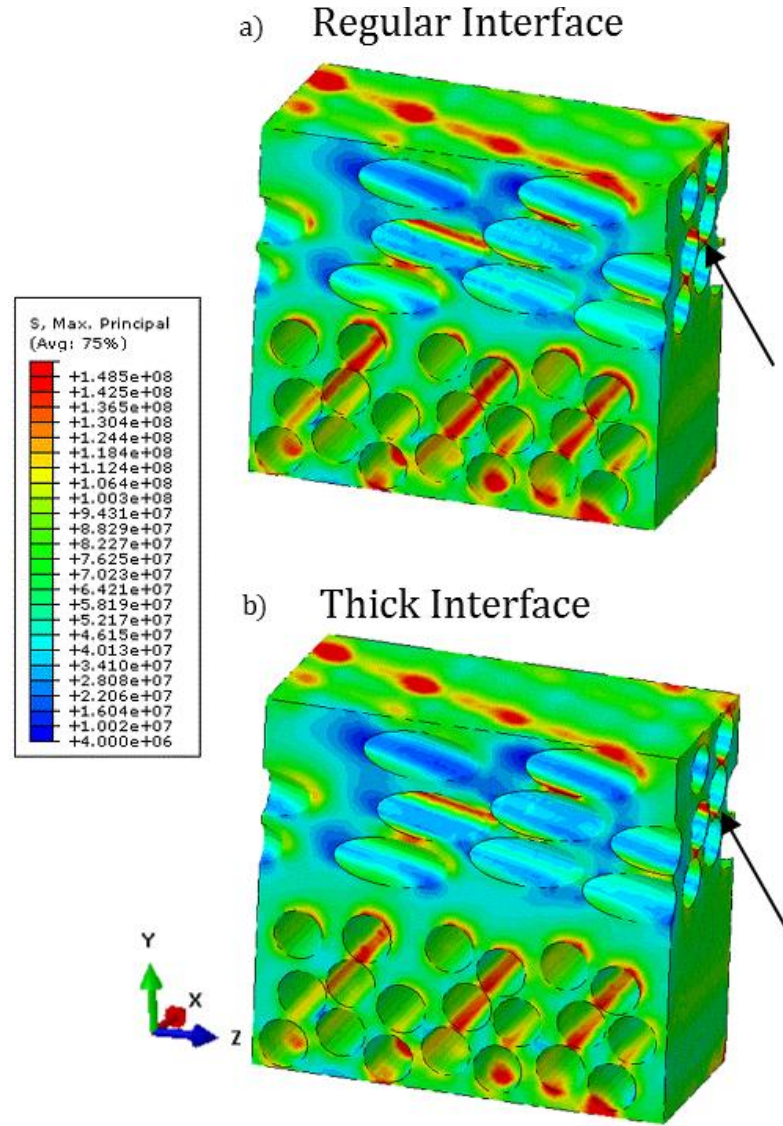


Figure 6.12: Stress contour plot of maximum principal stress (Pa) for the two dissimilar ply interface micro-models. The location of the highest maximum principal stress is indicated by a black arrow. The highest reported values were found to be located at the interior of the micro-model between proximal -25° fibers.

6.3 Discussion

The results from the first investigation found that the pre-cracking during thermal cooldown would only manifest at the 90° fiber boundaries, and was likely to occur facing the interlaminar region. The matrix regions surrounding the -25° fibers did not exhibit strong concentrations of free edge stresses and was likely due to the reduced mismatch in material stiffness and thermal properties between the fiber and matrix at the free edge as a result of the -25° orientation. These results correlated well with the experimental observations, which saw pre-cracking only occurring only on the face of the 90° fibers. In addition, the observance of minimal free edge stresses at the -25° ply provides an understanding to why previous experimental work by Dustin [31] found the dissimilar ply interface had shorter crack lengths as compared to the $90/90$ ply interface. As with the $90/90$ ply interface, the increased matrix thickness between the plies caused an increase in the free edge matrix stresses at the dissimilar ply interface.

The study of the dissimilar ply interface under mechanical loading only provided understanding into why progressive cracking was not typically observed to originate the dissimilar ply interface. First, the highest reported maximum principal stresses were not found at the free edge or between proximal 90° fibers, as was the case for the $90/90$ interface micro-models. Instead, the highest propensity for matrix cracking was between proximal -25° fibers at the interior of the micro-model. These -25° ply cracks may be the initiator of delamination when no transverse cracking was observed as the initiator. Second, the location of the highest matrix stresses between the 90° fibers in the dissimilar interface micro-model differed from that of the $90/90$ ply interfaces. Due to the high interlaminar shear stresses at the $-25/90$ interface, the highest matrix stresses between proximal 90° fibers occurred in an orientation that would align

the crack in with the z-direction (note that the 90/90 interfaces would have caused transverse cracks).

7. Conclusions and Future Work

7.1 Free edges in a semi-concurrent scheme

In this work, a semi-concurrent approach was developed with the inclusion of a free edge at the micro-scale. A method was proposed to preserve elastic strain energy across the scales despite the non-periodic boundary conditions in order to maintain a coupled kinematic and constitutive coupling between the macro and micro-scales. The implementation into ABAQUS was demonstrated for a simple 2D representative volume element with cohesive damage between the fiber and matrix under a simple macroscopic tensile load. The results found that a comparison between the periodic and free edge results varied significantly with RVE size. A medium size (16 fiber micro-model) tended to have the largest variation due to premature failing of cohesive surfaces in the periodic analysis. The study found that the free edge tended to decrease damage development due to local relaxation at the free edge boundary condition. The study also highlighted that prior to damage development, the response of the two different boundary conditions (purely periodic or free edge) were very similar, with minimal differences in the overall elastic energy. Thus, for the 3D elastic analysis of the full composite laminate, the implementation of the z-scalar approach for energy preservation across the scales may not be necessary.

7.2 Multiscale analysis of the 90/90 ply interface

The one-way 3D multiscale analysis was critical in providing understanding of the influence of the local microstructure in the creation of pre-cracks during manufacture as well as understanding the role of the pre-cracks for damage development. It was found that the matrix rich region between 90° plies caused a dramatic increase in the free edge stresses at the boundary of the 90° fibers facing the interlaminar region during thermal cooldown. These free edge

residual stresses are the likely cause of pre-cracks found in the experimental literature. The effect of the matrix-rich interlaminar interface helped to explain the higher propensity of pre-cracks to form near the 90/90 ply interfaces rather than within a given lamina.

By examining the thicker N=3 laminate, it was found that the meso-scale strain gradient (which varied along different 90/90 ply interfaces) played a minimal role in altering the free edge matrix stresses. Thus, pre-cracks were likely to form at any 90/90 ply interface, and not only at the interfaces that experience the highest meso-scale free edge deformations. Again, cracking was experimentally seen at all 90/90 interfaces and not just those situated closest to the dissimilar ply interface.

Under extensional loading, the interlaminar microstructure played a significantly different role. Increased matrix content at the interlaminar interface decreased both the free edge stresses in the matrix as well as the matrix stresses at the interior of the micro-model. Thus, transverse cracking (oriented parallel to the load) was likely to occur at regions that aren't susceptible to the pre-cracking during thermal cooldown. This helps to explain why the pre-cracks during manufacture did not influence the development of progressive damage, since the transverse cracks would likely develop in areas away from the interlaminar interface.

7.3 Multiscale analysis of the -25/90 interface

The analysis of the dissimilar ply interface found that the strong concentration of free edge stresses in the matrix only occurred at the boundary of the 90° fiber. The orientation of the -25° decreased the mismatch in material stiffness and thermal properties between the fiber and matrix and did not contribute to the development of high free edge matrix stresses even with the thicker interlaminar interface. This phenomenon of pre-cracking only at the 90° fiber surface facing the interlaminar region was observed experimentally. In addition, the results from the

mechanical loading investigation found that the highest stresses at the dissimilar ply interface did not occur at the free edge or between 90° fibers. The highest propensity for matrix failure at the -25/90 interface was at the interior of the model between proximal -25° fibers. This cracking may not have been observed as readily as the transverse cracks in the 90° plies, and could explain why visible damage tended to occur in the 90 plies before reaching the dissimilar ply interface. The orientation of the highest matrix stresses in the 90 ply region of the -25/90 interface were found between 90° fibers aligned with the z-direction, which differed from the 90/90 interfaces which saw the highest matrix stresses aligned parallel to the z-direction. This could help explain the occurrence of transverse cracking occurring in the 90 plies before progressing to the dissimilar ply boundary.

7.4 Future work

The current multiscale framework was successful in providing qualitative understanding of the influence of the local microstructure on the propensity for damage development at the free edge. The next step in this research is to extend the elastic multiscale analysis to model damage initiation and development to quantitatively assess the efficacy of the approach in predicting actual effective crack length sizes. The implementation of damage and non-linearity at the micro-scale will require the use of the proposed scale coupling for use with the free edge boundary conditions at the micro-scale. By utilizing the scale coupling, a quantitative analysis of the effect of free edge cracking on damage development could provide a homogenized response for meso- or macro-scale analysis that includes the effect of the microstructural free edge for use in other multiscale simulations. Other future work should also explore more complex loading configurations for the composite laminate, which may be difficult to test experimentally, and investigate the effect of the laminate free edge microstructure.

APPENDICES

A.1 UMAT subroutine for semi-concurrent scheme

```

C      UMAT SUBROUTINE FOR COARSE SCALE MATERIALS by CHRIS CATER 2011
C
      SUBROUTINE UMAT (STRESS, STATEV, DDSDE, SSE, SPD, SCD,
1      RPL, DDSDDT, DRPLDE, DRPLDT,
2      STRAN, DSTRAN, TTIME, DTIME, TEMP, DTEMP, PREDEF, DPRED, CMNAME,
3      NDI, NSHR, NTENS, NSTATV, PROPS, NPROPS, COORDS, DROT, PNEWDT,
4      CELENT, DFGRD0, DFGRD1, NOEL, NPT, LAYER, KSPT, KSTEP, KINC)
C
      USE IFPORT
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CMNAME
C
      DIMENSION STRESS (NTENS), STATEV (NSTATV),
1      DDSDE (NTENS, NTENS),
2      DDSDDT (NTENS), DRPLDE (NTENS),
3      STRAN (NTENS), DSTRAN (NTENS), TTIME (2), PREDEF (1), DPRED (1),
4      PROPS (NPROPS), COORDS (3), DROT (3, 3), DFGRD0 (3, 3), DFGRD1 (3, 3)
C
      INTEGER i, j
      INTEGER*4 com1
      DOUBLE Precision STRANT (8), CONT (6, 6), STRESSAVG (4)
C *****
C      Output information to the MSG file
C      This includes NTENS, step number, and increment number.
      WRITE (UNIT=7, FMT=*) 'NTENS is ', NTENS
      WRITE (UNIT=7, FMT=*) 'KSTEP is ', KSTEP
      WRITE (UNIT=7, FMT=*) 'KINC is ', KINC
      WRITE (UNIT=7, FMT=*) (TTIME (i), i=1, 2)
C
C      Output the Deformation Gradient (DFGRD1)
C      to the message file for easy checking.
C
      WRITE (UNIT = 7, FMT = *) 'Strains are '
      WRITE (UNIT = 7, FMT = *) (DSTRAN (i), i=1, 4)
      WRITE (UNIT = 7, FMT = *) 'FTENS is '
      DO i=1, 3
         WRITE (UNIT = 7, FMT = *) (DFGRD1 (i, j), j=1, 3)
      END DO
C
C      Write the deformation gradient to output CSV file
C
      OPEN (UNIT = 32, FILE='/mnt/home/caterchr/ftens.csv')
      DO i = 1, 3
         WRITE (UNIT = 32, FMT=100) (DFGRD1 (i, j), j=1, 3)
      END DO
100  FORMAT (E22.10, ', ', E22.10, ', ', E22.10)
      CLOSE (32)
C
C      Write the strain total to output CSV file
C

```

```

        DO i=1,4
            STRANT(i)=STRAN(i) + DSTRAN(i)
        END DO
        OPEN (UNIT = 33, FILE='/mnt/home/caterchr/strant.csv')
        WRITE (UNIT = 33, FMT=101) (STRANT(j),j=1,4)
101    FORMAT (E22.10)
        CLOSE(33)
C
C
C *****
C    Write the increment number and element number to ext file
C
C    CALL system('del increm.csv')
        OPEN (UNIT = 40, FILE = '/mnt/home/caterchr/increm.csv')
        WRITE (UNIT = 40, FMT=200) KINC, NOEL
200    FORMAT (I5, ', ', I5)
        CLOSE(40)
C
C
C *****
C    Call the Python script to initiate the UNIT CELL BVP
C
C    com1=SYSTEMQQ('ls -la')
        com1=SYSTEMQQ('/mnt/home/caterchr/Execute.bat')
C
C
C *****
C    Read the constitutive tensor from CSV file
C
        OPEN (Unit = 33, FILE = '/mnt/home/caterchr/contTensor.csv')
        DO i = 1,4
            READ (Unit = 33, FMT=*) (CONT(i,j),j=1,4)
        END DO
        CLOSE(33)
C
C
C *****
C    Read the macroscopic stress tensor from CSV file
C
        OPEN (UNIT=34, FILE = '/mnt/home/caterchr/Stress.csv')
        READ (Unit=34, FMT = *) (STRESSAVG(i), i=1,4)
        CLOSE(34)
C
C
C *****
C    Set STRESS and DDSDDDE arrays accordingly
C
        DO i = 1,4
            STRESS(i) = STRESSAVG(i)
        END DO
        DO j = 1,4
            DO i = 1,4
                DDSDDDE(i,j) = CONT(i,j)
            END DO
        END DO
C
C    Output the Stress and Jacobian tensors to

