Progressive damage analysis of realistic woven composites using an automated conformal finite element mesh model

by

Agniprobho Mazumder

B.E. (Honors), Indian Institute of Engineering Science and Technology, India, 2016

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Alan Levin Department of Mechanical and Nuclear Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Abstract

An automated procedure for generating conformal finite element volume meshes of woven composite micro-geometries is developed in order to improve micro-mechanics analysis of woven composites. The Digital Element Approach (DEA) facilitates the generation of realistic fiber-level micro-geometries of woven composite unit cells using a dynamic relaxation technique. A triangular surface mesh is then created to represent the fiber-level unit cell at the meso-scale, or yarn-level. The unit cell domain is first discretized into a uniform cuboid element mesh using 8-node hexahedron elements. Numerical imperfections such as yarn interpenetrations or narrow gaps between yarns are then removed by an efficient intersection search algorithm. A sequential process of nodal rearrangement and element splitting at the yarn-yarn and yarn-matrix interface is employed to transform the initial mesh domain into a conformal finite element mesh. The conformal meshing ensures that the element boundaries match the meso-scale geometry boundaries perfectly and that the shape functions across shared element faces are compatible. The quality of each element is examined and a mesh repair algorithm is employed to improve element quality if required. The mesh generation strategy uses realistic micro-geometries with irregular cross sections and yarn paths, and hence no idealized assumptions are made, which does not limit its application – it can be used for any general woven architecture or geometry.

Next, the stress and failure response of the unit cell meshed models is studied using a progressive damage model that is implemented using a user-defined material subroutine. The yarns in the woven composite domain are assumed to be comprised transversely isotropic elements while the surrounding polymeric matrix is assumed to be purely isotropic. Hence the yarn elements can be modeled as unidirectional composite laminates using this meso-scale idealization. The local orientations of the yarn elements are determined according to the yarn

central path from their micro-geometry. A recursive stiffness degradation method is employed to model damage evolution in yarns, while damage initiation is governed by the maximum strain criteria. This damage evolution model also takes into account combined failure modes. In case of the matrix, the maximum stress criterion is used with the instant stiffness degradation method. The validity of the model is established by comparing the results with available experimental and numerical data for two very different unit cells.

Finally, a purpose-driven design study is performed. A comparative analysis is presented where three different woven architectures – 3D-woven layer-to-layer, 3D-woven orthogonal and multi-layer 2D-woven unit cells – with similar overall fiber volume fraction and yarn fiber volume fraction are analyzed. The in-plane stiffness and strength results are compared and discussed. Failure mechanisms are discussed in detail and validated using experimental observations in literature.

Progressive damage analysis of realistic woven composites using an automated conformal finite element mesh model

by

Agniprobho Mazumder

B.E. (Honors), Indian Institute of Engineering Science and Technology, India, 2016

A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Alan Levin Department of Mechanical and Nuclear Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Approved by:

Major Professor
Dr. Youqi Wang

Copyright

Abstract

An automated procedure for generating conformal finite element volume meshes of woven composite micro-geometries is developed in order to improve micro-mechanics analysis of woven composites. The Digital Element Approach (DEA) facilitates the generation of realistic fiber-level micro-geometries of woven composite unit cells using a dynamic relaxation technique. A triangular surface mesh is then created to represent the fiber-level unit cell at the meso-scale, or yarn-level. The unit cell domain is first discretized into a uniform cuboid element mesh using 8-node hexahedron elements. Numerical imperfections such as yarn interpenetrations or narrow gaps between yarns are then removed by an efficient intersection search algorithm. A sequential process of nodal rearrangement and element splitting at the yarn-yarn and yarn-matrix interface is employed to transform the initial mesh domain into a conformal finite element mesh. The conformal meshing ensures that the element boundaries match the meso-scale geometry boundaries perfectly and that the shape functions across shared element faces are compatible. The quality of each element is examined and a mesh repair algorithm is employed to improve element quality if required. The mesh generation strategy uses realistic micro-geometries with irregular cross sections and yarn paths, and hence no idealized assumptions are made, which does not limit its application – it can be used for any general woven architecture or geometry.

Next, the stress and failure response of the unit cell meshed models is studied using a progressive damage model that is implemented using a user-defined material subroutine. The yarns in the woven composite domain are assumed to be comprised transversely isotropic elements while the surrounding polymeric matrix is assumed to be purely isotropic. Hence the yarn elements can be modeled as unidirectional composite laminates using this meso-scale idealization. The local orientations of the yarn elements are determined according to the yarn

central path from their micro-geometry. A recursive stiffness degradation method is employed to model damage evolution in yarns, while damage initiation is governed by the maximum strain criteria. This damage evolution model also takes into account combined failure modes. In case of the matrix, the maximum stress criterion is used with the instant stiffness degradation method. The validity of the model is established by comparing the results with available experimental and numerical data for two very different unit cells.

Finally, a purpose-driven design study is performed. A comparative analysis is presented where three different woven architectures – 3D-woven layer-to-layer, 3D-woven orthogonal and multi-layer 2D-woven unit cells – with similar overall fiber volume fraction and yarn fiber volume fraction are analyzed. The in-plane stiffness and strength results are compared and discussed. Failure mechanisms are discussed in detail and validated using experimental observations in literature.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my sincere gratitude to my major professor and academic advisor Dr. Youqi Wang for her support, advice and intellectual guidance throughout my Ph.D. research. I am also thankful for her discussions and constructive criticism. I would also like to thank my committee members Dr. X.J. Xin, Dr. William Hsu, Dr. Chian-Fong Yen and Dr. Raj Kumar Pal for their invaluable time. It was a great pleasure for me to work in the composites group with my colleague Dr. Qibang Liu. This research work was financially supported by the U.S. Army Research Laboratory (ARL).

I would like to extend a special thank you to Dr. Zheng Li at GE Research Center and Dr. Yong Yu at GE Aviation for their insights and input on my research and for giving me the opportunity to work alongside them, as well as Dr. Lun Li for his help with ANSYS.

I would like to thank my friends who have made this journey a smooth one – Aritri Majumdar, Somnath Chakravorty, Joydeep Mitra, Nandini Sarkar, Zayed Ahmed, Ranojoy Chatterjee, Ayyappan Elangovan, Asrar Ahmed, Aditya Narkar, Sandeep Dasari, Mahima Suresh, Rajarshi Pal Chowdhury, Chandrima Shyam, Sidney & Jamison Friesen, Mohammad Bisheh, David Stuckenschmidt, Jeremy Ensz and Rishav Mukherjee. Thank you for the wonderful times we had.

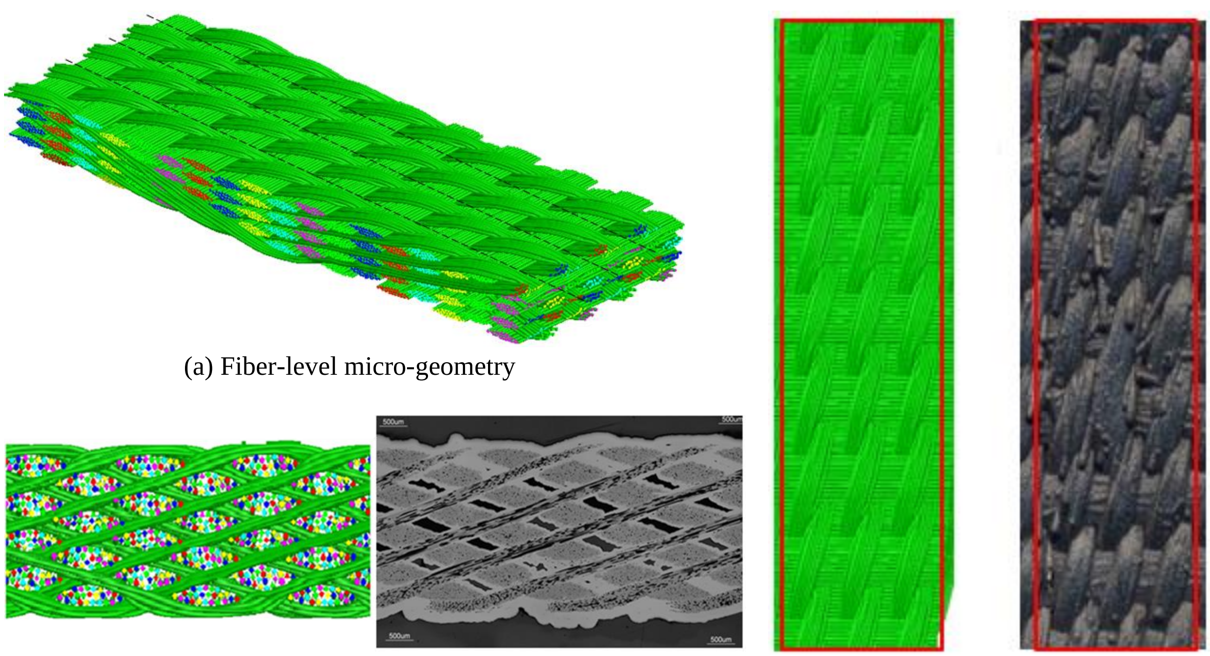
Finally, I would like to show immense appreciation to Grace Benedick, my parents – Manabendra Nath Mazumder and Arita Mazumder and my grandmother, Gouri Mazumder, for their love, support and encouragement in completion of this degree.

"*It's simply beyond words. It's incalculable.*" – Michael Scott.

Chapter 1 - Introduction

Fabric-reinforced composites are widely used as structural components. This calls for more accurate and reliable numerical tools to model these fabric micro-geometries. More accurate modeling of the textile preforms results in better prediction of their micromechanical properties. The woven architecture plays a key role in determining their mechanical response and applications. There has been a rise in the use of advanced fabric-reinforced composites for industrial applications since the 1990s, and with the advancement of numerical computing power, there has been an onset of development computer design tools for modeling their micro-geometries. Since modeling the entire fabric reinforced composite is computationally expensive, a “unit cell” of the fabric is generally used for determining the micromechanical behavior of such composites. A unit cell is the smallest repeating unit in a composite that is able to represent its bulk behavior as a whole. WiseTex, a software package developed by Verpoest *et al*. [1, 2] uses the weaving pattern to define the fabric unit cell topology. The required input parameters of bending rigidity and yarn transverse stiffness are usually difficult to determine. Additionally, yarn cross-sections are idealized as elliptical or lenticular in shape and yarn path is assumed to be a polynomial function. Another popular open-source fabric unit cell modeling software is TexGen [3]. It relies heavily on empirical or experimental data to determine yarn cross-section.

Finite element modeling of the unit cell is done at the meso-scale, i.e., using the yarn-level micro-geometries. Thus, a detailed representation of the unit cell micro-geometry is desired for better prediction of their mechanical behavior [4, 5]. A fiber-level micromechanics model termed as the digital element approach (DEA) was developed by Wang and her coworkers [6, 7], where yarns are modeled as a bundle of digital fibers, and each digital fiber is discretized as a series of interconnected rod-elements joined by frictionless pins. The yarns are woven into a pattern to form the unit cell. A digital fiber represents multiple actual fibers, which significantly reduces the computation cost for achieving a relaxed unit cell micro-geometry. The unit cell topology, fiber tension, inter-fiber compression and contact forces determine the fabric unit cell micro-geometry. Plastic deformation and rearrangement of fibers within the yarn during a dynamic relaxation process, to achieve a minimum total energy, generates a more realistic fiber-level micro-geometry. Figure 1.1 shows a fiber level unit cell generated using DFCA and a comparison with a test sample (MRD specimen 1748-01-002 "AngInt"). The DEA Fabric and Composite Analyzer or DFCA [8] (previously known as DEA Fabric Mechanics Analyzer, DFMA) implements this approach. Figure 1.2(a) shows a fiber-level micro-geometry of a 3D woven orthogonal unit cell generated using DFCA. The fiber-level micro-geometry can be converted into yarn-level micro-geometry (Figure 1.2(b)), where the geometry of each yarn is



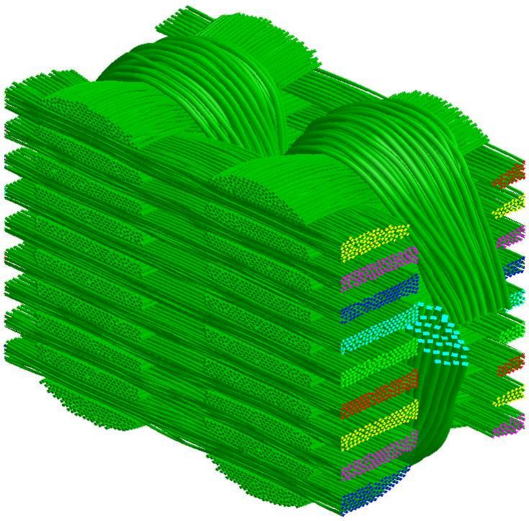
(a) Fiber-level micro-geometry

(b) Weft cross-section comparison

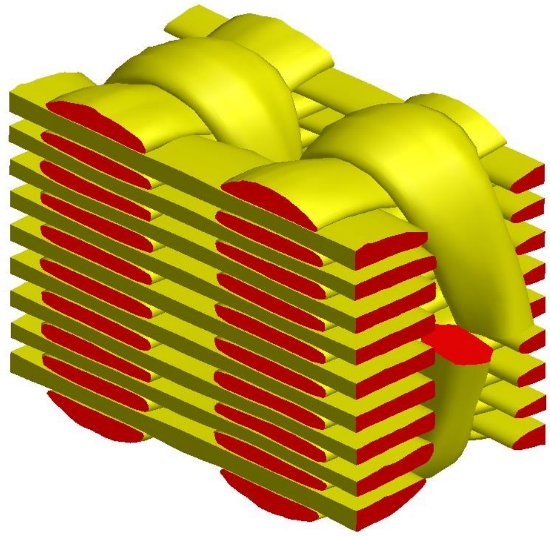
(c) Top view comparison

Figure 1.1: 3D woven angle interlock fabric unit cell generated using DFCA

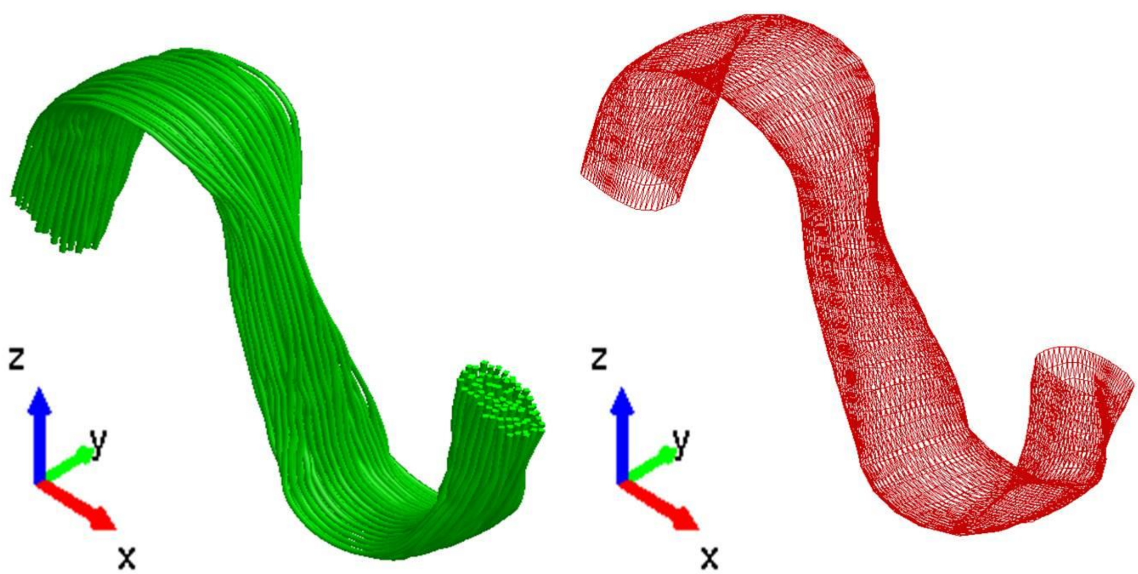
defined as a triangular surface mesh, as shown in Figure 1.2(c). The DEA approach has also been used to create other textile composite design software such as the Virtual Textile Morphology Suite (VTMS) software [9].



(a) Fiber-level micro-geometry

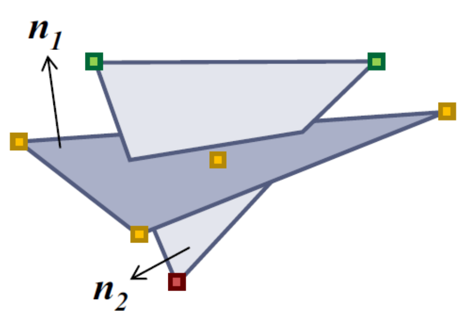


(b) Yarn-level micro-geometry

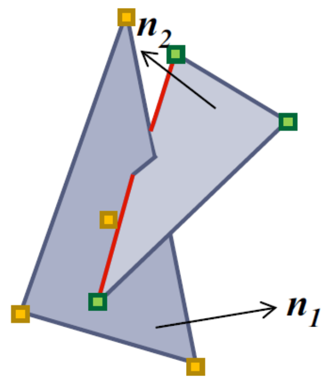


(c) Triangular surface mesh of a single yarn

Figure 1.2: Illustration of yarn-level micro-geometry generation



(a) Surface penetration



(b) Edge penetration

Figure 1.3: Instances of yarn element interpenetration [10]

Such numerically generated micro-geometries inevitably exhibit numerical imperfections such as yarn interpenetrations and narrow gaps between yarns. An instance of such yarn element interpenetrations is shown in Figure 1.3. While the yarn surface element interpenetrations make it problematic to directly input these unit cells into commercial finite element analysis (FEA) tools for mesh generation and solution, the narrow gaps in between yarns result in narrow slivers of matrix elements with high aspect ratios that could lead to stress concentration in those regions leading to premature failure of the composite unit cell. Furthermore, several manual preprocessing steps are required to define the local orientations of the orthotropic yarn elements, mesh the surrounding isotropic matrix and assembly of the reinforcing yarns and surrounding matrix. Hence the yarn-level micro-geometries thus generated need to be further preprocessed for FEA.

The use of FEA-based techniques for mesh generation of complex geometries is widely used since the advent of high-performance computing. The different methods that have been proposed to resolve the aforementioned issues and generate a unit cell mesh from yarn-level

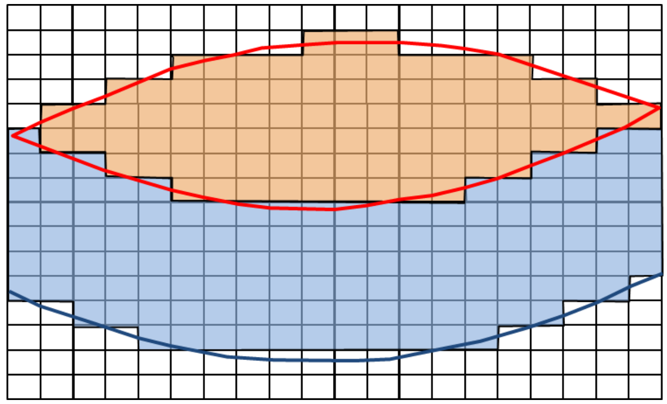
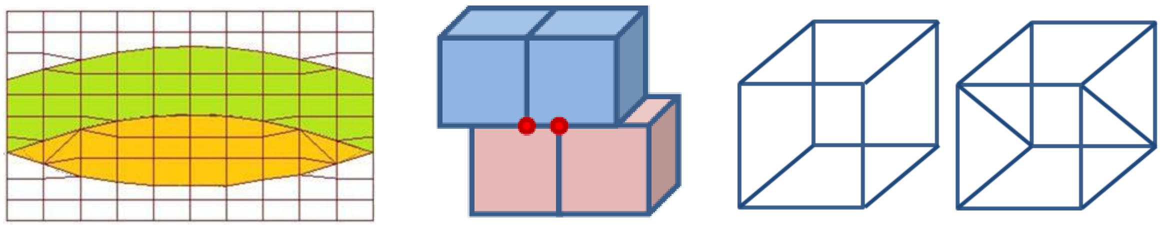


Figure 1.4: General voxel mesh illustration for a plain woven cross-section

micro-geometries can be grouped into two major categories: (1) non-conformal mesh generation techniques and (2) conformal mesh generation techniques.

Among the non-conformal mesh based techniques, the most commonly used ones adopt a voxel based meshing strategy [11, 12, 13, 14], as shown by a simplified illustration in Figure 1.4. In this method, the unit cell domain is discretized into a grid of uniformly sized hexahedra. Elements are either assigned to the yarn or the matrix based on the volume fraction of elements that lie inside the yarn boundary. These techniques have been shown to generate good results for linear elastic analysis of the woven composite unit cell. However, the irregular yarn boundaries lead to stress concentration at the sharp element edges and cause interlocking. Additionally their ability to predict non-linear response and damage of the fabric unit cells is not very well understood [15]. Other works in this category involve that of Nakai *et al*. [16] who proposed a mesh superposition technique and Durville [17] who proposed a meshing technique that connected the yarn reinforcements to the surrounding elastic matrix. Wucher *et al*. [18] proposed a procedure where the mesh was refined at yarn-yarn and yarn-matrix interfaces. Although this approach solved the yarn inter-penetration problem but the zigzag shapes of the yarn boundaries with sharp edges, would induce irregular stress distribution.



(a) Smooth boundary (b) Nodal mismatch (c) Adjacent element face discontinuity

Figure 1.5: Conformal mesh conditions

The conformal meshing technique on the other hand must satisfy three conditions: (1) yarn surfaces match element boundaries as shown in Figure **1.5**(a), (2) every node on one side of the interface matches a node on the other side of the interface – nodal mismatches, as shown in Figure **1.5**(b), are not permitted, and, (3) every element surface on one side of the interface matches an element surface on the other side of the interface – element face discontinuity, as shown in Figure 1.5(c), is not permitted, in order to preserve the shape function continuity across element boundaries. Thus this method required no additional interpolation at the interface for by the yarn and surrounding matrix boundaries. Also, the smooth surfaces ensure that no artificial stress concentrations are introduced in the numerical analysis. This results in faster and more accurate computation of the unit cell response. Different researchers have proposed various conformal mesh based techniques over the years. Notable works include those by Drach *et al*. [10, 19], who used the DFCA unit cell micro-geometries for generating the unit cell mesh. Two methods for correcting yarn interpenetrations were proposed to generate a modified yarn surface mesh. A final volume mesh was generated based on the modified surface meshes. Grail *et al*. [15] generated conformal yarn surface meshes by eliminating narrow gaps and yarn-interpenetrations near regions where yarn contact was detected. A tetrahedral volume mesh was generated from the surface mesh using Distenes' meshing tools. Lomov *et al*. [20] performed a

3-step penetration correction procedure to remove yarn interpenetrations from solid yarn models generated using WiseTex. A hierarchial pixel-voxel mesh of a desired resolution was developed by Kim *et al*. [21] that combined attributes of voxel-based meshing with conformal meshing techniques. McQuien [22] used separate mesh models for the reinforcing yarns and surrounding matrix to analyze stress distribution within the unit cells using a technique termed as the Independent Mesh Method (IMM). Fish *et al*. [23] created a multi-scale designer equipped with a built-in parametric library of idealized unit cell models with the option to generate user defined simplistic unit cell models that permit optimization and stochastic simulations. A similar method as WiseTex to generate fabric unit cell geometries was proposed by Ha *et al*. [24]. A mesh adaptation method was used to generate the final unit cell mesh. Bergan [25] generated the unit cell micro-geometry using VTMS to create a volume mesh of the reinforcing lines using the surface mesh data. The surrounding matrix in the domain was modeled and meshed separately followed by an assembly procedure to obtain the entire unit cell. Joglekar *et al*. [26] used DFCA to generate the fabric unit cell micro-geometry and obtained a conformal volume mesh using ABAQUS™.

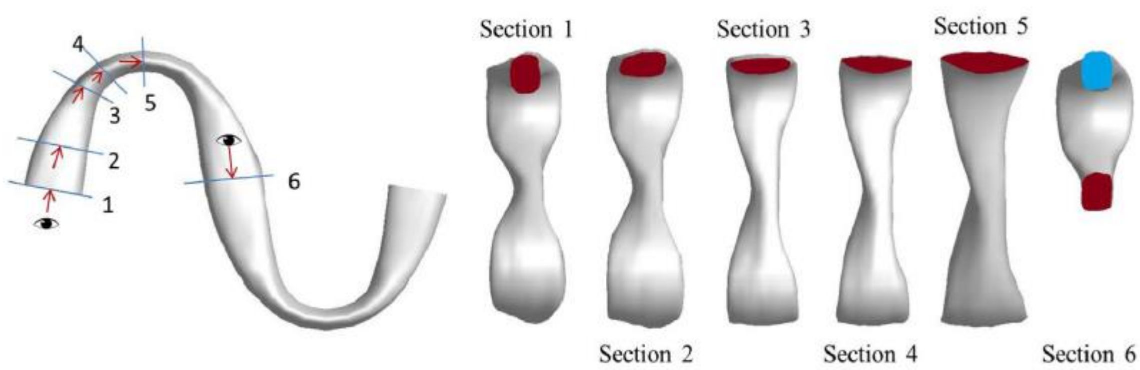


Figure 1.6: Illustration of yarn cross-section variation for a realistic (non-idealized) yarn representation

There are major research shortcomings in the area of creating a finite element model for analysis of realistic textile composite micro-geometries. This research is novel and unique in the field of textile composite mechanics as it addresses and solves the following major problems:

1. Existing textile composite modeling software either relies on micro-CT scans or empirical models to define the micro-geometry of the reinforcing yarns. With the development of DEA incorporated into DFCA, predictive modeling of complex realistic fabric micro-geometries was made possible at the fiber level. There are no idealized assumptions made for the yarn path or cross-section shapes as the digital fibers can rearrange themselves within the yarn under a combination of tension, inter-fiber compression and damping forces to give rise to yarns with varying cross-section shapes and paths as illustrated in Figure 1.6.
2. Generating a realistic micro-geometry is just the first step of the analysis process – generating a good quality conformal finite element representation of the textile composite micro-geometry and analysis of the composite unit cell behavior under various loading conditions are natural next steps. However most of the methods that exist thus far to achieve the same have several shortcomings:
    a. The voxel-based mesh methods as discussed before are generally used to solve the problem, but again these are prone to give rise to stress risers at the interface due to the zig-zag pattern of the mesh and incorrectly predict the strength and non-linear behavior.
    b. In other conformal meshing cases the meshing operation is performed on either the reinforcing yarns or the surrounding matrix alone, or the entities are meshed separately, requiring an additional manual assembly procedure. The

assembly procedure however does not ensure conformity at the material boundaries – there are still nodal mismatches and displacement compatibility across element interfaces is not ensured. Thus, this requires interface elements or “Cohesive Zone” elements which require additional set of input parameters to define their stiffness and strength, which makes the model more complex.

c. Most methods generate the mesh using tetrahedral (Tet4) elements only which has further disadvantages. Using just a tetrahedral mesh vs. using a combined Hex8/Tet4 Tetrahedral element mesh increases the size of the model drastically. Further Hex8 elements have better convergence behavior and solution speeds.

A conformal meshing procedure was developed in this research which has the following major advantages:

a. The entire mesh generation process is automated and is based on realistic fabric unit cells from DFCA. This also bridges the gap between DEA and FEA, all incorporated into a single textile composite modeling code.

b. The major drawbacks are addressed and eliminated –

    - Eliminate yarn interpenetrations and narrow gaps,
    - Generate a conformal mesh with no nodal mismatch and displacement compatibility between elements,
    - Generate the unit cell mesh consisting of yarns and the surrounding matrix without the need for manual assembly,
    - Use a Hex8 dominant mesh and split to lower order elements only when necessary

It also opens a new door making failure and strength analysis based on realistic micro-geometries possible without having to go for CT-scan images, which could thus result in significant cost reduction.

In this research, first an automated conformal volume mesh generation strategy for realistic fabric micro-geometries is presented that matches the yarn-yarn and yarn-matrix interfaces. The mesh generation process is however very challenging due to the complex yarn shapes and local variations in cross-sections that are crucial to the composite unit cell property prediction but are very difficult to capture accurately After a lot of improvements over the last 5 years, a procedure was obtained that could capture these complexities in a simplified sequence of steps.

The unit cell micro-geometry is obtained using the fiber-level micromechanics model, DEA. The micro-geometry generation procedure and the yarn-level micro-geometry description is discussed briefly in Chapter 2 - Unlike most textile composite simulations, no assumption of idealized cross-sections is made: DFCA uses a multi-step and multi-level dynamic relaxation procedure to generate a fiber-level micro-geometry; hence, the yarn-level micro-geometry derived from the prior is significantly more accurate and realistic than other approaches. The surface mesh data at the mesoscale is used to generate the conformal unit cell mesh. The meshing process, which is discussed in detail in Chapter 3 - , consists of nine major steps:

1. Discretize the periodic unit cell domain into a structured uniform cuboid FE mesh, consisting of 8-node hex elements with 24 degrees of freedom
2. Identify the intersecting points between yarn surface elements and mesh grid-lines in the x-, y- and z-directions

3. Correct yarn surface interpenetrations and narrow artificial gaps due to numerical imperfections
4. Modify the initial mesh created in step 1 in order to match yarn boundaries to the finite element mesh
5. Split elements at the interface between two or more materials. A rule is established for the splitting process to guarantee shape function compatibility
6. Ensure periodicity of the unit cell mesh domain boundaries
7. Perform a mesh smoothing operation to increase mesh accuracy
8. Check element validity and quality. Improve element quality by further node shifting or face splitting if required
9. Assign local element orientations based on yarn curvature

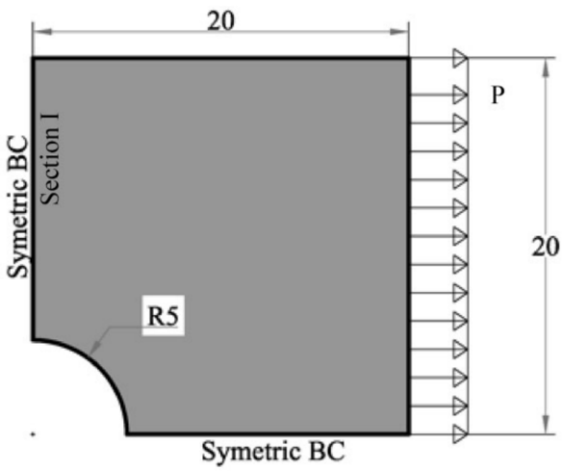
The woven composite unit cell model creation, mesh generation, material assignment and periodic boundary conditions have all been implemented using DFCA [27]. The unit cell finite element model generation is completely automatic.

Next, the unit cell meshed model is used to conduct a progressive damage modeling study. Although elastic or linear response of woven composites is very well established in theory, modeling their non-linear damage response remains a significant challenge due to the complex three-dimensional micro-geometries of woven composites. Additionally, the different failure mechanisms of fiber-reinforced composites such as fiber (yarn)-matrix debonding, fiber rupture, matrix cracking, matrix crushing and fiber pull-out make it more complex to model their non-linear behavior. These failure mechanisms make the load-deflection curve non-linear. Over the years, several damage modeling techniques have been proposed in literature for studying the non-linear failure behavior of woven composites. The different techniques range from the simple ply-discount methods to the more sophisticated strategies based on Continuum Damage Mechanics (CDM) models that derive from fracture mechanics theories.

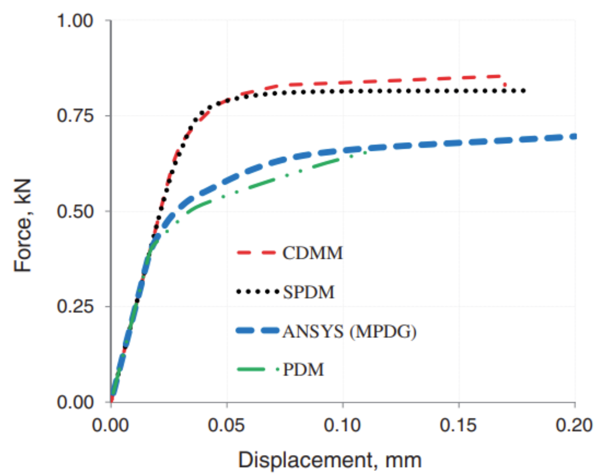
Several stiffness reduction based damage modeling techniques have been proposed in literature to model the damage evolution in woven composites. Blacketter *et al*. [28] studied the failure behavior of plain woven composites. In their implementation, damage initiation was governed by the maximum stress criterion and damage evolution was modeled using a ply-discount (instant stiffness reduction) technique, considering different failure modes of the composite yarns. Li *et al*. [29] studied the response of woven composites where they used the stiffness reduction method for modeling the damage evolution in the yarns and the matrix. Maximum stress criterion governed damage initiation for an iso-strain state, while that for an iso-stress state was governed by the maximum strain criterion. ANSYS offers an instant stiffness reduction damage evolution law called Material Property Degradation (MPDG) that is based on the ply-discount approach. MPDG has been used by Ferreira *et al*. [30], to study the non-linear response of non-crimp fabric composites under compressive load using the maximum strain criterion, and by Mazumder *et al*. [31] to study the response of woven composites under tensile load using the maximum stress criterion. The stress-strain response of a 2 by 2 twill weave composite was studied by Scida *et al*. [32] using a stiffness reduction technique. Tsai-Wu criterion governed damage initiation in the yarns, while that in the matrix was governed by the von Mises criterion. A modified stiffness reduction technique was also used by Yen [33] to study ballistic impact response in woven composites. Damage initiation was modeled using a generalized Hashin criterion. A scale value (S, greater than 1) was used to account for the stress concentration factor at the interface in order to model the yarn-matrix debonding. Engelstad *et al*. [34] studied the post-buckling response of graphite epoxy panels in axial compression using an instant stiffness reduction damage progression model corresponding to the maximum stress and Tsai-Wu failure criteria. Knight *et al*. [35] modeled the non-linear behavior of laminated composites using an incremental stiffness reduction technique for different failure initiation criteria. Moas *et al*. [36] used the maximum strain and Tsai-Wu criteria to model damage initiation. Damage progression was modeled using an instant stiffness reduction approach and a statistical approach where only the stiffness loss associated with fiber tensile failure was proportional to the probability of fiber failure and the other failure modes assumed a ply-discount approach. Reddy *et al*. [37] also used the stiffness reduction approach to study the non-linear behavior of composite laminates under uni-axial tension.

Other researchers have used the CDM based damage modeling technique. CDM models represent the material as a continuum having smooth, continuous field equations. It was described as "as a branch of continuum solid mechanics characterized by the introduction of a special (internal) field variable representing, in an appropriate (statistical) sense, the distribution of microcracks locally" by Krajcinovic [38]. Talreja [39] and Chang *et al*. [40] were the earliest to propose such a model for laminated composites. Barbero *et al*. [41] proposed a meso-scale CDM model coupled with thermodynamics for studying the response of 2D plain woven composites. The damage model was applied only to the yarns and the matrix was considered elastic. Although the model showed good agreement in the initial non-linear phase upto 0.84% strain, the response of the woven composite beyond that upto failure (at approximately 1.5-1.6% strain) was not reported. Wang *et al*. [42] developed an anisotropic damage model for 2D woven composites that was based on the Murakami-Ohno tensor, where the material fracture energy governed damage evolution. A CDM model with cohesive zone elements was used by Bergan [25] to model damage inside yarns and on the yarn-matrix interface. The use of contact elements did not require the model to be conformal at the interface between the yarn and matrix elements. Matrix damage was accounted for by plasticity. A micro-mechanical damage model to simulate response of woven composites under tension was proposed by Saleh *et al*. [43] where material parameters governed the damage evolution. A 3D mosaic concept was developed by Bogdanovich [44] where the woven composite structure was represented as an assembly of arbitrary number of distinct homogeneous anisotropic material blocks. In his implementation, damage initiation was governed by the maximum strain criterion and a CDM based damage evolution model with nine different failure modes associated with each Mosaic block at the meso-scale was used for modeling the progressive damage response. Two different damage models were proposed by Bednarcyk *et al*. [45] with different computational efficiency and fidelity levels to study the progressive damage behavior of 3D woven composites under various loading conditions. Matzenmiller *et al*. [46] proposed a model (called the MLT model) based on a Weibull function [47] to describe the statistical nature of internal defects and the ultimate strength of a fiber bundle within a composite lamina. In the MLT model, the Weibull factor controlled the degree of strain-softening in the damage phase. Non-linearity in the pre-damage phase was controlled by the ultimate strength values. Although several damage modeling techniques for woven composites have been proposed, at present no single method has been shown to be superior to predict non-linear response for a broad range of woven architectures.

El-Sisi *et al*. [48] present a comparison of the different 3D progressive damage modeling approaches by studying the behavior of a composite plate containing the source of a stress concentration as shown in Figure **1.7**(a). It was observed that the instant stiffness reduction damage models (Ply Discount Model or PDM, and MPDG) are the most sensitive to mesh size, whereas the recursive stiffness reduction model (Simple Progressive Damage Model or SPDM) and Continuum Damage Mechanics Model (CDMM) showed the least dependence on mesh size. The failure strength is also under-predicted in the instant stiffness reduction models because element failure is sudden and no progression of failure inside individual elements is considered. Figure 1.7(b) shows the comparison of force-displacement curves for the different models. In the case of SPDM and CDM, the element failure is a much more gradual process due to which the effect of high stress concentration in elements, causing catastrophic failure, is minimized. The material fracture energy inputs required to define the CDM model are however difficult to derive. Hence, a recursive stiffness reduction model for progressive damage analysis in fiber-reinforced composites shows good promise.



(a) Model: plate with a hole



(b) Force-displacement relation comparison

Figure 1.7: Comparison of different damage models [48]

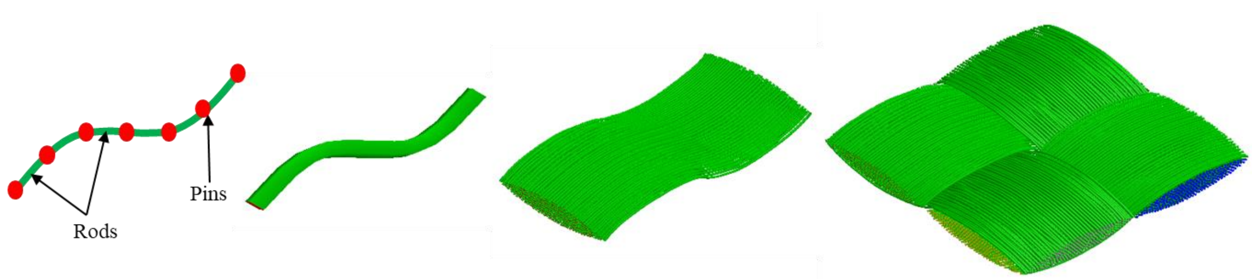
The existence of significant research on the damage modeling of unidirectional composite laminates prompted a mesoscale idealization that could be used to study the woven fabric micro-geometries. In this approach, the yarn elements are modeled as unidirectional fiber reinforced composite laminates, in their local coordinate systems. Thus well-established failure theories for unidirectional laminates can be used with the current woven composite finite element mesh. A recursive stiffness reduction method based on a combination of nine different failure modes is implemented to study damage progression of the as-modeled transversely isotropic yarns. The maximum strain criterion governs the initiation of failure. The isotropic matrix failure is governed by a ply-discount approach. The damage model is implemented using a user-defined subroutine. The progressive damage response of 2D and 3D woven composites are then investigated using the proposed recursive stiffness reduction damage model. The model is validated against experimental data and a comparison of the results is presented to support the model. The progressive failure process is monitored step-by-step and failure mechanisms are studied in detail. The procedure is presented in Chapter 4 -

Finally, a comparative study is presented in Chapter 5 - , where different 3D woven fabric unit cell architectures with the same constituent materials and fiber volume fraction are studied. Their stiffness and strength response in the warp and weft directions are compared. The observations and failure mechanisms are discussed in detail. It is expected that this study will help inform design decisions for the choice of a particular woven composite architecture for different applications in the future.

Chapter 2 - Fabric unit cell micro-geometry generation

This chapter briefly discusses the digital element approach (DEA) concepts, summarizes the unit cell micro-geometry generation process and presents some modifications for transverse contact calculation and unit cell periodic boundary conditions. For a more detailed discussion on micro-geometry generation, the reader is directed to references [6, 7, 49]. Apart from its major application in textile mechanics, DEA has also been used to generate random polymer morphologies [77].

The concept of digital element was introduced by Wang *et al*. [50, 51]. Fabric micro-geometries at the fiber-level were nearly impossible to model numerically due to the multitude of degrees of freedom of the model, which made it computationally very expensive. The digital element approach (DEA) overcame this bottleneck by using a “digital fiber” which represents multiple actual fibers. The idea is that a fabric micro-geometry is comprised of yarns woven in a pattern, where the yarns can be discretized as a bundle of digital fibers. A digital fiber in turn consists of a series of interconnected rod elements joined by frictionless pins. This is shown in Figure 2.1.



(a) Digital fiber description (b) Single digital fiber (c) Single yarn made of multiple digital fibers (d) Plain woven unit cell

Figure 2.1: Multi-scale representation of unit cell micro-geometry

The frictionless pins each represent a “node”. The user-defined weaving pattern in DFCA generates the initial fabric unit cell structure, which needs to be relaxed in order to achieve a stable state of the unit cell. The relaxation is achieved by calculating and minimizing the total energy due to the nodal forces. The three nodal forces considered are fiber tension, inter-fiber contact and damping. The dynamic relaxation process transfers the potential energy into kinetic energy, which is subsequently dissipated by the damping forces. The nodal coordinates, forces and velocities are calculated and updated using a central difference algorithm. The process continues until a minimum total energy state of the unit cell is reached, at which point the simulation ends.

