DETECTION AND RESOLUTION OF INTERPENETRATIONS OF WOVEN TOWS

A Thesis

by

COLLIN WISENBAKER BLAKE

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

MASTER OF SCIENCE

Chair of Committee,	John Whitcomb
Committee Members,	Darren Hartl
	Terry Creasy
Head of Department,	Ivett Leyva

August 2022

Major Subject: Aerospace Engineering

ABSTRACT

Woven composite tows can be approximated by creating surfaces using the Virtual Textile Morphology Suite (VTMS) developed at the Air Force Research Lab (AFRL). These surfaces have interpenetrations between tow surface meshes which must be resolved in order to have strict, compatible mesh between all domains. A compatible mesh is desirable to reduce the complexity of the model and allow for a wider range of FEA tools to be used for analysis. To detect these interpenetrations, the surfaces were approximated using Non-Uniform Rational B-Spline (NURBS) surfaces with the SISL library from SINTEF. The interpenetration regions were then identified by B-Spline curves which, when included during the mesh generation process, allowed the surface interpenetrations to be removed and replaced with a compatible mesh between tows. The meshes are strictly tied together to investigate the effects of removing the thin slices of matrix from between tows in close proximity. These resulting meshes were subjected to a simple in-plane loading and compared to another method for removing interpenetrations that shrinks the tow cross-sections until they no longer penetrate. The predicted stresses show that the new method can create small regions of high magnitude stress in the tow local to the edge of the connected region between tows, and that high mesh refinement around these regions can increase the magnitude of these stress concentrations. In regions away from the boundary of the connected regions, both models predict similar stress responses. Also, the analysis predicts less matrix volume at high von Mises stress due to the lack of matrix between tows in the NURBS method meshes. While the analysis shows evidence of singularities, the size of the concentrations and the similar overall response as the previous method show that the new method has some merit, particularly when considering the potential use cases for compatible, connected regions between tow meshes.

ACKNOWLEDGEMENTS

I can not thank my advisor Dr. Whitcomb for his support and patience throughout the long endeavor that has been this thesis. He has afforded me many opportunities to expose myself to others that have become friends and advisors that I would have other wise never gotten the opportunity to meet. While I am excited to move forward through this stage of my life I will always remember and embrace the teachings and standards he has instilled in me during this process. Also, I would like to thank Dr. Darren Hartl and Dr. Terry Creasy for their time and effort so that this research may be at the best standard it can be.

I would also like to thank Dr. David Mollenhauer, Eric Zhou and Dr. Craig Przybyla for allowing me to work under their guidance for my two summers at AFRL. I had many great experiences in Ohio that I would have never gotten to make without their kindness.

To my collleagues Dr. Keith Ballard and Scott McQuien, I vastly appreciate your help and mentorship through my years as an undergrad and graduate student. It is rare to have such a quality group in one place that always have an answer and don't mind giving it more than once.

Lastly, I would like to thank my family and beautiful wife Alyssa, who always seem to have faith in me.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis committee consisting of Dr. John Whitcomb and Dr. Darren Hartl of the Department of Aerospace Engineering and Dr. Terry Creasy of the Department of Materials Science and Engineering.

The data analyzed for Chapter 5 was provided by Dr. Keith Ballard. All other work conducted for the thesis was completed by the student independently.

Funding Sources

This work was made possible by the AFRL Research Collaboration Program under Contract Number FA8650-13-C-5800. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the Air Force Research Lab.

NOMENCLATURE

AFRL	Air Force Research Lab
VTMS	Virtual Textile Morphology Suite
DFMA	Digital Fabric Mechanics Analyzer
NURBS	Non-Uniform Rational B-Spline
SISL	The SINTEF Spline Library

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

The use of composite materials as both a functional and decorative material has increased the need to better understand these materials in all of their forms. Composite materials, by definition, are a combination of two or more materials to achieve a material that performs better than both materials separately. A common example is steel-reinforced concrete, where the concrete's inability to sustain tensile forces is remedied by the steel reinforcement. A different composite commonly found in aerospace industries is fiber matrix composites, whose lightweight but strong tensile properties are favorable where weight is a primary constrain. Fiber matrix composites consist of fibers (numbering from a few hundred to millions) for strength combined with a resin that maintains the form of the composite material. From unidirectional ply composites to intricate woven textile composites, the need for understanding material properties and mechanics for fiber-matrix composites has never been higher.