```

```

C      the message file for debugging
C
      WRITE (UNIT = 7, FMT = *) 'Stresses are '
      WRITE (UNIT = 7, FMT = *) (STRESS(i),i=1,4)
      WRITE (UNIT = 7, FMT = *) 'Jacobian elements are '
      DO i = 1,4
        WRITE (UNIT = 7, FMT = *) (DDSDDE(i,j),j=1,4)
      END DO

      RETURN
      END

```

A.2 Python Script for RVE Generation

```
# Script for Random Fiber Placement
# Christopher Cater, Michigan State University 2012
#     Date of last modification: 3/1/2013

# Code Notes
# The code is split into two parts.
#     1) Determine fiber locations, allowing some interpenetration of
#     fibers
#     2) Perturb the fibers until there is no interpenetration

# Import necessary modules
import csv #For reading/writing CSV files
from math import pi
from math import pow
from math import cos
from math import sin
from numpy import zeros
from numpy import array
import random

##### CODE INPUT #####
numFibers = 16 # Total number of fibers
volFraction = .62 # Volume fraction target with set number of fibers
RVEsize = 8 #Length Per Edge
penTol = .2 #Initial tolerance for penetration as a fraction of fiber radius

#Placeholder for current x and y coordinates
xcord = 0.0
ycord = 0.0

#Fiber radius (based on square array and desired volume fraction)
fibRadius = round(pow(((volFraction*RVEsize**2)/(pi*numFibers)),.5),2)

##### DEFINE FUNCTIONS #####

#Define Functions
def xCoord(node):
    return centerCoords[node-1][0]
def yCoord(node):
    return centerCoords[node-1][1]
def distance(node1,node2):
    tmp = pow((xCoord(node1)-xCoord(node2))**2+(yCoord(node1)-
yCoord(node2))**2,.5)
    return tmp
def distanceCoords(x1,y1,x2,y2):
    tmp = pow((x1-x2)**2+(y1-y2)**2,.5)
    return tmp
def unitVector(node1,node2):
    mag = distance(node1,node2)
    z = [(xCoord(node1)-xCoord(node2))/mag, (yCoord(node1)-yCoord(node2))/mag]
    return z
def multVector(vector,scalar):
    for i in range(0,len(vector)):
```



```

        vector[i] = scalar*vector[i]
    return vector

def inList(a,b):
    inListFlag = 0
    for row in b:
        if a==row:
            inListFlag=1
    if inListFlag==1:
        return True
    else:
        return False

def deleteRow(a,b):
    tempHolder=[]
    for i in range(0,len(a)):
        if i!=b:
            tempHolder.append(a[i])
    return tempHolder

#Check if a fiber exists outside of the RVE
# Tests all edges in one test defined below:
def outsideRVE(fiber):
    if (xCoord(fiber)>(RVESize+fibRadius) or xCoord(fiber)<=-fibRadius
        or yCoord(fiber)>(RVESize+fibRadius) or yCoord(fiber)<=-fibRadius):
        return True
    else:
        return False

def checkNonZero(array):
    rows = len(array)
    columns = len(array[0])
    testZero = 0
    for i in range(0,rows):
        for j in range(0,columns):
            if array[i][j]!=0:
                testZero=1
    if testZero==1:
        return True
    elif testZero==0:
        return False

def checkNonZeroVector(vector):
    columns = len(vector)
    testZero = 0
    for i in range(0,columns):
        if vector[i]!=0:
            testZero=1
    if testZero==1:
        return True
    elif testZero==0:
        return False

##### DETERMINE FIBER PLACEMENTS (INITIAL) #####
#Begin iterations
#Set iteration parameters
instVector = zeros([numFibers,4])

```

```

fiberLink=[1]
fiberNumber = 2
iteration = 1
numPenFibers = 0 #Placeholder for intial number of instances of penetrating
fibers
#First Fiber (completely within RVE)
centerCoords = [[random.uniform(0+fibRadius*1.1,RVesize-fibRadius*1.1),
                  random.uniform(0+fibRadius*1.1,RVesize-fibRadius*1.1)]]
#Loop based on fiber number
while fiberNumber < (numFibers)+1:
    #Generate random fiber
    xcord = random.uniform(0+fibRadius*.1,RVesize-fibRadius*.1)
    ycord = random.uniform(0+fibRadius*.1,RVesize-fibRadius*.1)
    #Test for clearance within tolerance other fibers
    flag = 0
    #Debug output
    testNumber = 0
    for i in range(1,len(centerCoords)+1):
        test = distanceCoords(xcord,ycord,xCoord(i),yCoord(i))
        testNumber = testNumber+1
        if test > ((1-penTol)*fibRadius*2.005):
            if test < 2*fibRadius:
                numPenFibers=numPenFibers+1
                flag = flag+0
            elif test<=((1-penTol)*fibRadius*2.005):
                flag = flag+1
    if flag==0:
        #Test for fiber on RVE edge
        toAdd = [[xcord,ycord]]
        instFlag=[0,0,0,0]
        # TOP
        if ycord > RVesize-fibRadius:
            xtemp = xcord
            ytemp = ycord-RVesize
            toAdd.append([xtemp,ytemp])
            instFlag[0]=1
        # BOTTOM
        if ycord < fibRadius:
            xtemp = xcord
            ytemp = ycord+RVesize
            toAdd.append([xtemp,ytemp])
            instFlag[1]=1
        # LEFT
        if xcord < fibRadius:
            xtemp = xcord+RVesize
            ytemp = ycord
            toAdd.append([xtemp,ytemp])
            instFlag[2]=1
            #If already exiting on top, add on bottom right
            if instFlag[0]==1:
                toAdd.append([xtemp,ytemp-RVesize])
            #If already exiting on bottom, add on top right
            elif instFlag[1]==1:
                toAdd.append([xtemp,ytemp+RVesize])
        # RIGHT
        if xcord > RVesize-fibRadius:
            xtemp = xcord-RVesize

```

```

        ytemp = ycord
        toAdd.append([xtemp,ytemp])
        instFlag[3]=1
        #If already exiting on top, add on bottom left
        if instFlag[0]==1:
            toAdd.append([xtemp,ytemp-RVesize])
        #If already exiting on bottom, add on top left
        elif instFlag[1]==1:
            toAdd.append([xtemp,ytemp+RVesize])
    # Check any added fibers for penetration
    for nodes in toAdd:
        for rows in centerCoords:
            test = distanceCoords(nodes[0],nodes[1],rows[0],rows[1])
            if test > ((1-penTol)*fibRadius*2.005):
                if test < 2*fibRadius:
                    numPenFibers=numPenFibers+1
                    flag = flag+0
            elif test<=((1-penTol)*fibRadius*2.005):
                flag = flag+1
    if flag == 0:
        instVector[fiberNumber-1]=instFlag
        for row in toAdd:
            centerCoords.append(row)
            fiberLink.append(fiberNumber)
        print 'Fiber Number: ', fiberNumber
        fiberNumber = fiberNumber+1
        print 'Number of iterations: ', iteration
        print 'Number of tests: ', testNumber
        iteration = 1
    else:
        iteration = iteration+1
    if iteration > 5000:
        print 'Max iterations exceeded'
        break

print fiberLink
print centerCoords
print instVector

oldarrayCenterCoords=array(centerCoords)

#Output array to CSV file
writeFile = open('randomFiber.csv','wb')
writer = csv.writer(writeFile)
writer.writerows(oldarrayCenterCoords)
writeFile.close()
##### PERTURB FIBER PLACEMENTS #####

# The perturbation of the fiber locations will be a function of two
# unique parameters. The first relates the force with the displacement
# between the two fibers. Currently, that relationship will be a
# a scalar value which multiplies the square of the distance. Cut-off
# distance will be equal to the fiber diameter. The second parameter
# is the size of the time increment. This time parameter determines
# the amount of time passing between each perturbation of the RVE.
# NOTE: Unit-mass is assumed in order to determine F=ma

```

```

#Parameters: 1) force scale, 2) time scale
fscale =100
tscale = .001
time = 0.0 #Current time
ttime = 100.0 #Time for stop
oldtime=0

#Define functions for computing the forces
def Force(node1,node2):
    forceScalar = fscale*(distance(node1,node2)-2.1*fibRadius)**2
    if distance(node1,node2)<(2.1*fibRadius):
        forceTemp = multVector(unitVector(node1,node2),forceScalar)
    else:
        forceTemp = [0,0]
    return forceTemp
def SumForce(vector):
    rows = len(vector)
    columns = len(vector[0])
    holder = 0
    for i in range(0,rows):
        for j in range(0,columns):
            holder = holder+vector[i][j]
    return holder

#Create Force vector and velocity vector to store
forceVect = zeros([numFibers,2])
veloVect = zeros([numFibers,2])

#Build initial force vector
for i in range(1,len(centerCoords)+1):
    for j in range(1,len(centerCoords)+1):
        if i!=j:
            for k in range(0,2):
                forceVect[fiberLink[i-1]-1][k] = forceVect[fiberLink[i-1]-1][k]+Force(i,j)[k]
oldForceVect=forceVect

#Begin iteration
while checkNonZero(oldForceVect):
    #Update the force vectors
    forceVect = zeros([numFibers,2])
    #print instVector
    #print fiberLink
    for i in range(1,len(centerCoords)+1):
        for j in range(1,len(centerCoords)+1):
            if i!=j:
                for k in range(0,2):
                    # print i,j,k
                    forceVect[fiberLink[i-1]-1][k] = forceVect[fiberLink[i-1]-1][k]+Force(i,j)[k]
                    # forceVect[fiberLink[i-1]-1][k] = 1
    #Compute the resulting displacements (1 increment)
    #print forceVect
    # newDisp = origDisp + vel*time + 1/2*a*time^2
    for i in range(1,len(centerCoords)+1):
        for j in range(0,2):

```

```

        centerCoords[i-1][j] = centerCoords[i-1][j] +(
            + veloVect[fiberLink[i-1]-1][j]*tscale+forceVect[fiberLink[i-
1]-1][j]*tscale**2)
    #Update velocity vector
    for i in range(1, len(centerCoords)+1):
        for j in range(0,2):
            if forceVect[fiberLink[i-1]-1][j]==0:
                veloVect[fiberLink[i-1]-1][j] = .5*veloVect[fiberLink[i-1]-
1][j]
            else:
                veloVect[fiberLink[i-1]-1][j] = veloVect[fiberLink[i-1]-
1][j]+(
                    forceVect[fiberLink[i-1]-1][j]*tscale)
#Check if a fiber which was once completely inside the RVE
# now exits an edge. If so, create the instances
# add the coordinates to the master coordinates
# list, modify the fiberLink and instVector vectors. Also,
# check if a fiber is now the only instance, although there
# is "1" flag in the instVector slot (meaning it's instances
# have been deleted)
    holdTemp2 = 0
    for i in range(1,numFibers+1):
        #print 'i is ',i
        ind = fiberLink.index(i)
        #print 'ind is ',ind
        xcord = xCoord(ind+1)
        ycord = yCoord(ind+1)
        instFlag=0
        toAdd=[]
        # TOP
        if instVector[i-1][0]==0:
            if ycord > (RVEsize-fibRadius):
                xtemp = xcord
                ytemp = ycord-RVEsize
                toAdd.append([xtemp,ytemp])
                instVector[i-1][0]==1
                instFlag=1
                #If already exiting on left, add on bottom right
                if instVector[i-1][2]==1:
                    toAdd.append([xtemp+RVEsize,ytemp])
                #If already exiting on right, add on bottom left
                elif instVector[i-1][3]==1:
                    toAdd.append([xtemp-RVEsize,ytemp])
        # BOTTOM
        if instVector[i-1][1]==0:
            if ycord < fibRadius:
                xtemp = xcord
                ytemp = ycord+RVEsize
                toAdd.append([xtemp,ytemp])
                instVector[i-1][1]==1
                instFlag=1
                #If already exiting on left, add on top right
                if instVector[i-1][2]==1:
                    toAdd.append([xtemp+RVEsize,ytemp])
                #If already exiting on right, add on top left
                elif instVector[i-1][3]==1:
                    toAdd.append([xtemp-RVEsize,ytemp])