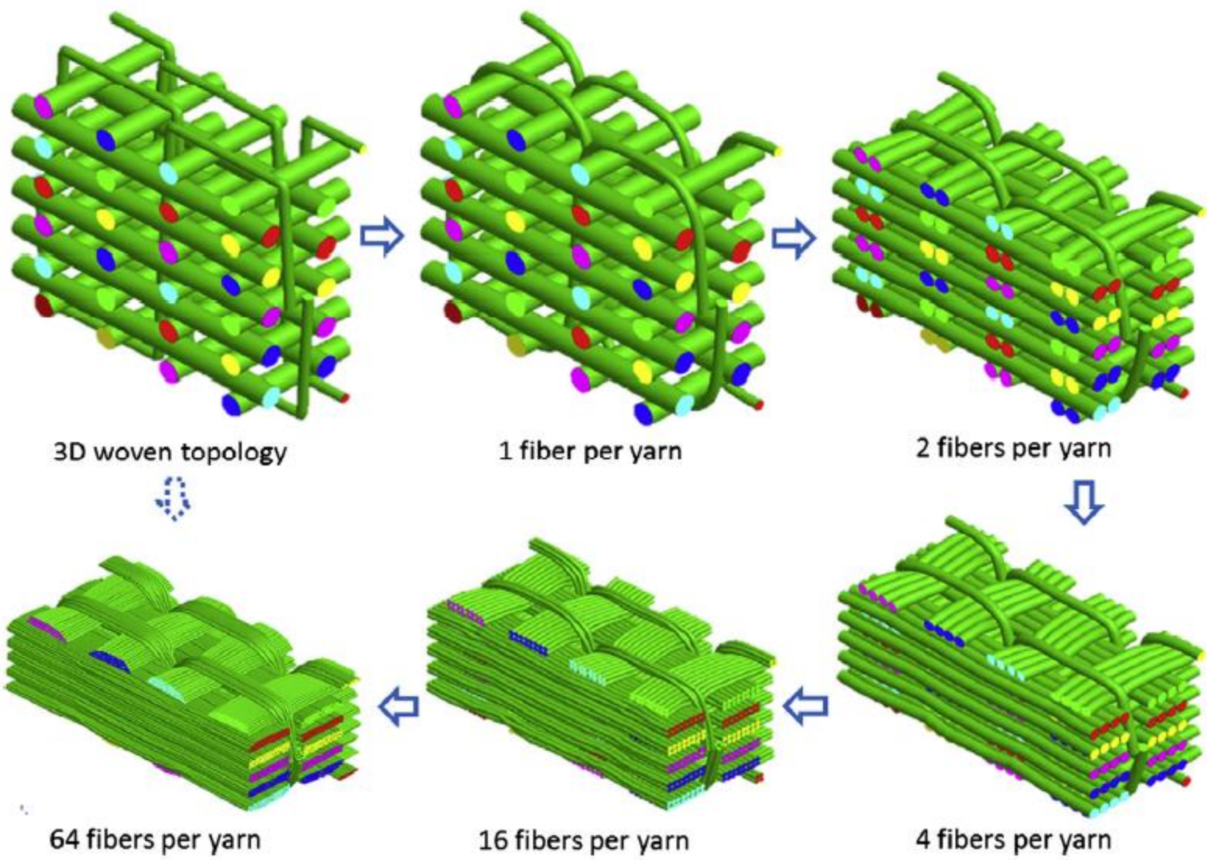


Figure 2.2: Multi-level simulation in DFCA [7]

The dynamic relaxation process in DFCA adopts a multi-level simulation approach, as shown in Figure 2.2, where the first step of the relaxation process is performed with each yarn consisting of 1 digital fiber. Subsequent relaxation steps are performed where the yarns are split into more and more digital fibers. It has been found that 19-60 digital fibers are sufficient to accurately represent the fabric unit cell micro-geometry. The yarn area is determined by the total fiber cross-section area and this remains the same in the whole process. Significant computation time is saved by using a multi-level relaxation process compared to using a single level relaxation with a fine element mesh. The following subsections discuss some minor modifications made to the micro-geometry generation process.

2.1 Modified contact calculation

The contact forces are calculated when two nodes come within a potential contact zone. As shown in Figure 2.3(a), the nodes of the two digital fibers will be in contact when the contact zone radii of the two nodes, shown by the dotted circles overlap. In other words, contact between two nodes belonging to different fibers occur when the distance between them, which is the contact distance denoted by L_C, is less than the sum of the contact zone radius (equation (2.1)).

$$\begin{gathered} L_C < (R1' + R2') \\ R1' + R2' = 1.1(R1 + R2) = D \end{gathered} \tag{2.1}$$

The contact zone radius is taken to be 10% greater than the actual radius, to include nodes that may be in contact in the subsequent simulation steps. The area between the solid circle and the dotted circle is known as the potential contact zone. The contact force is calculated using a modified Hertz contact theory given by equation (2.2), where E_T is the transverse contact stiffness.

$$F_C = \frac{2\sqrt{2}E_T l_e (D\Delta)^{1.5}}{3L_C D} \tag{2.2}$$

The factor Δ used in the above equation is a variable based on the contact distance and is given as:

$$\begin{aligned} \Delta &= 0.05D, \quad L_C > 0.95D \\ &= D - L_C, \quad L_C \leq 0.95D \end{aligned} \tag{2.3}$$

The mathematical representation of the response is shown in Figure 2.3(b). The indentation δ is the difference between the contact zone diameter (D) and the actual contact distance (L_C). The contact force curve has a linear "cushion" upto 5% indentation, and beyond that, the force increases at an increasing rate.

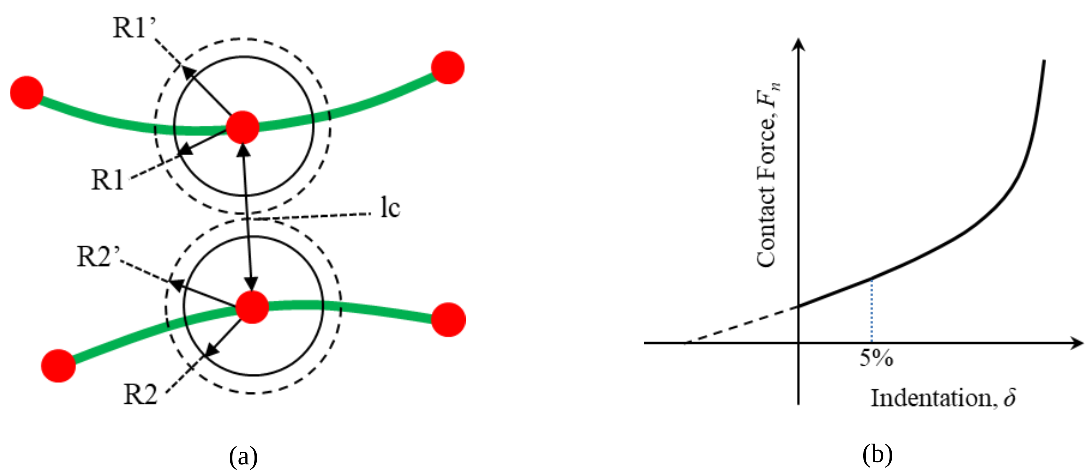


Figure 2.3: Illustration of contact force calculation

2.2 Unit cell periodic boundary condition

The unit cell is the smallest repeating unit in a fabric assembly that is capable of representing the bulk behavior of the woven composite. Thus the unit cell boundaries must be periodic, which means that the nodal coordinates of the start and end nodes of a digital fiber

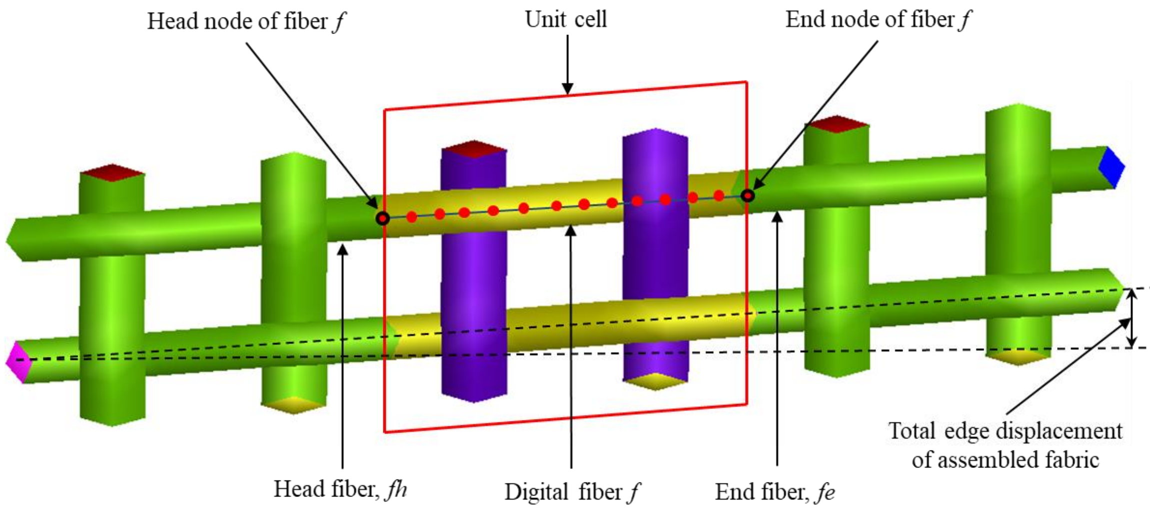


Figure 2.4: Illustration of periodicity conditions for a simple sheared unit cell

within a yarn must match a corresponding fiber within a unit cell. This idea which is implemented in DFCA is illustrated by a simple example as shown in Figure 2.4. The plain woven sheared unit cell shown has just one fiber-per-yarn for the ease of demonstration. Each fiber within the unit cell has a corresponding head fiber and an end fiber. The head and end fibers ensure periodicity of the unit cell and are used to create the assembled fabric. Similarly, the edge displacement, as shown in Figure 2.4, for a sheared unit cell needs to be incorporated into the periodicity calculations. The edge displacement can be applied either to the weft or the warp yarns. Denoting the three major axes for a fabric micro-geometry as being parallel to the warp (x-direction), weft (y-direction) and through-the thickness (z-direction) directions, equation (2.4) gives the generalized periodic relationship between the nodal coordinates of the head and end nodes of a fiber.

$$\begin{Bmatrix} x_{f_{end}} \\ y_{f_{end}} \\ z_{f_{end}} \end{Bmatrix} = \begin{Bmatrix} x_{fe_{head}} \\ y_{fe_{head}} \\ z_{fe_{head}} \end{Bmatrix} = \begin{Bmatrix} x_{f_{head}} \\ y_{f_{head}} \\ z_{f_{head}} \end{Bmatrix} + \begin{Bmatrix} d^{weft} \\ l^{weft} \\ 0 \end{Bmatrix} = \begin{Bmatrix} x_{fh_{end}} \\ y_{fh_{end}} \\ z_{fh_{end}} \end{Bmatrix} + \begin{Bmatrix} d^{weft} \\ l^{weft} \\ 0 \end{Bmatrix}, weft\ direction$$

$$\begin{Bmatrix} x_{f_{end}} \\ y_{f_{end}} \\ z_{f_{end}} \end{Bmatrix} = \begin{Bmatrix} x_{fe_{head}} \\ y_{fe_{head}} \\ z_{fe_{head}} \end{Bmatrix} = \begin{Bmatrix} x_{f_{head}} \\ y_{f_{head}} \\ z_{f_{head}} \end{Bmatrix} + \begin{Bmatrix} l^{warp} \\ d^{warp} \\ 0 \end{Bmatrix} = \begin{Bmatrix} x_{fh_{end}} \\ y_{fh_{end}} \\ z_{fh_{end}} \end{Bmatrix} + \begin{Bmatrix} l^{warp} \\ d^{warp} \\ 0 \end{Bmatrix}, warp\ direction \tag{2.4}$$

In the above equation, f refers to the digital fiber within the unit cell with fh and fe being the corresponding head and end fibers. f_{head} and f_{end} are the head and end nodes of digital fiber f respectively. l^{warp} and $l^{weft}$are the dimensions of the unit cell in the warp and weft directions, whereas d^{warp} and d^{weft} are the respective edge displacements. A similar procedure was implemented for incorporating periodicity in the yarn-level micro-geometry, with the end and end nodes replaced by the vertices of the head and end cross-sectional profiles along the yarn path.

2.3 Yarn surface mesh modification

The mesoscale yarn-level micro-geometry of the unit cell is available to export from DFCA as a STL (Standard Tessellated Language) file, which contains the coordinates of points

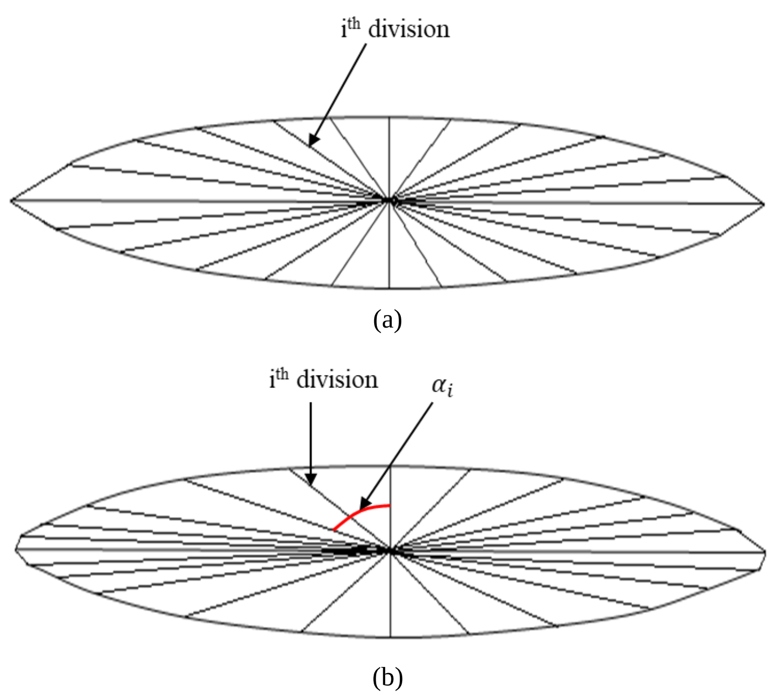


Figure 2.5: Illustration of yarn geometry preprocessing

defining each yarn surface organized in sets of cross-sectional profiles with a certain number of points per profile. A subroutine to transform this STL data file into a triangular surface mesh in terms of nodal positions and element connectivity was implemented. Each element of the surface mesh is a three-node triangle with nine degrees of freedom.

The cross-sectional profiles are evenly distributed along the yarn path. Initially, the profile boundary was divided into iso-length segments as shown in Figure 2.5(a). However, sudden changes in cross-section shapes such as areas around the sharp yarn curvatures were not captured smoothly as a consequence. Although the problem can be solved by refining the mesh, but that increases the size of the model drastically. Thus, the angles between the cross-section divisions were adjusted accordingly. The angles were redistributed in each quadrant according to a geometric progression series for each yarn profile, based on the suddenness of the cross-section change, as illustrated by equation (2.5). The result after adjustment is shown in Figure 2.5(b).

$$
\begin{aligned}
\alpha_i &= \frac{\pi}{2} - \theta^{0.25k-i}, \qquad quadrant\ I \\
\alpha_i &= \theta^{i-0.25k} + \frac{\pi}{2}, \qquad quadrant\ II \\
\alpha_i &= \frac{3\pi}{2} - \theta^{3k-i}, \qquad quadrant\ III \\
\alpha_i &= \theta^{i-3k} + \frac{3\pi}{2}, \qquad quadrant\ IV \\
\theta &= \frac{\pi^{4/k}}{2}
\end{aligned}
\tag{2.5}
$$

In the equation above, α is the adjusted angle as shown in Figure 2.5(b), i is the division number, k is the total number of divisions per yarn cross-section profile and θ is the geometric progression series angle ratio.

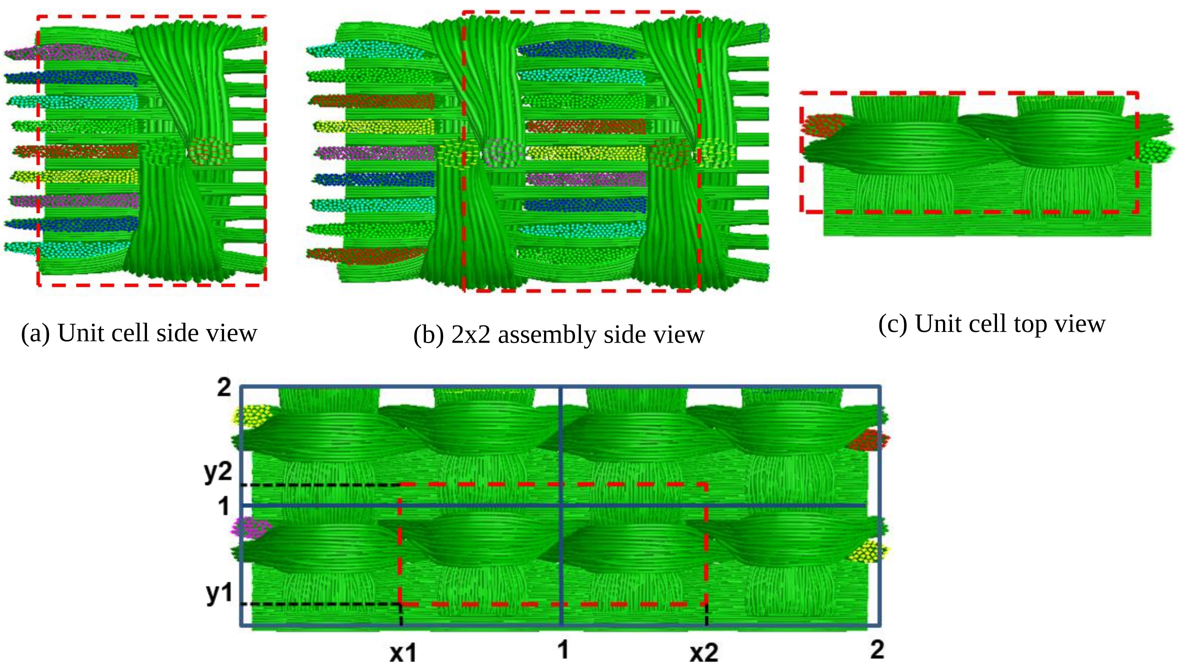
The yarn-level unit cell thus generated can now be used for generating the conformal volume mesh.

Chapter 3 - Conformal finite element mesh generation of woven composite unit cells

This chapter is focused on the finite element modeling of the unit cells from realistic micro-geometries generated using DFCA [52, 53]. As previously mentioned, no idealized assumptions are made for the yarn shapes during the meshing process. The following sub-sections discuss each part of the meshing procedure in greater detail.

3.1 Mesh domain definition

The unit cell generated in DFCA is periodic. However, during the micro-geometry generation, some yarns might move in and out of the initial bounding box or unit cell domain due



(a) Unit cell side view

(b) 2x2 assembly side view

(c) Unit cell top view

(d) 2x2 assembly top view

Figure 3.1: Mesh domain definition illustration

to the dynamic relaxation procedure. This is shown in Figure 3.1(a). In order to make sure that there are no artificial voids inside the unit cell domain, the initial domain is defined at a user defined location inside the unit cell assembly (2x2 or 3x3), as shown in Figure 3.1(d). The input parameters x1 and y1 control location of the unit cell domain definition. Once the domain is defined, it is discretized into a grid of uniformly-sized 8-node hexahedral elements, with 24 degrees of freedom, as illustrated in Figure 3.2. This element definition is equivalent to the SOLID185 element in ANSYS or the C3D8 element in ABAQUS^{TM}. The initial mesh domain is defined in terms of the number of divisions in the x-, y- and z-directions, which are taken as input parameters from the user.

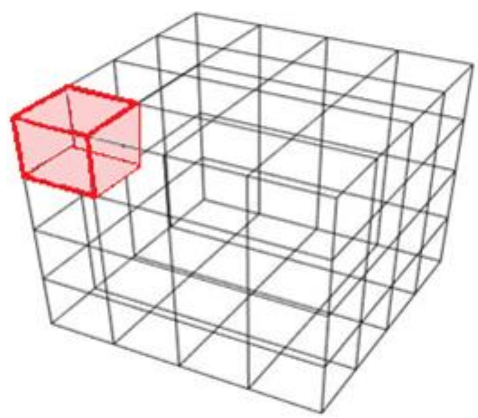
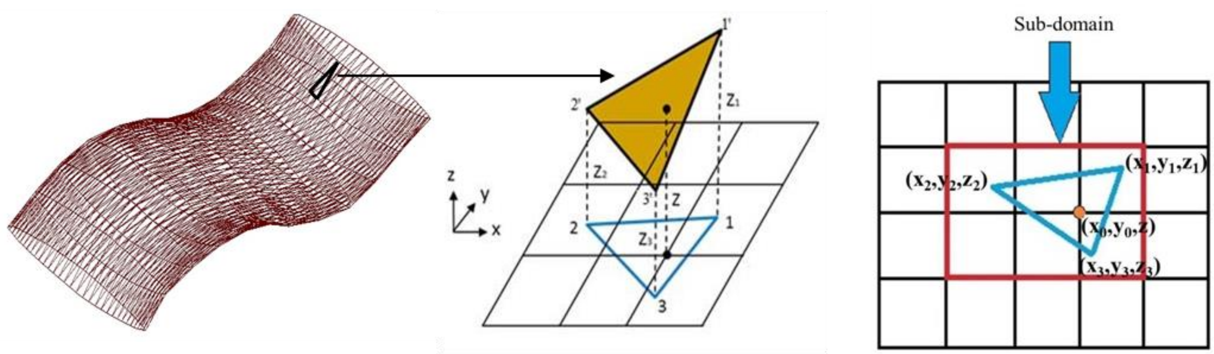


Figure 3.2: Domain discretization

There are two broad mesh refinement categories for a finite element domain: the h-refinement, which improves accuracy by increasing the number of elements in the domain and the p-refinement which uses higher order elements to improve accuracy. There also exists a mesh adaptation strategy termed as the hp-refinement which is a combination of the two. At present, only the h-refinement option is incorporated, however in the future, the p-refinement or even the hp-refinement could be adopted.

3.2 Determining intersection points

The yarn surface mesh is used to determine the intersection points. The intersection search starts with the mesh grid lines in the z-direction, followed by the x- and y-directions. Since there are a multitude of surface elements in the unit cell assembly, searching through all of them to determine intersection is computationally very expensive. To tackle this, an efficient search algorithm is developed, as illustrated in Figure 3.3.



(a) Yarn surface mesh

(b) Projection of surface mesh element on the xy-mesh grid plane

(c) Projection top view and sub-domain definition

Figure 3.3: Intersection search in the z-direction

In the search procedure, a triangular surface element is first checked if it lies within the unit cell domain bounds. For each element inside the domain, it is projected onto the xy-plane of the mesh grid when searching for intersection in the z-direction, as shown in Figure 3.3(b). A subdomain is then defined based on the maximum and minimum x- and y-coordinates of the 3 vertices of the corresponding surface element as shown in Figure 3.3(c). The z-grid lines that lie inside this subdomain are checked for potential intersection with the surface element. In case a z-grid line intersects a surface element, the intersection point position in the xy-plane given by (x_0, y_0) are determined and it must lie inside the surface element projected area. The z-coordinate of this intersection point is then interpolated using shape functions as shown in equation (3.1).

$$\begin{Bmatrix} N_1 \\ N_2 \\ N_3 \end{Bmatrix} = \begin{Bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{Bmatrix}^{-1} \begin{Bmatrix} x_0 \\ y_0 \\ 1 \end{Bmatrix}$$
$$z = N_1 z_1 + N_2 z_2 + N_3 z_3 \tag{3.1}$$

In the above equation, N_is are the shape functions ($0 \leq N_i \leq 1$) and (x_i, y_i, z_i) are the

Algorithm 3.1: Intersection point search algorithm (Z-direction)

```
Get total yarns as Y
Get total surface elements per yarn as SurfaceElements[yarn]
Define mesh domain as D
Define sub-domain bounding box as B
Define dynamic array of z-grid lines as zLine[]
for each yarn do {Yarns loop}
    for each surface element per yarn do {Yarn surface elements loop}
        Read X, Y, Z coordinates of the vertices of each element
        Store minimum and maximum X coordinates in minx and maxX respectively
        Store minimum and maximum Y coordinates in minY and maxY respectively
        Get bounding columns c1 and c2 from minX and maxX respectively
        Get bounding layers l1 and l2 from minY and maxY respectively
        Define B by c1, c2, l1, l2
        if B is inside D then
            for each Z-grid line inside B do {Sub-domain grid line loop}
                Calculate shape functions N1, N2, N3
                if N1, N2, N3 >= 0 then
                    Grid-line intersects surface element
                    Calculate intersection point as interpolated Z-coordinate
                    Store yarn ID and intersection points in zLine[]
                end if
            end for
        end if
    end for
end for
for each Z-grid line do {Domain grid line loop}
    Sort intersection points from top to bottom
    Keep only two intersection points per yarn per grid line, delete additional points
    Update dynamic array zLine[]
end for
```

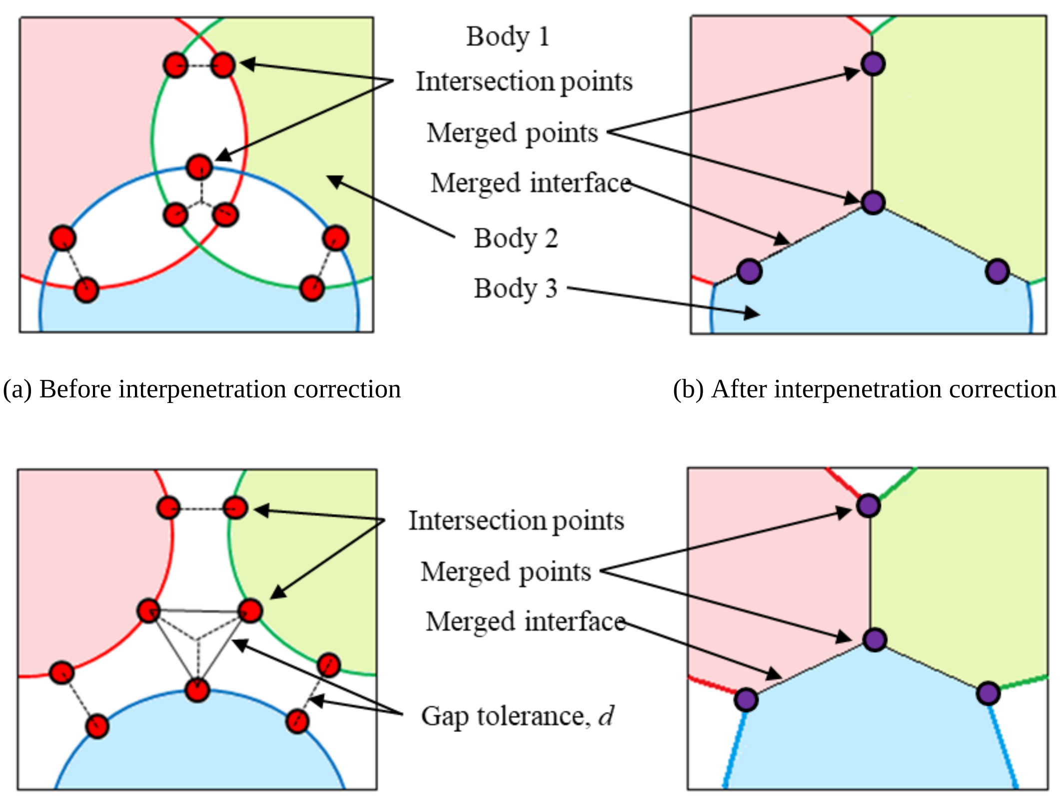
vertex coordinates of the surface element for $i = 1,2,3$. (x_0, y_0) is the location of the z-gridline in consideration on the projected plane. In case a z-gridline intersects a surface element, all the shape functions must be greater than or equal to zero. The above equation also shows the calculation of the interpolated z-coordinate using the shape functions. The implementation of the intersection search algorithm is given in Algorithm 3.1.

A similar procedure is implemented to determine the intersection points for the x- and y-mesh gridlines with the surface elements. For determining the intersection points in the x-direction, the sub-domain is defined in the y-z plane, while for the y-direction, the sub-domain is defined in the x-z plane. The order of determining the intersection points currently follows the z-x-y convention, because the unit cell is modelled with the z-axis representing the through-the-thickness direction. The intersection points are stored in computer memory for further preprocessing.

3.3 Removing numerical imperfections in the micro-geometry

The yarn surface mesh generated using DFCA has numerical imperfections like element interpenetrations and narrow gaps. While the element interpenetrations make it difficult to import the surface mesh model into commercial FEA software, narrow gaps lead to tiny slivers of elements, usually the surrounding resin or matrix, in between the yarns, with high aspect ratios and thus poor element quality. The high aspect ratio elements also lead to artificial stress concentrations within the unit cell, causing premature failure under loading. These imperfections can occur in any random direction within the unit cell. In order to remove these imperfections, the closest intersection points are merged into a single shared interface point. Figure 3.4 shows a simplified illustration of this procedure.

Yarn interpenetration, or "interference", is detected if all or part of a surface element of one yarn lies within the boundary of another yarn. Figure **3.4**(a) shows a simplified illustration of the occurrence of element interpenetration and the procedure to remove it is shown in Figure **3.4**(b). The three bodies interpenetrating each other are shown in red, green and blue. The interpenetration zone is shown in white. The intersection points are shown by the red dots. The dotted black lines show the lines along which the intersection points will be merged into a common interface. The purple dots in Figure 3.4(b) show the merged points and the solid black lines show the common interface after merge, with the interpenetrations eliminated. As shown, multiple surface elements might interfere at a given location.



(a) Before interpenetration correction

(b) After interpenetration correction

(c) Before gap correction

(d) After gap correction

Figure 3.4: Simplified illustration of interpenetration and gap correction

In a similar way, narrow gaps between yarns are corrected by merging the closest intersection points together. If the distance between two intersection points is less than the gap tolerance, d, the points are merged onto a shared interface as shown in Figure 3.4(c) and Figure 3.4(d). The gap tolerance is taken as 10% of the minimum element edge length. The interferences and gaps are corrected according to equation (3.2).

$$X_i = \frac{\sum x_i}{N} \tag{3.2}$$

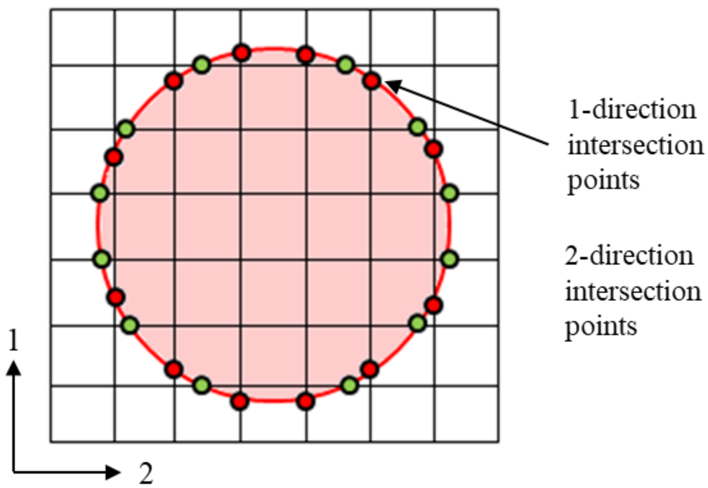
In the equation above, N is the number of intersection points that are merged, and the coordinates of the intersection points are given by $x_i = x, y, z$, where i denotes the directions parallel to the Cartesian axes. X_i are the coordinates (x, y, z) of the merged intersection points.

The intersection points data structure is then updated – the points that were merged into a single intersection point are deleted and the new merged points are added. The updated points are stored in memory for re-meshing the initial uniform mesh grid.

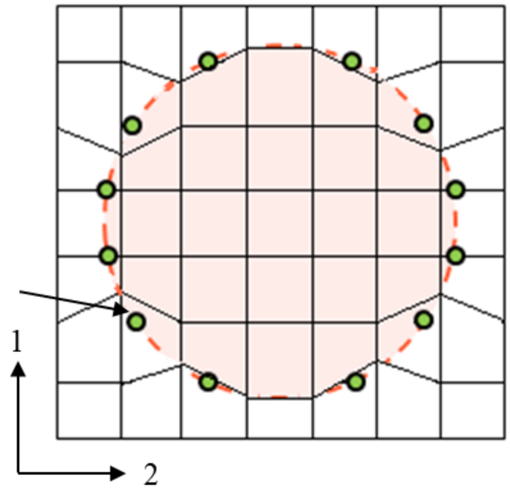
3.4 Modifying the mesh grid or "re-meshing"

The intersection points thus determined need to be defined as mesh nodes. A node is a finite element entity where degrees of freedom are assigned. Intersection points are geometric entities which just define their spatial locations. Thus the initial mesh grid connectivity needs to be modified to move nodes to these intersection points so as to match the yarn-yarn and yarn-matrix boundaries. This is the first step in converting a voxelized mesh domain into a conformal one.

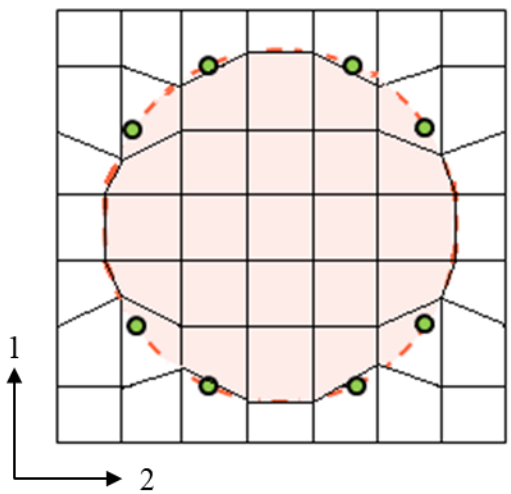
The re-meshing procedure is performed in the z-, x- and y- directions respectively in that order. The process is discussed using a simplified 2D illustration as shown in Figure 3.5. In this case only 2 major directions are considered – 1 and 2. The intersection points that define the



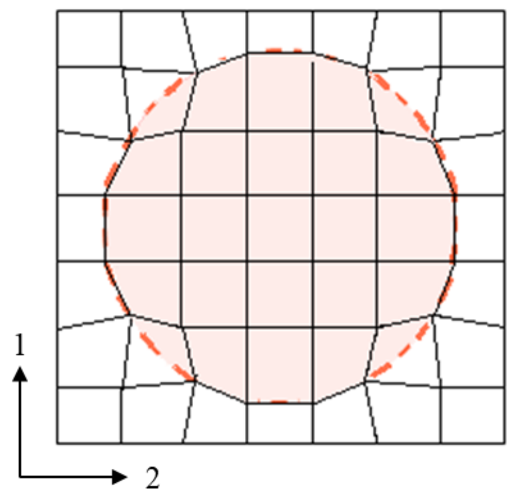
(a) All intersection points



(b) Re-meshing in direction 1



(c) Re-meshing in direction 2

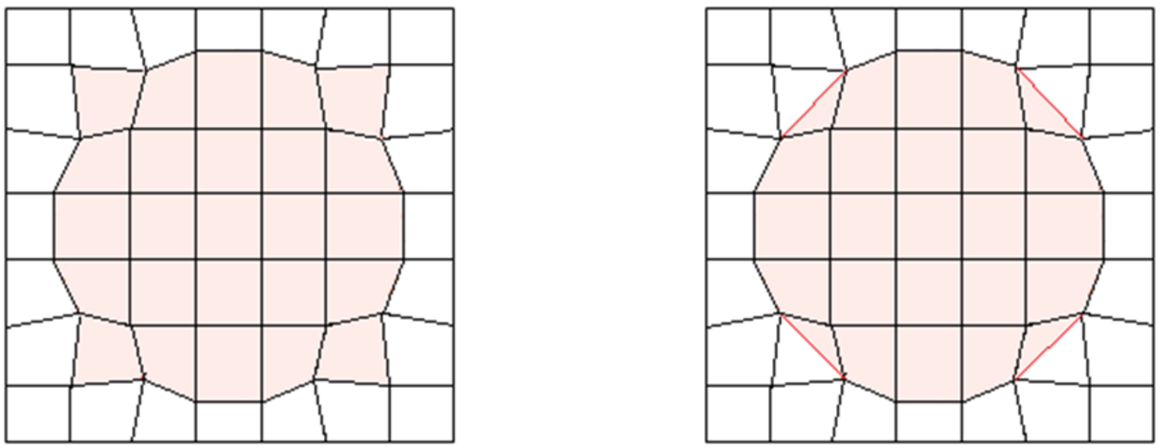


(d) Merge remaining points

Figure 3.5: Schematic of the re-meshing procedure

boundary of the geometry are shown in Figure **3.5**(a). The red dots denote the intersection points along the 1-direction grid lines and the green dots denote the intersection points along the 2-direction gridlines. If an intersection point lies between two adjacent mesh grid nodes along the same gridline, the node closer to the intersection point is moved to that point. This is illustrated by Figure **3.5**(b), for the 1-direction and Figure **3.5**(c) for the 2-direction. For the remaining intersection points, as shown in Figure **3.5**(c), the closest nodes have already moved previously.

Hence for these points, the closest nodes are moved to the mid-point between these nodes and the corresponding intersection point, as shown in Figure 3.5(d). This completes the re-meshing procedure, but the mesh is still not conformal. For this 2D case, a conformal mesh would look like the one shown in Figure 3.6(b), where the lines marked in red show the split in the elements to match the geometry boundary clearly. The next section in this chapter discusses the element splitting process in greater detail.



(a) Non-conformal geometry representation

(b) Conformal geometry representation

Figure 3.6: Simplified illustration of the next step

3.5 Element splitting at the interface

The initial 8-node hex element can consist of more than one material after the re-meshing process, based on the number of yarn geometries the element shares. The yarn-yarn or yarn-matrix interface divides the material among the element volume. Thus to ensure a conformal mesh, the element must be split at the interface. This is done in 3 steps:

1. Assign nodal material types
2. Split element faces
3. Split elements and assign element material type

3.5.1 Assigning nodal material types

The nodal material type assignment procedure is illustrated in Figure 3.7. The points marked by the red dots, namely, $u_1$and l_1 are the upper and lower intersection points of yarn 1 and $u_2$and l_2 are the upper and lower intersection points of yarn 2 along the z-grid line as shown. The green dot denotes the merged interface point after interference correction between yarns 1 and 2. It subsequently becomes the upper intersection point for yarn 1 and lower intersection point for yarn 2. The shared interface is shown by the dotted red line. Nodes i and j which lie inside the boundary of yarn 2 are assigned the nodal material type of yarn 2. Material type numbers are assigned to each node in the mesh grid depending on their position inside or outside a yarn. The nodal material type numbers or IDs follow a 0 based indexing, where 0 denotes the matrix. The upper and lower intersection points of each yarn along a grid line are termed as "interface" node with an additional interface ID.

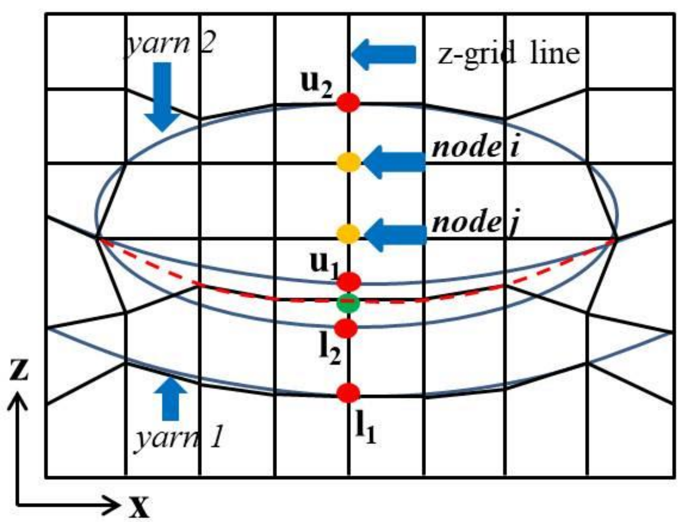


Figure 3.7: Nodal material type assignment

3.5.2 Splitting element faces

Next, the six faces of the 8-node hex element are checked if it needs splitting, based on the nodal material type. A face consists of 4 nodes and based on the node type ID, seven face-splitting cases are identified as shown in Figure 3.8. Let m and n be the nodal material types

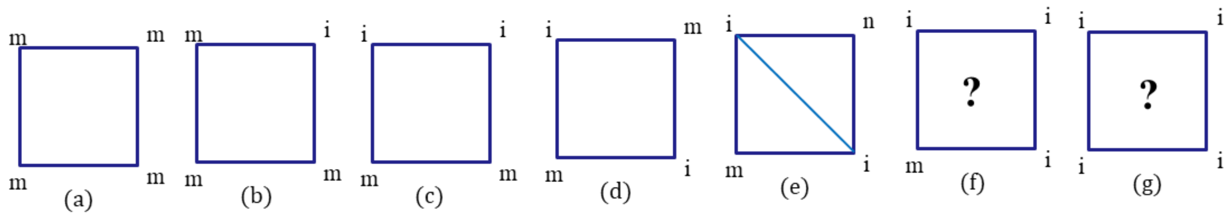


Figure 3.8: Face splitting cases

belonging to either the yarn or the matrix, *i* the interface node type constituting yarn boundaries.