There are various techniques that may be used to analyze a composite material and generally fall into three categories: experimental, analytical, and numerical. Experimental analysis involves manufacturing a composite material to known specifications and applying a specific load or strain on that material and studying the response. Experimental analysis of woven fiber composites has been conducted extensively. [1][2][3] Many of these tests focused on identifying the different macroscopic properties and responses of composites. However, only a limited amount of insight was provided by these tests because of the physical limitations. This method of analysis is generally expensive in both material cost and amount of time invested.

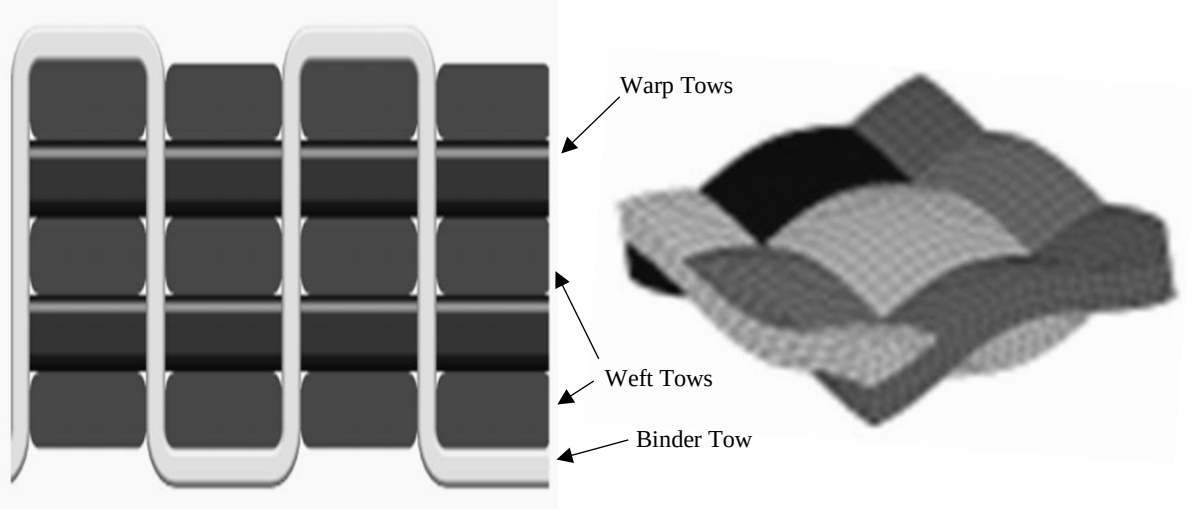
Analytical methods consist of developing mathematical equations that describe the problem such that it can be solved without a numerical method. Analytical models such as the fiber

undulation model, which describes how a fiber bundle (or tow) moves through the composite (generally as a sinusoid) [4], and the bridging model, which describes how a load is transferred between interwoven fiber bundles [5], have been developed. Together, these analytical models can predict the behavior of a woven composite and have been validated with experimental data. These models were later implemented as computer-based code to analyze woven composites. [6] Although an exact answer to a problem in the form of an analytical solution is desirable, it can be difficult or impossible to formulate such a solution. In this case, numerical analysis can be used to achieve an acceptable approximation of the result.

Numerical analyses involve the use of algorithms to approximate and solve continuous mathematical problems. A numerical analysis will create and use a set of approximations and/or iterative processes to approximate a solution with an allowable amount of error. These models are then verified against real world data to determine the accuracy of the model. A widely used numerical analysis to solve textile composites is the Finite Element Analysis. Finite Element Analysis (FEA) breaks up a large, complex problem into a collection of smaller, simpler pieces, referred to as a mesh. These smaller pieces can be reassembled so that the differential equations that described the problem can now be replaced with a collection of linear algebraic equations that approximate the problem. The result is a problem that is easily solved numerically. FEA can be used in composite research by creating two and three-dimensional models that mimic the actual geometry of a composite and assigning the appropriate bodies accurate material properties. The model is given certain boundary conditions and a result can be computed. An example of this type of model is shown in [7]. These results can provide insights into macroscale (overall) material properties, stress concentrations, deformation responses, energy absorption and other attributes that may be of interest. Results are primarily affected by how accurately the physical shape of the

object can be modeled. Although a finite element mesh (the discretized version of the physical body) is an approximation the exact shape, accurate results of the deformation of the body can be achieved as the geometric error is reduced.