```

```

# LEFT
if instVector[i-1][2]==0:
    if xcord < fibRadius:
        xtemp = xcord+RVESize
        ytemp = ycord
        toAdd.append([xtemp,ytemp])
        instVector[i-1][2]==1
        instFlag=1
        #If already exiting on top, add on bottom right
        if instVector[i-1][0]==1:
            toAdd.append([xtemp,ytemp-RVESize])
        #If already exiting on bottom, add on top right
        elif instVector[i-1][1]==1:
            toAdd.append([xtemp,ytemp+RVESize])
# RIGHT
if instVector[i-1][3]==0:
    if xcord > (RVESize-fibRadius):
        xtemp = xcord-RVESize
        ytemp = ycord
        toAdd.append([xtemp,ytemp])
        instVector[i-1][3]=1
        instFlag=1
        #If already exiting on top, add on bottom left
        if instVector[i-1][0]==1:
            toAdd.append([xtemp,ytemp-RVESize])
        #If already exiting on bottom, add on top left
        elif instVector[i-1][1]==1:
            toAdd.append([xtemp,ytemp+RVESize])
if instFlag != 0:
    #print centerCoords[0:i]
    #print centerCoords[i:]
    centerCoords=centerCoords[0:ind+1]+toAdd+centerCoords[ind+1:]
    holdTemp2=1
    print toAdd
    instVector[i-1]=1
    temp = fiberLink[0:ind+1]
    print centerCoords
    if len(toAdd)>=1:
        for j in range(0,len(toAdd)):
            temp.append(i)
    print 'Added ',len(toAdd),' instances to fiber ',i, ' at index
',ind
    fiberLink = temp+fiberLink[ind+1:]
    print fiberLink
# Time condition to end the "While" perturbation loop
oldtime = oldtime+tscale
time=time+tscale
if oldtime >= 0.5:
    oldtime=0
    openName = 'randomFiber_'+str(time)+'.csv'
    writeFile = open(openName,'wb')
    writer=csv.writer(writeFile)
    arrayCenterCoords=array(centerCoords)
    writer.writerows(arrayCenterCoords)
    writeFile.close()
print time
#if holdTemp2==1:

```

```

        #print centerCoords
        oldForceVect=forceVect
        if time >= ttime:
            break

#   If a fiber is outside the RVE the following is done:
#       1) The row is deleted from centerCoords
#       2) The row is deleted from fiberLink
print centerCoords
holdTemp=0
for i in range(1,len(centerCoords)+1):
    if outsideRVE(i-holdTemp):
        centerCoords=deleteRow(centerCoords,i-1-holdTemp)
        print 'deleting ',(i-1-holdTemp), ' for fiber ',fiberLink[i-1-
holdTemp]
        fiberLink=deleteRow(fiberLink,i-1-holdTemp)
        #           #print fiberLink
        #           #print centerCoords
        holdTemp = holdTemp+1
        #           #print centerCoords

arrayCenterCoords=array(centerCoords)

#Output array to CSV file
writeFile = open('randomFiber1.csv','wb')
writer = csv.writer(writeFile)
writer.writerows(arrayCenterCoords)
writeFile.close()

```

A.3 Python Script for standard computational homogenization

```
# HEADING
# DEVELOPED BY CHRISTOPHER REYNALDO CATER AT MICHIGAN
# STATE UNIVERSITY

## Computational homogenization script for periodic
## boundary conditions.

# Removed deletion of unit-cell databases

## Input File Name for Unit-cell below:
inputFile='Modified-input-36Fibers-Periodic.inp'

#-----
# IMPORT PYTHON MODULES
#-----
from sys import path
import csv #For reading/writing CSV files
import subprocess
from odbAccess import *
from job import *
path.append('C:\Programs\Abaqus\Custom\Python24\Lib\site-packages')
from os import remove, rename
from numpy import zeros
from numpy import linalg
from numpy import where

#-----
# IMPORT COARSE SCALE INFO
#-----

## Important: Adjust the file names and paths accordingly
incredImport = open('incred.csv','rb')
incredReader = csv.reader(incredImport)
incred = incredReader.next()
for i in range(0,len(incred)):
    incred[i]=int(incred[i])

incredImport.close()

ftens = zeros([3,3])
ftensImport = open('ftens.csv','rb')
ftensReader = csv.reader(ftensImport)
for i in range(0,3):
    ftens[i] = ftensReader.next()
ftensImport.close()

#-----
# IMPORT UNIT CELL BOUNDARY NODE INFORMATION
#-----

boundImport = open('boundNodes.csv','rb')
boundReader = csv.reader(boundImport)
boundNodes = []
for row in boundReader:
```



```

        boundNodes.append(row)

boundImport.close()

numBoundNodes = len(boundNodes)

for i in range(0,len(boundNodes)): #Convert string to float
    for j in range(0,len(boundNodes[0])):
        boundNodes[i][j] = float(boundNodes[i][j])

#-----
# Calculate Boundary Displacements
#-----

# Define a useful function COLUMN for extracting
# the column of a particular matrix.

def column(matrix,i):
    return [row[i] for row in matrix]

# -----

dispBoundNodes = zeros([numBoundNodes,3])

for i in range(0,numBoundNodes):
    for j in range(0,3):
        hold = 0
        for p in range(0,3):
            hold = hold + ftens[j][p]*boundNodes[i][p+1]
        dispBoundNodes[i][j]=hold-boundNodes[i][j+1]

#-----
# MODIFY MODEL, WRITE JOB INPUT, & SUBMIT JOB
#-----

#-----
# Create a container for the corner nodes: Contained in the
# first four elements of the boundNodes array
#
cornerNodes=[int(boundNodes[0][0]),int(boundNodes[1][0]),
              int(boundNodes[2][0]),int(boundNodes[3][0])]
#
#-----

element = increm[1]
curIncrem = increm[0]
oldIncrem = curIncrem - 1
jobName = "".join(['UnitCellComplete',str(element),'-',str(curIncrem)])
oldJobName = "".join(['UnitCellComplete',str(element),'-',str(oldIncrem)])
delJobName = "".join(['UnitCellComplete',str(element),'-',str(oldIncrem-1)])

# Define the boundSect which contains the boundary conditions for the
# analysis steps - this is irrespective of whether or not one is using
# restart input files

```

```

boundSect = "".join(['** BOUNDARY CONDITIONS\r\n**\r\n*Boundary\r\nMatrix-
1.LL,1,1\r\n',
                    'Matrix-1.LL,2,2\r\nMatrix-1.LR,1,1,',
                    str(disBoundNodes[1][0]),'\r\nMatrix-1.LR, 2, 2, ',
                    str(disBoundNodes[1][1]),'\r\nMatrix-1.UL, 1, 1, ',
                    str(disBoundNodes[2][0]),'\r\nMatrix-1.UL, 2, 2, ',
                    str(disBoundNodes[2][1]),'\r\nMatrix-1.UR, 1, 1, ',
                    str(disBoundNodes[3][0]),'\r\nMatrix-1.UR, 2, 2, ',
                    str(disBoundNodes[3][1]),'\r\n**\r\n**\r\n'])

#THE NEXT SECTION OF CODE IS CONDITIONALLY BASED ON THE INCREMENT NUMBER
# For curIncr=1, the standard input file is modified

if curIncr==1:
# Read in the Unit Cell input file
# Remove the Boundary Conditions Section and
# insert the disBoundNodes conditions
# Submit the corresponding job in Abaqus
    inputRead = open(inputFile,'rb')
    whole = inputRead.read()
    inputRead.close()
    indOne = whole.find('** BOUNDARY CONDITIONS')
    firstSect = whole[0:indOne]
    indTwo = whole.find('** OUTPUT REQUESTS')
    secondSect = whole[indTwo:]
    whole = "".join([firstSect,boundSect,secondSect])

# For curIncr=2, the restart input file is modified

if curIncr==2:
# Read in the Unit Cell restart file.
# Modify the Restart section to contain an updated increment number
    inputRead = open('RestartCompPeriodic.inp','rb')
    whole = inputRead.read()
    inputRead.close()
    indOne = whole.find('** BOUNDARY CONDITIONS')
    indTwo = whole.find('** OUTPUT REQUESTS')
    indThree = whole.find('*Restart')
    indFour = whole.find('** STEP: Analysis')
    firstSect = whole[0:indThree]
    secondSect = whole[indFour:indOne]
    thirdSect = whole[indTwo:]
    insert = "".join(['**\n**\n*Restart, READ, STEP=',str(oldIncr),'\n',
                    '**\n**\n'])
    whole = "".join([firstSect,insert,secondSect,boundSect,thirdSect])

# For curIncr>3, all n-2 job files are deleted and the restart file is
modified

elif curIncr>=3:
# Perform operations similar to procedure in Incr=2, however also
# Delete job files from increment n-2, where n is current increment
# extDelete =
['.com','.dat','.mdl','.msg','.odb','.prt','.res','.sta','.stt']
    extDelete = ['.com','.dat','.mdl','.msg','.prt','.res','.sta','.stt']
    for ext in extDelete:
        subprocess.call("".join(['del ',delJobName,ext]),shell=True)

```

```

inputRead = open('RestartCompPeriodic.inp','rb')
whole = inputRead.read()
inputRead.close()
indOne = whole.find('** BOUNDARY CONDITIONS')
indTwo = whole.find('** OUTPUT REQUESTS')
indThree = whole.find('*Restart')
indFour = whole.find('** STEP: Analysis')
firstSect = whole[0:indThree]
secondSect = whole[indFour:indOne]
thirdSect = whole[indTwo:]
insert = "".join(['**\n**\n*Restart, READ, STEP=',str(oldIncr),
                  '\n**\n**\n'])
whole = "".join([firstSect,insert,secondSect,boundSect,thirdSect])

#Note: This section of code to follow is introduced to combat issues with
# Abaqus errors during the "initialization" step (zero strain). Thus the
# Analysis step section will be removed in what follows.
if (float(displacementBoundNodes[1][0])==0.0 and float(displacementBoundNodes[1][1])==0.0
    and float(displacementBoundNodes[2][0])==0.0 and
float(displacementBoundNodes[2][1])==0.0)):
    indOne = whole.find('** STEP: Analysis')
    indTwo = whole.find('** STEP: OneOne')
    firstSect = whole[0:indOne]
    secondSect = whole[indTwo:]
    whole = "".join([firstSect,secondSect])
# End of this section of code to counter the errors in the initialization.

#### PERTURBATION STEP DEFINITIONS
perturb=.01

# Modify the OneOne perturbation boundary conditions
indOne = whole.find('** STEP: OneOne')
indTwo = whole.find('*Static',indOne)
indThree = whole.find('** OUTPUT REQUESTS',indOne)
firstSect = whole[0:indTwo]
secondSect = whole[indThree:]
boundSect = "".join(['*Static\r\n** BOUNDARY
CONDITIONS\r\n**\r\n*Boundary\r\nMatrix-1.LL,1,1\r\n',
                    'Matrix-1.LL,2,2\r\nMatrix-1.LR,1,1,',
                    str(displacementBoundNodes[1][0]+
                        (boundNodes[1][1]+displacementBoundNodes[1][0])*perturb),
                    '\r\nMatrix-1.LR , 2 , 2, ',str(displacementBoundNodes[1][1]),
                    '\r\nMatrix-1.UL, 1, 1, ',str(displacementBoundNodes[2][0]),
                    '\r\nMatrix-1.UL, 2, 2, ',str(displacementBoundNodes[2][1]),
                    '\r\nMatrix-1.UR, 1, 1, ',
                    str(displacementBoundNodes[3][0]+
                        (boundNodes[3][1]+displacementBoundNodes[3][0])*perturb),
                    '\r\nMatrix-1.UR, 2, 2, ',str(displacementBoundNodes[3][1]),
                    '\r\n**\r\n**\r\n'])
whole= "".join([firstSect,boundSect,secondSect])

# Modify the TwoTwo perturbation boundary conditions
indOne = whole.find('** STEP: TwoTwo')
indTwo = whole.find('*Static',indOne)

```

```

indThree = whole.find('** OUTPUT REQUESTS',indOne)
firstSect = whole[0:indTwo]
secondSect = whole[indThree:]
boundSect = "".join(['*Static\r\n** BOUNDARY
CONDITIONS\r\n**\r\n*Boundary\r\nMatrix-1.LL,1,1\r\n',
    'Matrix-1.LL,2,2\r\nMatrix-1.LR,1,1,',
    str(displacementBoundNodes[1][0]),
    '\r\nMatrix-1.LR , 2 , 2, ',
    str(displacementBoundNodes[1][1]),'\r\nMatrix-1.UL , 1, 1, ',
    str(displacementBoundNodes[2][0]),'\r\nMatrix-1.UL, 2, 2, ',
    str(displacementBoundNodes[2][1])+
    (displacementBoundNodes[2][2]+displacementBoundNodes[2][1])*perturb),
    '\r\nMatrix-1.UR, 1, 1, ',
    str(displacementBoundNodes[3][0]),'\r\nMatrix-1.UR, 2, 2, ',
    str(displacementBoundNodes[3][1])+
    (displacementBoundNodes[3][2]+displacementBoundNodes[3][1])*perturb), '\r\n**\r\n**\r\n'])
whole= "".join([firstSect,boundSect,secondSect])

# Modify the OneTwo perturbation boundary conditions
indOne = whole.find('** STEP: OneTwo')
indTwo = whole.find('*Static',indOne)
indThree = whole.find('** OUTPUT REQUESTS',indOne)
firstSect = whole[0:indTwo]
secondSect = whole[indThree:]
boundSect = "".join(['*Static\r\n** BOUNDARY
CONDITIONS\r\n**\r\n*Boundary\r\nMatrix-1.LL,1,1\r\n',
    'Matrix-1.LL,2,2\r\nMatrix-1.LR,1,1,',
    str(displacementBoundNodes[1][0]),
    '\r\nMatrix-1.LR , 2 , 2, ',
    str(displacementBoundNodes[1][1]),'\r\nMatrix-1.UL , 1, 1, ',
    str(displacementBoundNodes[2][0])+
    (displacementBoundNodes[2][2]+displacementBoundNodes[2][1])*perturb),
    '\r\nMatrix-1.UL, 2, 2, ',
    str(displacementBoundNodes[2][1]),'\r\nMatrix-1.UR, 1, 1, ',
    str(displacementBoundNodes[3][0])+
    (displacementBoundNodes[3][2]+displacementBoundNodes[3][1])*perturb),
    '\r\nMatrix-1.UR, 2, 2, ',
    str(displacementBoundNodes[3][1]),'\r\n**\r\n**\r\n'])
whole= "".join([firstSect,boundSect,secondSect])
inputWrite = open("".join([jobName,'.inp']), 'wb')
inputWrite.write(whole)
inputWrite.close()

if curIncr>=2:
    submitString = "".join(['abaqus job=',jobName,' oldjob=',oldJobName,'
int'])
else:
    submitString = "".join(['abaqus job=',jobName,' int'])
subprocess.call(submitString,shell=True)

#-----
# Calculate Macroscopic Stress
#-----