For each vertex node on the face, there are two adjacent nodes on the same face. If a node under consideration is of material type *m*, then the adjacent nodes must either belong to the same material *m* or be interface nodes *i*. As seen in the figure, for cases (a) to (d), no face split is required. Case (e) requires a face split with no face nodes, where the face is split into two

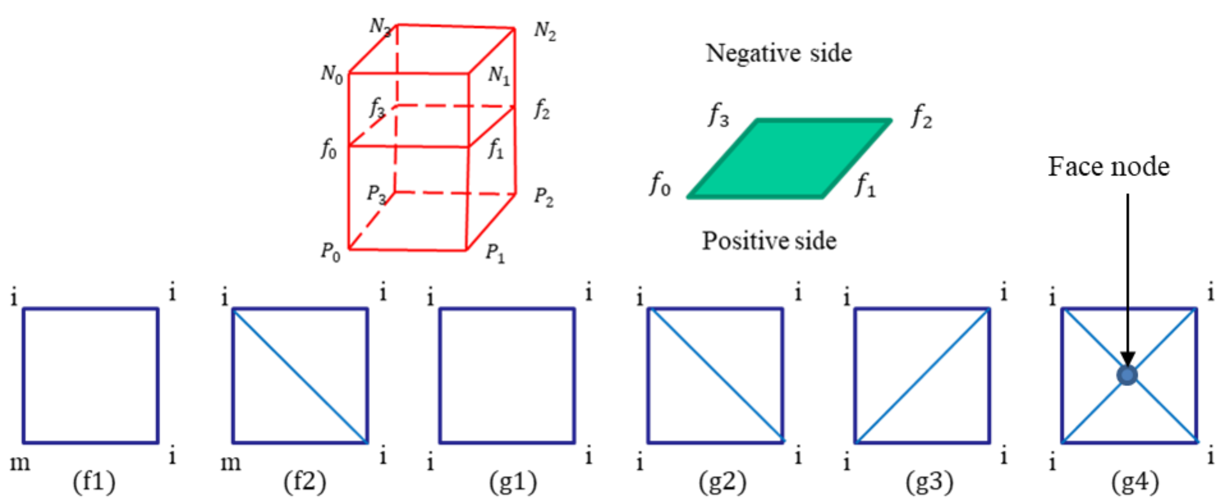
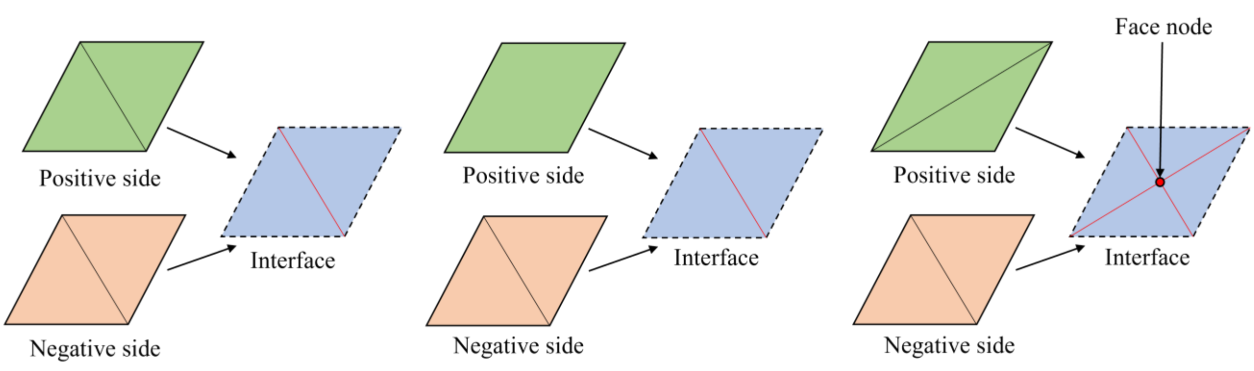


Figure 3.9: Face splitting subcases

triangles along a diagonal connecting the interface nodes – one half of the face belongs to material type *m* and the other half belongs to material type *n*. For cases (f) and (g), face splitting depends on the node type ID combinations of the elements sharing that face. This is shown in Figure 3.9, where based on the adjacent element nodal material type combinations, case (f) can have two and case (g) can have four subcases. For case (f1), if all the interface nodes marked as *i* belong to the interface of material type *m*, then no face split is required. However if the corner node diagonally across from the node of material type *m* does not belong to the interface of material type *m*, the face is split according to case (f2). For the subcases of case g, since all nodes belong to the interface, they are not split if all the nodes share a common interface material type as shown in case (g1). Otherwise, the face is split diagonally if 3 adjacent nodes share a common interface material type as shown in cases (g2) and (g3).

The splitting types of the two adjacent elements sharing the face are examined on both the positive and negative sides. If the face is split in an identical pattern on both sides, no further splitting is required as shown in Figure **3.10**(a). Otherwise, if the patterns are not identical, splitting is required. For example, if one side requires no splitting and the other side requires splitting, the face is split in accordance with the split side pattern, as shown in Figure 3.10(b). In



(a) Identical face split pattern (b) Non-identical face split pattern 1 (c) Non-identical face split pattern 2

Figure 3.10: Illustration of face split subcases

case both the positive and negative sides require splitting, but the pattern is different as shown by the example in Figure **3**.**10**(c), the face is split into four triangles as shown in case (g4) by introducing a mid-side face node, to ensure shape function compatibility across element boundaries. This procedure ensures interface conformity between adjacent elements.

3.5.3 Splitting elements

The initial mesh grid is discretized into 8-node hexahedral elements with 24 degrees of freedom. For stress analysis and post-processing, ANSYS is used. The equivalent element in is SOLID185, which is used for modeling solid structures. It allows for 3 types of degenerations when used in irregular regions – the 6 node prism, 5 node pyramid and 4 node tetrahedron, as shown in Figure 3.11 [54].

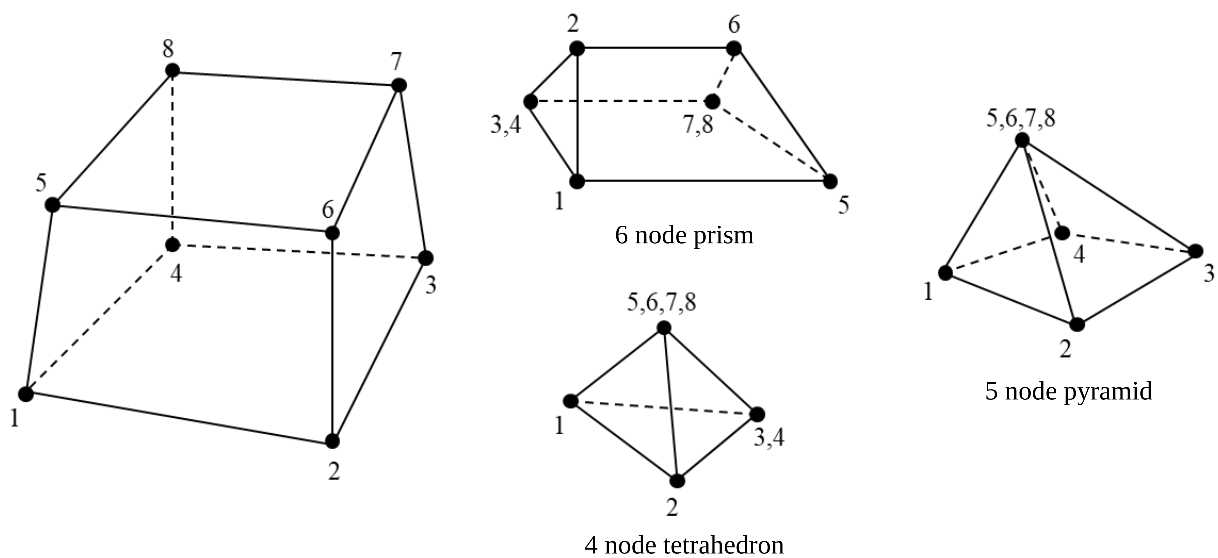


Figure 3.11: SOLID185 element description

The element splitting process is governed by the face-split pattern. Once the face splitting process is completed, the 8-node voxel element can be split into multiple material sections as shown in Figure 3.12, based on the nodal material type combinations. If none of the 6 faces of an

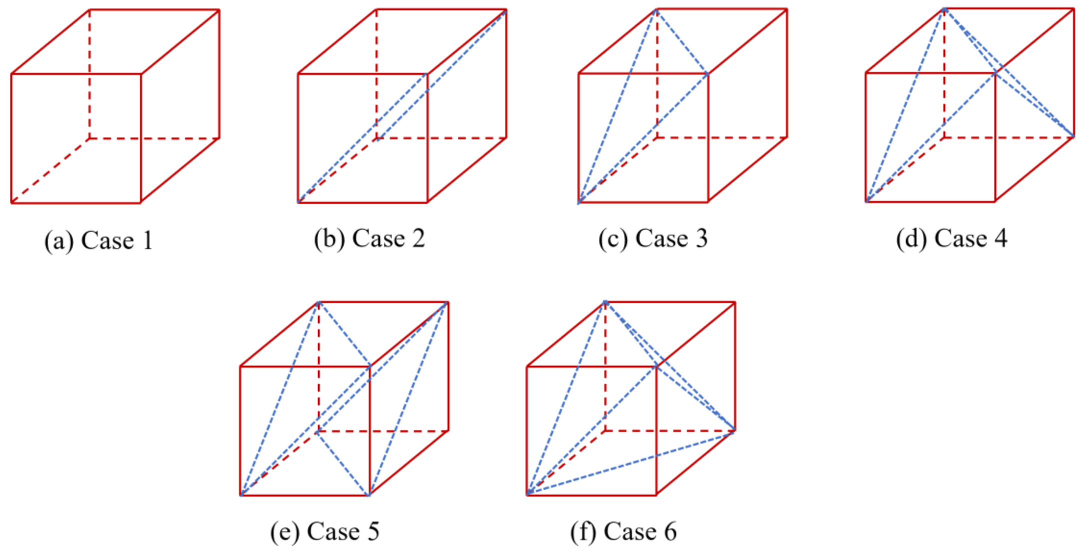


Figure 3.12: Material section splitting illustration

element was split in the previous step and all nodes share a common material type ID, say *m*, the element is not split further and is assigned material type *m*, as shown in Figure 3.12(a).

In all the other cases the element is split to material sections. The blue dotted lines show the split pattern. Case 1 represents 1 material section case, cases 2, 3 represent cases involving 2

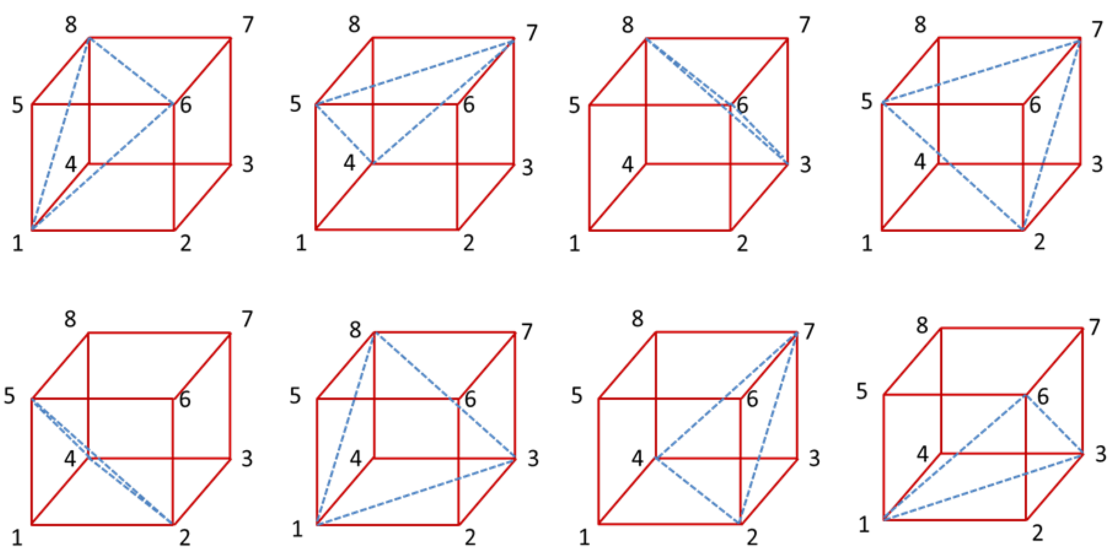


Figure 3.13: Different orientations of material section splitting case 2

material sections, cases 4, 5 are for 3 material sections and case 6 is for 4 or 5 material sections. For each of these cases, there may be several possibilities due to orientation of the nodal material combination and locations in space. An example is shown for Case 3 in Figure 3.13, where there are 8 possible combinations.

The next step is to split each of these material sections into elements that are compatible with the element degeneration definitions illustrated in Figure 3.11 as well as elements from the adjacent voxel elements on shared surface faces. In order to ensure the compatibility of elements from the adjacent voxel, the face split patterns of all material sections must remain unchanged during this process. Two examples are shown for material splitting case 2 in Figure 3.14. The dotted lines show the face split pattern and material interfaces. The original voxel is divided into two sections: a 7-node section and a 4-node tetrahedron element. The 7-node section must be further split based on the surface face split pattern. In case 1, there is a split on the right face. The

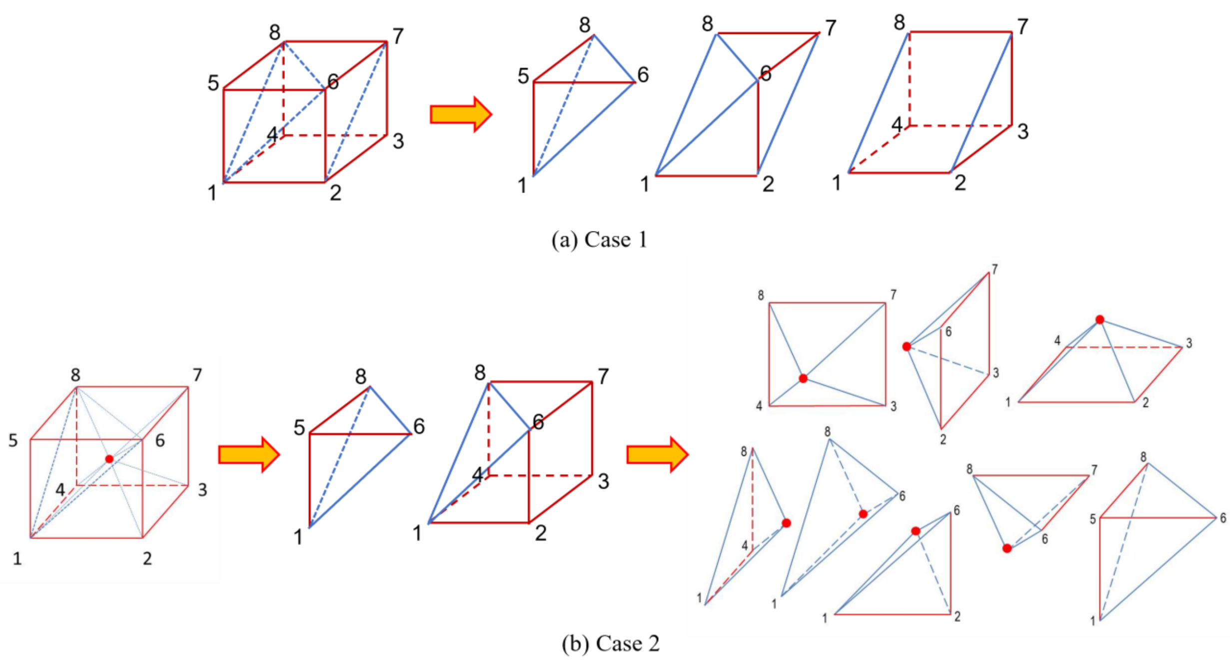


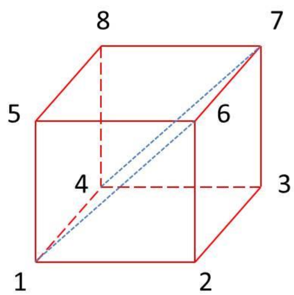
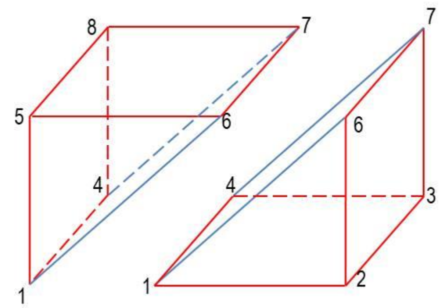
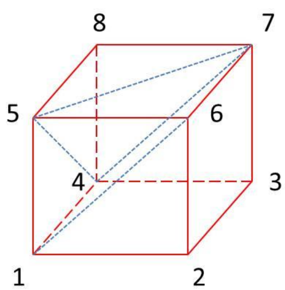
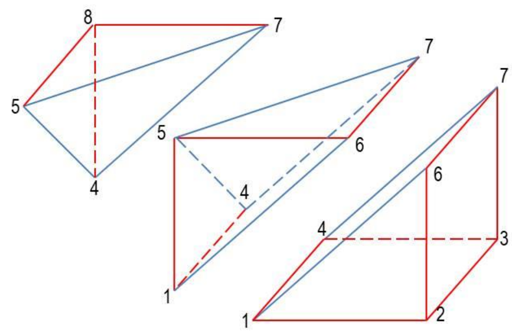
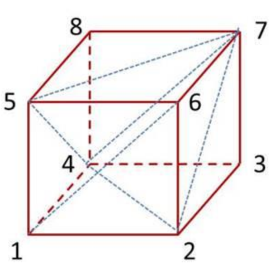
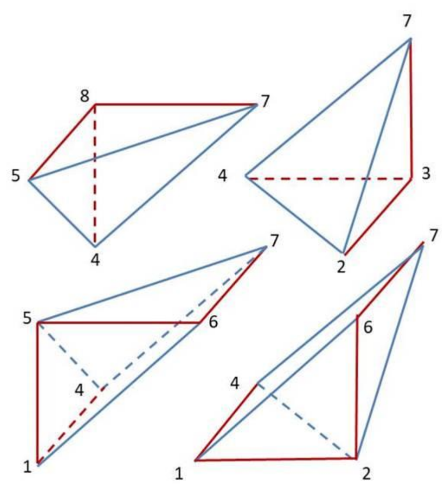
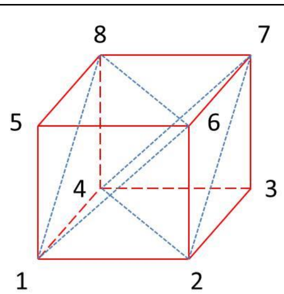
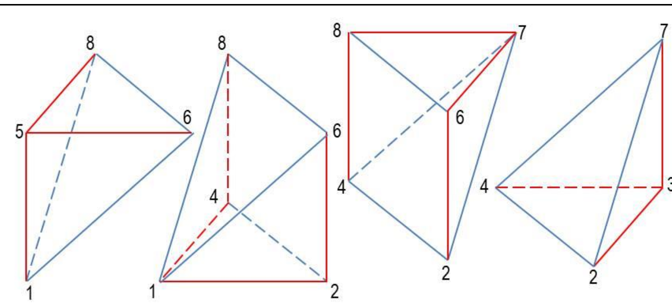
Figure 3.14: Further splitting of material sections for Case 2

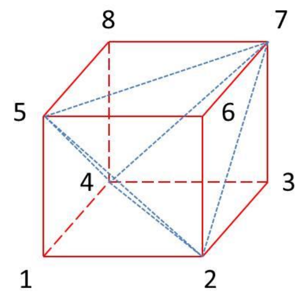
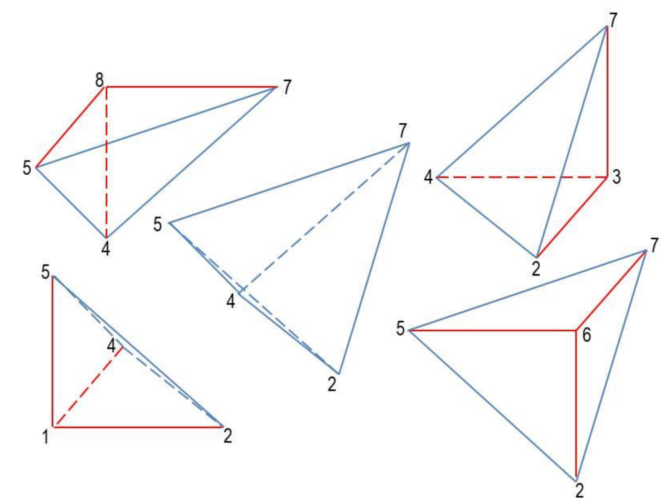
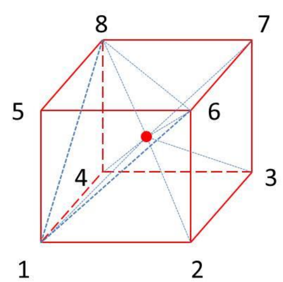
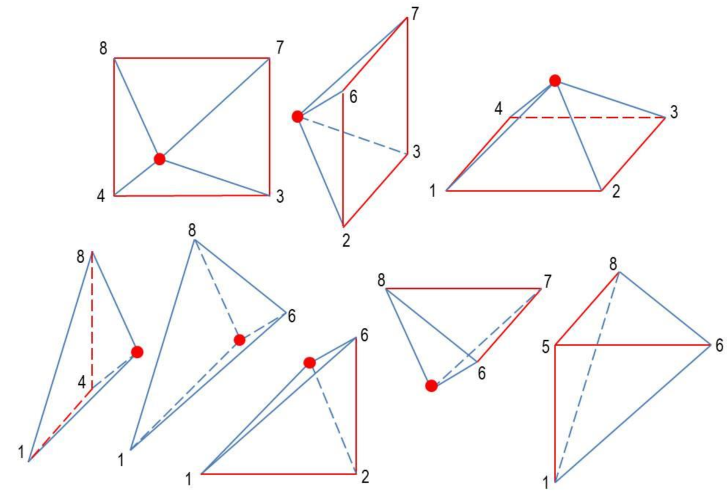
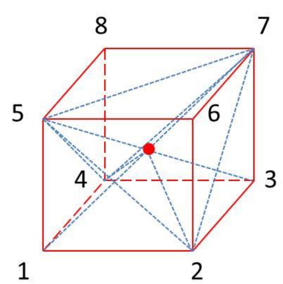
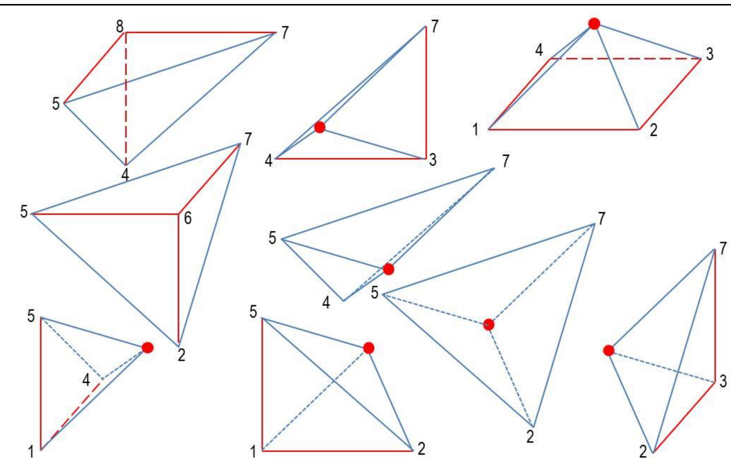
seven node section is split into a 6-node wedge element and a 5-node pyramid element. In case 2, the right face of the 7-node section is not split. In order to preserve the surface face split pattern, a node must be added inside the material section. Because the section has three quadrilateral surface faces and four triangle faces, it is divided into three pyramid elements and four tetrahedron elements. By the end of this process, a conformal FE mesh is created. This procedure makes certain that the voxel elements are not split unless absolutely necessary, ensuring the generation of a conformal volumetric mesh with the least number of nodes and elements. This streamlined element splitting procedure – from face splitting to material section splitting to element splitting –improves the computational efficiency of the meshing algorithm significantly.

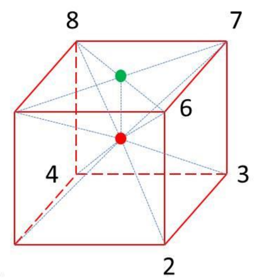
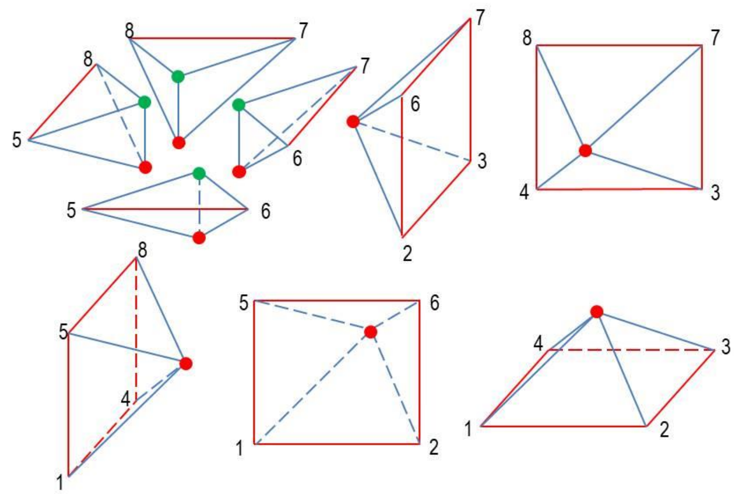
Table 3.1 summarizes the different possible element splitting patterns, as a reference for the reader. In the table, the first column represents the splitting pattern of the original 8-node element where the red and green dots represent additional body or face nodes that were added during splitting and the second column shows the elements after splitting. The element splitting cases according to the face split combinations and material section splits, as shown in the table can be summarized as follows:

1. Case 1: 2 prisms
2. Case 2: 1 tetrahedron, 1 pyramid, 1 prism
3. Case 3: 2 tetrahedrons, 2 pyramids
4. Case 4: 2 tetrahedrons, 2 pyramids
5. Case 5: 5 tetrahedrons
6. Case 6: 5 tetrahedrons, 3 pyramids
7. Case 7: 8 tetrahedrons, 1 pyramid
8. Case 8: 4 tetrahedrons, 5 pyramids

Table 3.1: Element splitting reference

	Split pattern	Elements after splitting
(1)		
(2)		
(3)		
(4)		

(5) 	
(6) 	
(7) 	

(8) 	

It is to be noted here that before splitting, the element can consist of a minimum of one to a maximum of five different materials. Once the elements are split, each element consists of only one material. The nodal material type of each element will either be the yarn or matrix interior node type, say *m*, and the interface node, *i*. Thus the material type of the element is assigned as follows:

1. If all the nodes share the common material type *m*, the element is assigned material type *m*

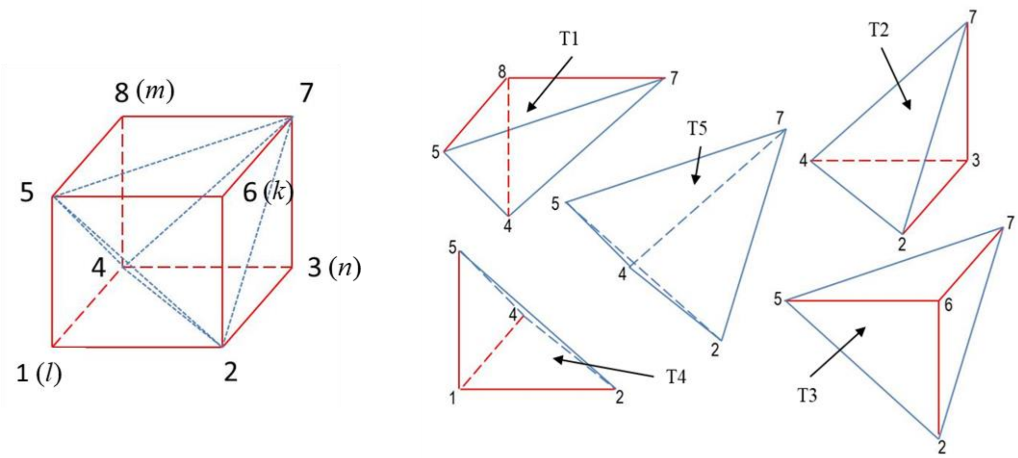


Figure 3.15: Material assignment example

2. If for an element, some nodes are of type *m* and others are of type *i*, the element is still assigned material type *m*
3. If all nodes are interface nodes and they do not share a common material type, the element is assigned the material type of the surrounding matrix

For instance, refer to Figure 3.15. It consists of five different material types – four yarns *m*, *n*, *k*, *l* sharing the interface and the surrounding matrix in between them. For this case, material section splitting case 6 is employed (Figure 3.12(f)). Here, nodes 7, 2, 5 and 4 are the interface nodes. Hence tetrahedron 1 (T1) belongs to yarn *m*, tetrahedron 2 (T2) belongs to yarn *n*, tetrahedron 3 (T3) belongs to yarn *k*, tetrahedron 4 (T4) belongs to yarn *l* and tetrahedron 5 (T5) belongs to the matrix. Finally the element connectivity, new nodes and elements and their respective material type IDs are updated.

3.6 Mesh smoothing

A mesh smoothing operation is performed once the initial conformal mesh is generated after the element splitting process. This is because, although the mesh thus generated is conformal at the yarn-yarn and yarn-matrix interfaces, it might still have yarns with irregular

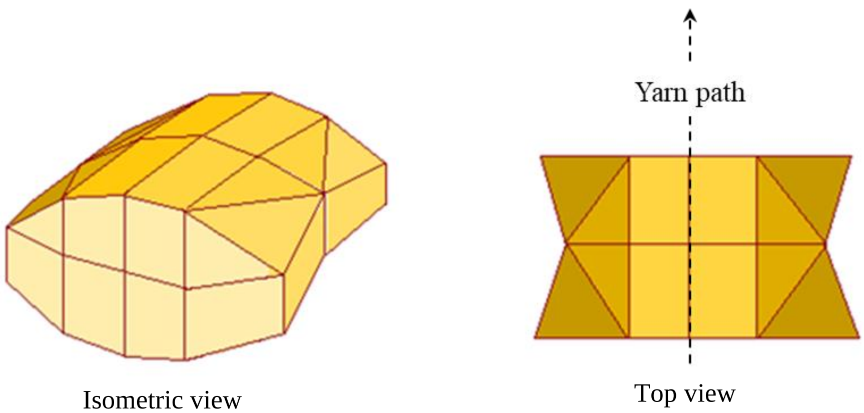


Figure 3.16: Yarn edge discontinuity

edges along its length. Most commonly it occurs at the narrowest yarn cross-sections as illustrated in Figure 3.16. It shows a yarn mesh over three adjacent mesh layers. As seen from the simplified illustration, at times, the edges in adjacent layers of a meshed model are not continuous along the yarn path.

So, first, mesh nodes at the end grid lines of the cross section are rearranged to move to an interpolated location, having the same distance Δz between them at all mesh sections, as shown in Figure 3.17. This ensures that the thickness of the narrow ends of each yarn cross section is uniform. The intermediate result after this step is shown by the second image from left in the figure.

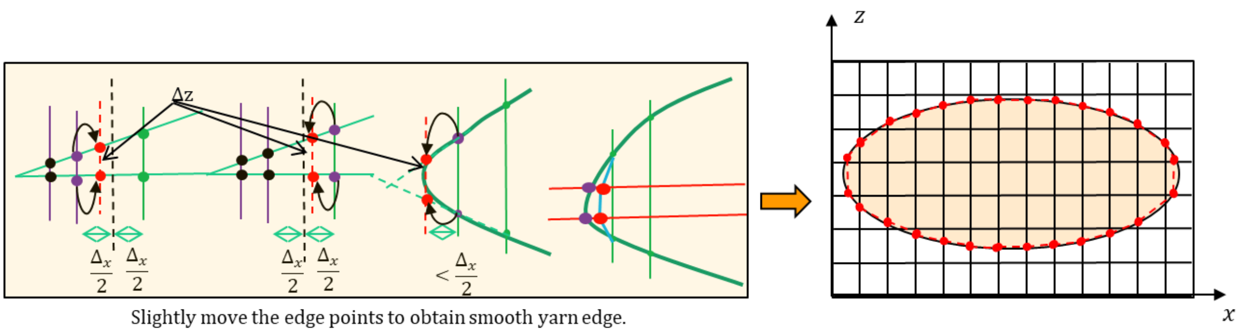


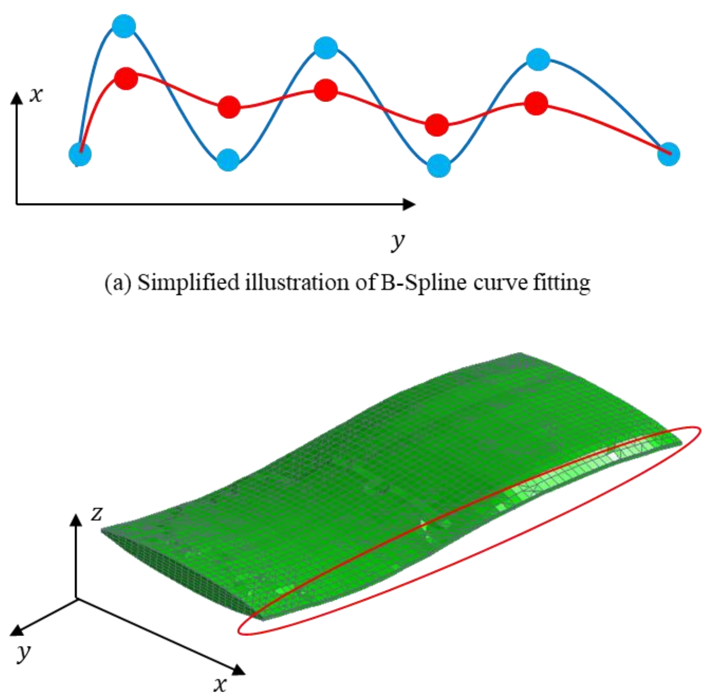
Figure 3.17: Edge modification for mesh smoothing

$$B_{i,0}(u) = \begin{cases} 1, & u_i \leq u \leq u_{i+1} \\ 0, & in\ other\ case \end{cases}$$

$$B_{i,2}(u) = \frac{u - u_i}{u_{i+2} - u_i} B_{i,1}(u) + \frac{u_{i+3} - u}{u_{i+3} - u_{i+1}} B_{i+1,1}(u) \tag{3.3}$$

$$S(u) = \sum_{i=0}^{n} P_i B_{i,2}(u)$$

Next, a B-spline curve is used to interpolate the end section nodes further into a smoother yarn end section that more closely matches the actual geometry. Refer to Figure 3.18(a). An exaggerated illustration of yarn end sections along the yarn path after the intermediate step is shown by the blue line, where the blue dots represent the end section nodes. A B-spline curve of degree 2, given by equation (3.3) is used to smooth this end section along the yarn path to the red line as shown. The end result is shown in Figure 3.18(b), where the red zone marks the region where the smoothing operation was performed. In the equation (**3**.**3**), P_i are the intersection points (control points), n is the total number of control points $B_{i,2}$ are the basis functions and $S(u)$ is the B-spline curve.



(a) Simplified illustration of B-Spline curve fitting

(b) Final result after mesh-smoothing procedure

Figure 3.18: Illustration of B-spline curve implementation for mesh smoothing

3.7 Unit cell mesh periodicity

To ensure that the unit cell mesh boundary is periodic, four imaginary columns (-2, -1, N+1, N+2) are added as shown in Figure 3.19, where the treatment in the x-direction for a plain woven composite cross-section is illustrated. Columns [0 … N] define the unit cell domain. All the intersecting points on columns [0, 1, 2] are copied to columns [N, N+1, N+2], while those in columns [N-2, N-1, N] are copied to columns [-2, -1, 0]. For the y-direction, a similar approach suffices. In the z-direction, the unit cell is periodic because a thin layer of matrix material constitutes the top and bottom layer, which means no nodes from those layers are rearranged.

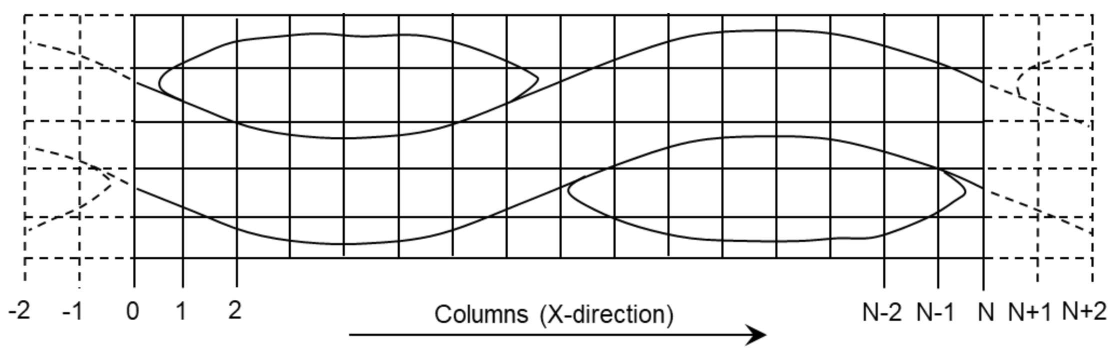


Figure 3.19: Periodic mesh boundary

3.8 Mesh repair

The conformal mesh generation procedure discussed above involves rearranging the nodal locations to match the yarn and matrix boundaries followed by splitting elements at the interface. Therefore, this might lead to elements of poor quality that cannot be used for further analysis, or bad shaped elements which could affect the results. Thus the finite element mesh quality must be inspected to assure that the global stiffness matrix is well-conditioned. This is done in 3 major steps:

1. Check element faces to ensure that quadrilateral faces are convex
2. Check the face warp factor for quadrilateral faces of an element

3. Jacobian-based mesh repair

The shape testing of elements is performed by computing the above parameters which are functions of geometry. The upper limit of these values can be modified using the SHPP command in ANSYS, if needs be.

3.8.1 Convex element face check

The quadrilateral faces of an element are checked to make sure that they are convex. In case a concave face of an element is detected, as shown in Figure 3.20, the face is split into two triangles, by the red line shown in Figure 3.20(c), that are convex. This step is carried out during

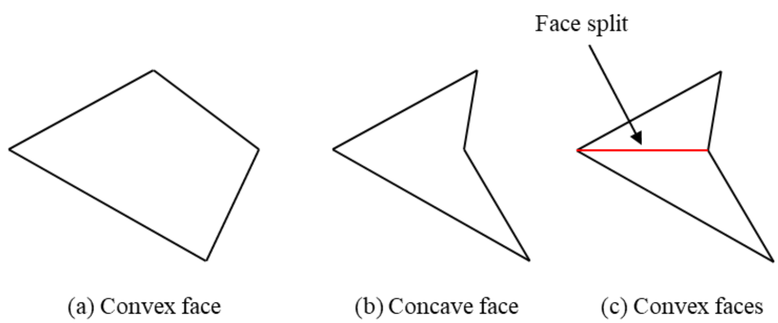


Figure 3.20: Concave face repair

the face-splitting procedure, as outlined in the previous section.

3.8.2 Face warp factor

The next step is to review the face warp factor of the quadrilateral faces of an element. The face warp factor (F_w) for a solid 3D element, such as the 8-node hex element used in this study, is defined as the edge height difference (h) divided by the square root of the projected area (A_p). This is given by equation (3.4) and illustrated in Figure 3.21(a).

$$F_w = \frac{h}{\sqrt{A_p}} \tag{3.4}$$

The warping factor is the maximum of the face warp factors calculated for the 6 quadrilateral faces of a 8-node hexahedron, 3 quadrilateral faces of a 6-node prism or the single quadrilateral face of a 5-node pyramid. A warping factor of 0 is ideal, while a factor of 0.6 is assumed to be the upper limit [55]. A physical representation of these is shown in Figure 3.21(b). Twisting the top face of the 3D brick element by 67.5º with respect to the base produces a warp factor of 0.6. If an element has a warping factor of greater than 0.6, the face with the highest face warp factor is split depending on the Jacobian determinant values of the split elements sharing the face. The shape parameters based on the Jacobian matrix of an element will be discussed in the next sub-section.

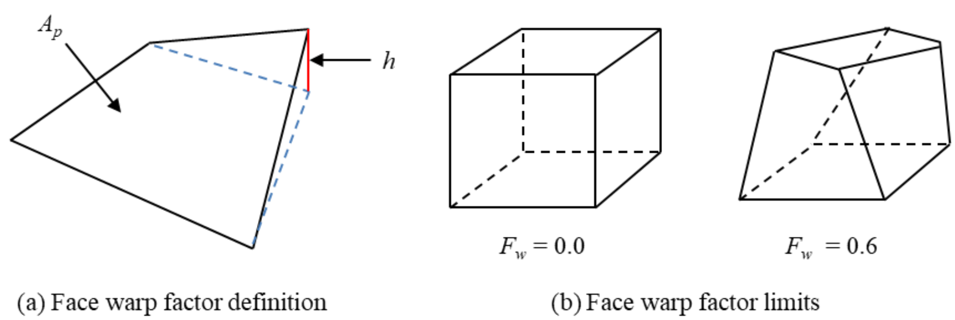


Figure 3.21: Face warping definition

3.8.3 Jacobian-based mesh repair

The 3D 8-node hexahedral linear isoparametric elements as shown in Figure 3.11 are used in this study. Isoparametric elements are those where the geometry and displacement fields are specified in parametric forms and interpolated using the same shape functions. The shape functions are polynomials of local coordinates ξ, η and ζ. For the 8 nodes of the 3D solid element, the local coordinates are given in Table 3.2.