```

```

#First open the output database
odbName = "".join([jobName, '.odb'])
odb = openOdb(odbName)

#The following "if" statement below checks to see if the displacements
# specified for all BC's are zero. In that case the analysis step
# is skipped and a zero stress tensor will automatically be outputted.

if (float(displacementBoundNodes[1][0])==0.0 and float(displacementBoundNodes[1][1])==0.0
    and float(displacementBoundNodes[2][0])==0.0 and
float(displacementBoundNodes[2][1])==0.0):
    avgStress = zeros(4)

else:
    # Import element stresses from ODB
    odbStress = odb.steps['Analysis'].frames[-1].fieldOutputs['S']
    numElem = len(odbStress.values)
    stressValues = zeros([numElem,4])
    for i in range(0,numElem):
        for j in range(0,4):
            stressValues[i][j] = odbStress.values[i].data[j]

# Extract element areas from the Analysis step
    odbArea = odb.steps['Analysis'].frames[-1].fieldOutputs['EVOL']
    elemArea = zeros(numElem)
    for i in range(0,numElem):
        elemArea[i] = odbArea.values[i].data

# Calculate total area
    totalArea = 0
    for i in range(0,numElem):
        totalArea = elemArea[i]+totalArea

# Define & Calculate Volume Average (For 2-D it is an Area Average)
    def dotproduct(a,b):
        if len(a)==len(b):
            hold=0
            for i in range(0,len(a)):
                hold = a[i]*b[i]+hold
        else:
            raise ValueError('Vectors do not have matching dimensions')
    return hold

    avgStress = zeros(4)
    for i in range(0,4):
        z = dotproduct(column(stressValues,i),elemArea)
        avgStress[i] = z/totalArea

# Export the 'avgStress' array to CSV file
writeFile = open('Stress.csv','w')
writer = csv.writer(writeFile)
writer.writerow(avgStress)
writeFile.close()

#-----
# Calculate Constitutive Tangent Operator

```

```

#-----

# It is important to note that when the non-linear geometry flag
# is active that ABAQUS expects the constitutive tensor to be of
# the form  $C = d\tau/d\epsilon$  where  $\tau$  is the Kirchhoff stress tensor

# Thus, the cauchy stresses output from the perturbation steps must be
# converted by multiplying by  $J = \det F$ 

detF = linalg.det(ftens)

# Import element stresses from ODB
# NOTE: There are three arrays; one for each direction (11,22,12)
odbStressOneOne = odb.steps['OneOne'].frames[-1].fieldOutputs['S']

# In case the Analysis step is skipped (during initialization), the num
# elem variabl needs to be defined
if (float(dispBoundNodes[1][0])==0.0 and float(dispBoundNodes[1][1])==0.0
    and float(dispBoundNodes[2][0])==0.0 and
float(dispBoundNodes[2][1]==0.0)):
    numElem = len(odbStressOneOne.values)
    # Set stressValues tensor to zero (if there was no analysis)
    stressValues = zeros([numElem,4])

oneOnestress = zeros([numElem,4])
for i in range(0,numElem):
    for j in range(0,4):
        oneOnestress[i][j] = odbStressOneOne.values[i].data[j]

odbStressTwoTwo = odb.steps['TwoTwo'].frames[-1].fieldOutputs['S']
twoTwostress = zeros([numElem,4])
for i in range(0,numElem):
    for j in range(0,4):
        twoTwostress[i][j] = odbStressTwoTwo.values[i].data[j]

odbStressOneTwo = odb.steps['OneTwo'].frames[-1].fieldOutputs['S']
oneTwostress = zeros([numElem,4])
for i in range(0,numElem):
    for j in range(0,4):
        oneTwostress[i][j] = odbStressOneTwo.values[i].data[j]

# Volume average the data and output the homogenized constitutive tensor

# If this is an intialization step, the element areas need to be extraced
here
if (float(dispBoundNodes[1][0])==0.0 and float(dispBoundNodes[1][1])==0.0
    and float(dispBoundNodes[2][0])==0.0 and
float(dispBoundNodes[2][1]==0.0)):
    # Extract element areas from the Analysis step
    odbArea = odb.steps['OneOne'].frames[-1].fieldOutputs['EVOL']
    elemArea = zeros(numElem)
    for i in range(0,numElem):
        elemArea[i] = odbArea.values[i].data

# Calculate total area

```

```

        totalArea = 0
        for i in range(0,numElem):
            totalArea = elemArea[i]+totalArea

contTensor = zeros([4,4])

for j in range(0,4):
    for i in range(0,numElem):
        contTensor[j][0] = contTensor[j][0] +
elemArea[i]*(oneOnestress[i][j]-stressValues[i][j])
        contTensor[j][1] = contTensor[j][1] +
elemArea[i]*(twoTwostress[i][j]-stressValues[i][j])
        contTensor[j][3] = contTensor[j][3] +
elemArea[i]*(oneTwostress[i][j]-stressValues[i][j])

for i in range(0,4):
    for j in range(0,4):
        contTensor[i][j] = (1/totalArea)*detF*contTensor[i][j]*(1/perturb)
contTensor[2][2] = 1

for j in range(0,4):
    contTensor[j][2] = contTensor[2][j]
contTensor[2][2]=51.6294665

symContTensor = zeros([4,4])

for i in range(0,4):
    for j in range(0,4):
        symContTensor[i][j] = .5*(contTensor[j][i]+contTensor[i][j])

# Export the 'symTensor' array to CSV file
writeFile = open('contTensor.csv','wb')
writer = csv.writer(writeFile)
writer.writerows(symContTensor)
writeFile.close()

# Export the 'symTensor' array to CSV file
# For post-processing
contName="".join([jobName,'ContTensor.csv'])
writeFile = open(contName,'wb')
writer = csv.writer(writeFile)
writer.writerows(symContTensor)
writeFile.close()

tauStress = zeros(4)
for i in range(0,4):
    tauStress[i] = detF*avgStress[i]

# Export the 'tauStress' array to CSV file
writeFile = open('StressTau.csv','w')
writer = csv.writer(writeFile)
writer.writerow(tauStress)
writeFile.close()

z = [detF]

writeFile = open('detF.csv','w')

```

```
writer = csv.writer(writeFile)
writer.writerow(z)
writeFile.close()
```


A.4 Python Script for z-scalar approach within ABAQUS

```
#####
# HEADING

## Computational homogenization script for periodic
## boundary conditions.

# Original Author: Christopher Cater
#Last Modified: 10-30-2012
#Changes: Modified script to include the non-periodic
#         boundary conditions and perturbation restart

#Note: 12-12-2012
# The odb deletion was suppressed to allow for post-
# processing of all unit cell ODB's.

# Also, the scalar variable is modified to allow for
# incorporation of the cohesive strain energy

#####
##### BEGIN CODE #####
#####

#USER INPUT#
#Input File name (Unit-cell)
fileInput='Modified-input-36Fibers.inp'

#-----
# IMPORT PYTHON MODULES
#-----
from sys import path
import csv #For reading/writing CSV files
import subprocess
from odbAccess import *
from job import *
path.append('/opt/software/NumPy/1.6.1--GCC-4.4.5/lib/python2.7/site-
packages')
from os import remove, rename
from numpy import zeros
from numpy import linalg

#-----
# IMPORT COARSE SCALE INFO
#-----

## Important: Adjust the file names and paths accordingly
incrimImport = open('incrim.csv','rb')
incrimReader = csv.reader(incrimImport)
incrim = incrimReader.next()
for i in range(0,len(incrim)):
    incrim[i]=int(incrim[i])

incrimImport.close()

# Import the Deformation Gradient
```

```

ftens = zeros([3,3])
ftensImport = open('ftens.csv','rb')
ftensReader = csv.reader(ftensImport)
for i in range(0,3):
    ftens[i] = ftensReader.next()
ftensImport.close()

# Import the macroscopic strain (total)

strant = zeros([4,1])
strantImport = open('strant.csv','rb')
strantReader = csv.reader(strantImport)
for i in range(0,4):
    strant[i] = strantReader.next()
strantImport.close()

#-----
# IMPORT UNIT CELL BOUNDARY NODE INFORMATION
#-----

#Imports boundary node list, nodes on the left, right,
# bottom and top edges to apply the BC's effectively.
boundImport = open('boundNodes.csv','rb')
boundReader = csv.reader(boundImport)
boundNodes = []
for row in boundReader:
    boundNodes.append(row)
boundImport.close()

leftImport = open('leftNodes.csv','rb')
leftReader = csv.reader(leftImport)
leftNodesAll = []
for row in leftReader:
    leftNodesAll.append(row) # "All" means it includes the corner nodes
leftImport.close()

rightImport = open('rightNodes.csv','rb')
rightReader = csv.reader(rightImport)
rightNodes = []
for row in rightReader:
    rightNodes.append(row)
rightImport.close()

bottomImport = open('bottomNodes.csv','rb')
bottomReader = csv.reader(bottomImport)
bottomNodes = []
for row in bottomReader:
    bottomNodes.append(row)
bottomImport.close()

topImport = open('topNodes.csv','rb')
topReader = csv.reader(topImport)
topNodes = []
for row in topReader:
    topNodes.append(row)
topImport.close()

```

```

numBoundNodes = len(boundNodes)
numLeftNodes = len(leftNodesAll)
numRightNodes = len(rightNodes)
numTopNodes = len(topNodes)
numBottomNodes = len(bottomNodes)

leftNodes = zeros([numLeftNodes,4])

for i in range(0,len(boundNodes)): #Convert string to float
    for j in range(0,len(boundNodes[0])):
        boundNodes[i][j] = float(boundNodes[i][j])

for i in range(0,len(leftNodes)): #Convert string to float
    for j in range(0,len(leftNodes[0])):
        leftNodes[i][j] = float(leftNodesAll[i][j])
for i in range(0,len(rightNodes)): #Convert string to float
    for j in range(0,len(rightNodes[0])):
        rightNodes[i][j] = float(rightNodes[i][j])

for i in range(0,len(bottomNodes)): #Convert string to float
    for j in range(0,len(bottomNodes[0])):
        bottomNodes[i][j] = float(bottomNodes[i][j])

for i in range(0,len(topNodes)): #Convert string to float
    for j in range(0,len(topNodes[0])):
        topNodes[i][j] = float(topNodes[i][j])

writeFile = open('leftNodesExport.csv','wb')
writer = csv.writer(writeFile)
writer.writerows(leftNodes)
writeFile.close()

#-----
# Define a usefule column function
#-----

# Define a useful function COLUMN for extracting
# the column of a particular matrix.

def column(matrix,i):
    return [row[i] for row in matrix]

#-----
# Calculate Boundary Displacements
#-----

dispBoundNodes = zeros([numBoundNodes,3])

for i in range(0,numBoundNodes):
    for j in range(0,3):
        hold = 0
        for p in range(0,3):
            hold = hold + ftens[j][p]*boundNodes[i][p+1]
        dispBoundNodes[i][j]=hold-boundNodes[i][j+1]

#-----
# MODIFY MODEL, WRITE JOB INPUT, & SUBMIT JOB

```

```

#-----

#-----
# ATTENTION::::INPUT NEEDED!!!!!!
#
# What nodes are in LowerLeft, LowerRight, and TopLeft?
#
cornerNodes=[int(boundNodes[0][0]),int(boundNodes[1][0]),
              int(boundNodes[2][0]),int(boundNodes[3][0])]
#
#-----

element = increm[1]
curIncrem = increm[0]
oldIncrem = curIncrem - 1
jobName = "".join(['UnitCellComplete',str(element),'-',str(curIncrem)])
oldJobName = "".join(['UnitCellComplete',str(element),'-',str(oldIncrem)])
delJobName = "".join(['UnitCellComplete',str(element),'-',str(oldIncrem-1)])

# Create the Insert for the Analysis Step (Boundary Conditions)
boundSect = '** BOUNDARY CONDITIONS\r\n**\r\n*Boundary\r\n'
for i in range(0,numBoundNodes):
    if boundNodes[i][0] in cornerNodes:
        if boundNodes[i][0] in column(rightNodes,0):
            boundSect = "".join([boundSect,'MATRIX-
1.',str(int(boundNodes[i][0])),', ',
                                str(2),', ',str(2),', ',
                                str(disBoundNodes[i][1]), '\r\n'])
        else:
            for j in range(0,2):
                boundSect = "".join([boundSect,'MATRIX-
1.',str(int(boundNodes[i][0])),', ',
                                    str(j+1),', ',str(j+1),', ',
                                    str(disBoundNodes[i][j]), '\r\n'])
            elif boundNodes[i][0] in column(leftNodes,0):
                boundSect = "".join([boundSect,'MATRIX-
1.',str(int(boundNodes[i][0])),', ',
                                    str(1),', ',str(1),', ',
                                    str(disBoundNodes[i][0]), '\r\n'])

#Note: This section of code to follow is introduced to combat issues with
# Abaqus errors during the "initialization" step (zero strain). Thus the
# Analysis step section will be removed in what follows.
if (float(disBoundNodes[1][0])==0.0 and float(disBoundNodes[1][1])==0.0
    and float(disBoundNodes[2][0])==0.0 and
float(disBoundNodes[2][1])==0.0):
    # End of this section of code to counter the errors in the
    initialization.
    inputRead = open(fileInput,'rb')
    whole = inputRead.read()
    indOne = whole.find('** STEP: Analysis')
    insertNoAnalysis = whole[0:indOne]
    inputRead.close()

# Read in the Unit Cell input file
# Remove the Boundary Conditions Section and
# insert the disBoundNodes conditions

```

```

# Submit the corresponding job in Abaqus
elif curIncrem==1:
    inputRead = open(fileInput,'rb')
    whole = inputRead.read()
    inputRead.close()
    indOne = whole.find('** BOUNDARY CONDITIONS')
    firstSect = whole[0:indOne]
    indTwo = whole.find('** OUTPUT REQUESTS')
    secondSect = whole[indTwo:]
    whole = "".join([firstSect,boundSect,secondSect])
    inputWrite = open("".join([jobName,'.inp']),'wb')
    inputWrite.write(whole)
    inputWrite.close()

# For curIncrem=2, the restart input file is modified

elif curIncrem==2:
    # Read in the Unit Cell restart file.
    # Modify the Restart section to contain an updated increment number
    inputRead = open('Restart.inp','rb')
    whole = inputRead.read()
    inputRead.close()
    indOne = whole.find('** BOUNDARY CONDITIONS')
    indTwo = whole.find('** OUTPUT REQUESTS')
    indThree = whole.find('*Restart')
    indFour = whole.find('** STEP: Analysis')
    firstSect = whole[0:indThree]
    secondSect = whole[indFour:indOne]
    thirdSect = whole[indTwo:]
    insert = "".join(['**\n**\n*Restart, READ, STEP=',str(oldIncrem),'\n',
                      '**\n**\n'])
    whole = "".join([firstSect,insert,secondSect,boundSect,thirdSect])

# For curIncrem>3, all n-2 job files are deleted and the restart file is
modified

elif curIncrem>=3:
    # Perform operations similar to procedure in Increm=2, however also
    # Delete job files from increment n-2, where n is current increment
    # extDelete =
    ['.com','.dat','.mdl','.msg','.odb','.prt','.res','.sta','.stt']
    # The above is commented so as to output the .odb for all steps
    extDelete = ['.com','.dat','.mdl','.msg','.prt','.res','.sta','.stt']
    for ext in extDelete:
        subprocess.call("".join(['rm -f ',delJobName,ext]),shell=True)
    inputRead = open('Restart.inp','rb')
    whole = inputRead.read()
    inputRead.close()
    indOne = whole.find('** BOUNDARY CONDITIONS')
    indTwo = whole.find('** OUTPUT REQUESTS')
    indThree = whole.find('*Restart')
    indFour = whole.find('** STEP: Analysis')
    firstSect = whole[0:indThree]
    secondSect = whole[indFour:indOne]
    thirdSect = whole[indTwo:]
    insert = "".join(['**\n**\n*Restart, READ, STEP=',str(oldIncrem),

```

```

        '\n**\n**\n'])
whole = "".join([firstSect,insert,secondSect,boundSect,thirdSect])

# Submit Job only if
if (float(displacementNodes[1][0])==0.0 and float(displacementNodes[1][1])==0.0
    and float(displacementNodes[2][0])==0.0 and
float(displacementNodes[2][1]==0.0)):
    placeholder='place'
else:
    inputWrite = open("".join([jobName, '.inp']), 'wb')
    inputWrite.write(whole)
    inputWrite.close()
    if curIncr>=2:
        submitString = "".join(['abaqus job=',jobName, ' oldjob=',oldJobName, '
int'])
    else:
        submitString = "".join(['abaqus job=',jobName, ' int'])
    subprocess.call(submitString,shell=True)

#-----
# Calculate Macroscopic Stress
#-----

#The following "if" statement below checks to see if the displacements
# specified for all BC's are zero. In that case the analysis step
# is skipped and a zero stress tensor will automatically be outputted.

if (float(displacementNodes[1][0])==0.0 and float(displacementNodes[1][1])==0.0
    and float(displacementNodes[2][0])==0.0 and
float(displacementNodes[2][1]==0.0)):
    avgStress = zeros(4)
    dispFree = zeros([len(rightNodes),3]) #empty placeholder array
    for i in range(0,len(rightNodes)): #Remainder of right side nodes
        dispFree[i][0] = rightNodes[i][0]
    dispSymm = zeros([len(leftNodes)+2,3])

    for i in range(0,len(leftNodes)):
        dispSymm[i][0]=leftNodes[i][0]

    writeFile = open('Stress.csv', 'wb')
    writer = csv.writer(writeFile)
    writer.writerow(avgStress)
    writeFile.close()
    zscale = 0 #Set to zero temporarily
    maEnergy=0 #Set to zero temporarily
    ucEnergy=0 #Set to zero temporarily

#First open the output database

else:
    odbName = "".join([jobName, '.odb'])
    odb = openOdb(odbName)
# Import element stresses from ODB
    odbStress = odb.steps['Analysis'].frames[-1].fieldOutputs['S']

```

```

numElem = len(odbStress.values)
stressValues = zeros([numElem,4])
for i in range(0,numElem):
    for j in range(0,4):
        stressValues[i][j] = odbStress.values[i].data[j]

# Extract element areas from the Analysis step
odbArea = odb.steps['Analysis'].frames[-1].fieldOutputs['EVOL']
elemArea = zeros(numElem)
for i in range(0,numElem):
    elemArea[i] = odbArea.values[i].data

# Calculate total area
totalArea = 0
for i in range(0,numElem):
    totalArea = elemArea[i]+totalArea

# Define & Calculate Volume Average (For 2-D it is an Area Average)
def dotproduct(a,b):
    if len(a)==len(b):
        hold=0
        for i in range(0,len(a)):
            hold = a[i]*b[i]+hold
    else:
        raise ValueError('Vectors do not have matching dimensions')
    return hold

avgStress = zeros(4)
for i in range(0,4):
    z = dotproduct(column(stressValues,i),elemArea)
    avgStress[i] = z/totalArea

# Extract the internal strain energy of the system
ucEnergy = odb.steps['Analysis'].historyRegions[
    'Assembly ASSEMBLY'].historyOutputs['ALLSE'].data[-1][1]

ucCohesEnergy = odb.steps['Analysis'].historyRegions['Assembly ASSEMBLY']
    .historyOutputs['ETOTAL'].data[-1][1]
#Note that this energy will be negative

# Compute the macroscopic internal strain energy
maEnergy = 0
for i in range(0,4):
    maEnergy = maEnergy + .5*strant[i]*avgStress[i]*totalArea

# Compute the z-scaling parameter
zscale = (ucEnergy-ucCohesEnergy)/maEnergy #minus signs accounts for
negative etotal

# Determine "true" energy based stressed tensor
trueStress = zeros(4)
for i in range(0,4):
    trueStress[i] = avgStress[i]*zscale

# Export the 'trueStress' array to CSV file
writeFile = open('Stress.csv','wb')

```

```

writer = csv.writer(writeFile)
writer.writerow(trueStress)
writeFile.close()

##### STEPS TO EXTRACT THE NODAL DISP #####
# This requires defining "regions" for left and right node sets

# Extract the displacements at free-edge
# These are the displacements at the end of the 'Analysis'

nodalDisp = odb.steps['Analysis'].frames[-1].fieldOutputs['U']
left = odb.rootAssembly.instances['MATRIX-1'].nodeSets['LEFTEDGESET']
right = odb.rootAssembly.instances['MATRIX-1'].nodeSets['RIGHTEDGESET']
leftDisp = nodalDisp.getSubset(region=left)
rightDisp = nodalDisp.getSubset(region=right)
dispFree = zeros([len(rightNodes),3]) #empty placeholder array

# Need to define the corner nodes separately from the
# free edge nodes since the corner nodes don't exist
# rightNodes array (they are in the boundNodes array)

for i in range(0,len(rightNodes)): #right side nodes
    for j in range(0,3):
        if j==0:
            dispFree[i][j] = int(right.nodes[i].label)
        elif j>0:
            dispFree[i][j] = rightDisp.values[i].data[j-1]

# Extract the displacements at symmetric-edge
# These are the displacements at the end of the 'Analysis'
# These are necessary for the shear perturbation step

dispSymm = zeros([len(left.nodes),3])

for i in range(0,len(left.nodes)):
    for j in range(0,3):
        if j==0:
            dispSymm[i][j]= int(left.nodes[i].label)
        elif j>0:
            dispSymm[i][j] = leftDisp.values[i].data[j-1]

#Close the ODB
odb.close()

#-----
# Modify Restart File for the Perturbation Steps
#-----

##### IMPORTANT #####
# Specify Perturbation Amount (small number ~10^-6)
perturb=.01
#####

# Open the Perturb.inp restart file
# & Insert the proper *Restart keyword entries
inputRead = open('Perturb.inp','rb')

```



```

whole = inputRead.read()
indThree = whole.find('*Restart')
indFour = whole.find('** STEP: OneOne')
if (float(displacementBoundNodes[1][0])==0.0 and float(displacementBoundNodes[1][1])==0.0
    and float(displacementBoundNodes[2][0])==0.0 and
float(displacementBoundNodes[2][1])==0.0)):
    whole = "".join([insertNoAnalysis,whole[indFour:]])

else:
    insert = "".join(['**\n**\n*Restart, READ, STEP=',str(curIncrement),'\n',
        '**\n**\n'])
    firstSect = whole[0:indThree]
    secondSect = whole[indFour:]
    whole = "".join([firstSect,insert,secondSect])

# Modify the OneOne perturbation boundary conditions
indOne = whole.find('** STEP: OneOne')
indTwo = whole.find('*Static',indOne)
indThree = whole.find('** OUTPUT REQUESTS',indOne)
firstSect = whole[0:indTwo]
secondSect = whole[indThree:]
boundSect = '*Static\r\n** BOUNDARY CONDITIONS\r\n**\r\n*Boundary\r\n'
for i in range(0,numBoundNodes):
    if boundNodes[i][0] in column(leftNodes,0): #Symmetric Edge
        if boundNodes[i][0] in cornerNodes: #Corner Nodes
            for j in range(0,1):
                boundSect = "".join([boundSect,'MATRIX-
1.',str(int(boundNodes[i][0])),', ',
                    str(j+1),', ',str(j+1),', ',
                    str(displacementBoundNodes[i][j]+(boundNodes[i][j+1]+
displacementBoundNodes[i][j])
                        *perturb),
                        '\r\n'])
            for j in range(1,2):
                boundSect = "".join([boundSect,'MATRIX-
1.',str(int(boundNodes[i][0])),', ',
                    str(j+1),', ',str(j+1),', ',
                    str(displacementBoundNodes[i][j]),
                    '\r\n'])
        else:
            boundSect = "".join([boundSect,'MATRIX-
1.',str(int(boundNodes[i][0])),', ',
                    str(1),', ',str(1),', ',
                    str(displacementBoundNodes[i][0]),
                    '\r\n'])
            elif boundNodes[i][0] in column(rightNodes,0): #Free Edge
                index = column(displacementFree,0).index(boundNodes[i][0])
                boundSect = "".join([boundSect,'MATRIX-
1.',str(int(boundNodes[i][0])),', ',
                    str(1),', ',str(1),', ',
                    str(displacementFree[index][1]+(boundNodes[i][1]+
displacementFree[index][1])*perturb),
                    '\r\n'])