Table 3.2: Local coordinates of nodes

Node i	ξ_i	η_i	ζ_i
1	-1	-1	-1
2	1	-1	-1
3	1	1	-1
4	-1	1	-1
5	-1	-1	1
6	1	-1	1
7	1	1	1
8	-1	1	1

The shape functions (N_i) of the linear 3D element are given by equation (3.5). ξ, η and ζ are the local coordinates of a point within the element and ξ_i, η_i and ζ_i are the local nodal coordinates.

$$N_i = \frac{1}{8}(1 + \xi\xi_i)(1 + \eta\eta_i)(1 + \zeta\zeta_i) \tag{3.5}$$

The material Jacobian matrix can then be calculated using equation (3.6) as:

$$[J] = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \tag{3.6}$$

The partial derivatives of the point global coordinates x, y and z with respect to the local coordinates are determined by differentiation of displacements expressed through shape functions and global nodal displacement values:

$$\begin{gathered} \frac{\partial x}{\partial \xi} = \sum \frac{\partial N_i}{\partial \xi} x_i, \quad \frac{\partial x}{\partial \eta} = \sum \frac{\partial N_i}{\partial \eta} x_i, \quad \frac{\partial x}{\partial \zeta} = \sum \frac{\partial N_i}{\partial \zeta} x_i \\ \frac{\partial y}{\partial \xi} = \sum \frac{\partial N_i}{\partial \xi} y_i, \quad \frac{\partial y}{\partial \eta} = \sum \frac{\partial N_i}{\partial \eta} y_i, \quad \frac{\partial y}{\partial \zeta} = \sum \frac{\partial N_i}{\partial \zeta} y_i \\ \frac{\partial z}{\partial \xi} = \sum \frac{\partial N_i}{\partial \xi} z_i, \quad \frac{\partial z}{\partial \eta} = \sum \frac{\partial N_i}{\partial \eta} z_i, \quad \frac{\partial z}{\partial \zeta} = \sum \frac{\partial N_i}{\partial \zeta} z_i \end{gathered} \tag{3.7}$$

The determinant of the Jacobian matrix, denoted by $\boldsymbol{D_J}$ is checked at each sampling location of an element. For the 3D 8-node element, the sampling locations are all the 8 corner nodes. $\boldsymbol{D_J}$ at a sampling location represents the magnitude of the mapping function between the element local coordinates and the real global coordinate space. In an ideally shaped element, $\boldsymbol{D_J}$ is nearly constant over the element. An element might need nodal coordinate correction, also termed as "mesh repair" if one of the following two cases is true:

1. $\boldsymbol{D_J}$ value is negative, which implies that the element is not valid for analysis. The mapping function for the considered element is not a bijection thus the singularity of the mapping function leads to modeling inconsistency. A geometrical representation of an invalid element is shown in Figure 3.22(b).
2. The Jacobian ratio at a sampling location, which is defined as the ratio of the maximum Jacobian determinant of the element to that of the Jacobian determinant at the current location, is greater than 30. This Jacobian Ratio limit is the default value,

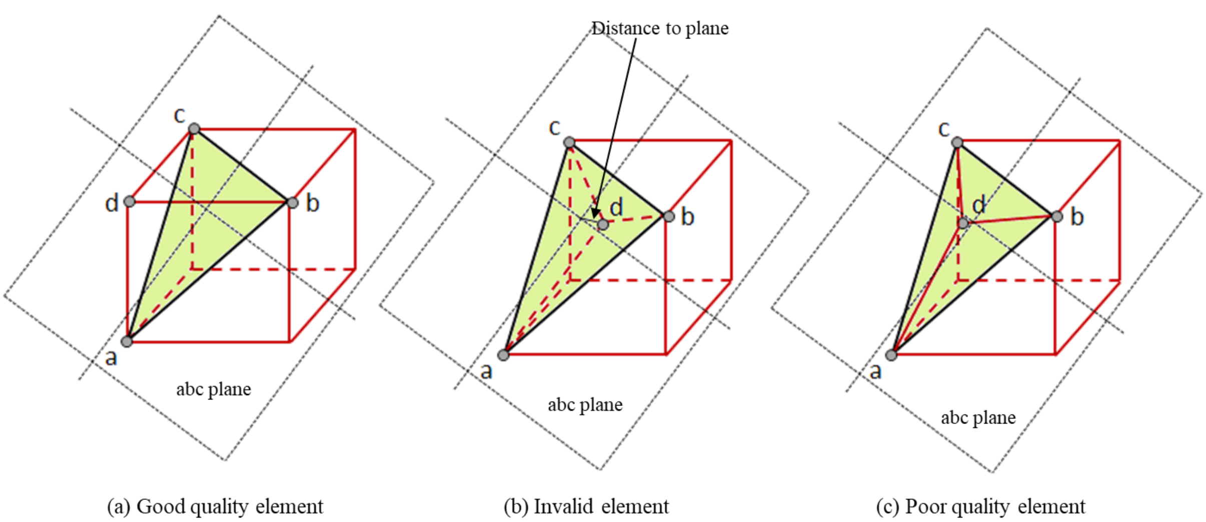


Figure 3.22: Element quality check: If signed distance of node **d** to the **abc** plane is negative, the element is invalid. If the distance of node **d** from the **abc** plane is almost equal to zero, the element is of poor quality

but can be changed if required. A good quality element has all its Jacobian ratio values within the interval [1, 30]. A geometrical representation of a poor quality element is shown in Figure 3.22(c).

Algorithm 3.2: Jacobian-based mesh repair algorithm

```
Get total elements after split as T
for element = 1 to T do {Elements loop}
    for node = 1 to 8 do {Sampling location global coordinate loop}
        Store node global coordinates for element in array CN[]
    end for
    Determine maximum Jacobian determinant (maxDJ) out of the 8 sampling locations
    for node = 1 to 8 do {Sampling location local coordinate loop}
        Calculate Jacobian determinant (DJ) of each sampling location
        if DJ < 0 or maxDJ/DJ > 30 then
            Mark node for need of nodal correction
            for increment = 1 to 1.2 do {Nodal correction loop}
                for UnitVector = 1 to 26 do {Direction vector loop}
                    CN[] += increment * UnitVector
                    Recalculate DJ with updated CN[]
                    if DJ > 0 and maxDJ/DJ < 30 then
                        Leave Nodal correction loop
                    else
                        Set CN[] back to previous value
                        Go to next UnitVector
                    end if
                end for
            end for
            Return Jacobian determinant value
            if node is repaired then
                Repair was successful
                Set nodal coordinates to global repaired coordinates CN[]
            else
                Repair failed
            end if
        end if
    end for
end for
```

In case any of the above conditions hold true, a Jacobian-based mesh repair algorithm similar to that implemented by Bucki *et al*. [56] is used to improve the element quality. The sampling location or node in consideration is moved in tiny increments in 26 different directions simultaneously, shown by the unit vectors from the center in Figure 3.23. It was found that the 26 direction vectors are sufficient for determining an optimal nodal correction. The Jacobian determinant and Jacobian ratio are recalculated for all the elements sharing that node. Once a nodal position is determined that satisfied both criteria for all the elements attached to the node, the nodal correction is accepted. The implementation of the mesh-repair algorithm for element validity and quality optimization is given in Algorithm 3.2.

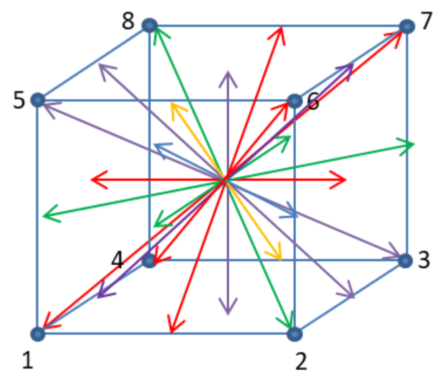


Figure 3.23: Unit vectors for optimal nodal correction for poor quality elements

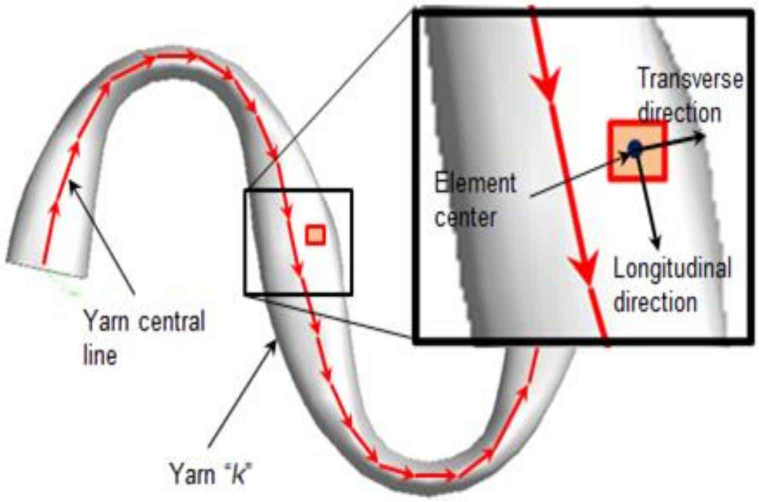
3.9 Local orientations and material properties

The finite element mesh of the composite unit cell consists of two constituent material geometries – the yarns and the surrounding matrix. The yarn elements are modeled as homogenous transversely isotropic, while the surrounding matrix elements are isotropic in nature. A transversely isotropic material is one which has one axis of symmetry, which in this case is the fiber longitudinal direction. DFCA fabric micro-geometry gives the overall fiber volume fraction of the unit cell, given as f_v. The unit cell total volume given by V_u can be calculated from the unit cell micro-geometry dimensions in DFCA. The volume of the reinforcing yarns, v_y, is obtained as the sum of the volumes of all the elements in the FE mesh that make up the yarn. Since each element in the FE model is a variation of the original 8 node hexahedral element, its volume can be calculated by calculating the volume of the hexahedral elements. An efficient algorithm for calculating the volume of a hexahedral element based on the vertex coordinates was proposed by Grandi [57]. The volume is computed using 60 FLOPS (floating point operations), using 3 determinants of vectors connecting the vertices. The same volume calculation algorithm is employed in this study to determine the volumes of each element, and subsequently the volume of the yarn. Thus, the mesoscale fiber volume fraction in yarns, v_f, which is also termed as the "packing density" of fibers in yarns, can be determined as:

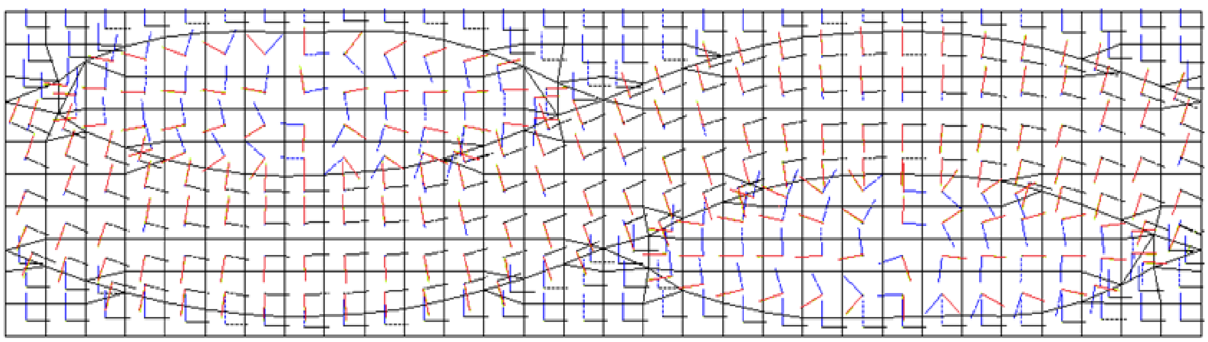
$$v_f = \frac{f_v V_u}{v_y} \tag{3.8}$$

As noted before, in this study, a mesoscale idealization is used – each yarn element is treated as a unidirectional fiber-reinforced laminate in its local coordinate system. Thus, the composite yarn element material properties can be determined using well-established composite micro-mechanics theories such as the Chamis' micro-mechanics model [58]. The mesoscale fiber volume fraction v_f is used to determine the same, as illustrated in Appendix A - The material properties of the transversely isotropic yarn elements vary with the yarn orientation unlike the isotropic matrix. Thus, the local coordinate system for each yarn element needs to be specified.

The yarn geometry in DFCA is defined by cross-sectional profiles along the yarn central line. The central line is further defined as a series of interconnected segments, joined at the center of each cross-sectional profile. Refer to Figure 3.24(a). The red line denotes the yarn central line, with each red arrow marking a segment. For an element that belongs to yarn k as shown in the figure, the segment closest to the element center is determined, in order to determine its local coordinate direction vectors. The fiber or longitudinal direction vector within the element is set parallel to the vector defining the closest central line segment. Let this unit vector components be denoted as (X_1, Y_1, Z_1). The transverse radial direction, the components of



(a) Determining element local coordinates



(b) Local coordinate system at a plain woven unit cell layer

Figure 3.24: Element local coordinate system

which are denoted as (X_2, Y_2, Z_2), is defined as the perpendicular unit vector directed from the segment to the element center. The third principal transverse unit vector (X_3, Y_3, Z_3), is given by the cross product of the first two vectors. The rotation matrix for the element local coordinate system is given by equation (3.9).

$$R = \begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ X_3 & Y_3 & Z_3 \end{bmatrix} \tag{3.9}$$

The convention adopted for the rotational degrees of freedom of a coordinate system is the 3-1-2 Euler angle sequence. Here, three rotations are implemented. The first one is about the original global Z-axis (or 3-axis), the second one is about the local X'-axis (or 1-axis) resulting from the first rotation and the third one is about the local Y''-axis (or 2-axix) resulting from the second rotation [59]. Thus, using the following notations, the rotation matrix for transformation can be created using the three rotations according to equation (3.10).

$c_1 = \cos\theta_1, \quad s_1 = \sin\theta_1$

$c_2 = \cos\theta_2, \quad s_2 = \sin\theta_2$

$c_3 = \cos\theta_3, \quad s_3 = \sin\theta_3$

$Z_1 = first\ rotation\ \theta_1\ about\ original\ Z - axis$

$X_2 = second\ rotation\ \theta_2\ about\ local\ X' - axis$

$Y_3 = third\ rotation\ \theta_3\ about\ local\ Y" - axis$

$$R = Z_1 X_2 Y_3 = \begin{bmatrix} c_1c_3 - s_1s_2s_3 & -c_2s_1 & c_1s_3 + s_1s_2c_3 \\ c_3s_1 + c_1s_2s_3 & c_1c_2 & s_1s_3 - c_1c_3s_2 \\ -c_2s_3 & s_2 & c_2c_3 \end{bmatrix} \tag{3.10}$$

Element local coordinate systems in ANSYS are defined using the LOCAL command and assigned using the ESYS command. The input parameters to define the local coordinate systems are the three rotation angles in degrees. Comparing equations (3.9) and (3.10), these rotation angles can be determined in terms of the local coordinate direction nit vectors of the element and are given in equation (3.11). Figure 3.24(b) shows the element local coordinate systems for a 2D plain-woven unit cell cross section. The matrix is isotropic which is why its local coordinate system is aligned with the global Cartesian coordinate system. The yarn elements on the other hand have their material orientations parallel to the yarn central line.

$$\theta_1 = \tan^{-1}\left(\frac{-Y_1}{Y_2}\right)$$

$$\theta_2 = \sin^{-1}(Y_3) \tag{3.11}$$

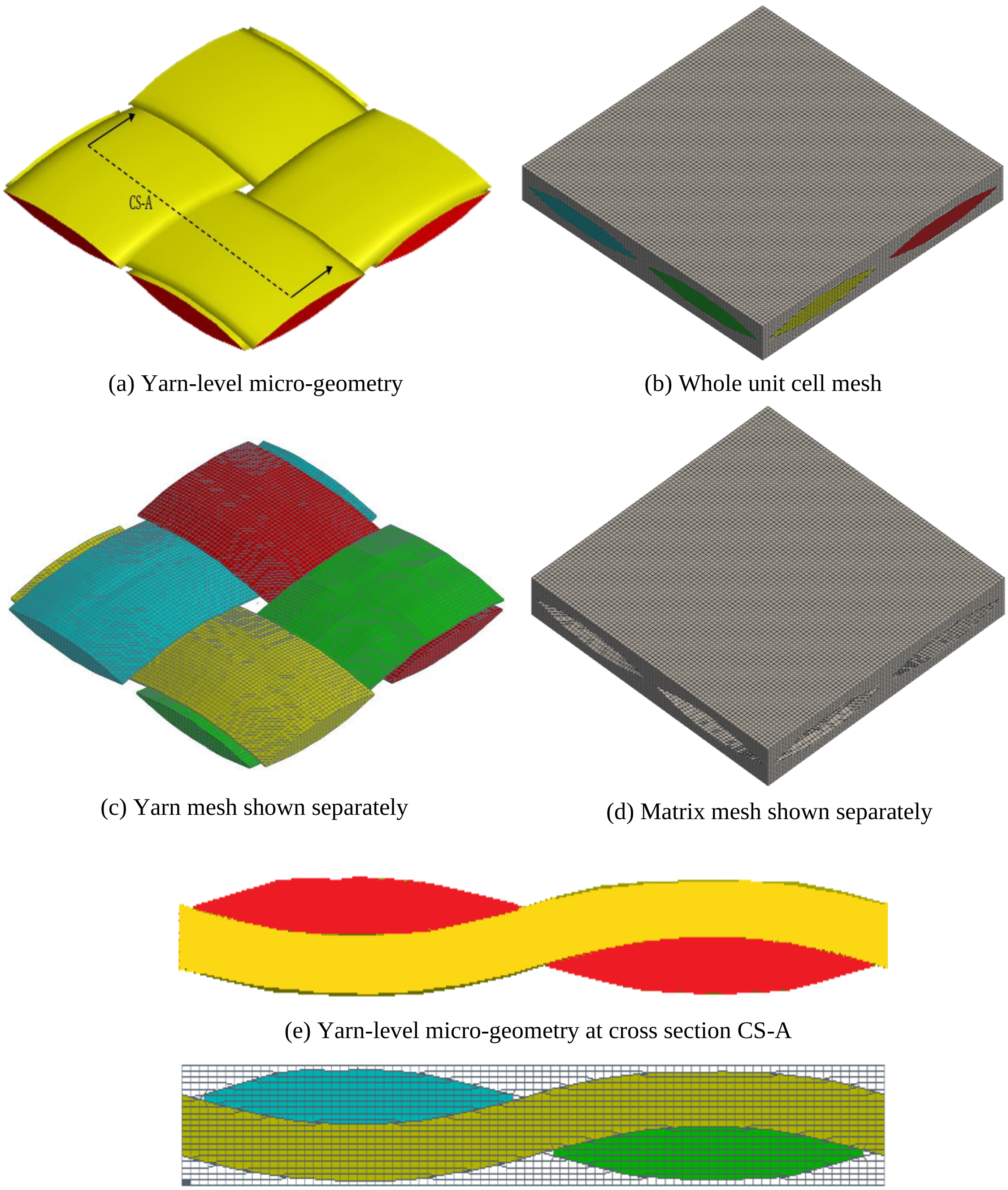
$$\theta_3 = \tan^{-1}\left(\frac{-X_3}{Z_3}\right)$$

3.10 Mesh validation

Once all of the steps above are completed, the final conformal finite element unit cell mesh is obtained. The mesh validation is implemented in three phases.

3.10.1 Geometry comparison

The meshed models of a 2D plain-woven and a 3D orthogonal-woven unit cell are presented in Figure 3.25, Figure 3.26 and Figure 3.27. The figures show the yarn-level geometry of the unit cells (a), the meshed models of the entire unit cell (b), the meshed models of the reinforcing yarns (c) and the meshed models of the surrounding matrix (d). Figure 3.25(a) and Figure 3.26(a) also show the warp cross-section locations to be examined for comparison. Figure 3.25(e) and Figure 3.27(a) shows the yarn-level micro-geometry cross-section while Figure 3.25(f) and Figure 3.27(b) shows the meshed cross-section at the same marked location for the 2D plain-woven unit cell and the 3D orthogonal-woven unit cell respectively. As is evident from these comparisons, the conformal FE mesh matches the exact yarn-yarn and yarn-matrix boundaries and is in very good agreement with the realistic micro-geometries obtained using the dynamic relaxation procedure in DFCA. The geometry and mesh parameters for the two example cases are listed in Table **3.3**.



(a) Yarn-level micro-geometry

(b) Whole unit cell mesh

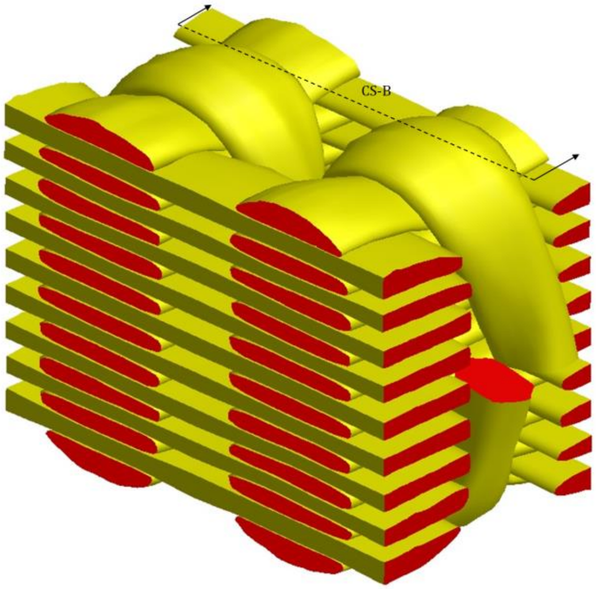
(c) Yarn mesh shown separately

(d) Matrix mesh shown separately

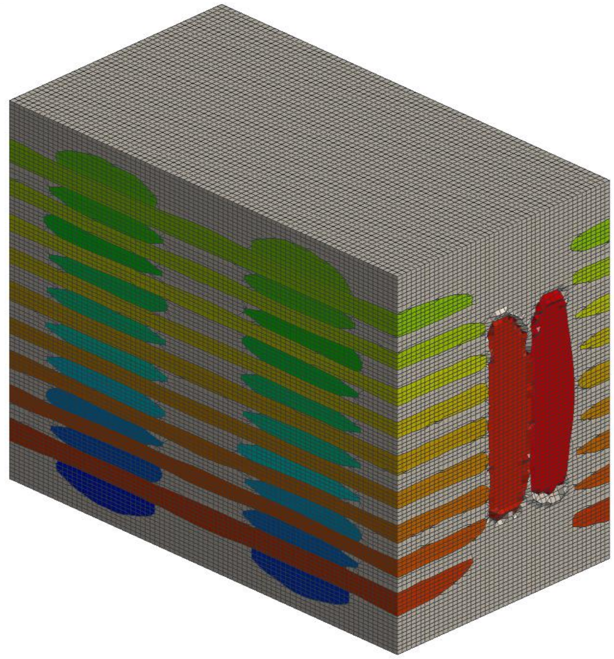
(e) Yarn-level micro-geometry at cross section CS-A

(f) Mesh at cross section CS-A

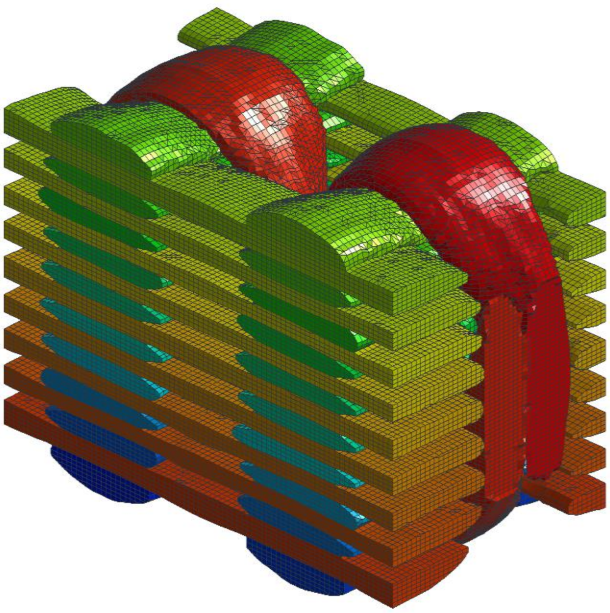
Figure 3.25: 2D plain-woven unit cell geometry comparison



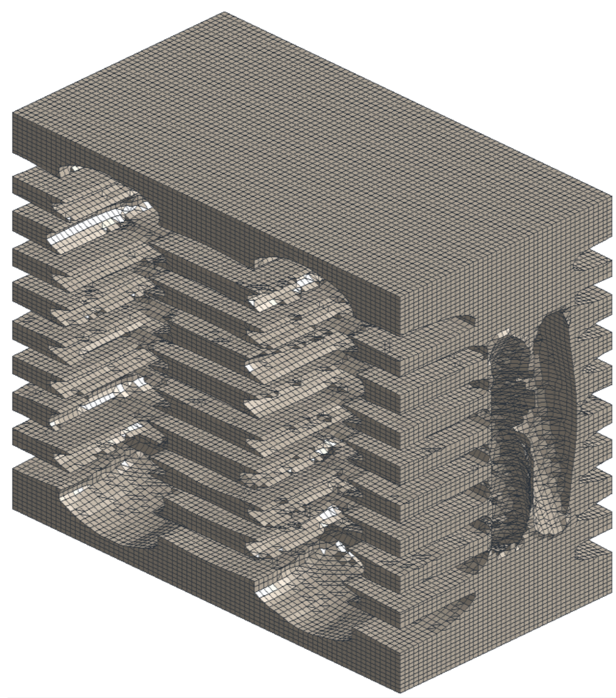
(a) Yarn-level micro-geometry



(b) Whole unit cell mesh

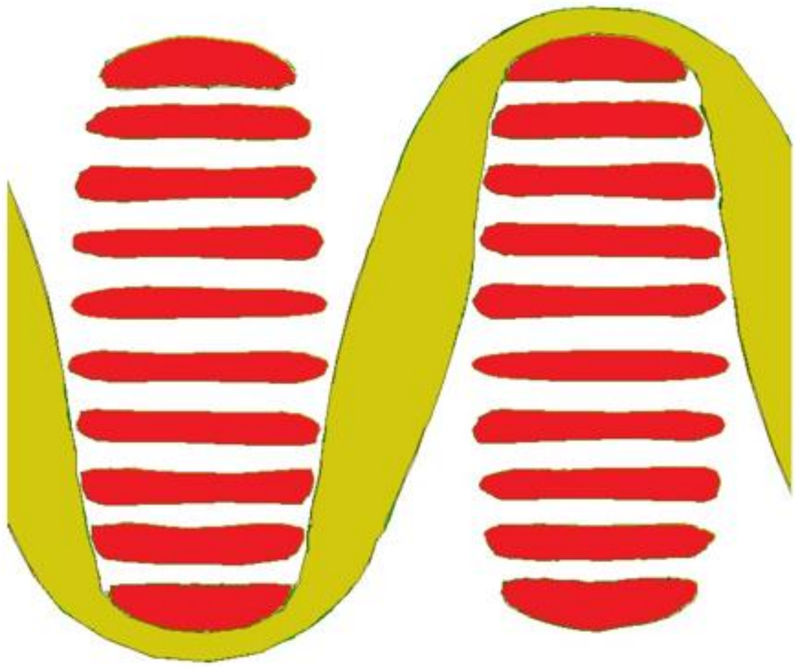


(c) Yarn mesh shown separately

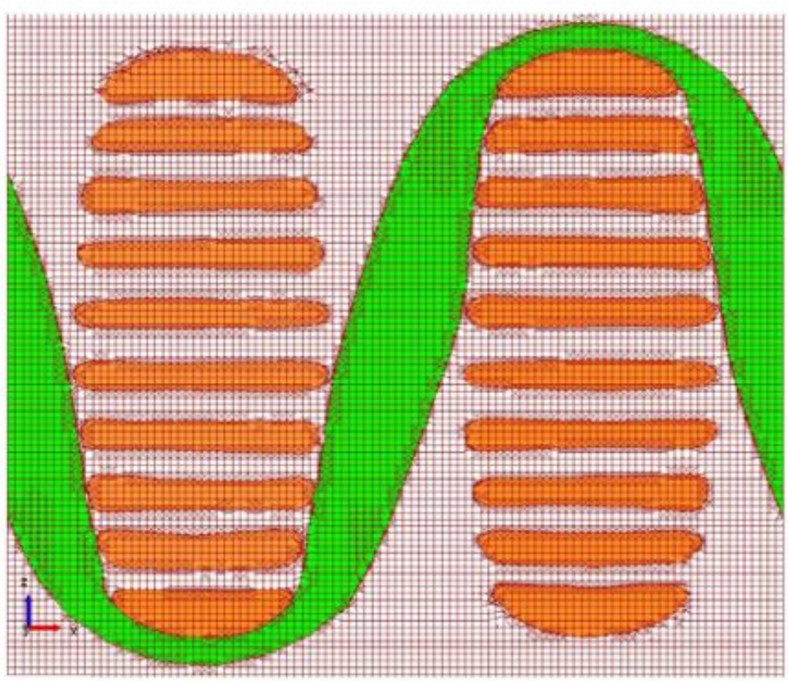


(d) Matrix mesh shown separately

Figure 3.26: 3D orthogonal woven unit cell geometry and mesh comparison



(a) Yarn-level micro-geometry at cross-section CS-B



(b) Mesh at cross-section CS-B

Figure 3.27: 3D woven orthogonal unit cell cross-section comparison

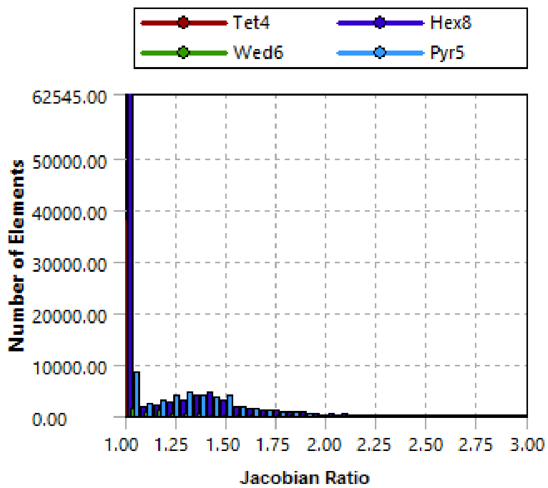
Table 3.3: Geometry and mesh parameters of the unit cells

	2D plain woven	3D orthogonal woven
Total yarn types	2	3
Weft layers	1	10
Weft columns	2	2
Warp sections	2	3
Cell length	3.00 mm	2.90 mm
Cell width	3.00 mm	1.58 mm
Cell height	0.30 mm	3.05 mm
Overall fiber volume fraction	45.2%	45%
Mesh domain columns (x-divisions)	80	90
Mesh domain layers (y-divisions)	80	50
Mesh domain rows (z-divisions)	20	100
Total nodes	150,049	525,939
Total elements	212,020	858,382

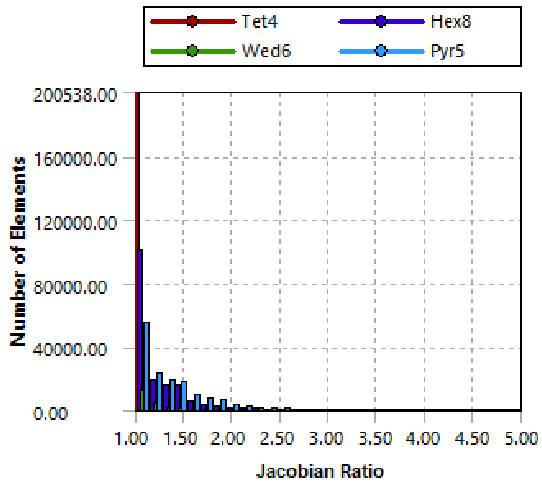
3.10.2 Mesh metrics

The meshed model of the unit cell can be exported to downstream FE codes from DFCA. DFCA has a built-in function that writes the mesh output into a file format readable by FE codes. It includes the nodal positions, element connectivities, element local orientations, material properties and periodic boundary constraints, which will be discussed in the next section.

The element quality results for the 2D plain woven and 3D orthogonal woven unit cells, presented in the previous section are examined. The Jacobian ratio value distribution as a measure of the mesh quality is shown in Figure 3.28. Table 3.4 also shows distributions of the Jacobian ratio values for the two unit cells. The percentage of total elements with a Jacobian ratio of greater than three in the 2D woven model and greater than five in the 3D woven model is less than 0.01%. Considering the fact that an element is of poor quality with a Jacobian ratio greater than 30 and good quality within a Jacobian ratio range of 1 to 10, with 1 representing an element of the best quality, it can be concluded that the proposed mesh generation algorithm can be used to generate realistic conformal meshed models of woven composite unit cells with excellent element quality.



(a) 2D plain woven unit cell



(b) 3D orthogonal woven unit cell

Figure 3.28: Element quality metrics

Table 3.4: Mesh metrics

Jacobian Ratio	1.0 – 1.5	1.5 – 2.0	2.0 – 2.5	2.5 – 3.0	3.0 – 4.0	4.0 – 5.0	> 5.0
Percentage of total elements (2D woven)	91.52%	6.57%	1.65%	0.24%	0.02%	–	–
Percentage of total elements (3D woven)	89.91%	7.98%	1.66%	0.36%	0.08%	0.008%	0.002%

3.10.3 Linear elastic analysis

The final phase of the mesh validation involves a comparison of linear elastic stiffness properties with available test and numerical data in literature. However first, periodic boundary constraints need to be specified for the unit cell mesh domain. This is because the unit cell is the smallest repeating unit of a woven fabric, which can be used to evaluate the bulk behavior of the system.

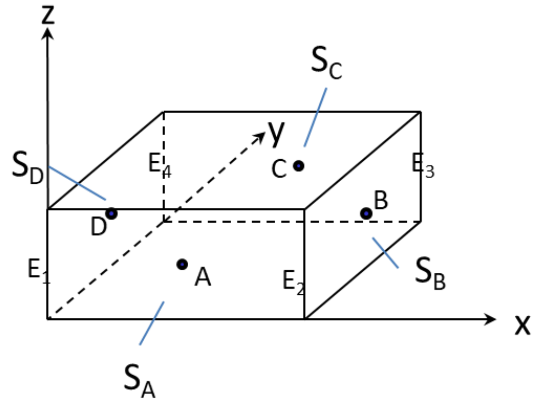


Figure 3.29: Illustration of periodic boundary constraints

Figure 3.29 shows a simplified illustration of a unit cell domain. Denoting *A*, *B*, *C*, and *D* as the mid-nodes of the faces S_A, S_B, S_C and S_D, and E_1, E_2, E_3 and E_4 as the four boundary edges, the following periodic constraints are imposed of the unit cell FE model to ensure displacement continuity at the unit cell boundary. The degrees of freedom in the x- and y-directions of the face nodes are tied to the corresponding mid nodes according to equation (3.12).

$$\begin{aligned} u_i(S_B) - u_i(S_D) &= u_i(B) - u_i(D), \quad i = x, y \\ u_i(S_C) - u_i(S_A) &= u_i(C) - u_i(A), \quad i = x, y \end{aligned} \tag{3.12}$$

The degrees of freedom in the z-direction for the face nodes are coupled as:

$$\begin{aligned} u_z(S_B) &= u_z(S_D) \\ u_z(S_C) &= u_z(S_A) \end{aligned} \tag{3.13}$$

The edges are considered separately. In a similar manner to the faces, the x- and y-direction degrees of freedom of the edges are tied to the mid-nodes of the faces as:

$$\begin{aligned} u_i(E_2) - u_i(E_1) &= u_i(B) - u_i(D), \quad i = x, y \\ u_i(E_3) - u_i(E_4) &= u_i(B) - u_i(D), \quad i = x, y \\ u_i(E_4) - u_i(E_1) &= u_i(C) - u_i(A), \quad i = x, y \\ u_i(E_3) - u_i(E_2) &= u_i(C) - u_i(A), \quad i = x, y \end{aligned} \tag{3.14}$$

Finally, the z-direction degree of freedom is coupled for all the edge nodes as:

$$u_z(E_1) = u_z(E_2) = u_z(E_3) = u_z(E_4) \tag{3.15}$$

In order to simulate the response of a thin lamina, the top and bottom face nodes (z-direction) are kept free. By enforcing uniform displacement conditions on the top and bottom face nodes, the response of a thick laminate can be simulated [60].

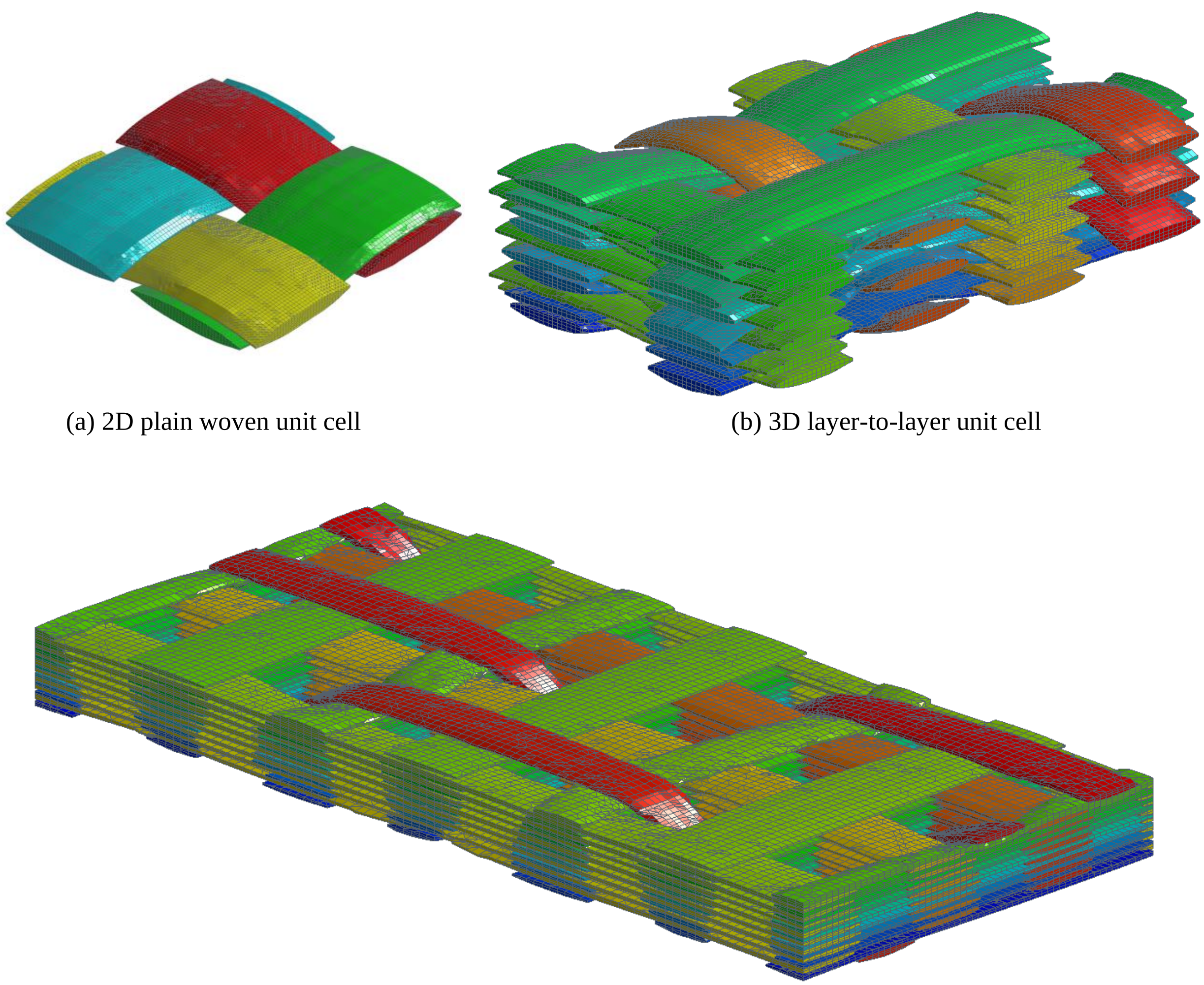
Once everything is set up, linear elastic stiffness results from the DFCA mesh model for different 2D and 3D woven unit cells are compared to existing results reported in literature. Three different unit cells are studied – a 2D plain weave laminate, a 3D layer-to-layer unit cell and a 3D orthogonal weave unit cell, with the same materials, fiber volume fractions and geometrical parameters as the ones in the experiments and numerical studies. The unit cell micro-geometry mesh for the reinforcing yarns for each of these three cases is shown in Figure 3.29. The matrix mesh or the whole unit cell mesh is not shown to keep the illustration simple. The dimensions, overall fiber volume fractions and materials for the fiber and matrix for each of the three test cases are listed in Table 3.5.