```

```

        boundSect = "".join([boundSect, 'MATRIX-
1.', str(int(boundNodes[i][0])), ', ', ' ',
                                str(2), ', ', ' ', str(2), ', ', ' ',
                                str(disFree[index][2]),
                                '\r\n'])

whole= "".join([firstSect, boundSect, secondSect])

# Modify the TwoTwo perturbation boundary conditions
indOne = whole.find('** STEP: TwoTwo')
indTwo = whole.find('*Static', indOne)
indThree = whole.find('** OUTPUT REQUESTS', indOne)
firstSect = whole[0:indTwo]
secondSect = whole[indThree:]
boundSect = '*Static\r\n** BOUNDARY CONDITIONS\r\n**\r\n*Boundary\r\n'
for i in range(0, numBoundNodes):
    if boundNodes[i][0] in column(leftNodes, 0): #Symmetric Edge
        if boundNodes[i][0] in cornerNodes: #Corner Nodes
            for j in range(0, 1):
                boundSect = "".join([boundSect, 'MATRIX-
1.', str(int(boundNodes[i][0])), ', ', ' ',
                                        str(j+1), ', ', ' ', str(j+1), ', ', ' ',
                                        str(disBoundNodes[i][j]),
                                        '\r\n'])
            for j in range(1, 2):
                boundSect = "".join([boundSect, 'MATRIX-
1.', str(int(boundNodes[i][0])), ', ', ' ',
                                        str(j+1), ', ', ' ', str(j+1), ', ', ' ',
                                        str(disBoundNodes[i][j]+(boundNodes[i][j+1]+
disBoundNodes[i][j]))
                                *perturb), '\r\n'])
            else:
                boundSect = "".join([boundSect, 'MATRIX-
1.', str(int(boundNodes[i][0])), ', ', ' ',
                                        str(1), ', ', ' ', str(1), ', ', ' ',
                                        str(disBoundNodes[i][0]),
                                        '\r\n'])
            elif boundNodes[i][0] in column(rightNodes, 0): #Free Edge
                index = column(disFree, 0).index(boundNodes[i][0])
                boundSect = "".join([boundSect, 'MATRIX-
1.', str(int(boundNodes[i][0])), ', ', ' ',
                                        str(1), ', ', ' ', str(1), ', ', ' ',
                                        str(disFree[index][1]),
                                        '\r\n'])
                boundSect = "".join([boundSect, 'MATRIX-
1.', str(int(boundNodes[i][0])), ', ', ' ',
                                        str(2), ', ', ' ', str(2), ', ', ' ',
                                        str(disFree[index][2]+(boundNodes[i][2]+
disFree[index][2])*perturb),
                                '\r\n'])

whole= "".join([firstSect, boundSect, secondSect])

# Modify the OneTwo perturbation boundary conditions
indOne = whole.find('** STEP: OneTwo')

```

```

indTwo = whole.find('*Static',indOne)
indThree = whole.find('** OUTPUT REQUESTS',indOne)
firstSect = whole[0:indTwo]
secondSect = whole[indThree:]
boundSect = '*Static\r\n** BOUNDARY CONDITIONS\r\n**\r\n*Boundary\r\n'
for i in range(0,numBoundNodes):
    print 'i is ', i
    if boundNodes[i][0] in column(leftNodes,0):
        print 'node number is ', boundNodes[i][0]
        if boundNodes[i][0] in cornerNodes:
            for j in range(0,1):
                boundSect = "".join([boundSect,'MATRIX-
1.',str(int(boundNodes[i][0])),', ',
                                str(j+1),', ',str(j+1),', ',
str(dispsymNodes[i][j]+(dispsymNodes[i][j+1]
+boundNodes[i][j+2])*perturb),
                                '\r\n'])
            for j in range(1,2):
                boundSect = "".join([boundSect,'MATRIX-
1.',str(int(boundNodes[i][0])),', ',
                                str(j+1),', ',str(j+1),', ',
                                str(dispsymNodes[i][j]),
                                '\r\n'])
        else:
            index = column(dispsym,0).index(boundNodes[i][0])
            boundSect = "".join([boundSect,'MATRIX-
1.',str(int(boundNodes[i][0])),', ',
                                str(1),', ',str(1),', ',
str(dispsymNodes[i][0]+(boundNodes[i][2]+dispsym[index][2])
                                *perturb),
                                '\r\n'])
            boundSect = "".join([boundSect,'MATRIX-
1.',str(int(boundNodes[i][0])),', ',
                                str(2),', ',str(2),', ',
                                str(dispsym[index][2]),
                                '\r\n'])
        elif boundNodes[i][0] in column(rightNodes,0): #Free Edge
            index = column(dispsym,0).index(boundNodes[i][0])
            boundSect = "".join([boundSect,'MATRIX-
1.',str(int(boundNodes[i][0])),', ',
                                str(1),', ',str(1),', ',
str(dispsym[index][1]+(boundNodes[i][2]+dispsym[index][2])
                                *perturb),
                                '\r\n'])
            boundSect = "".join([boundSect,'MATRIX-
1.',str(int(boundNodes[i][0])),', ',
                                str(2),', ',str(2),', ',
                                str(dispsym[index][2]),
                                '\r\n'])
whole= "".join([firstSect,boundSect,secondSect])

#Write Final Perturb.inp Input File &

```

```

inputWrite = open("".join(['Perturb-Mod.inp']), 'wb')
inputWrite.write(whole)
inputWrite.close()

#SUBMIT THE PERTURBATION JOB

submitString = "".join(['abaqus job=Perturb-Mod oldjob=', jobName, ' int'])
subprocess.call(submitString, shell=True)

#-----
# Calculate Constitutive Tangent Operator
#-----

# Open the Perturbation output database

odb = openOdb('Perturb-Mod.odb')

# Import element stresses from ODB
# NOTE: There are three arrays; one for each direction (11,22,12)
odbStressOneOne = odb.steps['OneOne'].frames[-1].fieldOutputs['S']

# In case the Analysis step is skipped (during initialization), the num
# elem variable needs to be defined
if (float(displacementBoundNodes[1][0])==0.0 and float(displacementBoundNodes[1][1])==0.0
    and float(displacementBoundNodes[2][0])==0.0 and
float(displacementBoundNodes[2][1]==0.0)):
    numElem = len(odbStressOneOne.values)

# Import element stresses from ODB
# NOTE: There are three arrays; one for each direction (11,22,12)
odbStressOneOne = odb.steps['OneOne'].frames[-1].fieldOutputs['S']

oneOnestress = zeros([numElem,4])
for i in range(0,numElem):
    for j in range(0,4):
        oneOnestress[i][j] = odbStressOneOne.values[i].data[j]

odbStressTwoTwo = odb.steps['TwoTwo'].frames[-1].fieldOutputs['S']
twoTwostress = zeros([numElem,4])
for i in range(0,numElem):
    for j in range(0,4):
        twoTwostress[i][j] = odbStressTwoTwo.values[i].data[j]

odbStressOneTwo = odb.steps['OneTwo'].frames[-1].fieldOutputs['S']
oneTwostress = zeros([numElem,4])
for i in range(0,numElem):
    for j in range(0,4):
        oneTwostress[i][j] = odbStressOneTwo.values[i].data[j]

# Volume average the data and output the homogenized constitutive tensor

# If this is an initialization step, the element areas need to be extracted
here
if (float(displacementBoundNodes[1][0])==0.0 and float(displacementBoundNodes[1][1])==0.0

```

```

    and float(dispBoundNodes[2][0])==0.0 and
float(dispBoundNodes[2][1]==0.0)):
# Extract element areas from the OneOne step
    odbArea = odb.steps['OneOne'].frames[-1].fieldOutputs['EVOL']
    elemArea = zeros(numElem)
    for i in range(0,numElem):
        elemArea[i] = odbArea.values[i].data

# Calculate total area
    totalArea = 0
    for i in range(0,numElem):
        totalArea = elemArea[i]+totalArea

# Set stressValues tensor to zero (if there was no analysis)
    stressValues = zeros([numElem,4])
    trueStress = zeros(4)

contTensor = zeros([4,4])
testContTensor = zeros([4,4])

for j in range(0,4):
    for i in range(0,numElem):
        contTensor[j][0] = contTensor[j][0] +
elemArea[i]*(oneOnestress[i][j]-stressValues[i][j])
        contTensor[j][1] = contTensor[j][1] +
elemArea[i]*(twoTwostress[i][j]-stressValues[i][j])
        contTensor[j][3] = contTensor[j][3] +
elemArea[i]*(oneTwostress[i][j]-stressValues[i][j])

# Test Code for variation in computing the constitutive tensor
#     Here the stresses will be averaged first before subtracting
#     the stresses from the analysis step.
#     -Previously it was done using element by element differences

for j in range(0,4):
    for i in range(0,numElem):
        testContTensor[j][0] = testContTensor[j][0] +
elemArea[i]*(oneOnestress[i][j])
        testContTensor[j][1] = testContTensor[j][1] +
elemArea[i]*(twoTwostress[i][j])
        testContTensor[j][3] = testContTensor[j][3] +
elemArea[i]*(oneTwostress[i][j])

for i in range(0,4):
    for j in range(0,4):
        contTensor[i][j] = (1/totalArea)*contTensor[i][j]*(1/perturb)*zscale
        testContTensor[i][j] = (1/totalArea)*testContTensor[i][j]*(1/perturb)
    testContTensor[i][0] = testContTensor[i][0]*zscale-trueStress[i]
    testContTensor[i][1] = testContTensor[i][1]*zscale-trueStress[i]
    testContTensor[i][3] = testContTensor[i][3]*zscale-trueStress[i]
for j in range(0,4):
    contTensor[j][2] = contTensor[2][j]
    testContTensor[j][2] = testContTensor[2][j]
contTensor[2][2] =11707248018.4
testContTensor[2][2] = 11707248018.4

```

```

symContTensor = zeros([4,4])
symTestContTensor = zeros([4,4])

for i in range(0,4):
    for j in range(0,4):
        symTestContTensor[i][j] =
            .5*(testContTensor[i][j]+testContTensor[j][i])
        symContTensor[i][j] = .5*(contTensor[i][j]+contTensor[j][i])

# Export the 'testContTensor' array to CSV file
writeFile = open('testContTensor.csv','wb')
writer = csv.writer(writeFile)
writer.writerows(testContTensor)
writeFile.close()

# Export the 'contTensor' array to CSV file for post-processing
previousfile="".join([jobName,'ContTensor.csv'])
writeFile = open(previousfile,'wb')
writer = csv.writer(writeFile)
writer.writerows(symContTensor)
writeFile.close()

# Export the 'symContTensor' array to CSV file
writeFile = open('contTensor.csv','wb')
writer = csv.writer(writeFile)
writer.writerows(symContTensor)
writeFile.close()

tauStress = zeros(4)
for i in range(0,4):
    tauStress[i] = linalg.det(ftens)*avgStress[i]

# Export the 'tauStress' array to CSV file
writeFile = open('StressTau.csv','wb')
writer = csv.writer(writeFile)
writer.writerow(tauStress)
writeFile.close()

#Output for debugging:
debug = "".join(['AvgStress is ',str(avgStress),'\n The z parameter is ',
str(zscale),
'\n The UC energy is ',str(ucEnergy),'\n The ma Energy is
',str(maEnergy),
'\n Total Area is ',str(totalArea), 'Disp Free Nodes
',str(dispFree)])
writeFile = open('Debug.txt','wb')
writeFile.write(debug)
writeFile.close()

```

A.5 3D Periodic linear constraint equations for the micro-scale model

This section is segmented into two columns for brevity.

** EQUATION KEYWORDS

** BUR

*Equation

3

BUR, 2, 1.

BUL, 2, -1.

BLR, 2, -1.

*Equation

3

BUR, 1, 1.

BUL, 1, -1.

BLR, 1, -1.

*Equation

3

BUR, 3, 1.

BUL, 3, -1.

BLR, 3, -1.

*Equation

3

FLR, 1, 1.

FLL, 1, -1.

BLR, 1, -1.

*Equation

3

FLR, 2, 1.

FLL, 2, -1.

BLR, 2, -1.

*Equation

3

FLR, 3, 1.

FLL, 3, -1.

BLR, 3, -1.

*Equation

4

FUR, 1, 1.

FLL, 1, -1.

BUL, 1, -1.

BLR, 1, -1.

*Equation

4

FUR, 2, 1.

FLL, 2, -1.

BUL, 2, -1.

BLR, 2, -1.

*Equation

4

FUR, 3, 1.

FLL, 3, -1.

BUL, 3, -1.

BLR, 3, -1.

**

** Constraint for the Periodic
Faces

** FRONT/BACK

*Equation

3

Thick_1_All-1.Back, 1, -1.

Thick_1_All-1.copyBack, 1, 1.

FLL, 1, -1.

*Equation

3

Thick_1_All-1.Back, 2, -1.

Thick_1_All-1.copyBack, 2, 1.

FLL, 2, -1.

*Equation

3

Thick_1_All-1.Back, 3, -1.

Thick_1_All-1.copyBack, 3, 1.

FLL, 3, -1.

** TOP/BOTTOM

*Equation

3

Thick_1_All-1.Bottom, 1, -1.

Thick_1_All-1.copyBottom, 1, 1.

BUL, 1, -1.

*Equation

3

Thick_1_All-1.Bottom, 2, -1.

Thick_1_All-1.copyBottom, 2, 1.

BUL, 2, -1.

*Equation

3

Thick_1_All-1.Bottom, 3, -1.

Thick_1_All-1.copyBottom, 3, 1.

BUL, 3, -1.

**

** Front/Bottom Edges

**FB1

*Equation

3

BLower, 1, -1.

BUpper, 1, 1.

BUL, 1, -1.

*Equation

3

BLower, 2, -1.

BUpper, 2, 1.

BUL, 2, -1.