Finally, the results for the predicted stiffness are presented in Table 3.6. As is evident from the table, the predicted values of stiffness from the DFCA model matches very well with the experimental as well as numerical results. For cases where both the experimental and numerical results are available, it can be observed that the DFCA unit cell model provides a much better prediction of the stiffness in comparison to the existing numerical simulations.

Hence, from the three mesh validation arguments discussed in this chapter, it can be concluded that the proposed mesh generation algorithm that is incorporated into DFCA improves the accuracy of micro-mechanical analysis of woven composites significantly, and thus can be used to generate finite element mesh of realistic woven composite unit cells with excellent element quality.

Table 3.5: Unit cell geometrical parameters and material properties

Model Type	Dimension	f_v	Fiber	Matrix
2D laminate	6.43 mm × 6.11 mm × 5.09 mm	0.44	AS4 12k	Vinyl ester
3D Layer-to-layer	5.08 mm × 8.47 mm × 4.00 mm	0.52	AS4 12k	RTM6 epoxy
3D Orthogonal	18.90 mm × 8.60 mm × 4.17 mm	0.50	AS4 12k	RTM6 epoxy



(a) 2D plain woven unit cell

(b) 3D layer-to-layer unit cell

(c) 3D orthogonal unit cell

Figure 3.30: Yarn mesh of the unit cells used for linear elastic comparison

Table 3.6: Comparison of predicted stiffness

Model Type	Load direction	Experiment	Numerical results	DFCA model
2D laminate	Warp tension	42.8 GPa [61]	41.5 GPa [41]	42.5 GPa
3D Layer-to-layer	Weft tension	N/A	53.3 GPa [62]	51.6 GPa
	Warp tension	N/A	35.2 GPa [62]	33.8 GPa
3D Orthogonal	Warp tension	57.7 GPa [63]	59.3 GPa [45]	57.3 GPa

Chapter 4 - Progressive damage analysis of woven composites

This chapter presents a predictive model for determining the damage response of textile composite unit cells. The failure initiation and evolution laws and the development of the progressive damage model is discussed in detail. As previously illustrated in Chapter 1 - material fracture energy inputs required for the Continuum Damage Mechanics (or CDM) approach are difficult to derive, whereas, the instant stiffness reduction approach (termed as the Material Property Degradation method, or MPDG) is very sensitive to mesh size and does not model the progression of damage within each element. Since MPDG is based on a ply-discount approach, the sudden loss of stiffness in a failed element leads to numerical instability in the model, making it difficult to achieve a converged solution. Furthermore, the failure strength of the model is also under-predicted due to sudden or instant element failure. For these reasons, a progressive damage model is proposed in this study that is based on a recursive stiffness reduction technique in order to study the failure behavior of woven composite micro-geometries. In this study, ANSYS Mechanical was employed for studying the progressive damage response of woven composites [76]. It has built-in commonly used damage initiation laws, some of which will be discussed later in this chapter, as well two built-in damage evolution laws that can be implemented by the user (MPDG, CDM). The progressive damage model is only used for the composite yarns, while for the isotropic matrix the conventional ply-discount approach is used to model damage evolution. This is because the reinforcing yarns are the main load-bearing components in a woven composite, while the function of the matrix is to bind the reinforcements together and transfer loads. The maximum strain failure criterion determines the onset of damage in the composite yarns, whereas the maximum stress criterion suffices for the matrix. A mesoscale homogenization is used for the yarns where each yarn element is represented as a unidirectional homogenous body with a local orientation. Thus, conventional failure criteria for unidirectional composites can be used to model the non-linear failure behavior of the yarns in the woven composite unit cell.

In this implementation, the yarns and the matrix are perfectly bonded and there are no voids in the woven composite unit cell. The following sections describe the constitutive material model, failure initiation criteria and the damage model implementation in greater detail.

4.1 Constitutive material model

The constitutive material models relate the state of strain to the state of stress. In point strain analysis, the stress state at a point can be readily computed by determining the point strains through kinematic relations. For a three-dimensional material model, which is of interest in this study, all the six stress and strain components are evaluated. This section describes the constitutive relations for the kinematic model. The engineering material constants in the subsequent equations are the linear elastic mechanical property values – longitudinal modulus (E_1), transverse moduli (E_2, E_3), shear moduli (G_{12}, G_{13}, G_{23}) and Poisson's ratios (ν_{12}, ν_{13}, ν_{23}). The 3D strain-stress relations for a linear-elastic orthotropic material implemented can be written as:

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{Bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & 0 & 0 & 0 \\ S_{21} & S_{22} & S_{23} & 0 & 0 & 0 \\ S_{31} & S_{32} & S_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & S_{66} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{Bmatrix} = [S]\{\sigma\} \tag{4.1}$$

As is evident from equation (4.1), the normal components are coupled to one another while the shear components are completely uncoupled. $[S]$ is the compliance matrix whose coefficients are defined in terms of the elastic engineering material constants according to equation (4.2).

$$
\begin{gathered}
S_{11}=\frac{1}{E_1}; \quad S_{22}=\frac{1}{E_2}; \quad S_{33}=\frac{1}{E_3} \\
S_{21}=S_{12}=-\frac{\nu_{12}}{E_1}=-\frac{\nu_{21}}{E_2} \\
S_{31}=S_{13}=-\frac{\nu_{13}}{E_1}=-\frac{\nu_{31}}{E_3} \\
S_{32}=S_{23}=-\frac{\nu_{23}}{E_2}=-\frac{\nu_{32}}{E_3} \\
S_{44}=\frac{1}{G_{23}}; \quad S_{55}=\frac{1}{G_{13}}; \quad S_{66}=\frac{1}{G_{12}}
\end{gathered}
\tag{4.2}
$$

In the above relations, the engineering shear strain $\gamma_{ij}=2\varepsilon_{ij}$ is used instead of the tensor shear strain ε_{ij} because the shear modulus G is defined by $\tau=G\gamma$ in mechanics of materials [64]. The stress-strain relations can thus be obtained directly from the compliance relations according to equation (4.3), where $[C]$ is the stiffness matrix.

$$
\{\sigma\}=\begin{Bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \tau_{23}\\ \tau_{13}\\ \tau_{12}\end{Bmatrix}=\begin{bmatrix}C_{11} & C_{12} & C_{13} & 0 & 0 & 0\\ C_{21} & C_{22} & C_{23} & 0 & 0 & 0\\ C_{31} & C_{32} & C_{33} & 0 & 0 & 0\\ 0 & 0 & 0 & C_{44} & 0 & 0\\ 0 & 0 & 0 & 0 & C_{55} & 0\\ 0 & 0 & 0 & 0 & 0 & C_{66}\end{bmatrix}\begin{Bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \gamma_{23}\\ \gamma_{13}\\ \gamma_{12}\end{Bmatrix}=[C]\{\varepsilon\}
\tag{4.3}
$$

The order of the stress and strain terms in the stiffness and compliance matrices is different for different FE codes. The stiffness components are also defined in terms of the elastic material constants and the non-zero terms are expressed according to equation (4.4).

$$C_{11} = \frac{1 - \nu_{23}\nu_{32}}{E_2 E_3 \Delta}; \;\; C_{22} = \frac{1 - \nu_{13}\nu_{31}}{E_1 E_3 \Delta}; \;\; C_{33} = \frac{1 - \nu_{12}\nu_{21}}{E_1 E_2 \Delta} = G_{12}$$

$$C_{12} = \frac{\nu_{21} + \nu_{31}\nu_{23}}{E_2 E_3 \Delta} = \frac{\nu_{12} + \nu_{32}\nu_{13}}{E_1 E_3 \Delta}$$

$$C_{13} = \frac{\nu_{31} + \nu_{21}\nu_{32}}{E_2 E_3 \Delta} = \frac{\nu_{13} + \nu_{12}\nu_{23}}{E_1 E_2 \Delta} \tag{4.4}$$

$$C_{23} = \frac{\nu_{32} + \nu_{12}\nu_{31}}{E_1 E_3 \Delta} = \frac{\nu_{23} + \nu_{21}\nu_{13}}{E_1 E_2 \Delta}$$

$$C_{44} = G_{23}; \;\; C_{55} = G_{13}; \;\;\; C_{66}$$

$$\Delta = \frac{1 - \nu_{12}\nu_{21} - \nu_{23}\nu_{32} - \nu_{31}\nu_{13} - 2\nu_{21}\nu_{32}\nu_{13}}{E_1 E_2 E_3}$$

The reciprocity relations for 3D elasticity are given as:

$$\begin{aligned} E_{11}\nu_{21} &= \nu_{12}E_{22} \\ E_{11}\nu_{31} &= \nu_{13}E_{33} \\ E_{22}\nu_{32} &= \nu_{23}E_{33} \end{aligned} \tag{4.5}$$

In the 3D elasticity approach, each of the Gaussian integration points within each 3D solid element is evaluated to determine the state of strain and to estimate the trial stresses.

4.2 Failure criteria and damage initiation laws

With the constitutive material model set up, the commonly implemented failure initiation criteria for progressive damage analysis are summarized here. The different failure criteria relate the material strength allowables, defined for uniaxial tension-compression and shear, to the general stress-strain state due to multiaxial loads and track the onset of material failure in elements. f is termed as the failure index which is a function of the stresses (or strains) and the material strength. Failure starts when $f \geq 1$. The inverse of failure index is termed as the strength ratio, $R = \frac{1}{f}$. Five different failure criteria are discussed briefly in the following sub-sections.

4.2.1 Maximum strain criterion

The ratios of the actual strains to the failure strains are compared in the element local coordinate system. The failure index or the failure criterion function is given as:

$$f = \max\left(\left|\frac{\varepsilon_{11}}{S_1}\right|, \left|\frac{\varepsilon_{22}}{S_2}\right|, \left|\frac{\varepsilon_{33}}{S_3}\right|, \left|\frac{\gamma_{23}}{S_4}\right|, \left|\frac{\gamma_{13}}{S_5}\right|, \left|\frac{\gamma_{12}}{S_6}\right|\right) \tag{4.6}$$

The strain limits are given as S_i, which have different values considering the tensile or compressive stress/strain state of the material point as:

$$\begin{aligned} \varepsilon_{11} \geq 0 \rightarrow S_1 = S_{1t}; \quad & \varepsilon_{11} < 0 \rightarrow S_1 = S_{1c} \\ \varepsilon_{22} \geq 0 \rightarrow S_2 = S_{2t}; \quad & \varepsilon_{22} < 0 \rightarrow S_2 = S_{2c} \\ \varepsilon_{33} \geq 0 \rightarrow S_3 = S_{3t}; \quad & \varepsilon_{33} < 0 \rightarrow S_3 = S_{3c} \end{aligned} \tag{4.7}$$

4.2.2 Maximum stress criterion

The ratios of the actual stresses to the failure stresses are compared in the element local coordinate system. The failure index or the failure criterion function is given as:

$$f = \max\left(\left|\frac{\sigma_{11}}{F_1}\right|, \left|\frac{\sigma_{22}}{F_2}\right|, \left|\frac{\sigma_{33}}{F_3}\right|, \left|\frac{\tau_{23}}{F_4}\right|, \left|\frac{\tau_{13}}{F_5}\right|, \left|\frac{\tau_{12}}{F_6}\right|\right) \tag{4.8}$$

The stress limits are given as F_i, which have different values considering the tensile or compressive stress/strain state of the material point as:

$$\begin{aligned} \sigma_{11} \geq 0 \rightarrow F_1 = F_{1t}; \quad & \sigma_{11} < 0 \rightarrow F_1 = F_{1c} \\ \sigma_{22} \geq 0 \rightarrow F_2 = F_{2t}; \quad & \sigma_{22} < 0 \rightarrow F_2 = F_{2c} \\ \sigma_{33} \geq 0 \rightarrow F_3 = F_{3t}; \quad & \sigma_{33} < 0 \rightarrow F_3 = F_{3c} \end{aligned} \tag{4.9}$$

4.2.3 Tsai-Wu failure criterion

The Tsai-Wu failure criterion is a quadratic failure criterion, where all the stress or strain components are combined into one expression [65]. The Tsai-Wu failure criterion in 3D represents a closed conical failure surface and is given by the expression:

$$
\begin{aligned}
f = {} & \frac{{\sigma_{11}}^2}{F_{1t}F_{1c}} + \frac{{\sigma_{22}}^2}{F_{2t}F_{2c}} + \frac{{\sigma_{33}}^2}{F_{3t}F_{3c}} + \frac{{\tau_{23}}^2}{{F_4}^2} + \frac{{\tau_{13}}^2}{{F_5}^2} + \frac{{\tau_{12}}^2}{{F_6}^2} + c_{XY}\frac{\sigma_{11}\sigma_{22}}{\sqrt{F_{1t}F_{1c}F_{2t}F_{2c}}} \\
& + c_{YZ}\frac{\sigma_{22}\sigma_{33}}{\sqrt{F_{2t}F_{2c}F_{3t}F_{3c}}} + c_{XZ}\frac{\sigma_{11}\sigma_{33}}{\sqrt{F_{1t}F_{1c}F_{3t}F_{3c}}} + \sigma_{11}\left(\frac{1}{F_{1t}} - \frac{1}{F_{1c}}\right) \\
& + \sigma_{22}\left(\frac{1}{F_{2t}} - \frac{1}{F_{2c}}\right) + \sigma_{33}\left(\frac{1}{F_{3t}} - \frac{1}{F_{3c}}\right)
\end{aligned}
\tag{4.10}
$$

The default values of the Tsai-Wu constants, c_{XY}, c_{YZ}, c_{XZ} are -1.

4.2.4 Hashin failure criterion

The Hashin failure criterion is used to predict failure in transversely-isotropic unidirectional composites [66]. In the 3D Hashin criterion, the critical tensile load in the fiber direction is determined by the failure index f_f as:

$$f_f = \left(\frac{\sigma_{11}}{F_{1t}}\right)^2 + \frac{1}{{F_5}^2}\left({\tau_{12}}^2 + {\tau_{13}}^2\right) \tag{4.11}$$

Under compressive loading in the fiber direction, failure is predicted by the independent stress condition:

$$f_f = \frac{\sigma_{11}}{F_{1c}} \tag{4.12}$$

The matrix failure due to transverse tension is predicted using the following expression:

$$f_m = \frac{1}{{F_{2t}}^2}\left({\sigma_{22}}^2 + {\sigma_{33}}^2\right) + \frac{1}{{F_4}^2}\left({\tau_{23}}^2 - \sigma_{22}\sigma_{33}\right) + \frac{1}{{F_5}^2}\left({\tau_{12}}^2 + {\tau_{13}}^2\right), \qquad \sigma_{22} + \sigma_{33} > 0 \tag{4.13}$$

The failure due to transverse compression is evaluated using the following expression:

$$f_m = \frac{1}{F_{2c}}\left[\left(\frac{F_{2c}}{2F_4}\right)^2 - 1\right](\sigma_{22} + \sigma_{33}) + \frac{1}{4{F_4}^2}({\sigma_{22}}^2 + {\sigma_{33}}^2) + \frac{1}{{F_4}^2}({\tau_{23}}^2 - \sigma_{22}\sigma_{33})$$
$$+ \frac{1}{{F_5}^2}({\tau_{12}}^2 + {\tau_{13}}^2), \qquad \sigma_{22} + \sigma_{33} < 0 \tag{4.14}$$

Additionally, delamination is predicted using the following expression:

$$f_d = \left(\frac{\sigma_{33}}{F_{3i}}\right)^2 + \left(\frac{\tau_{13}}{F_5}\right)^2 + \left(\frac{\tau_{23}}{F_4}\right)^2, \qquad i = c \; if \; \sigma_{33} < 0 \; else \; i = t \tag{4.15}$$

The most critical failure mode is then selected as the failure index as:

$$f = \max\left(f_f, f_m, f_d\right) \tag{4.16}$$

4.2.5 Puck failure criterion

The Puck failure criterion distinguishes between fiber failure (FF) and inter-fiber failure (IFF) [67]. IFF has two different modes: when transverse stress is positive, it is Mode A failure, when transverse stress is negative, it is Mode B failure. Mode A is when transverse cracks appear in the lamina under transverse tensile stress. Mode B denotes transverse cracks under in-plane shear stress and transverse compressive stress. The FF mode depends only on longitudinal tension or compression and thus uses the maximum stress criterion:

$$f_{FF} = \frac{\sigma_{11}}{F_{1t}}, \qquad \sigma_{11} > 0$$
$$f_{FF} = \frac{\sigma_{11}}{F_{1c}}, \qquad \sigma_{11} < 0 \tag{4.17}$$

The IFF criterion is based on the Mohr-Coulomb theory and assumes transverse isotropy of the unidirectional composites. Failure occurs only on the fracture plane where shear stresses

τ_{nt}, τ_{n1} and normal stress σ_n exist, as shown in Figure 4.1. These three stresses can be calculated according to equation (4.18).

$$\begin{gathered}\sigma_n = \sigma_{22}\cos^2\theta + \sigma_{33}\sin^2\theta + 2\tau_{23}\sin\theta\cos\theta \\ \tau_{nt} = (\sigma_{33} - \sigma_{22})\sin\theta\cos\theta + \tau_{23}(\cos^2\theta - \sin^2\theta) \\ \tau_{n1} = \tau_{31}\sin\theta + \tau_{21}\cos\theta\end{gathered} \tag{4.18}$$

Puck's IFF criterion distinguishes failure mechanisms based on tensile and compressive normal stresses. For a given combination of stress-state and strength value, the fracture angle θ is a variable. The problem is thus converted to finding the fracture angle which makes f_{IFF} maximum.

$$\begin{gathered}f_{IFF} = \sqrt{(1 - A^t)^2{\sigma_n}^2 + \left(\frac{\tau_{nt}}{R_2}\right)^2 + \left(\frac{\tau_{n1}}{R_1}\right)^2} + A^t\sigma_n, \qquad \sigma_n \geq 0 \\ f_{IFF} = \sqrt{(A^c)^2{\sigma_n}^2 + \left(\frac{\tau_{nt}}{R_2}\right)^2 + \left(\frac{\tau_{n1}}{R_1}\right)^2} + A^c\sigma_n, \qquad \sigma_n < 0 \\ A^{(t,c)} = \frac{p_{2(t,c)}}{R_2}\cos^2\varphi + \frac{p_{1(t,c)}}{R_1}\sin^2\varphi \\ \cos^2\varphi = \frac{{\tau_{nt}}^2}{{\tau_{nt}}^2 + {\tau_{n1}}^2} \\ R_2 = \frac{F_{2c}}{2(1 + p_{2c})}, \qquad R_1 = F_6\end{gathered} \tag{4.19}$$

The default values for the coefficients are available for carbon and glass fiber plies as:

$$\begin{gathered}\text{Carbon: } p_{1t} = 0.35, p_{1c} = 0.3, p_{2t} = 0.25, p_{2c} = 0.2 \\ \text{Glass: } p_{1t} = 0.3, p_{1c} = 0.25, p_{2t} = 0.2, p_{2c} = 0.2\end{gathered} \tag{4.20}$$

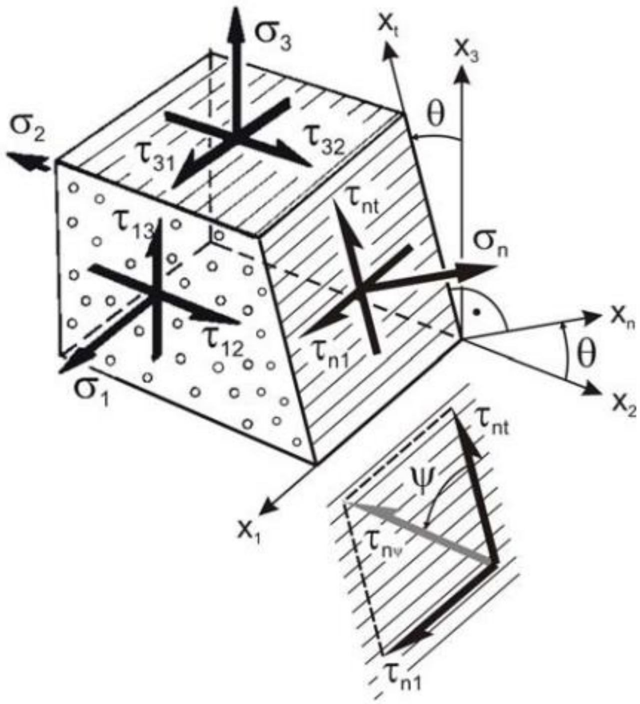
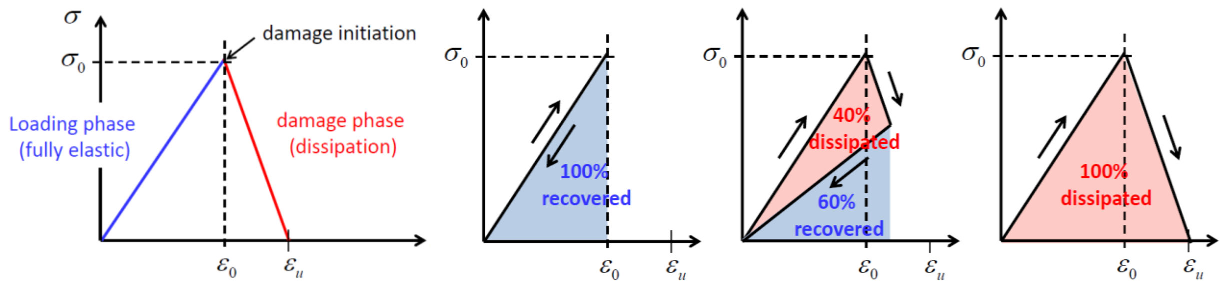


Figure 4.1: Stress components on fracture plane [68]

In this study, the maximum strain criterion is used for modeling damage initiation in the composite yarns and the maximum stress criterion is used for the isotropic matrix.

4.3 Damage modeling of fiber-reinforced composites

The start of damage in fiber-reinforced composites is governed by damage initiation criteria as described in the previous section. A composite laminate undergoes a damage process in which it loses its load carrying capacity gradually – as in the case of Continuum Damage Mechanics (CDM) or the recursive stiffness reduction models. In the elastic loading phase, the entire internal energy is recovered upon unloading. The damage phase however is dissipative in nature. 100% of the internal energy of the material can still be recovered upon unloading at the instant of damage initiation. This percentage gradually decreases to 0% as the material is strained further. This is illustrated in Figure 4.2.



(a) Damage process (b) Recoverable energy at loading phases

Figure 4.2: Damage model [69]

Material damage is then represented by using a scalar damage variable d. Consider the reduction of effective load carrying cross-section due to fiber breakage in Figure 4.3. The axial stress neglecting voids is represented as:

$$\sigma = \frac{F}{A} \tag{4.21}$$

If d denotes the volume fraction of failed fibers at a cross-section, the effective cross-section area is only $(1-d)A$. Thus the local stresses are higher and are given as:

$$\sigma_{local} = \frac{F}{(1-d)A} = \frac{\sigma}{(1-d)} = E\varepsilon \tag{4.22}$$

Thus, the macroscopic stress-strain relationship in terms of a "damaged modulus, E_d" is:

$$\sigma = (1-d)E\varepsilon, \qquad E_d = (1-d)E \tag{4.23}$$

Thus, it can be concluded that damage is associated with a reduction in stiffness.

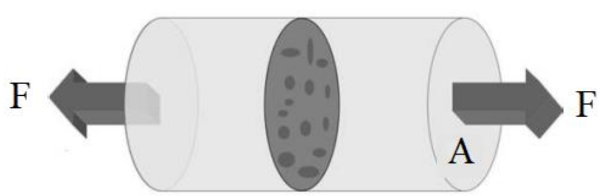


Figure 4.3: Axial stress

Refer again to Figure 4.2. Assuming a linearly decreasing relationship in the damage phase, the following relationship can be obtained:

$$\sigma = \sigma_0 \left(1 - \frac{\varepsilon - \varepsilon_0}{\varepsilon_u - \varepsilon_0}\right) \Rightarrow E_d = \frac{\sigma}{\varepsilon} = E \frac{\varepsilon_0}{\varepsilon} \frac{\varepsilon_u - \varepsilon}{\varepsilon_u - \varepsilon_0} \tag{4.24}$$

Thus, the damage variable d can be determined as:

$$d = 1 - \frac{E_d}{E} = \frac{\varepsilon_u}{\varepsilon} \frac{\varepsilon - \varepsilon_0}{\varepsilon_u - \varepsilon_0} \tag{4.25}$$

Next, the two damage evolution laws namely, the instant stiffness reduction method and the continuum damage mechanics method are briefly discussed.

4.3.1 Instant stiffness reduction

The instant stiffness reduction method is also called the ply-discount method or the material property degradation method. In this approach, material stiffness is instantly reduced, just once, based on the damage variables d. It requires four different damage variables, also called stiffness reduction factors as input – tensile fiber stiffness reduction factor (${d_f}^+$), compressive fiber stiffness reduction factor (${d_f}^-$), tensile matrix stiffness reduction factor (${d_m}^+$), compressive matrix stiffness reduction factor (${d_m}^-$). It should be noted that "fiber tension", "fiber compression", "matrix tension" and "matrix compression" in terms of the damage factors are misnomers used in FE codes and literature. These actually refer to the directions of the fiber-reinforced composite failure – fiber tension and compression represent *longitudinal tension* and *longitudinal compression*, matrix tension represents *transverse tension and in-plane shear*, matrix compression represents *transverse compression* [70]. Fiber tension and compression represent fiber dominated failure modes, while matrix tension and compression represent matrix dominated failure modes. The allowable values for the four damage factors are between 0 and 1 where 0 represents no reduction in material stiffness in the affected mode after damage initiation and 1 represents a complete loss of stiffness in the affected mode.

4.3.2 Continuum damage mechanics

In this method, the damage variables increase gradually based on the amount of dissipated energy for the various damage modes. It requires eight input parameters – energy dissipated per unit area from tensile fiber damage (${G_{fC}}^{+}$), viscous damping coefficient for tensile fiber damage (${\eta_f}^{+}$), energy dissipated per unit area from compressive fiber damage (${G_{fC}}^{-}$), viscous damping coefficient for compressive fiber damage (${\eta_f}^{-}$), energy dissipated per unit area from tensile matrix damage or transverse tensile damage (${G_{mC}}^{+}$), viscous damping coefficient for tensile matrix damage or transverse tensile damage (${\eta_m}^{+}$), energy dissipated per unit area from compressive matrix damage or transverse compressive damage (${G_{mC}}^{-}$), viscous damping coefficient for compressive matrix damage or transverse compressive damage (${\eta_m}^{-}$). The energy dissipated per unit area, or G_C, is specified individually for all damage modes. For a specific damage mode, G_C is given in terms of equivalent stress and displacement according to equation (4.26).

$$G_C = \int_0^{{u_e}^f} \sigma_e du_e \tag{4.26}$$

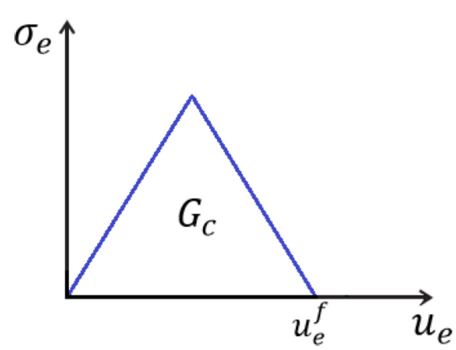


Figure 4.4: Equivalent stress vs. equivalent displacement

Four damage modes – fiber tension (fiber rupture), fiber compression (buckling), matrix tension (cracking) and matrix compression (crushing) – are accounted for. Four damage variables ($d_f{}^{+}$, $d_f{}^{-}$, $d_m{}^{+}$, $d_m{}^{-}$) are used to measure damage, one for each mode respectively. The damage variables are calculated according to equation (4.25). The three major damage variables used for calculating the damaged stiffness matrix are determined as follows:

$$
\begin{gathered}
d_f = \begin{cases} d_f{}^{+}, & \textit{fiber tension failure} \\ d_f{}^{-}, & \textit{fiber compression failure} \end{cases} \\
d_m = \begin{cases} d_m{}^{+}, & \textit{matrix tension failure} \\ d_m{}^{-}, & \textit{matrix compression failure} \end{cases} \\
d_s = 1 - \left(1 - d_f{}^{+}\right)\left(1 - d_f{}^{-}\right)\left(1 - d_m{}^{+}\right)\left(1 - d_m{}^{-}\right), \qquad \textit{shear failure}
\end{gathered}
\tag{4.27}
$$

The damaged stiffness matrix for a general orthotropic material is thus defined in CDM as well as MPDG according to equation (4.28).

$$
[D]_d = \begin{bmatrix}
\dfrac{C_{11}}{(1-d_f)} & C_{12} & C_{13} & 0 & 0 & 0 \\
C_{21} & \dfrac{C_{22}}{(1-d_m)} & C_{23} & 0 & 0 & 0 \\
C_{31} & C_{32} & \dfrac{C_{33}}{(1-d_m)} & 0 & 0 & 0 \\
0 & 0 & 0 & \dfrac{C_{44}}{(1-d_s)} & 0 & 0 \\
0 & 0 & 0 & 0 & \dfrac{C_{55}}{(1-d_s)} & 0 \\
0 & 0 & 0 & 0 & 0 & \dfrac{C_{66}}{(1-d_s)}
\end{bmatrix}^{-1}
\tag{4.28}
$$

4.4 Recursive stiffness reduction model in USERMAT

The unit cell domain consists of the reinforcing yarns and the surrounding matrix. The matrix material is isotropic as previously stated. Damage initiation in the matrix is governed by the maximum stress criterion and damage evolution is governed by the instant stiffness reduction approach. In the case of tensile failure, the stiffness is reduced by 99%, while in the case of compressive failure the stiffness is reduced by 80%. These values align well with the fact that matrix crushing under compression has a higher residual stiffness than matrix cracking under tension due to the ability of the polymeric matrix to undergo compressive loads despite initiation of fracture. However, for the orthotropic yarns, the material can fail due to a combination of tensile, compressive and shear failure modes in the local material coordinates. Hence damage evolution in the yarns is modeled using a recursive stiffness reduction technique implemented through the user-defined material subroutine, USERMAT [71] (Appendix B -).

While the instantaneous or single-step stiffness reduction approach degrades the material stiffness coefficients only once by the prescribed degradation factors, the recursive stiffness reduction approach degrades the material stiffness coefficients in a gradual manner and thus avoids numerical convergence issues associated with the sudden local change in material stiffness. Specifying a high value to stiffness reduction factors (close to 1) is similar to implementing the instant stiffness reduction approach with the same degradation factors. The recursive stiffness reduction approach in this study degrades the respective coefficients of the stiffness matrix directly rather than degrading the engineering properties individually. Thus the issue of maintaining reciprocity relations for degraded engineering constants is avoided.

Within user material subroutine, the user is given various results from the solution procedure that is passed to the subroutine as arguments and may be used to determine new trial values for the stresses and stiffness coefficients based on the current deformation state. It should be noted here that the USERMAT subroutine actually defines a trial stress state for the end of the solution increment and thus, damage can evolve as the nonlinear solution procedure iterates to find the solution for the current increment. These new trial values are returned to the FE solver for additional computations.

In addition to material property data and user state variables, total strains from the previous solution increment i and current strain increment for a given material point are also passed as input arguments from the solver to the USERMAT subroutine. Thus, at each solution step, the current total strain is first calculated using the strain increment for that step. So at the $i+1^{\text{th}}$ step, the current total strain is given as:

$$\{\varepsilon\}^{i+1} = \{\varepsilon\}^{i} + \{\Delta\varepsilon\}^{i} \tag{4.29}$$

The calculated current total strain is used to determine the current total stress using equation (4.30). $[C]^i$ in the equation refers to the stiffness matrix from the previous converged solution step. The stiffness matrix coefficients are initially calculated as functions of the material properties using equation (4.4), and are updated as damage occurs.

$$\{\sigma\}^{i+1} = [C]^{i}\{\varepsilon\}^{i+1} \tag{4.30}$$

This stress is then used to determine the strength ratios R. If the strength ratio(s) is less than one, failure occurs at the material point in consideration. The damage factor(s) corresponding to the failure modes are updated according to equation (4.31) once failure is detected. k_m represents the amount by which the stiffness matrix coefficients are reduced, under a stress state m corresponding to tension (T), compression (C), shear (S). k_m are set to 1-5% in the current implementation of the non-linear material model and can assume different values for tension, compression and shear failure types.

$$\left(D_j\right)^{i+1} = \left(D_j\right)^{i} - k_m, \;\; j = 1,..,9, \;\; m = T, C, S \tag{4.31}$$

There are nine different failure modes considered in this implementation:

1. 1(T): Longitudinal in-plane tensile failure
2. 1(C): Longitudinal in-plane compressive failure
3. 2(T): Transverse in-plane tensile failure
4. 2(C): Transverse in-plane compressive failure
5. 3(T): Transverse out-of-plane tensile failure
6. 3(C): Transverse out-of-plane compressive failure
7. 4: Shear failure 2-3 plane
8. 5: Shear failure 1-3 plane
9. 6: Shear failure 1-2 plane

The failure modes and identifiers are listed in Table 4.1. The stiffness matrix coefficients are then updated based on the updated damage factors as shown in the table. Off-diagonal terms in the stiffness matrix along the same row and column are also degraded in the same manner as the diagonal terms. This updated stiffness matrix ($[C]^{i+1}$) is used to re-compute the current stress. It is to be noted here, that once a failure mode is detected at a material point, recursive stiffness reduction for that particular failure mode continues to be applied at that point in the subsequent steps. The material point can additionally fail due to other modes in the subsequent steps. The damage factors are initially set to 1 (no damage) at the start of solution and are updated until it reaches a specified minimum value (5% to 20% of initial value), after which they are held constant.

The current implementation of the recursive damage model in USERMAT subroutine requires 22 input parameters for defining material properties and analysis options that are read into the property array `prop`. The variable `PDA_Criteria` sets the failure initiation criteria for the solution to maximum strain (`PDA_Criteria` = 1) or maximum stress (`PDA_Criteria` = 2). Other failure criteria can be included in the subroutine implementation in the future. The strain limits are calculated from the stress limits inside the subroutine. Outputs from this subroutine at every solution increment are the stress vector (`stress`) and the material stiffness matrix at each material point (`dsdePl`), along with a set of user state variables (`ustatev`), which keep track of the damage factors and failure flags corresponding to the different failure modes. `ustatev(1)` – `ustatev(9)` store and update the damage factor values for the nine failure modes and `ustatev(10)` – `ustatev(18)` store the failure flags for each of the nine modes. It also stores the overall failure initiation flag (`ustatev(19)`) which assumes a value of either zero if no failure has occurred, or, one if at least one element has started to fail. All failure flags are initially set to zero. The input and output variables are listed in Table 4.2. The output variables are passed back into the FE solver. The overall flowchart of this process is shown in Figure 4.5.

Table 4.1: Damage evolution law

Failure index	Damage factors for terms of constitutive matrix: Maximum stress/strain failure criteria								
	C_{11}	C_{12}	C_{13}	C_{22}	C_{23}	C_{33}	C_{44}	C_{55}	C_{66}
1(T): 1	$D_{1T}C_{11}$	$D_{2T}C_{12}$	$D_{3T}C_{13}$	–	–	–	–	–	–
1(C): 2	$D_{1C}C_{11}$	$D_{2C}C_{12}$	$D_{2C}C_{13}$	–	–	–	–	–	–
2(T): 3	–	$D_{2T}C_{12}$	–	$D_{4T}C_{22}$	$D_{5T}C_{23}$	–	–	–	–
2(C): 4	–	$D_{2C}C_{12}$	–	$D_{4C}C_{22}$	$D_{5C}C_{23}$	–	–	–	–
3(T): 5	–	–	$D_{3T}C_{13}$	–	$D_{5T}C_{23}$	$D_{6T}C_{33}$	–	–	–
3(C): 6	–	–	$D_{3C}C_{13}$	–	$D_{5C}C_{23}$	$D_{6C}C_{33}$	–	–	–
4: 7	–	–	–	–	–	–	$D_7 C_{44}$	–	–
5: 8	–	–	–	–	–	–	–	$D_8 C_{55}$	–
6: 9	–	–	–	–	–	–	–	–	$D_9 C_{66}$

Table 4.2:USERMAT input and output variables

Input variables	
`prop` array index: variable	Description
1: E1	Longitudinal modulus in 1-direction
2: E2	Transverse modulus in 2-direction
3: E3	Transverse modulus in 3-direction
4: nu23	Poisson's ratio, $\nu 23$
5: nu13	Poisson's ratio, $\nu 13$
6: nu12	Poisson's ratio, $\nu 12$
7: G23	Shear modulus, G23
8: G13	Shear modulus, G13
9: G12	Shear modulus, G12
10: F1t	Ultimate tensile strength in 1-direction
11: F2t	Ultimate tensile strength in 2-direction
12: F3t	Ultimate tensile strength in 3-direction
13: F1c	Ultimate compressive strength in 1-direction
14: F2c	Ultimate compressive strength in 2-direction
15: F3c	Ultimate compressive strength in 2-direction
16: F23	Ultimate shear strength in-plane
17: F13	Ultimate shear strength out-of-plane
18: F12	Ultimate shear strength out-of-plane
19: kt	Stiffness coefficient reduction factor, tension
20: kc	Stiffness coefficient reduction factor, compression
21: ks	Stiffness coefficient reduction factor, shear
22: `PDA_Criteria`	Damage initiation criteria
Output variables	
Array/matrix (dimension)	Description
`ustatev(19)`	Damage factors and subsequent failure flags
`stress(6)`	Stress components for the material point
`dsdePl(6,6)`	Material stiffness matrix

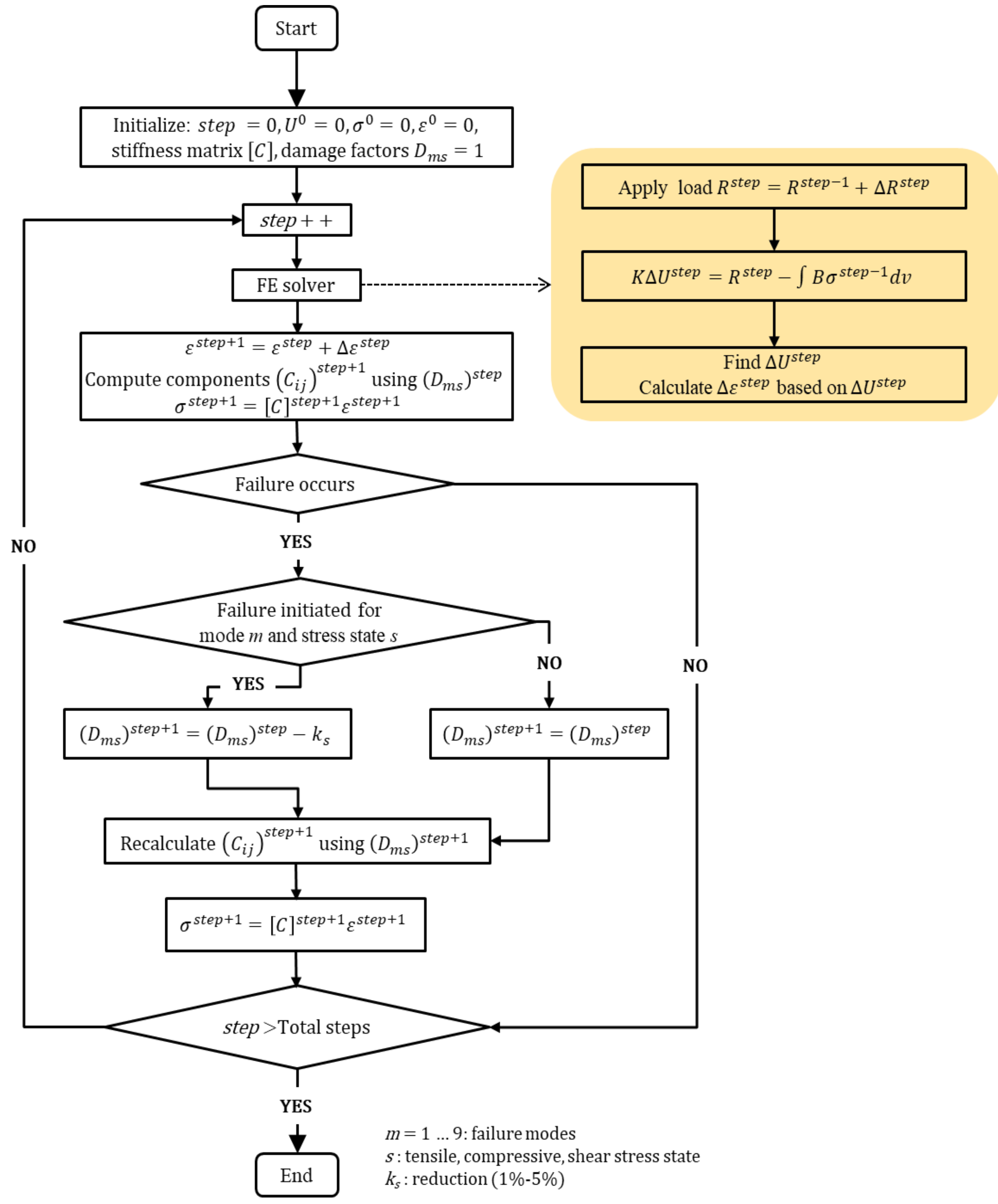


Figure 4.5: Solution process flowchart

4.5 Solution procedure

The stress-strain response is divided into three stages as shown in Figure 4.6. In Stage I, a single step is used to reach the point A in the stress-strain curve. Point A is determined after a linear elastic analysis, as the fraction of load after which the local material stress within an element exceeds its strength. Stage II is divided into *m* sub-steps and represents the response between the load points u_A and u_B. This is the initial non-linear stage where transverse and shear failure occur and thus the material stiffness reduces. Point B represents the start of catastrophic failure and the solution ends at Point C. Stage III, which represents the loading stage from B to C is divided into *n* sub-steps ($n >> m$) and represents the stage of fiber rupture failure where the yarns lose their load-carrying capacity. This is done to reduce solution time and improve convergence after failure initiation.

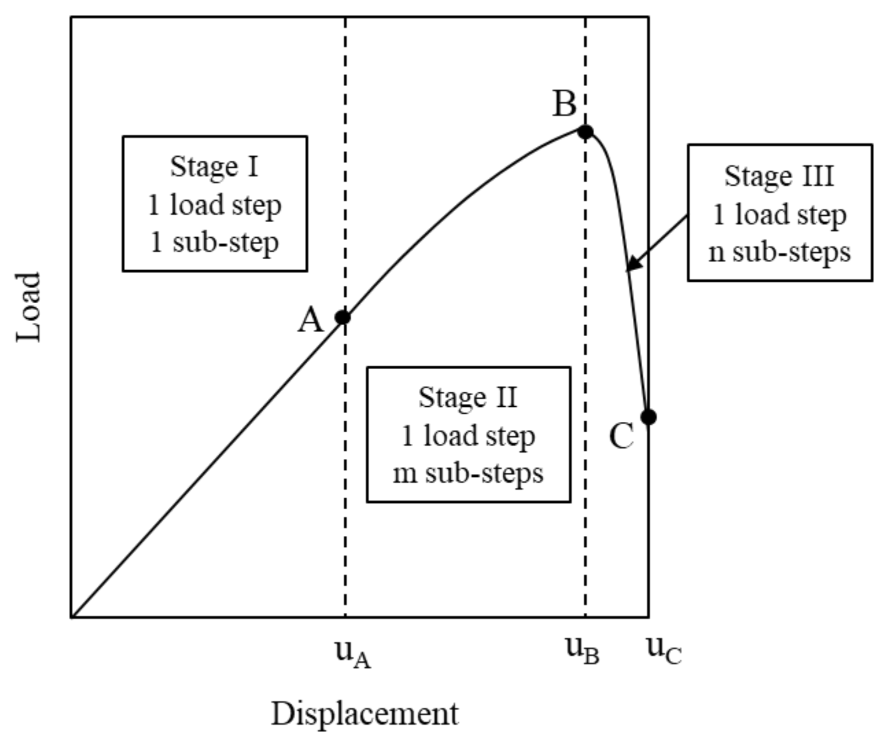


Figure 4.6: Loading stages and time-stepping

4.6 Results and discussion

In this section, the non-linear behavior of two different unit cells are analyzed and compared with existing experimental data in literature. Prior to that, the recursive stiffness reduction model is tested with a single element.

4.6.1 Single element test

The material model is compatible with 3D solid elements (8-node hex and 20-node hex elements). The current implementation uses SOLID185 elements. In its simplest form, the single 8-node element is tested under uniaxial load in the X-direction (fiber direction) considering it as an orthotropic homogenized unidirectional composite. The ultimate stress at failure of the material obtained from Chamis' micromechanics model was 2531 MPa and the model captures it very accurately at 2530 MPa as shown in Figure 4.7. The loading stages can be discerned from the distribution of the sub-steps in the different loading phases as described in the previous section.

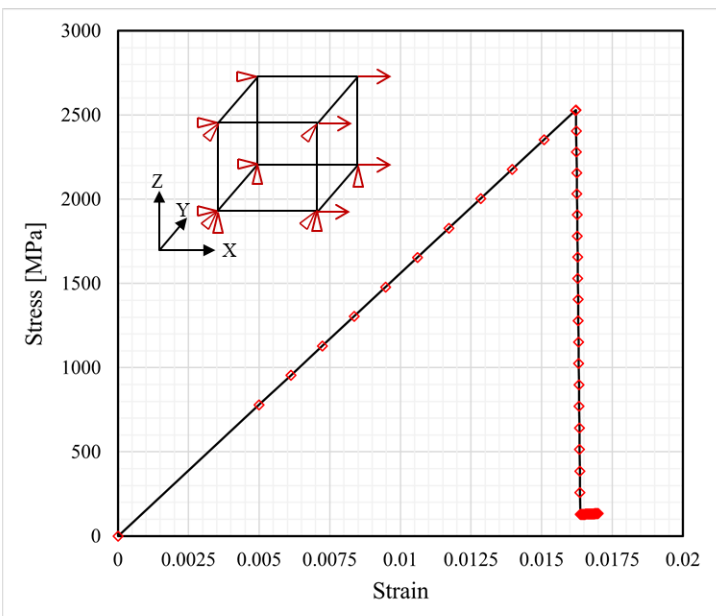
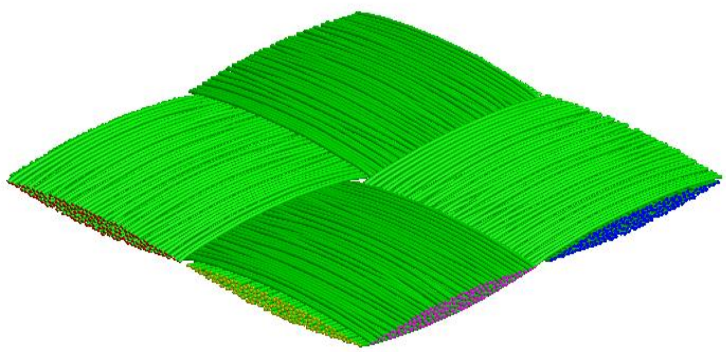


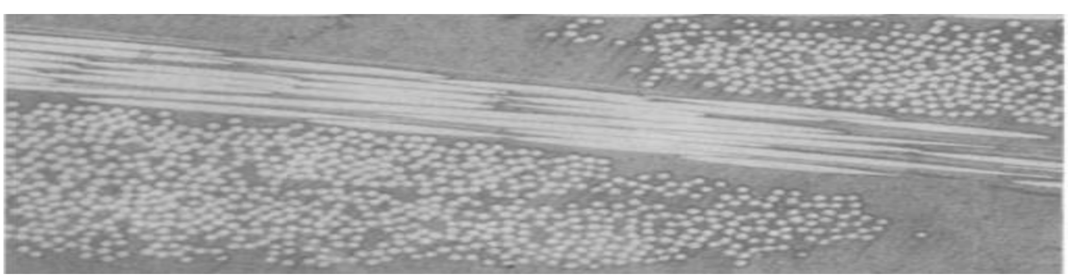
Figure 4.7: Single element stress test using recursive damage model

4.6.2 Plain woven unit cell model

The fabric unit cell model for a plain woven laminate is analyzed subjected to a uniaxial tensile load in the warp direction. The unit cell micro-geometry generated in DFCA is shown in Figure 4.8(a). The experimental and numerical results reported by Blacketter *et al*. [28] are used for comparison. The boundary conditions for a laminate as outlined in Chapter 3 - are employed. The aspect ratio of the yarns is defined by the yarn cross-sectional height to width ratio. An aspect ratio of approximately 1:10 is used which is similar to the experimental samples reported in literature that were measured using photomicrographs. Figure 4.8(b) shows the photomicrograph cross-section of the unit cell.

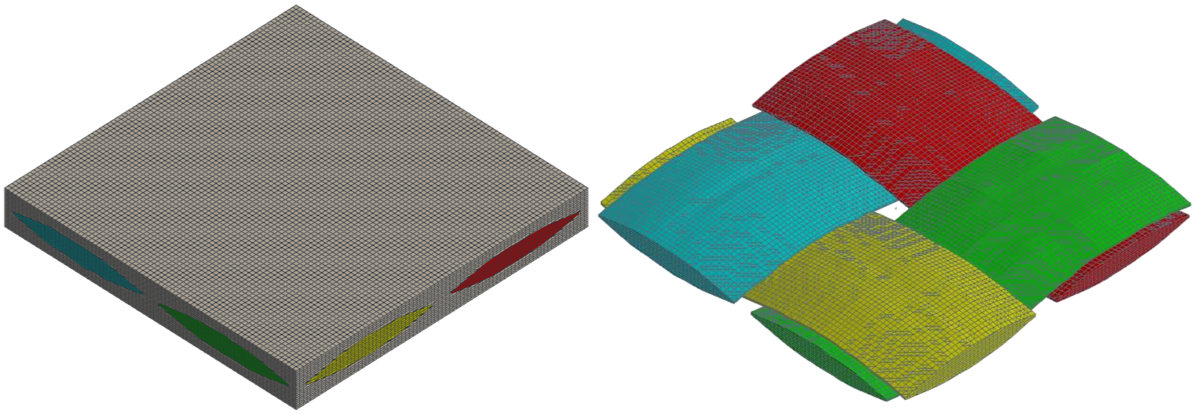


(a) DFCA unit cell fiber-level micro-geometry



(b) Cross-sectional view of plain woven fabric-reinforced composite [28]

Figure 4.8: Plain woven unit cell micro-geometry representation



(a) Plain woven unit cell meshed model (b) Plain woven unit cell yarn mesh

Figure 4.9: Plain woven unit cell finite element mesh

The final finite element mesh comprised of 121,632 nodes and 171,650 elements as shown in Figure 4.9. The overall fiber volume fraction of the fabric unit cell reported in literature was 60%. A slightly lower value was obtained for the similar unit cell model using DFCA. The higher value in the experimental sample was most likely due to the vacuum bagging process during layup and manufacturing, while for the micro-geometry in DFCA, the micro-geometry is generated based on the internal force relaxation procedure. Since the actual cross-sectional area of the experimental sample was not reported in the literature, a corrected cross-sectional area, A^e, was calculated to determine the nominal stress σ_x, according to equation (**4.32**). A is the nominal area and f_v is the overall fiber volume fraction from DFCA, whereas A^e and ${f_v}^e$are those from the experimental data. Table 4.3 lists the material properties of the fiber and the matrix used in this analysis.

$$A v_f = A^e {v_f}^e \tag{4.32}$$

Table 4.3: Material properties of the fiber and matrix

	Elastic properties (GPa)			Stress limits (MPa)	
	AS4 fiber	3501-6 epoxy		AS4 fiber	3501-6 epoxy
E_1	221	4.4	F_{1t}	3585	159
$E_2 = E_3$	13.8	4.4	$F_{2t} = F_{3t}$	400	159
$G_{12} = G_{13}$	13.8	1.7	F_{1c}	3585	159
G_{23}	5.5	1.7	$F_{2c} = F_{3c}$	400	159
$v_{12} = v_{13}$	0.2	0.34	$F_5 = F_6$	1793	110
v_{23}	0.25	0.34	F_4	200	110

The nominal stress σ_x is obtained by dividing the sum of the forces in the loading direction by the corrected nominal cross-section area A^e. The loading direction is the x-direction in this case. The fiber volume fraction in yarns was reported in literature as 70%. This value was used to calculate the composite yarn properties using the Chamis' micromechanics equations. The nominal stress vs strain response of the woven composite for the current model is compared with numerical and experimental results [28] as shown in Figure 4.10. As seen, the DFCA unit cell model has an excellent agreement with the test model for initial stiffness. The stiffness reported in the experiments was 64 GPa and the predicted stiffness from the DFCA FE model is 63.6 GPa. The DFCA FE model used in conjunction with the recursive stiffness reduction model also matches the experimental data very well in predicting the failure strength and is more accurate than the numerical simulation reported by Blacketter *et al*. The experimental result predicts a failure strength of 758 MPa, whereas the current implementation using the recursive stiffness reduction damage model predicts failure at 749 MPa, which is within a 1.2% error margin. The predicted strain at failure is 1.19% compared with 1.18% for the experimental sample.

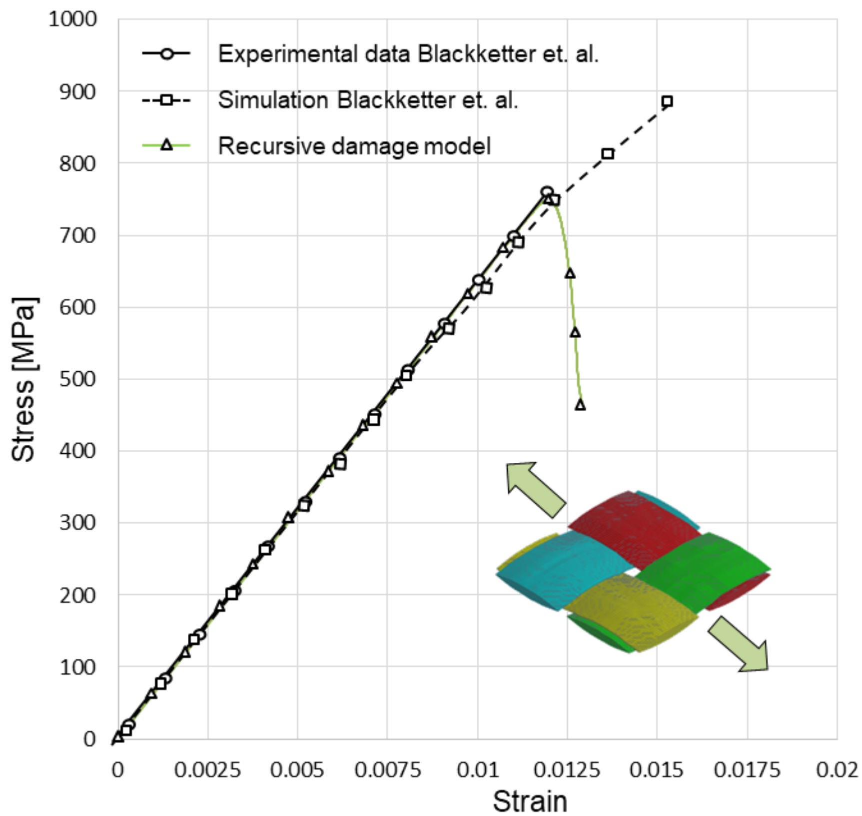
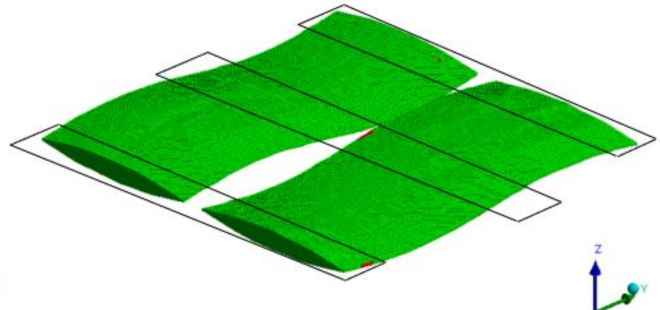


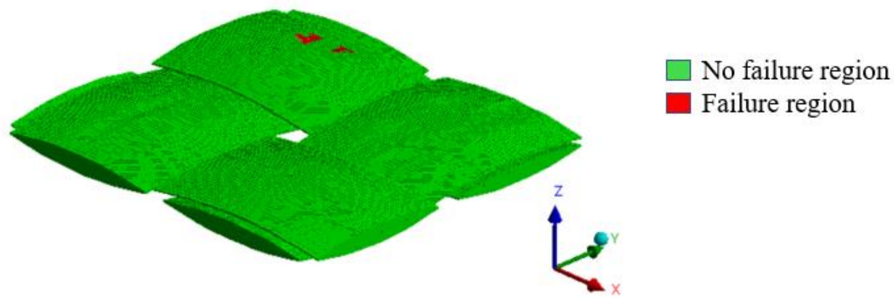
Figure 4.10: Stress-strain response of plain woven laminate under tension

Blackketter *et. al*. observed in their experiments that some fibers fail prior to catastrophic failure, transferring loads to nearby fibers. Transverse failure of yarns in test samples preceded catastrophic longitudinal failure due to fiber rupture, which was found to occur at points of maximum yarn curvature. This phenomenon is captured very well in the present model, as illustrated in Figure **4.12**, which shows the distribution of numbers of failed Gauss points in the model vs the applied strain for the three major failure modes. Transverse yarn failure due to in-plane shear emerges at approximately 1% strain and remains the dominant failure mode until strain reaches approximately 1.12%, as shown in Figure **4.12**(a). After that point, transverse and longitudinal tension failures occur. Figure 4.12(b) shows the progression of damage due to the

(a) Longitudinal tensile failure

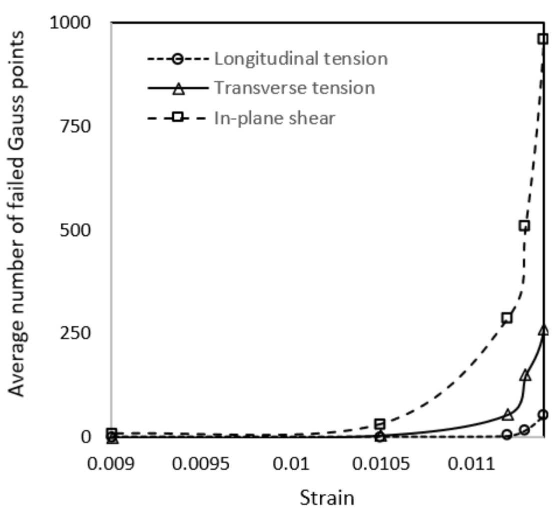


(b) Transverse tensile failure

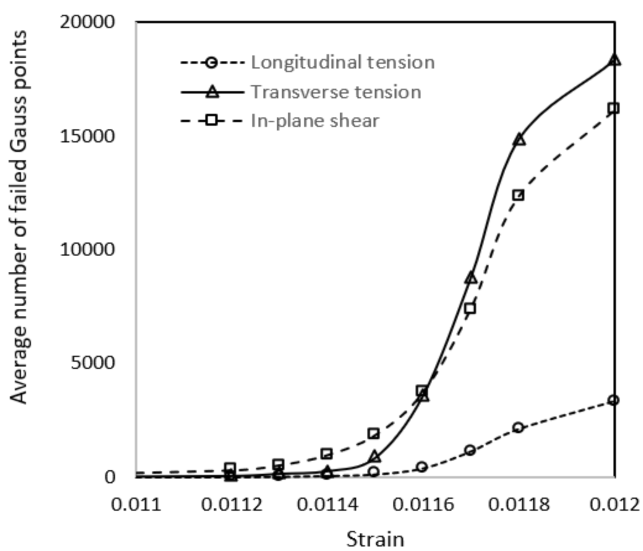


(c) In-plane shear failure

Figure 4.11: Failure initiation locations



(a) Damage initiation



(b) Damage progression

Figure 4.12: Distribution of failure occurrence as a function of the applied strain in the 2D woven model

three failure modes post failure initiation. Figure **4.11** shows the failure initiation locations corresponding to the three failure modes. Both longitudinal tension failure and in-plane shear failure initiate on the top and bottom surfaces of yarn, where fiber orientations are either in the x- or y-directions, and fiber curvatures reach minimum, as shown in Figure **4**.**11**(a) and Figure **4**.**11**(c). As seen in Figure 4.11(b), yarn transverse failures (weft yarns in this case) commence at points of maximum yarn curvature (marked by the rectangular zones), which is validated by the experimental observations.

4.6.3 3D orthogonal woven unit cell model

Next, a 3D orthogonal woven fabric unit cell is analyzed. The unit cell fiber-level micro-geometry generated using DFCA is shown in Figure 4.13. A qualitative comparison of the DFCA generated geometry and CT data of the 3D woven unit cell cross sections is shown in Figure 4.14. The DFCA data is marked by the red lines and the two data sources are overlaid for the selected cross-sections. The overall agreement between the DFCA model and the CT scan data is very good considering the yarn locations, cross-section shapes, aspect ratios and yarn crimp. A unidirectional tensile load is applied in the warp direction. Test data designated as "SN006 RTA (Room Temperature Ambient) - AP (as-processed)" from Farrokh *et. al*. [72] are compared with the numerical results derived using the proposed recursive stiffness reduction damage model. The unit cell model in the test samples is 19.1 mm long, 8.5 mm wide and 3.175 mm thick. The reported overall fiber volume fraction of the unit cell is 52.6%. The test sample comprises of AS4-12k fibers impregnated with RTM6 epoxy resin. Material properties of the constituents are listed in Table 4.4. The cross-section area for the AS4-12k yarns is available from reference [73] which is used to generate the unit cell micro-geometry and FE mesh using DFCA. The overall fiber volume fraction in the numerical model is 51.6%, which is very close to the test samples.

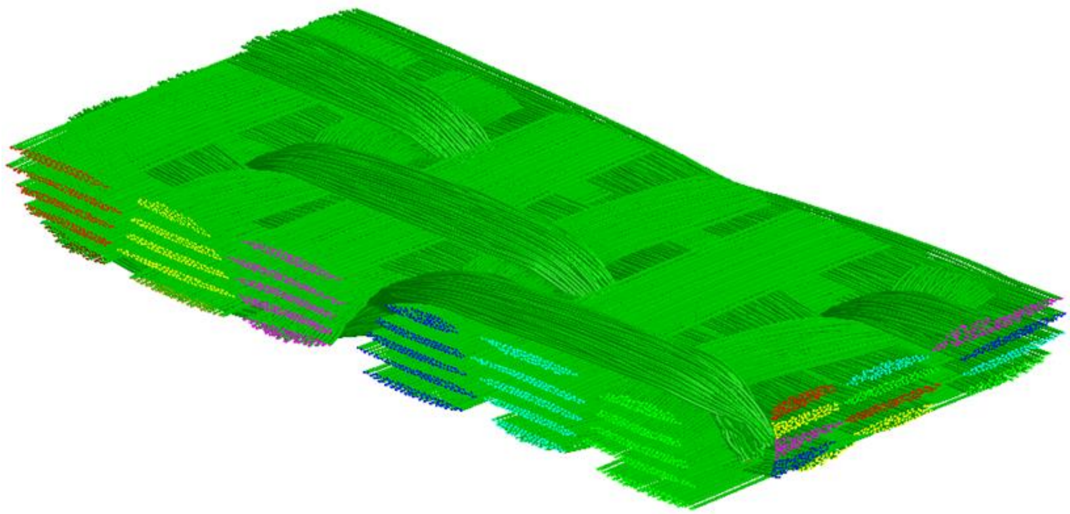
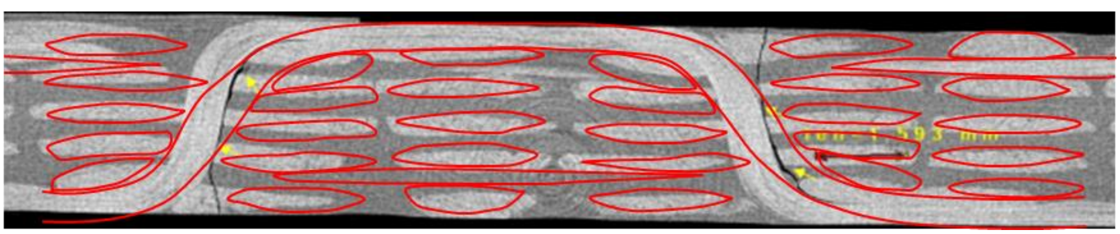
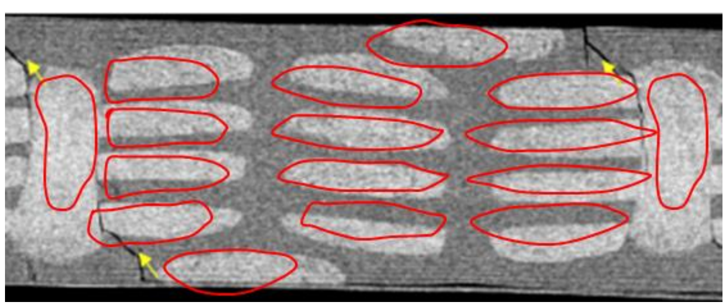


Figure 4.13: DFCA unit cell fiber-level micro-geometry



(a) Normal to weft yarns



(b) Normal to warp and binder yarns

Figure 4.14: Overlay of DFCA micro-geometry and CT data

Table 4.4: Material properties of the fiber and matrix

	Elastic properties (GPa)			Stress limits (MPa)	
	AS4 fiber	RTM6 epoxy		AS4 fiber	RTM6 epoxy
E_1	231	2.89	F_{1t}	3585	76
$E_2 = E_3$	15.0	2.89	$F_{2t} = F_{3t}$	400	76
$G_{12} = G_{13}$	15.8	1.07	F_{1c}	3585	110
G_{23}	5.8	1.07	$F_{2c} = F_{3c}$	400	110
$v_{12} = v_{13}$	0.21	0.35	$F_5 = F_6$	1793	49
v_{23}	0.30	0.35	F_4	200	49

The conformal finite element mesh of the unit cell generated using DFCA consists of 461,148 nodes and 858,363 elements as shown in Figure 4.15. Periodic boundary conditions as specified before were used. The available test data for comparison only includes uniaxial tensile test results in the warp direction. Figure 4.16 shows the comparison of the nominal stress-strain curves of the unit cell derived from the tensile tests [72], from the current progressive damage model, and from the instant stiffness reduction (or material property degradation, MPDG) model. As shown in the figure, the predicted warp direction stiffness (52.1 GPa) is in very good agreement with the test results. The average value of the two test results is 51.1 GPa. The predicted value of the failure strength from the recursive stiffness reduction model (548 MPa) is slightly lower than that of the test data average (626 MPa). As expected, the MPDG model predicts failure at a much lower strength because the failure is sudden and no damage progression within an element is considered. The current model shows some non-linearity in the curve at 0.54% strain, but the test data is almost perfectly linear up to failure.

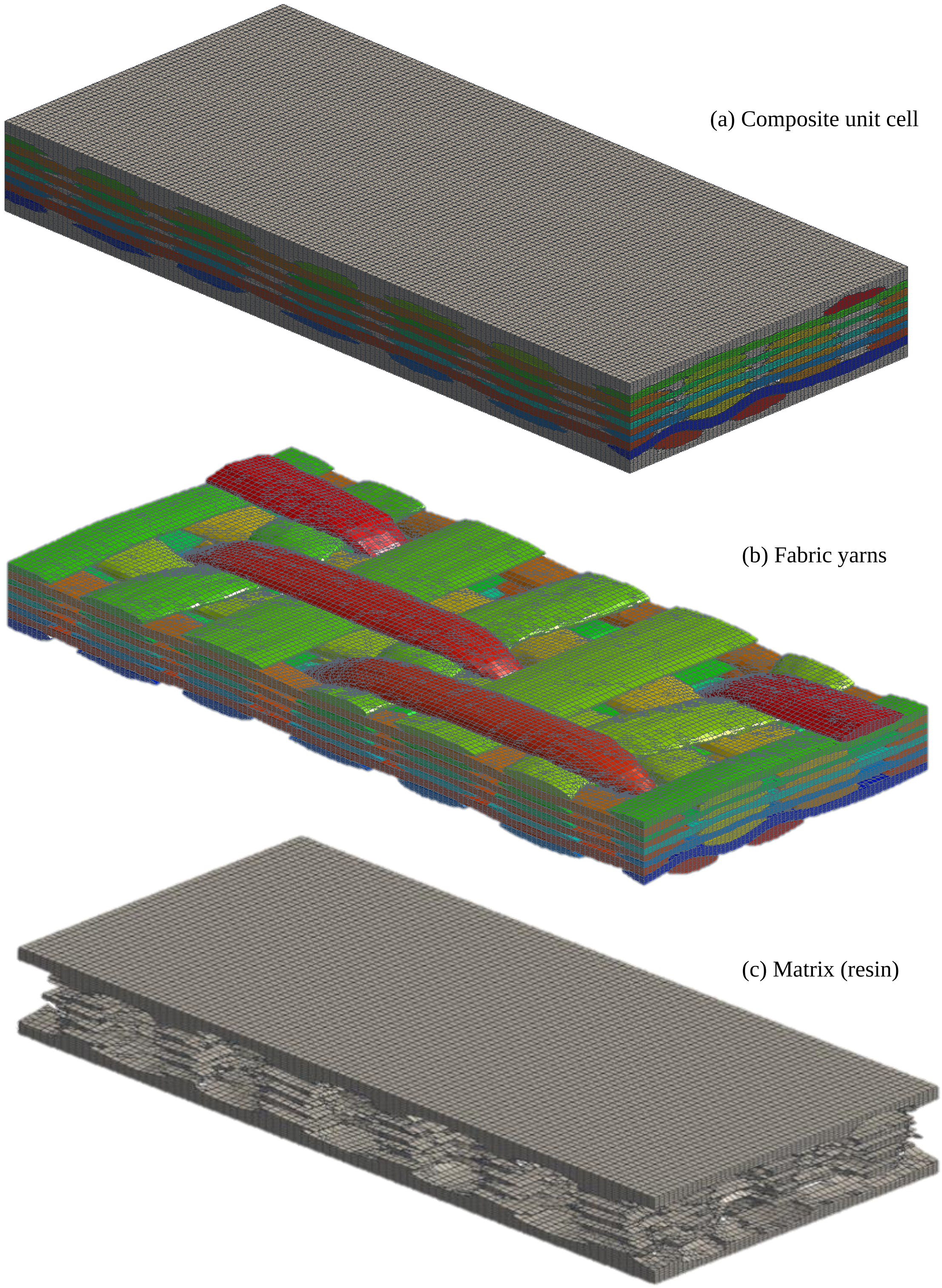


Figure 4.15: 3D orthogonal woven unit cell finite element mesh

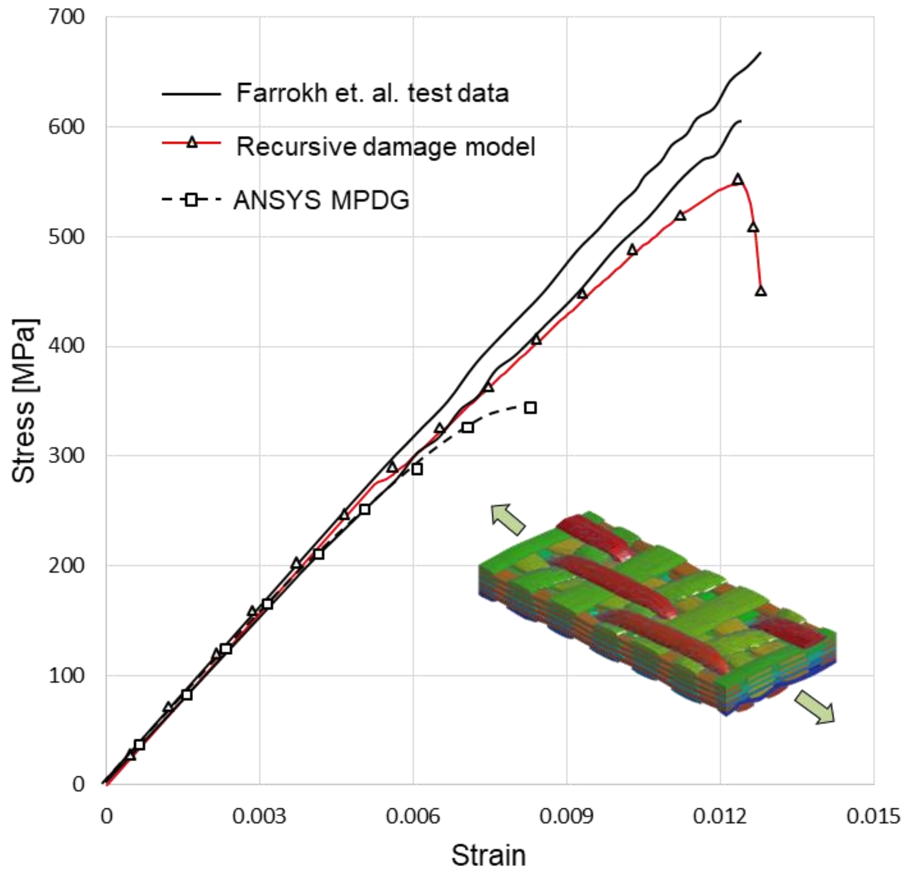
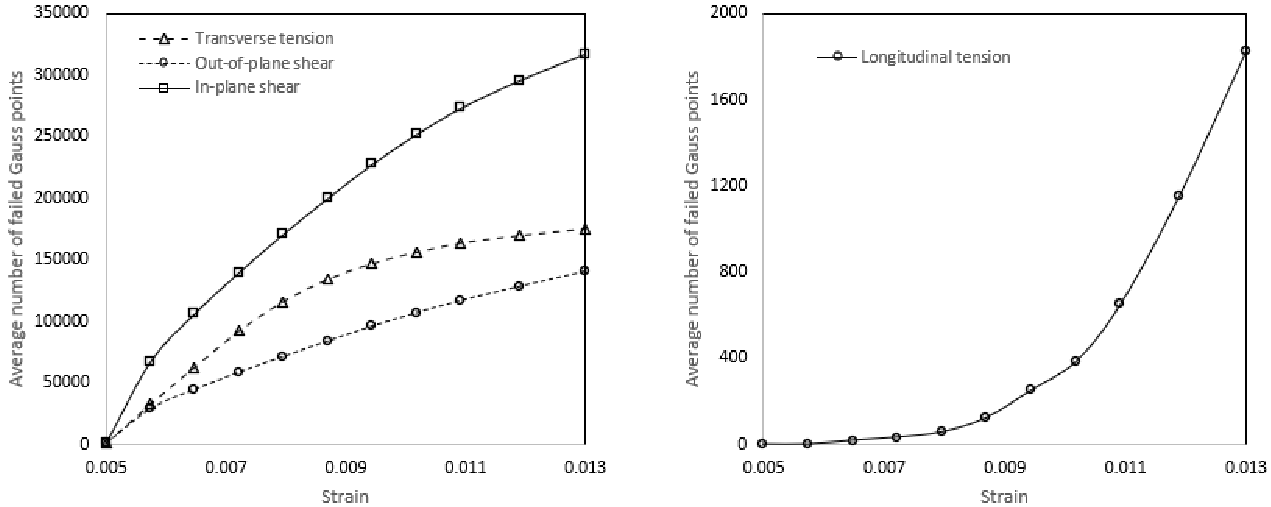


Figure 4.16: Stress-strain response of 3D orthogonal woven unit cell under tension

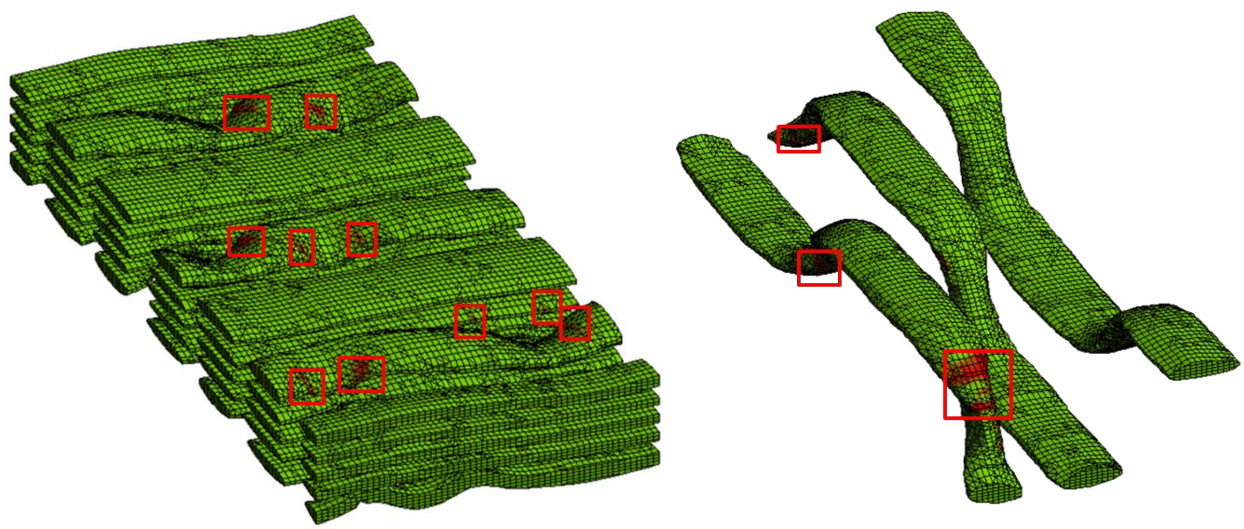
Farrokh *et al*. [72] do not discuss damage and failure mechanisms in their paper, whereby the current model provides useful insights into the damage progression process. The numerical result shows some non-linearity in the curve at 0.54% strain, while the test data is almost perfectly linear up to failure. Non-linearity found in the present study arises from failure due to transverse tension and shear in the weft and binder yarns, all which precedes the catastrophic fiber rupture in the warp yarns. At this pre- catastrophic fiber rupture stage, the yarns can still bear reduced load, with the failed fibers transferring loads to nearby fibers. Dominant failure modes initially are transverse tension, in-plane shear, and out-of-plane shear as shown in Figure 4.17(a). Longitudinal tensile failure develops at about 0.7% strain, becomes more prominent



(a) Transverse tensile and shear damage progression (b) Longitudinal tensile damage progression

Figure 4.17: Distribution of failure occurrence as a function of applied strain for the 3D woven model

around 1% strain, and finally leads to fiber rupture as shown by the damage progression curve in Figure 4.17(b).



(a) Weft yarns (b) Binder yarns

Figure 4.18: Damage due to transverse tension and shear at 0.54% strain

Transverse tension and shear failure materialize at locations where the weft and binder yarn paths are unaligned with the loading direction, as shown by the red zones in Figure 4.18(a) and Figure 4.18(b), respectively. At approximately 1.24% strain, most warp yarns fail, as shown in Figure 4.19, causing the load to drop, and matching the test data very well.

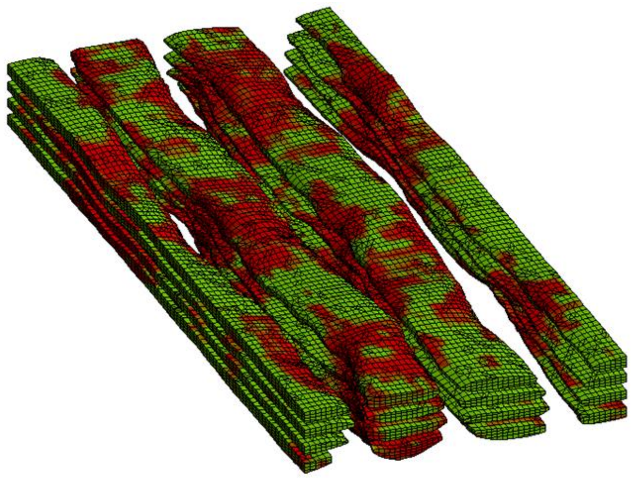


Figure 4.19: Start of catastrophic failure in warp yarns at 1.24% strain

The two examples presented here for the 2D and 3D woven composite unit cells show good agreement with available tests and numerical data and thus validate the proposed conformal FE mesh and the progressive damage model formulation.

Chapter 5 - A comparative study of the response of equivalent woven composite unit cells

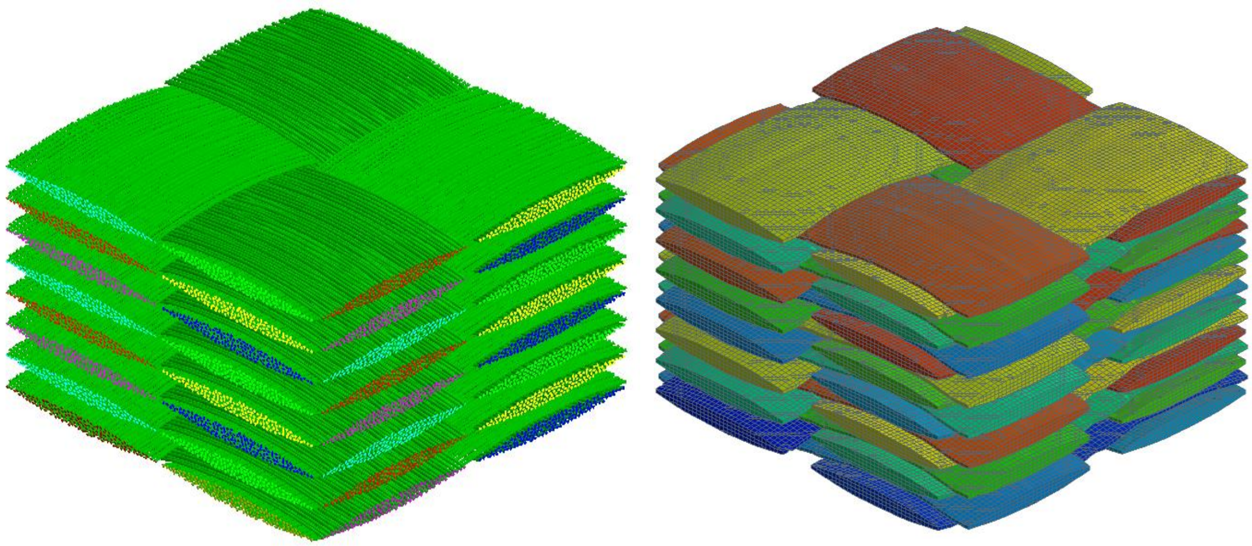
Different weaving processes result in very different fiber architectures of 2D and 3D woven fabrics, and consequently lead to substantially different properties of the resulting woven fabric composites [74]. The mechanical properties are also dependent on the type of fiber and resin system used. In this chapter, a comprehensive comparative numerical study of the in-plane tensile response of three different woven composite systems is performed – 2D multi-layer plain-woven (2DPW), 3D layer-to-layer (3DLTL) and 3D Cartesian orthogonal woven (3DCOW) composite unit cell architectures. A detailed discussion of the failure mechanisms and damage progression is presented, which is captured very accurately by the proposed progressive damage model. It enables an understanding of the complex non-linearity in the tensile response for both the fiber as well as matrix dominated failure modes. The 2D plain-woven multi-layer unit cell is the same one as reported by Blacketter *et. al*. [28] as well in the previous chapter, and hence forms the basis of the numerical comparison, since experimental data for the two 3D woven composite unit cells is not available.