*Equation

3

BLower, 3, -1.

BUpper, 3, 1.

BUL, 3, -1.

```

**
**FB2
*Equation
3
BUpper, 1, -1.
FUpper, 1, 1.
FLL, 1, -1.
*Equation
3
BUpper, 2, -1.
FUpper, 2, 1.
FLL, 2, -1.
*Equation
3
BUpper, 3, -1.
FUpper, 3, 1.
FLL, 3, -1.
**
**FB3
*Equation
3
FUpper, 1, 1.
FLower, 1, -1.
BUL, 1, -1.
*Equation
3
FUpper, 2, 1.
FLower, 2, -1.
BUL, 2, -1.
*Equation
3
FUpper, 3, 1.
FLower, 3, -1.
BUL, 3, -1.
**

```

```

**
** Top/Bottom Edges
**TB3
*Equation
3
RUpper, 1, 1.
RLower, 1, -1.
BUL, 1, -1.
*Equation
3
RUpper, 2, 1.
RLower, 2, -1.
BUL, 2, -1.
*Equation
3
RUpper, 3, 1.
RLower, 3, -1.
BUL, 3, -1.
**
**
*Equation
3
FRight, 1, 1.
BRight, 1, -1.
FLL, 1, -1.
*Equation
3
FRight, 2, 1.
BRight, 2, -1.
FLL, 2, -1.
*Equation
3
FRight, 3, 1.
BRight, 3, -1.
FLL, 3, -1.

```


A.6 3D Free edge analysis linear constraint equations

This section is segmented into two columns for brevity.

```
** EQUATION KEYWORDS
** BUR
*Equation
3
BUR, 2, 1.
BUL, 2, -1.
BLR, 2, -1.
*Equation
3
BUR, 1, 1.
BUL, 1, -1.
BLR, 1, -1.
*Equation
3
BUR, 3, 1.
BUL, 3, -1.
BLR, 3, -1.
*Equation
3
FUL, 1, 1.
FLL, 1, -1.
BUL, 1, -1.
*Equation
3
FUL, 2, 1.
FLL, 2, -1.
BUL, 2, -1.
*Equation
3
FUL, 3, 1.
FLL, 3, -1.
BUL, 3, -1.
*Equation
3
FLR, 1, 1.
FLL, 1, -1.
BLR, 1, -1.
*Equation
3
FLR, 2, 1.
FLL, 2, -1.
BLR, 2, -1.
*Equation
3
FLR, 3, 1.
FLL, 3, -1.
BLR, 3, -1.
*Equation
4
FUR, 1, 1.
FLL, 1, -1.
BUL, 1, -1.
BUL, 1, -1.
BLR, 1, -1.
BUL, 1, -1.
BLR, 1, -1.
**
** Constraint for the Periodic
Faces
** FRONT/BACK
*Equation
3
Thick_1_All-1.Back, 1, -1.
Thick_1_All-1.copyBack, 1, 1.
FLL, 1, -1.
*Equation
3
Thick_1_All-1.Back, 2, -1.
Thick_1_All-1.copyBack, 2, 1.
FLL, 2, -1.
*Equation
3
Thick_1_All-1.Back, 3, -1.
Thick_1_All-1.copyBack, 3, 1.
FLL, 3, -1.
** TOP/BOTTOM
*Equation
3
Thick_1_All-1.Bottom, 1, -1.
Thick_1_All-1.copyBottom, 1, 1.
BUL, 1, -1.
*Equation
3
Thick_1_All-1.Bottom, 2, -1.
Thick_1_All-1.copyBottom, 2, 1.
BUL, 2, -1.
*Equation
3
Thick_1_All-1.Bottom, 3, -1.
Thick_1_All-1.copyBottom, 3, 1.
BUL, 3, -1.
** LEFT/RIGHT
*Equation
```

```

3
Thick_1_All-1.Left, 1, -1.
Thick_1_All-1.copyLeft, 1, 1.
BLR, 1, -1.
*Equation
3
Thick_1_All-1.Left, 2, -1.
Thick_1_All-1.copyLeft, 2, 1.
BLR, 2, -1.
*Equation
3
Thick_1_All-1.Left, 3, -1.
Thick_1_All-1.copyLeft, 3, 1.
BLR, 3, -1.
**
** Front/Bottom Edges
**FB1
*Equation
3
BLower, 1, -1.
BUpper, 1, 1.
BUL, 1, -1.
*Equation
3
BLower, 2, -1.
BUpper, 2, 1.
BUL, 2, -1.
*Equation
3
BLower, 3, -1.
BUpper, 3, 1.
BUL, 3, -1.
**
**FB2
*Equation
3
BUpper, 1, -1.
FUpper, 1, 1.
FLL, 1, -1.
*Equation
3
BUpper, 2, -1.
FUpper, 2, 1.
FLL, 2, -1.
*Equation
3
BUpper, 3, -1.
FUpper, 3, 1.
FLL, 3, -1.
**
**FB3
*Equation
3
FUpper, 1, 1.
FLower, 1, -1.
BUL, 1, -1.
*Equation

```

```

3
FUpper, 2, 1.
FLower, 2, -1.
BUL, 2, -1.
*Equation
3
FUpper, 3, 1.
FLower, 3, -1.
BUL, 3, -1.
**
**
** Top/Bottom Edges
**TB1
*Equation
3
LLower, 1, -1.
LUpper, 1, 1.
BUL, 1, -1.
*Equation
3
LLower, 2, -1.
LUpper, 2, 1.
BUL, 2, -1.
*Equation
3
LLower, 3, -1.
LUpper, 3, 1.
BUL, 3, -1.
**
**TB2
*Equation
3
LUpper, 1, -1.
RUpper, 1, 1.
BLR, 1, -1.
*Equation
3
LUpper, 2, -1.
RUpper, 2, 1.
BLR, 2, -1.
*Equation
3
LUpper, 3, -1.
RUpper, 3, 1.
BLR, 3, -1.
**
**
**TB3
*Equation
3
RUpper, 1, 1.
RLower, 1, -1.
BUL, 1, -1.
*Equation
3
RUpper, 2, 1.
RLower, 2, -1.

```

```

BUL, 2, -1.
*Equation
3
RUpper, 3, 1.
RLower, 3, -1.
BUL, 3, -1.
**
**
** Left/Right Edges
** LR1
*Equation
3
BLeft, 1, -1.
FLeft, 1, 1.
FLL, 1, -1.
*Equation
3
BLeft, 2, -1.
FLeft, 2, 1.
FLL, 2, -1.
*Equation
3
BLeft, 3, -1.
FLeft, 3, 1.
FLL, 3, -1.
**
** LR2
*Equation
3
FLeft, 1, -1.

```

```

FRight, 1, 1.
BLR, 1, -1.
*Equation
3
FLeft, 2, -1.
FRight, 2, 1.
BLR, 2, -1.
*Equation
3
FLeft, 3, -1.
FRight, 3, 1.
BLR, 3, -1.
**
** LR1
*Equation
3
FRight, 1, 1.
BRight, 1, -1.
FLL, 1, -1.
*Equation
3
FRight, 2, 1.
BRight, 2, -1.
FLL, 2, -1.
*Equation
3
FRight, 3, 1.
BRight, 3, -1.
FLL, 3, -1.

```

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Ayranci C, Carey J. 2D braided composites: A review for stiffness critical applications. *Compos Struct* 2008;85:43–58.
- [2] Jones R. Residual strength of composites with multiple impact damage. *Compos Struct* 1994;28:347–56. doi:10.1016/0263-8223(94)90117-1.
- [3] Edlund U, Volgers P. A composite ply failure model based on continuum damage mechanics. *Compos Struct* 2004;65:347–55. doi:10.1016/j.compstruct.2003.11.010.
- [4] Degrieck J, Van Paepegem W. Fatigue damage modeling of fibre-reinforced composite materials: Review. *Appl Mech Rev* 2001;54:279. doi:10.1115/1.1381395.
- [5] Pineda EJ, Waas AM, Bednarczyk BA, Collier CS, Yarrington PW. A Multiscale Progressive Damage and Failure Modeling Approach for Laminated Fiber Reinforced Composites. *Adv Math Model Exp Methods Mater Struct* 2010:43–56.
- [6] Talreja R. Damage analysis for structural integrity and durability of composite materials. *Fatigue Fract Eng Mater Struct* 2006;29:481–506. doi:10.1111/j.1460-2695.2006.00974.x.
- [7] Pagano NJ, Pipes RB. Some observations on the interlaminar strength of composite laminates. *Int J Mech Sci* 1973;15:679–92. doi:10.1016/0020-7403(73)90099-4.
- [8] Mittelstedt C, Becker W. Free-edge effects in composite laminates. *Appl Mech Rev* 2007;60:217–45.
- [9] Wang ASD, Crossman FW. Some New Results on Edge Effect in Symmetric Composite Laminates. *J Compos Mater* 1977;11:92–106. doi:10.1177/002199837701100110.
- [10] Nailadi CL, Adams DF, Adams DO. An experimental and numerical investigation of the free edge problem in composite laminates. *J Reinf Plast Compos* 2002;21:3–39. doi:10.1177/0731684402021001280.
- [11] Lorriot T, Marion G, Harry R, Wagnier H. Onset of free-edge delamination in composite laminates under tensile loading. *Compos Part B Eng* 2003;34:459–71. doi:10.1016/S1359-8368(03)00016-7.
- [12] Limam O, Foret G, Ehrlacher A. Ultimate strength of free-edge composite laminates under tensile loading: A limit analysis approach. *Compos Part B Eng* 2006;37:286–91. doi:10.1016/j.compositesb.2006.01.001.
- [13] Lagunegrand L, Lorriot T, Harry R, Wagnier H, Quenisset JM. Initiation of free-edge delamination in composite laminates. *Compos Sci Technol* 2006;66:1315–27. doi:10.1016/j.compscitech.2005.10.010.

- [14] Byron Pipes R, Goodsell J, Ritchey A, Dustin J, Gosse J. Interlaminar stresses in composite laminates: Thermoelastic deformation. *Compos Sci Technol* 2010;70:1605–11. doi:10.1016/j.compscitech.2010.05.026.
- [15] Goodsell J, Pagano NJ, Kravchenko O, Pipes RB. Interlaminar stresses in composite laminates subjected to anticlastic bending deformation. *J Appl Mech Trans ASME* 2013;80. doi:10.1115/1.4007969.
- [16] Raju IS, Crews Jr JH. Interlaminar Stress Singularities at a Straight Free Edge in Composite Laminates. DTIC Document; 1980.
- [17] Robbins, JR. DH, Reddy JN. Variable Kinematic Modelling of Laminated Composite Plates.pdf. *Int J Numer Methods Eng* 1996;39:2238–317.
- [18] Nguyen V-T, Caron J-F. A new finite element for free edge effect analysis in laminated composites. *Comput Struct* 2006;84:1538–46. doi:10.1016/j.compstruc.2006.01.038.
- [19] Nguyen V-T, Caron J-F. Finite element analysis of free-edge stresses in composite laminates under mechanical and thermal loading. *Compos Sci Technol* 2009;69:40–9. doi:10.1016/j.compscitech.2007.10.055.
- [20] Lo SH, Zhen W, Cheung YK, Wanji C. An enhanced global–local higher-order theory for the free edge effect in laminates. *Compos Struct* 2007;81:499–510. doi:10.1016/j.compstruct.2006.09.013.
- [21] Carreira RP, Caron J-F, Diaz Diaz A. Model of multilayered materials for interface stresses estimation and validation by finite element calculations. *Mech Mater* 2002;34:217–30.
- [22] Diaz Diaz A, Caron J-F, Pedro Carreira R. Software application for evaluating interfacial stresses in inelastic symmetrical laminates with free edges. *Compos Struct* 2002;58:195–208.
- [23] Tahani M, Nosier A. Free edge stress analysis of general cross-ply composite laminates under extension and thermal loading. *Compos Struct* 2003;60:91–103. doi:10.1016/S0263-8223(02)00290-8.
- [24] Nosier A, Bahrami A. Free-edge stresses in antisymmetric angle-ply laminates in extension and torsion. *Int J Solids Struct* 2006;43:6800–16. doi:10.1016/j.ijsolstr.2006.02.006.
- [25] Nosier A, Maleki M. Free-edge stresses in general composite laminates. *Int J Mech Sci* 2008;50:1435–47. doi:10.1016/j.ijmecsci.2008.09.002.
- [26] Romera JM, Cantera MA, Adarraga I, Mujika F. Application of the submodeling technique to the analysis of the edge effects of composite laminates. *J Reinf Plast Compos* 2013;32:1099–111. doi:10.1177/0731684413482995.