The 2D multi-layer plain-woven laminate consists of eight stacked plain woven fabric composite plies, whereas the 3D woven architectures are single-ply unit cell systems. The three woven composite unit cells are "equivalent" –

1. Materials – the unit cells consist of AS4 12k carbon fiber yarns impregnated with 3501-epoxy resin, whose properties are listed in Table **4.3**
2. Overall fiber volume fraction of the woven composite unit cells is 45%
3. Fiber volume fraction in yarns of the unit cells is 70%

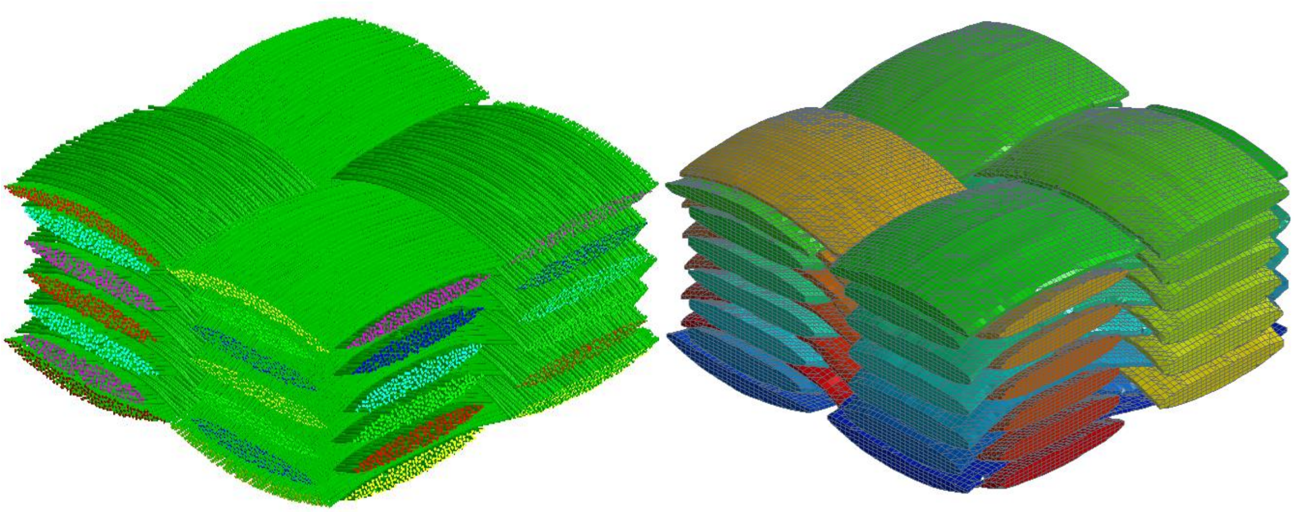
The three different woven unit cells are shown in Figure 5.1 to Figure 5.3 which show the fiber-level micro-geometry as well the conformal FE mesh models of the yarn reinforcements.

The analyses are performed on the composite unit cells which consist of the composite yarns surrounded by the isotropic matrix (resin) system, but only the reinforcing yarns are shown for simplicity.



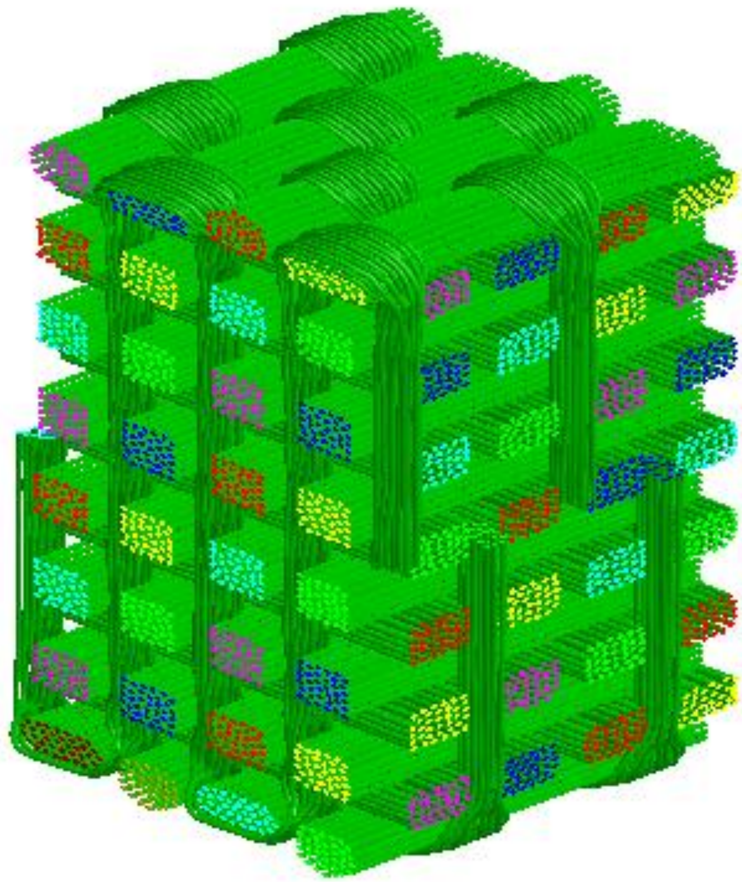
(a) Fiber-level micro-geometry (b) Yarn mesh

Figure 5.1: 2D multi-layer plain woven architecture

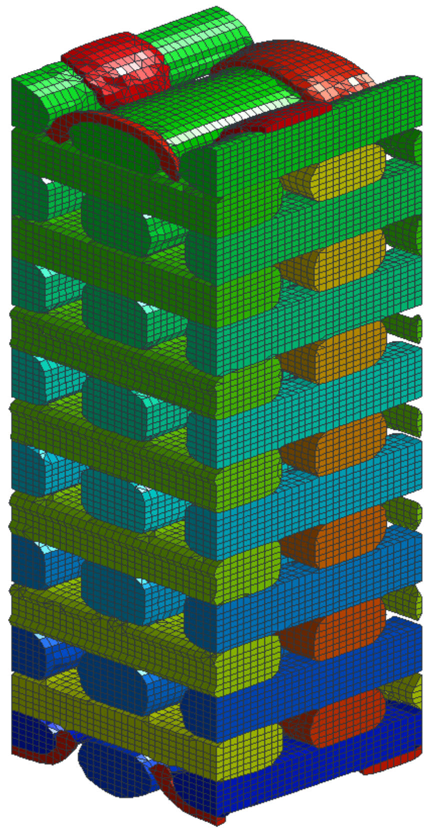


(a) Fiber-level micro-geometry (b) Yarn mesh

Figure 5.2: 3D layer-to-layer architecture



(a) Fiber-level micro-geometry



(b) Yarn mesh

Figure 5.3: 3D Cartesian orthogonal architecture

Table 5.1 lists the DFCA design summary for the three woven composite architectures. All the woven architectures have a square planar configuration and similar overall and yarn fiber volume fractions.

Table 5.1: DFCA design summary (FVF: fiber volume fraction; AD: areal density)

Woven architecture	Thickness (m)	Overall FVF	Yarn FVF	AD (g/m^2)	Weft col./in.	Warp col./in.	Fiber material
3DCOW	0.0134	0.45	0.7	15329	18	18	AS4 12k – warp, weft AS4 3k – binder
3DLTL	0.0043	0.45	0.7	4893	8	8	AS4 12k
2DPW	0.0024	0.45	0.7	2736	8	8	AS4 12k

The linear elastic stiffness under uniaxial tensile loading in the warp and weft directions are compared first for the three architectures and the results are listed in

Table **5.2**. The yarns in a fabric interlace, that is, they follow a curved path. Crimp in a fabric is the measure of this yarn waviness. Crimp of yarn is measured in terms of crimp ratio (CR). It is defined as the mean difference between the straightened yarn length and curved yarn length while in the fabric and is expressed as a percentage, given by equation (5.1).

$$CR = \frac{Uncrimped\ length - Crimped\ length}{Crimped\ Length} \times 100 \tag{5.1}$$

Table 5.2: Mechanical properties and architecture comparison (FVF: fiber volume fraction; CR: crimp ratio; b: binder)

Woven architecture	E_1 (warp)	E_2 (weft)	Weft FVF	Warp FVF	Max. CR weft	Min. CR weft	Max. CR warp	Min. CR warp
3DCOW	49.8 GPa	54.6 GPa	0.206	0.18 + 0.064b	0.3%	~0%	890.1%	~0%
3DLTL	34.4 GPa	40.4 GPa	0.24	0.21	3.5%	0.8%	3.5%	2.9%
2DPW	63.6 GPa	63.6 GPa	0.225	0.225	0.53%	0.53%	0.53%	0.53%

The overall fiber volume fraction in a unit cell is the sum of the fiber volume fractions of the weft and warp yarns (and binder yarns, when applicable). As seen from the table above, the 2D plain woven (2DPW) laminate has the highest stiffness in both the warp and weft directions, followed by the 3D Cartesian orthogonal woven (3DCOW) composite and finally the 3D layer-to-layer (3DLTL) woven composite unit cells. This is due to the combined effect of the overall fiber volume fraction as well the crimp ratio in any particular direction. A higher fiber volume fraction means more fibers are available to bear load in that direction whereas a higher crimp ratio signifies a higher fraction of the fiber lengths are not aligned to the loading direction, thus reducing their load bearing capacity. A high fiber volume fraction and low crimp ratio in the loading direction leads to higher stiffness, and thus the plain woven laminate performs better in that respect compared to the other two configurations.

On comparing the linear elastic stiffness in the weft and warp directions of each woven configuration, it can be observed that the 2DPW configuration has similar values in both directions due to symmetry. The 3DLTL configuration has a higher weft fiber volume fraction as well comparatively lower weft crimp ratio which is why it has higher stiffness in the weft direction compared to the warp direction. The 3DCOW configuration has a higher overall fiber volume fraction in the warp direction, but the crimp in the orthogonal binder yarns is very high which is why the crimp ratio plays a dominating role in determining the stiffness in this case – the stiffness in the weft direction is thus higher compared to the warp direction owing to a higher percentage of yarns with a lower crimp ratio.

Next, the non-linear response under uniaxial tension in the weft and warp directions for the three configurations are compared and the failure mechanisms are examined in detail. Figure 5.4 shows the stress vs. strain response of the different woven composite configurations under in-plane uni-axial tensile loading. The 2DPW configuration shows an almost perfectly linear stress-strain response up to the failure strength of 749 MPa, which is validated by experimental observations in [28]. Since this is a symmetric planar configuration, the weft and warp responses are similar. Fiber rupture occurs at approximately 1.18% strain after which the load drops suddenly to 0. Transverse failure of yarns precedes catastrophic longitudinal failure due to fiber rupture, which was found to occur at points of maximum yarn curvature. A detailed discussion of the failure mechanism and damage curves for this configuration is presented in Chapter 4 - and it forms the basis of the comparison with the other two configurations.

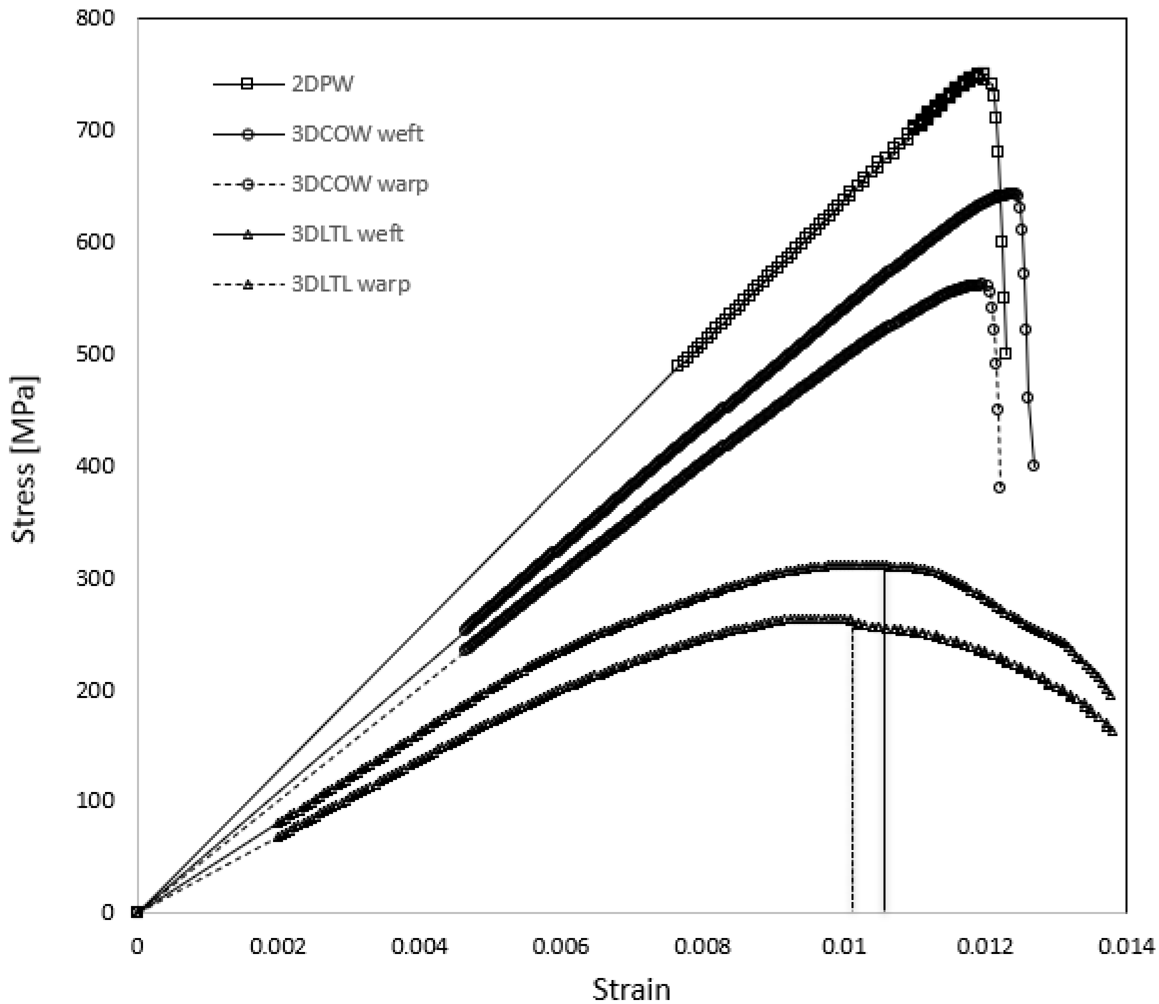


Figure 5.4: Comparison of the stress-strain response of the "equivalent" woven composite unit cells

The 3DCOW configuration shows a similar trend in the stress-strain response, where it is almost linear up to failure. As seen in Figure 5.4, the predicted value of the failure strength from the recursive stiffness reduction model is 642 MPa in the weft direction and 560 MPa in the warp direction. The predicted strengths in the two directions are consistent with the observation that higher fiber volume fraction and lower crimp ratio lead to higher stiffness and strength, as can be seen from this figure and Table **5**.**2**. Looking closely at the stress-strain response however, one can notice that a slight non-linearity arises at approximately 1% strain for both the weft and warp direction loading cases. This is due to transverse tension and shear in the weft and binder yarns when the unit cell is loaded in the warp direction, and transverse tension and shear in the warp and binder yarns when loaded in the weft direction. This is illustrated in Figure 5.5 and Figure 5.6 which shows the distribution of the total number of failed Gaussian points vs the applied strain. As can be seen in Figure 5.5, the major failure modes initially are transverse tension in the warp yarns, in-plane and out-of-plane shear for the weft loading case. On the other hand, for the warp loading case, as shown in Figure 5.6, initially transverse tension in the binder yarn and in-plane shear are the dominant failure modes. These failure modes precede the catastrophic unit cell failure due to fiber rupture, and the yarns can still bear reduced load, with the failed fibers transferring loads to nearby fibers. Catastrophic failure due to longitudinal tension becomes more prominent around 1.1% to 1.2% for the two loading cases. Consequently, the unit cells fail at 1.25% strain under uni-axial weft tension, and at 1.21% strain under uniaxial warp tension, causing the load to drop.

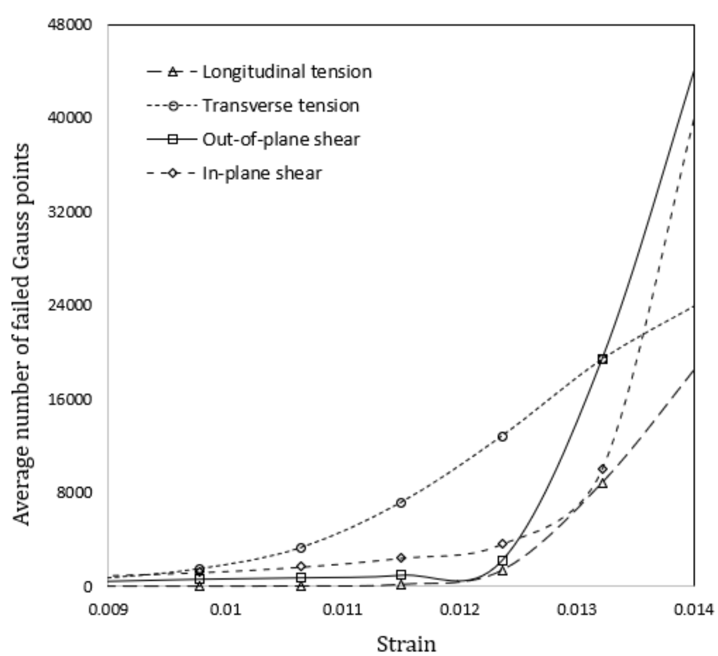


Figure 5.5: Distribution of failure occurrence for 3DCOW weft loading

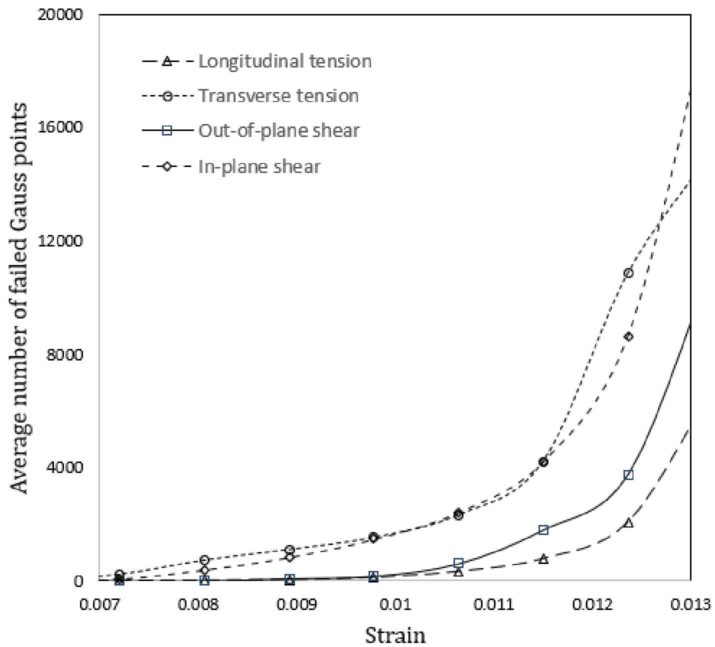


Figure 5.6: Distribution of failure occurrence for 3DCOW warp loading

The stress-strain response of the 3DLTL woven unit cell is very different from the other two configurations. Figure 5.4 shows that the predicted value of the ultimate strength from the recursive stiffness reduction model is 315 MPa in the weft direction and 264 MPa in the warp direction. The predicted strengths in the two directions are again consistent with the observation that higher fiber volume fraction and lower crimp ratio lead to higher stiffness and strength. The response is however of more interest. As observed, although the ultimate strengths of the two configurations are significantly lower compared to the two other configurations, the 3DLTL unit cells have a "softer" failure response in the sense that they have a large reserve energy after failure. Owing to the higher crimp ratio that exists in all the yarns (weft and warp), there is little to no failure of the unit cell due to fiber rupture and hence the load bearing capacity of this unit cell decreases gradually instead of a sudden abrupt failure. This results in higher energy absorption capacity of the 3DLTL configuration under in-plane loading.

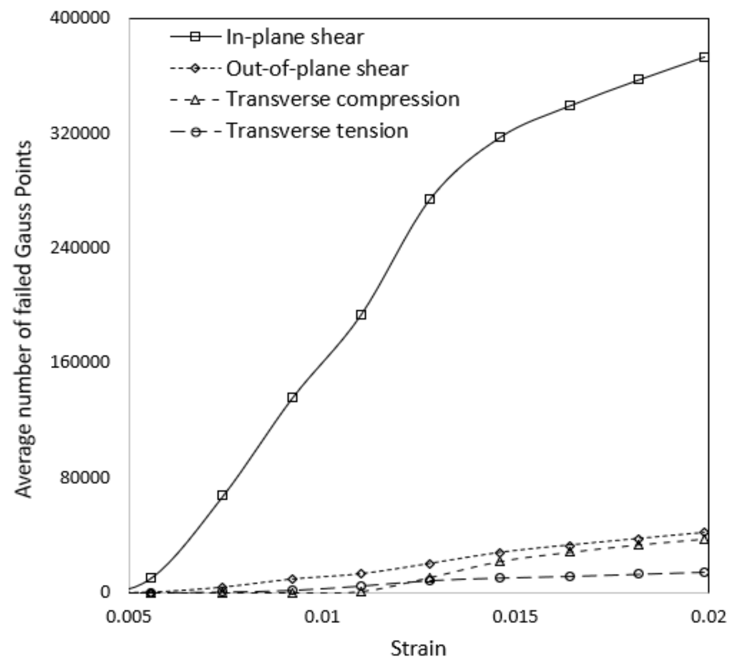


Figure 5.7: Distribution of failure occurrences for 3DLTL warp loading

Non-linearity in the stress-strain curve arises at approximately 0.5% strain for both the weft and warp direction loading cases. This is illustrated in Figure 5.8 for the weft direction

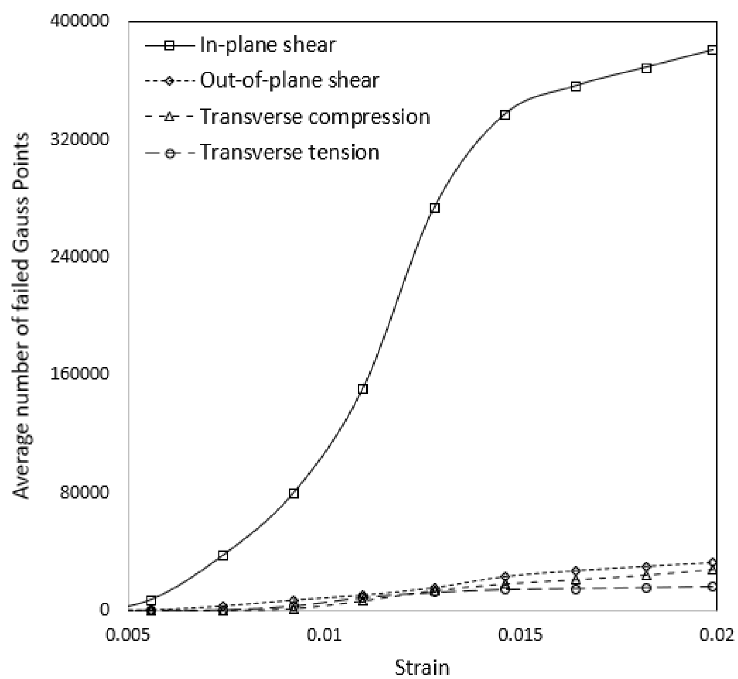


Figure 5.8: Distribution of failure occurrence for 3DLTL weft loading

tensile loading case and in Figure 5.7 for the warp direction tensile loading case. The pattern of failure is similar for the two loading cases. In-plane shear is the dominant failure mode which becomes prominent from 0.5% strain which leads to the nonlinearity in the stress-strain curve. Out-of-plane shear, transverse tension and transverse compression failure start to become prominent around 1% strain and their combined effect causes the load bearing capacity of the unit cell to drop at around 1.03% strain under uni-axial weft tension and 1% strain under uniaxial warp tension. It is to be noted here that since no catastrophic failure due to fiber rupture occurs, all the yarns can still bear reduced loads even after failure. Delamination is governed by the out-of-plane shear failure in all the cases above.

Figure **5.9** shows that the 3DLTL configuration has the highest energy absorbing capacity. Thus, as illustrated in this chapter, different unit cells have advantages in different areas – 2DPW architecture provides highest in-plane stiffness and strength, whereas the 3DLTL architecture has the highest reserve energy. 3DCOW architecture provides a good in-plane stiffness, strength and energy absorption capacity that lies in between. Based on the difference in

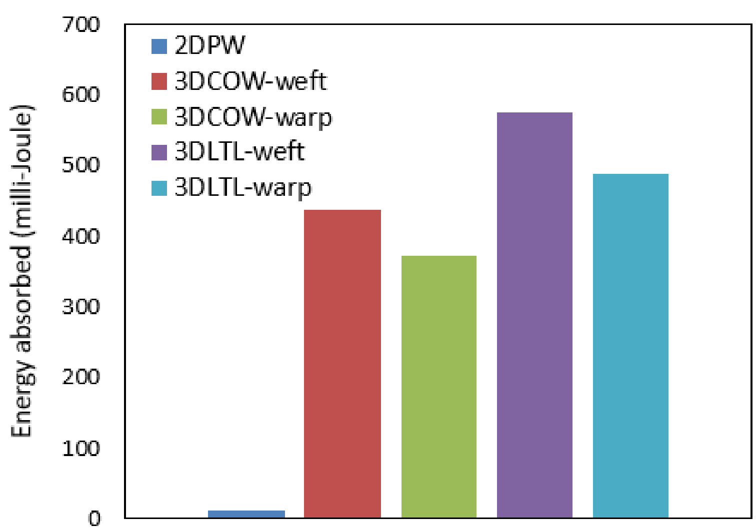


Figure 5.9: Reserve energy of the different woven configurations

woven architectures, the modes of failure also differ as seen in the 3 different cases studied in this chapter. Ultimately, the choice of the woven composite architecture depends on the design requirements.

Chapter 6 - Conclusions

6.1 Concluding remarks

An efficient procedure to generate conformal finite element volume meshes for realistic textile composite micro-geometries is developed using yarn surface geometry data derived from DFCA. Element boundaries match yarn surfaces accurately and shape function compatibility across element boundaries is enforced. Yarn interferences and narrow gaps are eliminated. Element quality improvements are performed to improve the quality and computation efficiency of the FE mesh. Both textile reinforcements and matrix components are considered to model the meshed textile composite unit cell and the components are perfectly bonded with no voids. Yarn geometry is generated using the digital element approach, which uses a multi-level dynamic relaxation procedure to generate textile composite micro-geometries that are more realistic than other existing modeling techniques. The developed model is shown to successfully capture yarn-yarn and yarn-matrix boundaries. It is used to generate several woven composite unit cell finite element models and is validated using experimental and numerical results in literature. The mesh can be input to commercial FEM software for composite stress analysis.

Then, a progressive damage model based on recursive stiffness reduction is presented to simulate the response of the woven composites to external loads. Combined tension, compression and shear failure modes are applied in numerical analysis. The recursive stiffness reduction procedure considers the progression of damage within each failed element. Hence, it presumes a more gradual process of element failure than the traditional ply-discount approach, in which element failure is considered as sudden. Stiffness and strength of two different unit cells are studied under uni-axial tensile load. The predicted values match the test results with high accuracy. Non-linearity observed in the stress-strain response of the 3D woven unit cell is discussed. Damage initiation and development are monitored step-by-step, enabling an understanding of complex failure mechanisms. A comparative study of different equivalent woven composite unit cells is also performed to study their stiffness and strength response and failure behavior. Multilayer plain woven composites show superior in-plane behavior while the layer-to-layer configuration has the highest energy absorption capacity. The orthogonal 3D woven configuration proves advantageous when both energy absorption and in-plane properties are important design factors.

6.2 Future work

The presented mesh generation capabilities have been incorporated into DFCA from version 2.0 onwards. To further improve the existing FE meshing procedure, the following approaches could be employed in the future.

6.2.1 Mesh refinement

Mesh refinement is the process of resolving a finite element model by using finer meshes to achieve a converged solution with higher accuracy. Two different methods exist for achieving the same, as shown in Figure 6.1:

1. *h-refinement*: this uses a finer mesh of the same element type by decreasing the characteristic length of elements and dividing each existing element into two or more elements without changing the type of elements used. The current version of the meshing algorithm uses h-refinement
2. *p-refinement*: this uses the same mesh but increases the displacement field accuracy in each element by increasing the degree of the highest complete polynomial (p)

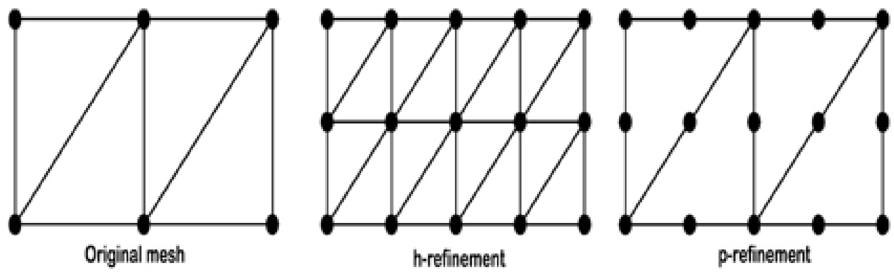
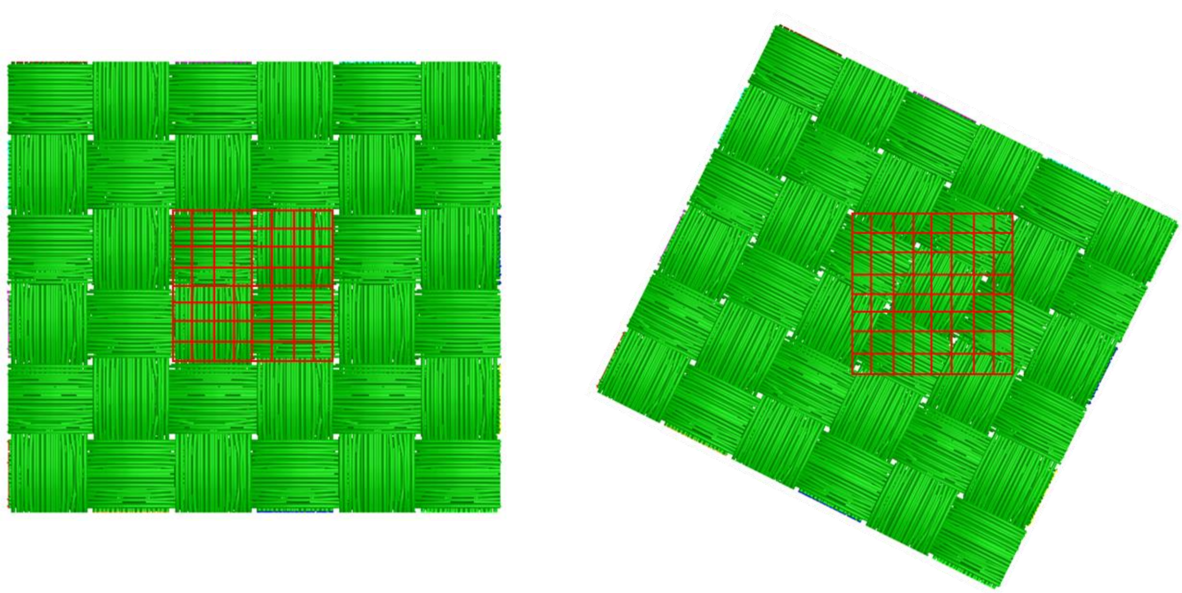


Figure 6.1: Mesh refinement

within an element without changing the number of elements used – that is, by using higher order elements. The difference between the two methods lies in how these elements are treated. The h-method uses many simple elements, whereas the p-method uses few complex elements.

A third method called the hp-refinement also exists which attempts to improve numerical convergence using a combination of the h- and p-methods. The p- and hp-refinements can be incorporated easily into the current meshing algorithm by changing the element type from SOLID185 to SOLID186, which is a 20-node brick element, by using mid-side nodes.



(a) Mesh domain with periodic boundary

(b) Mesh domain in rotated unit cell assembly without periodic boundary

Figure 6.2: Angle ply mesh domain illustration

6.2.2 Angle ply unit cell mesh

Currently unit cell finite element mesh for 0 and 90 degree plies can be generated with periodic mesh boundary. Further development is needed to generate unit cell angle ply mesh with periodic boundary, as shown in Figure 6.2.

6.2.3 Heat transfer and thermal conductivity analyses

A progressive damage model to study response to mechanical loads was developed and verified using test data in literature. Since woven carbon fiber composites are widely used in high temperature applications, the finite element model can be further used to study the heat transfer through the woven composite unit cells and determine the thermal conductivity and response to thermal loads.

References

1. Vandeurzen P, Ivens J, Verpoest I. A three-dimensional micromechanical analysis of woven-fabric composites: I. geometric analysis. Compos Sci Technol 1996;56(11):1303–15. doi: 10.1016/S0266-3538(96)00092-9.

2. Verpoest I, Lomov SV. Virtual textile composites software wisetex: integration with micro-mechanical, permeability and structural analysis. Compos Sci Technol 2005;65(15):2563–74. doi: 10.1016/j.compscitech.2005.05.031 20th Anniversary Special Issue.

3. Lin H, Brown LP, Long AC. Modelling and simulating textile structures using texgen. Advances in Textile Engineering Advanced Materials Research, vol. 331. Trans Tech Publications; 2011. p.44–7. doi: 10.4028/www.scientific.net/AMR.331.44.

4. Isart N., El Said B., Ivanov D.S., Hallett S. R., Mayugo J.A., Blanco N. Internal geometric modelling of 3d woven composites: A comparison between different approaches. Compos Struct 2015;132:1219–1230. doi:10.1016/j.compstruct.2015.07.007.

5. Ewert A., Drach B., Vasylevskyi K., Tsukrov I., Predicting the overall response of an orthogonal 3d woven composite using simulated and tomography-derived geometry, Compos Struct 2020; 243:112169. doi: 10.1016/j.compstruct.2020.112169.

6. Miao Y, Zhou E, Wang Y, Cheeseman BA. Mechanics of textile composites: microgeometry. Compos Sci Technol 2008;68(7):1671–8. doi: 10.1016/j.compscitech.2008.02.018.

7. Huang L, Wang Y, Miao Y, Swenson D, Ma Y, Yen C-F. Dynamic relaxation approach with periodic boundary conditions in determining the 3-d woven textile microgeometry. Compos Struct 2013;106:417–25. doi: 10.1016/j.compstruct.2013.05.057.

8. DFCA. www.fabricmechanics.com [accessed: 2022-04-20].

9. Zhou E., Mollenhauer D., Iarve E. Micro-geometric modeling of textile preforms with vacuum bag compression: An application of multichain digital element technique, in: 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2014. doi: 10.2514/6.2008-1868.

10. Drach A, Drach B, Tsukrov I, Bayraktar H, Goering J. Realistic fea modeling of 3d woven composites on mesoscale. Proceedings of the 19th International Conference on Composite Materials (ICCM-19). 2013.

11. Green S, Matveev M, Long A, Ivanov D, Hallett S. Mechanical modelling of 3d woven composites considering realistic unit cell geometry. Compos Struct 2014;118:284–93. doi: 10.1016/j.compstruct.2014.07.005.

12. Hello G, Schneider JD, Aboura Z. Numerical simulations of woven composite materials with voxel-FE models. In: Proceedings of the 16th European Conference on Composite Materials (ECCM 2014), Seville, Spain, 2014, p. (elec. proc.).

13. Potter E, Pinho S, Robinson P, Iannucci L, McMillan A. Mesh generation and geometrical modelling of 3d woven composites with variable tow cross-sections. Comput Mater Sci 2012;51(1):103–11. doi: 10.1016/j.commatsci.2011.06.034.

14. Joo Kim H, Swan C. Voxel based meshing and unitcell analysis of textile composites. Int J Numer Meth Eng 2003;56:977–1006. doi: 10.1002/nme.594.

15. Grail G, Hirsekorn M, Wendling A, Hivet G, Hambli R. Consistent finite element mesh generation for meso-scale modeling of textile composites with preformed and compacted reinforcements. Compos Part A: Appl Sci Manuf 2013;55:143–51. doi: 10.1016/j.compositesa.2013.09.001.

16. Nakai H, Kurashiki T, Zako M. Individual modeling of composite materials with mesh superposition method under periodic boundary condition. Proceedings of the 16th International Conferences on Composite Materials (ICCM-16). 2007.

17. Durville D. 15 – microscopic approaches for understanding the mechanical behavior of reinforcement in composites. In: Boisse P, editor. Composite Reinforcements for Optimum Performance Woodhead Publishing Series in Composites Science and Engineering Woodhead Publishing; 2011. p. 461–85. doi: 10.1533/9780857093714.4.461.

18. Wucher B, Hallstrm S, Dumas D, Pardoen T, Bailly C, Martiny P, Lani F. Nonconformal mesh-based finite element strategy for 3d textile composites. J Compos Mater 2017;51(16):2315–30. doi: 10.1177/0021998316669875.

19. Drach A, Drach B, Tsukrov I. Processing of fiber architecture data for finite element modeling of 3d woven composites. Adv Eng Softw 2014;72:18–27 special Issue dedicated to Professor Zdenk Bittnar on the occasion of his Seventieth Birthday: Part 2. doi: 10.1016/j.advengsoft.2013.06.006.

20. Lomov SV, Ivanov DS, Verpoest I, Zako M, Kurashiki T, Nakai H, Hirosawa S. Mesofe modelling of textile composites: road map, data flow and algorithms. Compos Sci Technol 2007;67(9):1870–91. doi: 10.1016/j.compscitech.2006.10.017.

21. Kim HJ, Swan CC. Algorithms for automated meshing and unit cell analysis of periodic composites with hierarchical tri-quadratic tetrahedral elements. Int J Numer Meth Eng 2003;58(11):1683–711. doi: 10.1002/nme.828.

22. McQuien JS. Evaluation of independent mesh modeling for textile composites. College Station TX, USA: Texas A&M University; 2017. [Master's thesis].

23. Fish J, Yuan Z, Crouch R, McAuliffe C, Plakomytis D, Wollschlager J. Multiscale material modeling with multiscale designer. Altair Engineering Inc; 2013. Tech. rep.

24. Ha MH, Cauvin L, Rassineux A. A methodology to mesh mesoscopic representative volume element of 3d interlock woven composites impregnated with resin. Comptes Rendus Mcanique 2016;344(4):267–83. doi: 10.1016/j.crme.2015.09.008. computational simulation of manufacturing processes.

25. Bergan A. An automated meshing framework for progressive damage analysis of fabrics using compdam. in: Proceedings of the American Society for Composites—Thirty-Fifth Technical Conference, 2020. doi: 10.12783/asc35/34953.

26. Joglekar S., Pankow M. Modeling of 3d woven composites using the digital element approach for accurate prediction of kinking under compressive loads. Compos Struct 2017;160:547–559. doi: 10.1016/j.compstruct.2016.10.070.

27. Wang Y., Mazumder A., Liu B., Yen C.F. Applications of digital element approach in textile mechanics and textile composite mechanics. in: Proceedings of the American Society for Composites - Thirty-Fifth Technical Conference, 2020; 1189–1198. doi: 10.12783/asc35/34924.

28. Blackketter DM., Walrath DE., Hansen AC. Modeling damage in a plain weave fabric-reinforced composite material. Journal of Composites Technology & Research 1993;15:136–142. doi: 10.1520/CTR10364J.

29. Li J., Zhao M., Gao X., Wan X., Zhou J. Modeling the stiffness, strength, and progressive failure behavior of woven fabric-reinforced composites. Journal of Composite Materials 2014;48:735–747. doi: 10.1177/0021998313477171.

30. Ferreiraa L., Graciania E., Parísa F. Progressive damage study of ncf composites under compressive loading. in: Proceedings of the 16th European Conference on Composite Materials (ECCM 2014), 2014.

31. Mazumder A., Liu Q., Wang Y., Yen C.-F. Stress and failure analysis of woven composites using an automated conformal mesh method. in: Proceedings of the American Society for Composites—Thirty-Sixth Technical Conference on Composite Materials, 2021. doi: 10.12783/asc36/35935.

32. Scida D., Aboura Z., Benzeggagh M., Bocherens E. A micromechanics model for 3d elasticity and failure of woven-fibre composite materials. Composites Science and Technology 1999;59:505–517. doi: 10.1016/S0266-3538(98)00096-7.

33. Yen C.-F. A ballistic material model for continuous-fiber reinforced composites. International Journal of Impact Engineering, 2012;46:11–22. doi: 10.1016/j.ijimpeng.2011.12.007.

34. Engelstad SP., Reddy JN., Knight NF. Jr. Postbuckling Response and Failure Prediction of Graphite-Epoxy Plates Loaded in Compression, AIAA Journal, 1992;30(8):2106-2112. doi: 10.2514/3.11187.

35. Knight NF. Jr., Rankin CC., Brogan FA. STAGS Computational Procedure for Progressive Failure Analysis of Laminated Composite Structures, International Journal of Non-Linear Mechanics, 2002;37(4-5):833-849. doi: 10.1016/S0020-7462(01)00101-9.

36. Moas E., Griffin OH. Jr. Progressive Failure Analysis of Laminated Composite Structures, presented at the AIAA/ASME/ASCE/AHS/ASC 38th Structures, Structural Dynamics, and Materials Conference, 1997. doi: 10.2514/6.1997-1186.

37. Reddy YS., Reddy JN. Three-Dimensional Finite Element Progressive Failure Analysis of Composite Laminates under Axial Compression, Journal of Composite Technology and Research, 1993;15(2):73-87. doi: 10.1520/CTR10358J.

38. Krajcinovic D. Continuum Damage Mechanics, Applied Mechanics Reviews, 1984;37(1):1-6.

39. Talreja R. Modeling of Damage Development in Composites using Internal Variable Concepts, Damage Mechanics in Composites, 1987;12:11-16.

40. Chang FK., Chang KY. A Progressive Damage Model for Laminated Composites Containing Stress Concentrations, Journal of Composite Materials, 1987;21(9):834-855. doi: 10.1177/002199838702100904.

41. Barbero E., Lonetti P., Sikkil K. Finite element continuum damage modeling of plain weave reinforced composites, Composites Part B: Engineering 2005;37:137–147. doi: 10.1016/j.compositesb.2005.06.001.

42. Wang L., Wu J., Chen C., Zheng C., Li B., Joshi SC., Zhou K. Progressive failure analysis of 2d woven composites at the meso-micro scale. Composite Structures 2017;178:395–405. doi: 10.1016/j.compstruct.2017.07.023.

43. Saleh MN., Lubineau G., Potluri P., Withers PJ., Soutis C. Micro-mechanics based damage mechanics for 3d orthogonal woven composites: Experiment and numerical modelling. Composite Structures 2016;156:115–124. 70th Anniversary of Professor J. N. Reddy. doi: 10.1016/j.compstruct.2016.01.021.

44. Bogdanovich AE. Multi-scale modeling, stress and failure analyses of 3-d woven composites. Journal of Materials Science 2006;41:6547–6590. doi: 10.1007/s10853-006-0197-2.

45. Bernadyck B.A., Pineda E., Ricks T.M., Mital S.K. Progressive damage Response of 3D woven composites via the multiscale recursive micromechanics solution with tailored fidelity. in: Proceedings of the American Society for Composites – Thirty-Sixth Technical Conference, 2021. doi: 10.12783/asc36/35943.

46. Matzenmiller A., Lubliner J., Taylor RL. A Constitutive Model for Anisotropic Damage in Fiber-Composites, Mechanics of Materials, 1995;20(2):125-152. doi: 10.1016/0167-6636(94)00053-0.

47. Weibull W., A Statistical Distribution Function of Wide Applicability, ASME Journal of Applied Mechanics, 1951;18:293-297. doi: 10.1115/1.4010337.

48. El-Sisi AEDA., El-Emam HM., Salim HA., Sallam HEDM. Efficient 3d modeling of damage in composite materials, Journal of Composite Materials 2015;49:817–828. doi: 10.1177/0021998314525983.

49. Huang L. Determining micro- and macro- geometry of fabric and fabric reinforced composites. Manhattan KS, USA: Kansas State University; 2013. [Ph.D. thesis].

50. Wang Y., Sun X. Determining the geometry of textile preforms using finite element analysis. in: Proceedings of the American Society for Composites, College Station, TX: 2000, p. 485–92.

51. Wang Y., Sun X. Digital element simulation of textile processes. Compos Sci Technol 2001;63:311–9. doi: 10.1016/S0266-3538(00)00223-2.

52. Mazumder A., Wang Y., Yen C.-F., A structured method to generate conformal FE mesh for realistic textile composite micro-geometry, Compos Struct 2020;239:112032. doi: 10.1016/j.compstruct.2020.112032.

53. Mazumder A., Wang Y., Yen C.-F., Conforming element mesh for realistic textile composite micro-geometry, in: Proceedings of the American Society for Composites - Thirty-Third Technical Conference, volume 4, 2018; 2445–2458. doi: 10.12783/asc33/26102.

54. SOLID185 Element description – homogenous structural geometry. ANSYS help v182. https://www.mm.bme.hu/~gyebro/files/ans_help_v182/ans_elem/Hlp_E_SOLID185.html [accessed: 2022-02-25].

55. Element Shape Testing, Ansys v182 help manual, 2017. Accessed:2022-02-07.

56. Bucki C., Lobos Y., Payan N., Hitschfeld N., Jacobian-based repair method for finite element meshes after registration, Engineering with Computers 2011;27:285–297. doi: 10.1007/s00366-010-0198-2.

57. Grandy J. Efficient computation of volume of hexahedral cells, Technical Report, Lawrence Livermore National Lab (LLNL), 1997. doi: 10.2172/632793.

58. Chamis C.C. Mechanics of composite materials: Past, present and future, Journal of Composites Technology & Research 1989;11:3–14. doi: 10.1520/CTR10143J.

59. ANSYS Mechanical APDL Modeling and Meshing Guide, ANSYS Inc., PA, USA.

60. Guagliano M., Riva E. Mechanical behaviour prediction in plain weave composites, The Journal of Strain Analysis for Engineering Design, 2001; 36:153–162. doi: 10.1243/0309324011512702.

61. Ito M., Chou T.W. An Analytical and Experimental Study of Strength and Failure Behavior of Plain Weave Composites. Journal of Composite Materials, 1998;32(1):2–30. doi: 10.1177/002199839803200101.

62. Drach B., Drach A., Tsukrov I., Penverne M., Lapusta, Yuri. Finite element models of 3D woven composites based on numerically generated micro-geometry of reinforcement. in: Proceedings of American Society for Composites – Twenty-Ninth Technical Conference, 2014.

63. Pineda E., Bernadyck B.A., Ricks T.M., Farrokh B., Jackson W. Multiscale Failure Analysis of a 3D woven Unit Cell Containing Defects. in: Proceedings of the American Society for Composites – Thirty-Fifth Technical Conference, 2020. doi: 10.12783/asc35/34928.

64. Beer FP., Johnston Jr. ER., DeWolf JT.. Mechanics of Materials, 3rd Edition. McGraw-Hill, Boston, MA, 2001.

65. Tsai SW., Wu EM. A general theory of strength for anisotropic materials. Journal of Composite Materials, 1971;5: 58–80. doi: 10.1177/002199837100500106.

66. Hashin Z. Failure Criteria for Unidirectional Fiber Composites. Journal of Applied Mechanics, 1980;47(2):329-334. doi: 10.1115/1.3153664.

67. Puck A., Schneider W. On Failure Mechanisms and Failure Criteria of Filament-wound Glass-Fiber/Resin Composites. Plastic and Polymer Technology. 1969;33-43.

68. Puck AP., Deuschle HM. Progress in the Puck Failure Theory for Fibre Reinforced Composites : Analytical solutions for 3 D-stress. 2013.

69. Mohr D. Damage of fiber-reinforced composites. ETH Zurich, Lecture 13, Fall 2015.

70. Barbero E., Shahbazi M. Determination of Material Properties for ANSYS Progressive Damage Analysis of Laminated Composites, Composite Structures 2017;176:768–779. doi: 10.1016/j.compstruct.2017.05.074.

71. ANSYS USER Material Subroutine USERMAT, https://www.cae.tntech.edu/~chriswilson/FEA/ANSYS/ansys_usermat.pdf, 1999. Accessed: 2022-02-07.

72. Farrokh B., Segal KN., Ricks TM., Miller SG., Rodini BT., Sleight DS. Uniaxial tensile properties of as4 3d woven composites with four different resin systems: Experimental results and analysis – property computations, in: Proceedings of The Composites and Advanced Materials Expo (CAMX), 2019.

73. HexTow AS4 Carbon fiber product datasheet, Hexcel, https://www.hexcel.com/Resources/DataSheets/Carbon-Fiber, 2022. Accessed: 2022-02-07.

74. Brandt J., Drechsler K., Mohamed M., Pu G. Manufacture and performance of carbon/epoxy 3-D woven composites. in: 37th International SAMPE symposium, Anaheim, CA; 1992.

75. ANSYS Mechanical APDL Programmer's Reference, ANSYS Inc., PA, USA

76. Mazumder, A., Liu, Q., Wang, Y., Yen, C.-F. A non-linear material model for progressive damage analysis of woven composites using a conformal meshing framework. Journal of Composite Materials, 2022; doi: doi.org/10.1177/00219983221113615

77. Liu, Q., Mazumder, A., Wang, Y., Xin, X., & Su, J. (2021, March). A mesoscopic level multi-scale approach to generate a complex polymer morphology. In: Electroactive Polymer Actuators and Devices (EAPAD) XXIII 2021;11587: 342-350. SPIE.

Appendix A - Material properties calculation

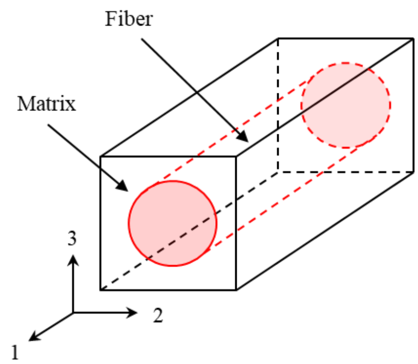


Figure A 1: Unidirectional transversely isotropic fiber reinforced composite

The material properties of each transversely isotropic composite yarn element are determined according to the Chamis' micro-mechanics model. Since transverse isotropy is assumed, the material can be described by five constants – longitudinal modulus E_1, transverse modulus E_2, shear moduli $G_{12}$and G_{23} and Poisson's ratio ν_{12}. Let the fiber longitudinal modulus be E_{f1}, transverse modulus be E_{f2}, in-plane shear modulus and Poisson's ratio be G_{f12} and ν_{f12} respectively, and out-of-plane shear modulus and Poisson's ratio be G_{f23} and ν_{f23} respectively. For the matrix, the Young's modulus, shear modulus and Poisson's ratio is denoted by E_m, G_m and ν_m respectively. Since a mesoscale idealization of unidirectional fiber-reinforced plies is used, the linear elastic properties can be calculated using the mesoscale fiber volume fraction (v_f) as shown below.

Longitudinal modulus: $$E_1 = v_f E_{f1} + (1 - v_f) E_m$$

Transverse modulus: $$E_2 = E_3 = \frac{E_m}{1 - \sqrt{v_f}\left(1 - \frac{E_m}{E_{f2}}\right)}$$

Shear modulus: $$G_{12} = G_{13} = \frac{G_m}{1 - \sqrt{v_f}\left(1 - \frac{G_m}{G_{f12}}\right)}$$

Shear modulus: $$G_{23} = \frac{G_m}{1 - \sqrt{v_f}\left(1 - \frac{G_m}{G_{f23}}\right)}$$

Poisson's ratio: $$\nu_{12} = \nu_{13} = v_f \nu_{f12} + (1 - v_f)\nu_m$$

Poisson's ratio: $$\nu_{23} = \frac{E_2}{2G_{23}} - 1$$

Micro-mechanics equations for the strength of the plies are calculated using the linear elastic properties and strengths of the fiber and matrix and the mesoscale fiber volume fraction. There are six major failure strength limits for a unidirectional fiber-reinforced lamina. Longitudinal tensile failure is largely governed by the fiber rupture, whereas buckling of fibers lead to longitudinal compressive failure. Transverse tensile failure of a unidirectional lamina occurs when a transverse crack propagates along the fiber direction and thus is a matrix dominated failure. Transverse compressive failure is governed by crushing of the matrix. In-

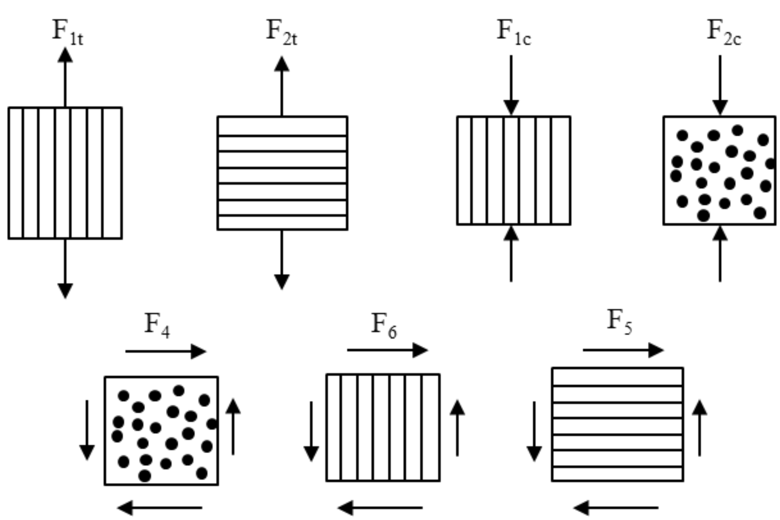


Figure A 2: Strength and failure directions of a unidirectional composite plane shear strength defines the strength at which the composite starts to fail due to matrix cracking along the fiber direction. Finally, the intra-laminar shear strength is also a matrix dominated property which causes splitting of the matrix without shearing off any fibers. The failure directions relative to the arrangement of the fibers within the composite are shown in Figure A 2. Denoting the fiber longitudinal tensile and compressive strengths as S_{f1t} and S_{f1c} respectively, and matrix tensile, compressive and shear strengths as S_{mt}, S_{mc} and S_{ms} respectively, the strengths of the unidirectional fiber-reinforced composite are given as:

Longitudinal tension: $$F_{1t} = \left\{v_f + \left(\frac{E_m}{E_{f1}}\right)(1 - v_f)\right\} S_{f1t}$$

Longitudinal compression: $$F_{1c} = v_f S_{f1c}$$

Transverse tension: $$F_{2t} = F_{3t} = \left\{1 - \left(\sqrt{v_f} - v_f\right)\left(1 - \frac{E_m}{E_{f2}}\right)\right\} S_{mt}$$

Transverse compression: $$F_{2c} = F_{3c} = \left\{1 - \left(\sqrt{v_f} - v_f\right)\left(1 - \frac{E_m}{E_{f2}}\right)\right\} S_{mc}$$

In-plane shear: $$F_6 = \left\{1 - \left(\sqrt{v_f} - v_f\right)\left(1 - \frac{G_m}{G_{f12}}\right)\right\} S_{ms}$$

Intra-laminar shear: $$F_4 = \left\{\frac{1 - \sqrt{v_f}\left(1 - \frac{G_m}{G_{f23}}\right)}{1 - v_f\left(1 - \frac{G_m}{G_{f23}}\right)}\right\} S_{ms}$$

Since a homogenous unidirectional ply is considered to calculate the material properties, in-plane shear strength F_6 and intra-laminar shear strength F_5 are numerically the same.

Appendix B - Setting up user-subroutine in ANSYS

ANSYS offers User Programmable Features, or UPFs, which allow the user to write their own subroutines in C, C++ or FORTRAN and link them to the program. In this case, a user-defined material model for progressive damage analysis was implemented using the subroutine USERMAT, the procedure for which is described here.

The analysis was done on a Dell system with four 4-core Intel Xeon E5-2637 v2 CPUs and 64 GB RAM. The OS was Windows 10.0 Pro 64-bit. ANSYS version 2020 R2 was used for all the analyses. The FORTRAN source file for the subroutine, `usermat.F`, exists in the subdirectory:

`<`*`ANSYS installation path`*`>\v202\ansys\customize\user`

Where, `<`*`ANSYS installation path`*`>` is the directory of ANSYS installation. This source file was modified to include the user-defined material model and was copied to the subdirectory it was linking in:

`<`*`ANSYS installation path`*`>\v202\ansys\custom\user\winx64`

The prerequisites for compiling the custom subroutines are described by the Installation and Licensing guide - Visual Studio with C++ compiler (version 2017 or higher) and Intel C++ and FORTRAN compilers (version 2019 or higher). One also needs to make sure that during the installation, the option "Customization files for User Programmable Features" must be selected. The Intel oneAPI Base and HPC toolkits (which replaced Intel Parallel Studio XE) were installed and integrated with Visual Studio 2019 on the system to meet the compiler requirements for ANSYS UPFs on Windows systems. The user defined subroutines can then be compiled using the Intel oneAPI command prompt for Intel 64 Visual Studio 2019 by running the `ANSCUST.bat` file from the subdirectory `<`*`ANSYS installation`* *path*>\v202\ansys\custom\user\winx64. This procedure then loads all object files along with default ANSYS objects and libraries and creates a custom ANSYS executable (ANSYS.exe), which resides in the above subdirectory. Details of the procedure as well as alternate methods are well documented [75]. Every time changes are made in any user-defined subroutines, they need to be compiled by running ANSCUST.bat to get the custom executable file.

Once the new executable is created, open ANSYS Mechanical and go to **Solve Process Settings...**. Click on **My Computer, Background > Advanced…** and paste the custom executable path (<*path*>\v202\ansys\custom\user\winx64\ANSYS.exe) to the space "**Custom Executable Name (with path)**". The custom ANSYS executable can now be used by specifying commands under the solid bodies in the model.

It is also highly recommended to debug the usermat code by testing on a simple model first. For debugging purposes, the most essential step is to make changes to the ANSCUST.bat file. There are certain arguments that need to be provided to enable debugging. Open the file in a text editor and add the following fields (in red) to the CUSTOMACROS line: **set** "CUSTMACROS=/DNOSTDCALL /DARGTRAIL /DPCWIN64_SYS /DPCWINX64_SYS /DPCWINNT_SYS /DCADOE_ANSYS /debug /Zi /warn:all /check:all /traceback /Qfp-stack-check /Zi". Next open the ansys.lrf file in a text editor and add a field called -debug after the line -manifest:embed. Once these two changes are done, recompile the ANSCUST.bat file. Then run the custom executable from APDL Product Launcher. Open Visual Studio and click **Continue without code….** Go to **File > Open File >** usermat.F. Click **Attach** and choose ANSYS Mechanical from the window. Select Code type and choose **Automatically determine the type of code to debug**. Next insert the breakpoints in the code to check the variables and run the script from APDL.

Appendix C - Progressive damage model using USERMAT

```
*deck,usermat      USERDISTRIB  parallel
subroutine usermat( matId, elemId,kDomIntPt, kLayer, kSectPt,
     &               ldstep,isubst,keycut,
     &               nDirect,nShear,ncomp,nStatev,nProp,
     &               Time,dTime,Temp,dTemp,
     &               stress,ustatev,dsdeEl,sedEl,sedPl,epseq,
     &               Strain,dStrain, epsPl, prop, coords,
     &               var0, defGrad_t, defGrad,
     &               tsstif, epsZZ,
     &               cutFactor, pVolDer, hrmflg, var3, var4,
     &               var5, var6, var7)
c*************************************************************************
c     *** primary function ***
c
c       User defined material constitutive model: 3D
c
c       The following demonstrates a USERMAT subroutine for
c       a progressive damage model for a fiber-matrix composite.
c       See "ANSYS user material subroutine USERMAT" for detailed
c       description of how to write a USERMAT routine.
c
c                                        Agniprobho Mazumder, January, 2022
c
c       If using this work, please cite [76]
c
c*************************************************************************
c
c     input arguments
c     ===============
c      matId     (int,sc,i)               material #
c      elemId    (int,sc,i)               element #
c      kDomIntPt (int,sc,i)               "k"th domain integration point
c      ldstep    (int,sc,i)               load step number: Step number
c      isubst    (int,sc,i)               substep number
c      nDirect   (int,sc,in)              # of direct components
c      nShear    (int,sc,in)              # of shear components
c      ncomp     (int,sc,in)              nDirect + nShear
c      nstatev   (int,sc,l)               Number of state variables
c      nProp     (int,sc,l)               Number of material ocnstants
c
c      Temp      (dp,sc,in)               temperature at beginning of time increment
c      dTemp     (dp,sc,in)               temperature increment
c      Time      (dp,sc,in)               time at beginning of increment (t)
c      dTime     (dp,sc,in)               current time increment (dt)
c      Strain    (dp,ar(ncomp),i)         Strain at beginning of time increment
c      dStrain   (dp,ar(ncomp),i)         Strain increment
c
c      prop      (dp,ar(nprop),i)         Material constants defined by TB,USER
c      coords    (dp,ar(3),i)             current coordinates
c      defGrad_t (dp,ar(3,3),i)           Deformation gradient at time t
c      defGrad   (dp,ar(3,3),i)           Deformation gradient at time t+dt
c
c     input output arguments
c     ======================
c      stress    (dp,ar(nTesn),io)        stress
c      ustatev   (dp,ar(nstatev),io)      user state variables
c            ustatev(1)                     - damage factor for C11
c            ustatev(2)                     - damage factor for C12
c            ustatev(3)                     - damage factor for C13
```

```
c           ustatev(4)                    - damage factor for C22
c           ustatev(5)                    - damage factor for C23
c           ustatev(6)                    - damage factor for C33
c           ustatev(7)                    - damage factor for C44
c           ustatev(8)                    - damage factor for C55
c           ustatev(9)                    - damage factor for C66
c           ustatev(10)                   - failure flag for 1-direction tension
c           ustatev(11)                   - failure flag for 1-direction compression
c           ustatev(12)                   - failure flag for 2-direction tension
c           ustatev(13)                   - failure flag for 2-direction compression
c           ustatev(14)                   - failure flag for 3-direction tension
c           ustatev(15)                   - failure flag for 3-direction compression
c           ustatev(16)                   - failure flag for 23-direction shear
c           ustatev(17)                   - failure flag for 13-direction shear
c           ustatev(18)                   - failure flag for 12-direction shear
c           ustatev(19)                   - overall failure initiation flag
c     sedEl    (dp,sc,io)               elastic work
c     sedPl    (dp,sc,io)               plastic work
c     epsPl    (dp,sc,io)               plastic strain
c     tsstif   (dp,ar(2),io)            transverse shear stiffness
c                                       tsstif(1) - Gxz
c                                       tsstif(2) - Gyz
c
c    output arguments
c    ================
c     keycut   (int,sc,io)              loading bisect/cut control
c                                       0 - no bisect/cut
c                                       1 - bisect/cut
c                                       (factor determined by sol. control)
c     dsdeEl   (dp,ar(ncomp,ncomp),io)  material jacobian matrix
c     epsZZ    (dp,sc,o)                strain epsZZ for plane stress,
c                                       define when accounting for thickness change
c                                       in shell and plane stress states
c     cutFactor(dp,sc,o)                time step size cut-back factor
c                                       define it if a smaller step size is wished
c                                       recommended value is 0~1
c
c*************************************************************************
c
c     ncomp   6   for 3D  (nshear=3)
c
c     stresss and strains, plastic strain vectors
c         11, 22, 33, 12, 23, 13    for 3D
c
c     material jacobian matrix
c       3D
c          dsdePl    |  1111   1122   1133   1112   1123   1113 |
c          dsdePl    |  2211   2222   2233   2212   2223   2213 |
c          dsdePl    |  3311   3322   3333   3312   3323   3313 |
c          dsdePl    |  1211   1222   1233   1212   1223   1213 |
c          dsdePl    |  2311   2322   2333   2312   2323   2313 |
c          dsdePl    |  1311   1322   1333   1312   1323   1313 |
c
c*************************************************************************
#include "impcom.inc"
c
INTEGER          matId, elemId,
     &                 kDomIntPt, kLayer, kSectPt,
     &                 ldstep,isubst,keycut,
     &                 nDirect,nShear,ncomp,nStatev,nProp
DOUBLE PRECISION
     &                 Time,    dTime,   Temp,    dTemp, hrmflg,
     &                 sedEl,   sedPl,   epseq,   epsZZ,   cutFactor
```

```
DOUBLE PRECISION
     &                 stress  (ncomp  ), ustatev (nStatev),
     &                 dsdePl  (ncomp,ncomp),
     &                 Strain  (ncomp  ), dStrain (ncomp  ),
     &                 epsPl   (ncomp  ), prop    (nProp  ),
     &                 coords  (3), pVolDer (3),
     &                 defGrad (3,3),     defGrad_t(3,3),
     &                 tsstif  (2)
EXTERNAL         OrthoElast3d, FailureCriteria3d
c
c**************** User defined part ************************************
c
c --- parameters
c
INTEGER          mcomp
DOUBLE PRECISION HALF, THIRD, ONE, TWO, SMALL, ONEHALF,
     &                 ZERO, TWOTHIRD, ONEDM02, ONEDM05, sqTiny
PARAMETER       (ZERO       = 0.d0,
     &                 HALF       = 0.5d0,
     &                 THIRD      = 1.d0/3.d0,
     &                 ONE        = 1.d0,
     &                 TWO        = 2.d0,
     &                 SMALL      = 1.d-08,
     &                 sqTiny     = 1.d-20,
     &                 ONEDM02    = 1.d-02,
     &                 ONEDM05    = 1.d-05,
     &                 ONEHALF    = 1.5d0,
     &                 TWOTHIRD   = 2.0d0/3.0d0,
     &                 mcomp      = 6
     &                 )
c
c --- local variables
c
c      sigElp        (dp,ar(6  ),l)            trial stress
c      sigDev        (dp,ar(6  ),l)            deviatoric stress tensor
c      betaT         (dp,sc     ,l)            Tensile damage factor
c      betaC         (dp,sc     ,l)            Compressive damage factor
c      betaS         (dp,sc     ,l)            Shear damage factor
c      PDA_Criteria  (dp,sc     ,l)            Max. Strain (1) or Max. Stress (2)
c
c --- temporary variables for solution purpose
c      i, j
c      threeOv2qEl, oneOv3G, qElOv3G, con1, con2, fratio
c
EXTERNAL         vzero, vmove, get_ElmData, get_ElmInfo
DOUBLE PRECISION sigElp(mcomp), G(mcomp), dsdeEL(mcomp,mcomp),
     &                 sigDev(mcomp), sigi(mcomp), strainEl(mcomp)

DOUBLE PRECISION var0, var1, var2, var3, var4, var5,
     &                 var6, var7

DATA G/1.0D0,1.0D0,1.0D0,0.0D0,0.0D0,0.0D0/
c
INTEGER          i, j, ncompgt, PDA_Criteria, drop
INTEGER          var_dbg1, var_dbg2
DOUBLE PRECISION pEl,   qEl, Inv1,
     &                 E1, E2, E3, nu12, nu13,
     &                 nu23, G12, G13, G23, nu21,
     &                 nu31, nu32, F1t, F2t, F3t,
     &                 F1c, F2c, F3c, F12, F13,
     &                 F23, betaT, betaC, betaS, Cpara,
     &                 C11, C12, C13, C22, C23,
     &                 C33, C44, C55, C66, E1t, E2t,
```

```
     &                  E3t, E1c, E2c, E3c, E12, E13, E23,
     &                  twoG,  threeG,  oneOv3G, qElOv3G, threeOv2qEl,
     &                  fratio,  con1,    con2, dperr(3)
c*********************************************************************
c
keycut   = 0
cutFactor = 0.d0
c
c User defined material properties are stored as
c  * First line:
c     prop(1) --> Young's modulus in 1-direction, E1
c     prop(2) --> Young's modulus in 2-direction, E2
c     prop(3) --> Young's modulus in 3-direction, E3
c     prop(4) --> Poisson's ratio, nu12
c     prop(5) --> Poisson's ratio, nu13
c     prop(6) --> Poisson's ratio, nu23
c
c  * Second line:
c     prop(7) --> Shear modulus, G12
c     prop(8) --> Shear modulus, G13
c     prop(9)  --> Shear modulus, G23
c     prop(10) --> Ultimate tens stress in 1-direction, F1t
c     prop(11) --> Ultimate comp stress in 1-direction, F1c
c     prop(12) --> Ultimate tens stress in 2-direction, F2t
c
c  * Third line:
c     prop(13) --> Ultimate comp stress in 2-direction, F2c
c     prop(14) --> Ultimate tens stress in 2-direction, F3t
c     prop(15) --> Ultimate comp stress in 2-direction, F3c
c     prop(16) --> Ultimate shear stress, F12
c     prop(17) --> Ultimate shear stress, F13
c     prop(18) --> Ultimate shear stress, F23
c
c  * Fourth line:
c     prop(19) --> beta tensile damage factor: reduce to, NOT reduce by
c     prop(20) --> beta compressive damage factor: reduce to, NOT reduce by
c     prop(21) --> beta shear damage factor: reduce to, NOT reduce by
c     prop(22) --> max strain or max stress failure critera (1 or 2)
c     prop(23) --> "not used"
c     prop(24) --> "not used"
c
c
c*** Read material properties
E1 = prop(1)
E2 = prop(2)
E3 = prop(3)
nu12 = prop(4)
nu13 = prop(5)
nu23 = prop(6)
G12 = prop(7)
G13 = prop(8)
G23 = prop(9)
nu21 = nu12 * E2 / E1
nu31 = nu13 * E3 / E1
nu32 = nu23 * E3 / E2
F1t = prop(10)
F2t = prop(11)
F3t = prop(12)
F1c = prop(13)
F2c = prop(14)
F3c = prop(15)
F23 = prop(16)
F13 = prop(17)
```

```
F12 = prop(18)
betaT = prop(19)
betaC = prop(20)
betaS = prop(21)
PDA_Criteria = prop(22)
c*** Calculate strain limits
E1t = F1t / E1
E2t = F2t / E2
E3t = F3t / E3
E1c = F1c / E1
E2c = F2c / E2
E3c = F3c / E3
E23 = F23 / G23
E13 = F13 / G13
E12 = F12 / G12
c
c
c *** define tsstif(1) since it is used for calculation of hourglass stiffness
tsstif(1) = G13
tsstif(2) = G23
c
c
c*** Compute current total strain
do i=1,ncomp
    strainEl(i) = Strain(i) + dStrain(i)
end do
c
c
c*** Assume no damage at the beginning of the analysis
if ( isubst .eq. 1 ) then
   ustatev(1) = ONE
   ustatev(2) = ONE
   ustatev(3) = ONE
   ustatev(4) = ONE
   ustatev(5) = ONE
   ustatev(6) = ONE
   ustatev(7) = ONE
   ustatev(8) = ONE
   ustatev(9) = ONE
   ustatev(10) = ZERO
   ustatev(11) = ZERO
   ustatev(12) = ZERO
   ustatev(13) = ZERO
   ustatev(14) = ZERO
   ustatev(15) = ZERO
   ustatev(16) = ZERO
   ustatev(17) = ZERO
   ustatev(18) = ZERO
   ustatev(19) = ZERO
end if
c
c
c *** Compute terms of stiffness matrix (3d)
Cpara = ONE / ( ONE - nu12*nu21 - nu23*nu32 - nu31*nu13
     *     - TWO*nu21*nu32*nu13 )
C11 = E1 * ( ONE - nu23*nu32 ) * Cpara
C12 = E1 * ( nu21 + nu31*nu23 ) * Cpara
C13 = E1 * ( nu31 + nu21*nu32 ) * Cpara
C22 = E2 * ( ONE - nu13*nu31 ) * Cpara
C23 = E2 * ( nu32 + nu12*nu31 ) * Cpara
C33 = E3 * ( ONE - nu12*nu21 ) * Cpara
C44 = G23
C55 = G13
```

```
C66 = G12
c
c
c*** Compute Stresses and stiffness matrix
call vzero(sigElp, 6)
call OrthoElast3d ( C11, C22, C33, C12, C23, C13, C44, C55,
     *     C66, strainEl, sigElp, ustatev, dsdeEl, isubst, elemId )
c
c
c*** Failure evaluation
call FailureCriteria3d  ( ustatev, F1t, F2t, F3t, F1c, F2c,
     *     F3c, F12, F23, F13, E1t, E2t, E3t, E1c, E2c,
     *     E3c, E23, E13, E12, sigElp, strainEL, betaT, betaC,
     *     betaS, isubst, elemId, PDA_Criteria )
c
c
c*** Re-compute Stresses and stiffness matrix
call vzero(sigElp, 6)
call OrthoElast3d ( C11, C22, C33, C12, C23, C13, C44, C55,
     *     C66, strainEl, sigElp, ustatev, dsdeEl, isubst, elemId )
c
c
c***  Update stresses, strains and material Jacobian matrix
do i=1,ncomp
   do j=1,ncomp
      dsdePl(j,i) = dsdeEl(j,i)
   end do
   stress(i) =  sigElp(i)
end do
return
end
************************************************************
*   OrthoElast3d: Orthotropic elasticity - 3d             *
************************************************************
subroutine OrthoElast3d ( C11, C22, C33, C12, C23, C13, C44, C55,
     *     C66, strnD, stressD, ustatev, dsdeEl, isubst, elemId )
*
#include "impcom.inc"
*
*  Orthotropic elasticity, 3D case -
*
double precision ZERO, one, two
parameter( ZERO = 0.d0, one = 1.d0, two =2.d0 )
double precision strnD(6), stressD(6), ustatev(19),
     *     dsdeEL(6,6)
double precision C11, C12, C13, C22, C23, C33,
     *     C44, C55, C66
double precision dC11, dC12, dC13, dC22, dC23, dC33,
     *     dG23, dG13, dG12
integer i, j, isubst, elemId
*     -- Compute damaged stiffness according to table
*
dC11 = ustatev(1) * C11
dC12 = ustatev(2) * C12
dC13 = ustatev(3) * C13
dC22 = ustatev(4) * C22
dC23 = ustatev(5) * C23
dC33 = ustatev(6) * C33
dG23 = ustatev(7) * C44
dG13 = ustatev(8) * C55
dG12 = ustatev(9) * C66
c
*  -- Stress update
```

```
stressD(1) = dC11 * strnD(1) + dC12 * strnD(2) + dC13 * strnD(3)
stressD(2) = dC12 * strnD(1) + dC22 * strnD(2) + dC23 * strnD(3)
stressD(3) = dC13 * strnD(1) + dC23 * strnD(2) + dC33 * strnD(3)
stressD(4) = dG23 * strnD(4)
stressD(5) = dG13 * strnD(5)
stressD(6) = dG12 * strnD(6)
*
c  -- Update material constitutive matrix
dsdeEl(1,1) = dC11
dsdeEl(1,2) = dC12
dsdeEl(1,3) = dC13
dsdeEl(1,4) = ZERO
dsdeEl(1,5) = ZERO
dsdeEl(1,6) = ZERO
dsdeEl(2,2) = dC22
dsdeEl(2,3) = dC23
dsdeEl(2,4) = ZERO
dsdeEl(2,5) = ZERO
dsdeEl(2,6) = ZERO
dsdeEl(3,3) = dC33
dsdeEl(3,4) = ZERO
dsdeEl(3,5) = ZERO
dsdeEl(3,6) = ZERO
dsdeEl(4,4) = dG23
dsdeEl(4,5) = ZERO
dsdeEl(4,6) = ZERO
dsdeEl(5,5) = dG13
dsdeEl(5,6) = ZERO
dsdeEl(6,6) = dG12
do i=1,5
  do j=i+1,6
    dsdeEl(j,i)=dsdeEl(i,j)
  end do
end do
return
end
************************************************************
*   Failure Criteria definition: Evaluate Maximum Strain   *
*   failure or Maximum Stress failure criterion for yarn   *
************************************************************
subroutine FailureCriteria3d ( ustatev, f1t, f2t, f3t, f1c, f2c,
     *     f3c, f12, f23, f13, e1t, e2t, e3t, e1c, e2c,
     *     e3c, e23, e13, e12, stress, strain, betaT, betaC,
     *     betaS, isubst, eId, PDA_Criteria )
*
#include "impcom.inc"
c
external vmax, vzero
double precision ZERO, one, two, ONETENTH
parameter (ZERO       = 0.d0,
     &           one        = 1.d0,
     &           two        = 2.d0,
     &           minimum    = 1.d-20
     &          )
integer iloc, PDA_Criteria, eId
double precision fc(9), vect(9)
double precision stress(6), strain(6), ustatev(19)
double precision f1t, f2t, f3t, f1c, f2c, f3c,
     *     f23, f13, f12, betaT, betaC, betaS, vmax,
     *     e1t, e2t, e3t, e1c, e2c, e3c, e23, e13, e12
integer ndmg, isubst
*
if ( PDA_Criteria .eq. 1) then
```

```
   vect(1) = strain(1)/e1t
   vect(2) = strain(1)/e1c
   vect(3) = strain(2)/e2t
   vect(4) = strain(2)/e2c
   vect(5) = strain(3)/e3t
   vect(6) = strain(3)/e3c
   vect(7) = (abs (strain(4)))/e23
   vect(8) = (abs (strain(5)))/e13
   vect(9) = (abs (strain(6)))/e12
else if ( PDA_Criteria .eq. 2) then
   vect(1) = stress(1)/f1t
   vect(2) = stress(1)/f1c
   vect(3) = stress(2)/f2t
   vect(4) = stress(2)/f2c
   vect(5) = stress(3)/f3t
   vect(6) = stress(3)/f3c
   vect(7) = (abs (stress(4)))/f23
   vect(8) = (abs (stress(5)))/f13
   vect(9) = (abs (stress(6)))/f12
end if
fc(1) = vmax (vect(1),9,iloc)
c
c
c***  Check if the strength ratio is greater than 1, then damage is detected
c     If damage is detected, update user state variables (ustatev). Once a failure
c     mode (1 to 9) is detected, recursive degradation continues to be applied
if (fc(1) .gt. 1 .or. ustatev(19) .eq. ONE) then
   if (vect(1) .eq. fc(1) .or. ustatev(10) .eq. ONE) then
      if (ustatev(1) .gt. betaT) then
           ustatev(1) = ustatev(1) - betaT
      else
           ustatev(1) = minimum
      end if
      if (ustatev(2) .gt. betaT) then
           ustatev(2) = ustatev(2) - betaT
      else
           ustatev(2) = minimum
      end if
      if (ustatev(3) .gt. betaT) then
           ustatev(3) = ustatev(3) - betaT
      else
           ustatev(3) = minimum
      end if
      ustatev(10) = ONE
   end if
   if (vect(2) .eq. fc(1) .or. ustatev(11) .eq. ONE) then
      if (ustatev(1) .gt. betaC) then
           ustatev(1) = ustatev(1) - betaC
      else
           ustatev(1) = minimum
      end if
      if (ustatev(2) .gt. betaC) then
           ustatev(2) = ustatev(2) - betaC
      else
           ustatev(2) = minimum
      end if
      if (ustatev(3) .gt. betaC) then
           ustatev(3) = ustatev(3) - betaC
      else
           ustatev(3) = minimum
      end if
      ustatev(11) = ONE
   if (vect(3) .eq. fc(1) .or. ustatev(12) .eq. ONE) then
```

```
   if (ustatev(2) .gt. betaT) then
        ustatev(2) = ustatev(2) - betaT
   end if
   if (ustatev(4) .gt. betaT) then
        ustatev(4) = ustatev(4) - betaT
   end if
   if (ustatev(5) .gt. betaT) then
        ustatev(5) = ustatev(5) - betaT
   end if
   ustatev(12) = ONE
end if
if (vect(4) .eq. fc(1) .or. ustatev(13) .eq. ONE) then
   if (ustatev(2) .gt. betaC) then
        ustatev(2) = ustatev(2) - betaC
   end if
   if (ustatev(4) .gt. betaC) then
        ustatev(4) = ustatev(4) - betaC
   end if
   if (ustatev(5) .gt. betaC) then
        ustatev(5) = ustatev(5) - betaC
   end if
   ustatev(13) = ONE
end if
if (vect(5) .eq. fc(1) .or. ustatev(14) .eq. ONE) then
   if (ustatev(3) .gt. betaT) then
        ustatev(3) = ustatev(3) - betaT
   end if
   if (ustatev(5) .gt. betaT) then
        ustatev(5) = ustatev(5) - betaT
   end if
   if (ustatev(6) .gt. betaT) then
        ustatev(6) = ustatev(6) - betaT
   end if
   ustatev(14) = ONE
end if
if (vect(6) .eq. fc(1) .or. ustatev(15) .eq. ONE) then
   if (ustatev(3) .gt. betaC) then
        ustatev(3) = ustatev(3) - betaC
   end if
   if (ustatev(5) .gt. betaC) then
        ustatev(5) = ustatev(5) - betaC
   end if
   if (ustatev(6) .gt. betaC) then
        ustatev(6) = ustatev(6) - betaC
   end if
   ustatev(15) = ONE
end if
if (vect(7) .eq. fc(1) .or. ustatev(16) .eq. ONE) then
   if (ustatev(7) .gt. betaS) then
        ustatev(7) = ustatev(7) - betaS
   end if
   ustatev(16) = ONE
end if
if (vect(8) .eq. fc(1) .or. ustatev(17) .eq. ONE) then
   if (ustatev(8) .gt. betaS) then
        ustatev(8) = ustatev(8) - betaS
   end if
   ustatev(17) = ONE
end if
if (vect(9) .eq. fc(1) .or. ustatev(18) .eq. ONE) then
   if (ustatev(9) .gt. betaS) then
        ustatev(9) = ustatev(9) - betaS
   end if
```

```
      ustatev(18) = ONE
   end if
   ustatev(19) = ONE
end if
return
end
```