- [27] Esquej R, Castejon L, Lizaranzu M, Carrera M, Miravete A, Miralbes R. A new finite element approach applied to the free edge effect on composite materials. *Compos Struct* 2013;98:121–9. doi:10.1016/j.compstruct.2012.09.043.
- [28] Lecomte-Grosbras P, Paluch B, Brieu M. Characterization of free edge effects: Influence of mechanical properties, microstructure and structure effects. *J Compos Mater* 2013;47:2823–34. doi:10.1177/0021998312458817.
- [29] Dustin JS, Byron Pipes R. Free-edge singularities meet the microstructure: Important considerations. *Compos Sci Technol* 2012; 72:933–7. doi:10.1016/j.compscitech.2012.03.005.
- [30] Lecomte-Grosbras P, Paluch B, Brieu M, Saxce GD, Sabatier L. Interlaminar shear strain measurement on angle-ply laminate free edge using digital image correlation. *Compos Part Appl Sci Manuf* 2009;40:1911–20. doi:10.1016/j.compositesa.2009.07.011.
- [31] Dustin JS. Stress and Strain Field Singularities, Micro-Cracks, and Their Role in Failure Initiation at the Composite Laminate Free-Edge. Purdue University, 2012.
- [32] Belytschko T, Song J-H. Coarse-graining of multiscale crack propagation. *Int J Numer Methods Eng* 2009;n/a – n/a. doi:10.1002/nme.2694.
- [33] Sun CT, Vaidya RS. Prediction of composite properties from a representative volume element. *Compos Sci Technol* 1996;56:171–9.
- [34] Kim BR, Lee HK. Elastoplastic modeling of circular fiber-reinforced ductile matrix composites considering a finite RVE. *Int J Solids Struct* 2010;47:827–36. doi:10.1016/j.ijsolstr.2009.11.015.
- [35] Hill R. Elastic properties of reinforced solids: Some theoretical principles. *J Mech Phys Solids* 1963;11:357–72. doi:10.1016/0022-5096(63)90036-X.
- [36] Hill R. Theory of Mechanical Properties of Fibre-Strengthened Materials: III, Self-Consistent Mode. *Journal of the Mechanics of Physics and Solids* 1965;13.
- [37] Eshelby JD. The Determination of the Elastic Field of an Ellipsoidal Inclusion, and Related Problems. *Proc R Soc Lond Ser Math Phys Sci* 1957;241:376–96. doi:10.1098/rspa.1957.0133.
- [38] Mori T, Tanaka K. Average stress in matrix and average elastic energy of materials with misfitting inclusions. *Acta Metall* 1973;21:571–4. doi:10.1016/0001-6160(73)90064-3.
- [39] Paley M, Aboudi J. Micromechanical analysis of composites by the generalized cells model. *Mech Mater* 1992;14:127–39. doi:10.1016/0167-6636(92)90010-B.
- [40] Whitney J, McCullough R. Micromechanical Materials Modeling. vol. 2. Delaware Composites Design Encyclopedia. Technomic Publishing Company; 1990.

- [41] Chung PW, Tamma KK, Namburu RR. Asymptotic expansion homogenization for heterogeneous media: computational issues and applications. *Compos Part Appl Sci Manuf* 2001;32:1291–301.
- [42] Kouznetsova V. Computational Homogenization. Grad Sch Mech GraSMech Eindh Lect Notes 2004.
- [43] González C, LLorca J. Mechanical behavior of unidirectional fiber-reinforced polymers under transverse compression: microscopic mechanisms and modeling. *Compos Sci Technol* 2007;67:2795–806.
- [44] BANERJEE S, SANKAR B. Mechanical Properties of Hybrid Composites using Finite Element Method Based Micromechanics n.d.
- [45] Aitharaju VR, Averill RC. Three-dimensional properties of woven-fabric composites. *Compos Sci Technol* 1999;59:1901–11. doi:10.1016/S0266-3538(99)00049-4.
- [46] Huang Y, Xu L, Ha SK. Prediction of three-dimensional composite laminate response using micromechanics of failure. *J Compos Mater* 2012;46:2431–42. doi:10.1177/0021998312449888.
- [47] Sun XS, Tan VBC, Tay TE. Micromechanics-based progressive failure analysis of fibre-reinforced composites with non-iterative element-failure method. *Comput Struct* 2011;89:1103–16.
- [48] LLorca J, González C, Molina-Aldareguía JM, Segurado J, Seltzer R, Sket F, et al. Multiscale Modeling of Composite Materials: a Roadmap Towards Virtual Testing. *Adv Mater* 2011:n/a – n/a. doi:10.1002/adma.201101683.
- [49] Canal LP, Segurado J, LLorca J. Failure surface of epoxy-modified fiber-reinforced composites under transverse tension and out-of-plane shear. *Int J Solids Struct* 2009;46:2265–74.
- [50] Yan CK. On homogenization and de-homogenization of composite materials. Drexel University, 2003.
- [51] Chung S. Effects of interlaminar stress gradients on free edge delamination in composite laminates. Drexel University, 2003.
- [52] Soden P., Kaddour A., Hinton M. Recommendations for designers and researchers resulting from the world-wide failure exercise. *Compos Sci Technol* 2004;64:589–604. doi:10.1016/S0266-3538(03)00228-8.
- [53] Christensen RM. The World Wide Failure Exercise II Examination of Results. *J Reinf Plast Compos* 2013;32:1668–72. doi:10.1177/0731684413498634.

- [54] Nilakantan G, Keefe M, Bogetti TA, Gillespie Jr. JW. Multiscale modeling of the impact of textile fabrics based on hybrid element analysis. *Int J Impact Eng* 2010;37:1056–71. doi:10.1016/j.ijimpeng.2010.04.007.
- [55] Ghosh S, Lee K, Raghavan P. A multi-level computational model for multi-scale damage analysis in composite and porous materials. *Int J Solids Struct* 2001;38:2335–85.
- [56] Raghavan P, Moorthy S, Ghosh S, Pagano NJ. Revisiting the composite laminate problem with an adaptive multi-level computational model. *Compos Sci Technol* 2001;61:1017–40. doi:10.1016/S0266-3538(00)00230-X.
- [57] Liu WK, McVeigh C. Predictive multiscale theory for design of heterogeneous materials. *Comput Mech* 2007;42:147–70. doi:10.1007/s00466-007-0176-8.
- [58] Rao MP, Nilakantan G, Keefe M, Powers BM, Bogetti TA. Global/Local Modeling of Ballistic Impact onto Woven Fabrics. *J Compos Mater* 2009;43:445–67. doi:10.1177/0021998308097684.
- [59] Bensoussan A, Lions J-L, George Papanicolaou. *Asymptotic Analysis for Periodic Structures*. vol. 5. Elsevier; 1978.
- [60] Bigaud D, Hamelin P. Stiffness and failure modelling of 2D and 3D textile-reinforced composites by means of imbricate-type elements approaches. *Comput Struct* 2002;80:2253–64. doi:10.1016/S0045-7949(02)00248-1.
- [61] Kanouté P, Boso DP, Chaboche JL, Schrefler BA. Multiscale Methods for Composites: A Review. *Arch Comput Methods Eng* 2009;16:31–75. doi:10.1007/s11831-008-9028-8.
- [62] Bahei-El-Din YA, Rajendran AM, Zikry MA. A micromechanical model for damage progression in woven composite systems. *Int J Solids Struct* 2004;41:2307–30.
- [63] Aboudi J, Pindera MJ, Arnold SM. Higher-order theory for periodic multiphase materials with inelastic phases. *Int J Plast* 2003;19:805–47.
- [64] Oskay C, Fish J. Eigendeformation-based reduced order homogenization for failure analysis of heterogeneous materials. *Comput Methods Appl Mech Eng* 2007;196:1216–43.
- [65] Fish J, Oskay C. A nonlocal multiscale fatigue model. *Mech Adv Mater Struct* 2005;12:485–500.
- [66] Feyel F, Chaboche JL. FE² multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials. *Comput Methods Appl Mech Eng* 2000;183:309–30.
- [67] Geers MGD, Kouznetsova VG, Brekelmans WAM. Multi-scale computational homogenization: Trends and challenges. *J Comput Appl Math* 2010;234:2175–82.

- [68] Alfano G. On the influence of the shape of the interface law on the application of cohesive-zone models. *Compos Sci Technol* 2006;66:723–30.
- [69] Volokh KY. Comparison between cohesive zone models. *Commun Numer Methods Eng* 2004;20:845–56. doi:10.1002/cnm.717.
- [70] Huynh DBP, Belytschko T. The extended finite element method for fracture in composite materials. *Int J Numer Methods Eng* 2009;77:214–39. doi:10.1002/nme.2411.
- [71] Dolbow J, Moës N, Belytschko T. Discontinuous enrichment in finite elements with a partition of unity method. *Finite Elem Anal Des* 2000;36:235–60. doi:10.1016/S0168-874X(00)00035-4.
- [72] Kouznetsova VG, Geers MGD, Brekelmans WAM. Multi-scale second-order computational homogenization of multi-phase materials: A nested finite element solution strategy. *Comput Methods Appl Mech Eng* 2004;193:5525–50. doi:10.1016/j.cma.2003.12.073.
- [73] Andrews EW, Garnich MR. Stresses around fiber ends at free and embedded ply edges. *Compos Sci Technol* 2008;68:3352–7. doi:10.1016/j.compscitech.2008.09.001.
- [74] Garnich MR, Dalgarno RW, Kenik DJ. Effects of moisture on matrix cracking in a cryo-cycled cross-ply laminate. *J Compos Mater* 2011;45:2783–95. doi:10.1177/0021998311410468.
- [75] Fish J, Wagiman A. Multiscale finite element method for a locally nonperiodic heterogeneous medium. *Comput Mech* 1993;12:164–80.
- [76] Dustin JS, Pipes RB. Multi-Scale Modeling of Free-Edge Micro-Cracks with XFEM, American Institute of Aeronautics and Astronautics; 2014. doi:10.2514/6.2014-0995.
- [77] Van der Sluis O, Schreurs P, Brekelmans W, Meijer H. Overall behaviour of heterogeneous elastoviscoplastic materials: effect of microstructural modelling. *Mech Mater* 2000;32:449–62.
- [78] Inglis HM, Geubelle PH, Matouš K. Boundary condition effects on multiscale analysis of damage localization. *Philos Mag* 2008;88:2373–97. doi:10.1080/14786430802345645.
- [79] Kaczmarczyk Ł, Pearce CJ, Bićanić N. Scale transition and enforcement of RVE boundary conditions in second-order computational homogenization. *Int J Numer Methods Eng* 2008;74:506–22. doi:10.1002/nme.2188.
- [80] Kaczmarczyk L, Pearce CJ, Bicanic N, de Souza Neto E. Numerical multiscale solution strategy for fracturing heterogeneous materials. *Comput Methods Appl Mech Eng* 2010;199:1100–13.
- [81] González C, LLorca J. Multiscale modeling of fracture in fiber-reinforced composites. *Acta Mater* 2006;54:4171–81.

- [82] LLorca J, Cox BN. Virtual fracture testing of composite materials and structures. *Int J Fract* 2009;158:97–97. doi:10.1007/s10704-009-9383-y.
- [83] Aboudi J, Ryvkin M. The effect of localized damage on the behavior of composites with periodic microstructure. *Int J Eng Sci* 2011.
- [84] Coenen EWC, Kouznetsova VG, Geers MGD. Novel boundary conditions for strain localization analyses in microstructural volume elements. *Int J Numer Methods Eng* 2012;90:1–21. doi:10.1002/nme.3298.
- [85] Larsson F, Runesson K, Saroukhani S, Vafadari R. Computational homogenization based on a weak format of micro-periodicity for RVE-problems. *Comput Methods Appl Mech Eng* 2011;200:11–26. doi:10.1016/j.cma.2010.06.023.
- [86] González C, LLorca J. Mechanical behavior of unidirectional fiber-reinforced polymers under transverse compression: Microscopic mechanisms and modeling. *Compos Sci Technol* 2007;67:2795–806. doi:10.1016/j.compscitech.2007.02.001.
- [87] Xie D, Waas AM. Discrete cohesive zone model for mixed-mode fracture using finite element analysis. *Eng Fract Mech* 2006;73:1783–96.
- [88] Canal LP, Segurado J, LLorca J. Failure surface of epoxy-modified fiber-reinforced composites under transverse tension and out-of-plane shear. *Int J Solids Struct* 2009;46:2265–74.
- [89] Inglis HM, Geubelle PH, Matouš K. Boundary condition effects on multiscale analysis of damage localization. *Philos Mag* 2008;88:2373–97. doi:10.1080/14786430802345645.
- [90] Kouznetsova VG, Geers M, Brekelmans WAM. Size of a representative volume element in a second-order computational homogenization framework. *Int J Multiscale Comput Eng* 2004;2.
- [91] Aboudi J, Arnold SM, Bednarczyk BA. *Micromechanics of Composite Materials: A Generalized Multiscale Analysis Approach*. Butterworth-Heinemann; 2012.
- [92] Shokrieh MM, Kamali Shahri SM. Modeling residual stresses in composite materials. *Residual Stress*. Compos. Mater., Elsevier; 2014, p. 173–93.
- [93] Kim YK, White SR. Stress relaxation behavior of 3501-6 epoxy resin during cure. *Polym Eng Sci* 1996;36:2852–62. doi:10.1002/pen.10686.
- [94] Micromechanics of failure: a new approach to modelling composite failure and life | JEC Composites 2014. <http://www.jeccomposites.com/news/composites-news/micromechanics-failure-new-approach-modelling-composite-failure-and-life> (accessed February 12, 2015).
- [95] Yuan Z, Fish J. Toward realization of computational homogenization in practice. *Int J Numer Methods Eng* 2008;73:361–80. doi:10.1002/nme.2074.