Before a finite element analysis on a woven composite can be conducted, the geometry of the problem must first be defined. A perfect replication of a real woven composite is desirable. The real tows of woven composite consist of many individual fibers that follow the same relative path intertwined with other tows. However, modelling each individual fiber is too computationally expensive when considering multiple tows and layers where the fibers number in the millions. Instead, the woven composite models approximate these tows as homogenous volumes. Simplified and idealized woven fabric geometry models are commonly used to simulate the complex geometry of a woven fabric composite. The most simplistic models consist of rounded rectangles as a tow cross-section, and model the tow paths as simple curves or straight lines (Figure 1.1a). Idealized woven geometries better approximate the tow cross-sections as lenticular (equal and bi-convex) cross-sections (Figure 1.1b) and sinusoidal tow path configurations, but still reduce the



a.) Woven textile with simplistic tow cross-sections

b.) Woven textile with idealized cross-sections and compatible surface mesh with matrix removed

Figure 1.1: Simplified (a) and lenticular (b) woven tow cross sections

complexity of an actual woven composite. Idealized woven composite cross sections do not vary along the length of the tow and there is perfect mesh compatibility between tow surfaces. These models of the tow geometry can be assigned anisotropic material properties defined by a coordinate system that runs along the tow path which is defined by the centroids of the cross-sections along the length of the tow. The tow properties are generally homogenized with a high fiber volume fraction and represent pre-impregnated fiber bundles. When the surrounding matrix is added to the model, a model of a woven fabric composite is completed. The idealized tows are modified to fit together so that no two tows occupy the same space, which is required to create a finite element mesh.

Substantial experimental and computational research has been conducted on unidirectional ply composites as well as idealized textile woven composites. [8][9] However, there is still a need for non-idealized textile composite tow representations for computational analysis. Idealized tows for woven fabrics can be great predictors of macroscopic properties and give some insight into stress concentrations. Because these tows are idealized, they do not allow for the small changes in tow shape that can cause stress concentrations and lead to damage initiation. More realistic tow geometry may yield better predictions of the response of realistic woven composites.

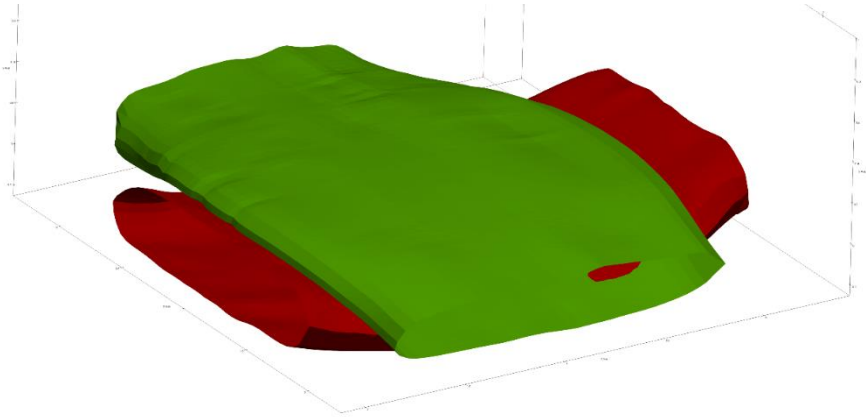
One method for creating more realistic woven geometries, developed by Wang et al. [10][11], attempts to simulate the process that manufacturers employ to create the fabrics. This method, along with other useful functionalities, has been compiled into a software called Digital Fabric Mechanics Analyzer, or DFMA. This method models the physics of this process by chaining digital elements together to simulate the interaction of fibers in a tow during production and weaving. The resulting woven geometries exhibit variations that are desired when attempting to model non-idealized woven composites. These variations include changes in cross-sectional area

and shape, tow path direction, tow thickness, and weave pattern. [12] Eric Zhou continued this work at the Air Force Research Laboratory (AFRL) and has developed the Virtual Textile Morphology Suite (VTMS). VTMS implements the chain digital element method, creating woven composite geometry as well as approximating the tows as surfaces. VTMS can then create volumetric meshes from these surfaces, creating a volumetric approximation of a non-idealized woven composite tow. It is important to note that these volumetric meshes are not compatible, nor is a compatible matrix mesh provided. AFRL uses VTMS with its own finite element software, B-Spline Analysis Method (BSAM), to conduct analyses of woven composites. BSAM uses the Independent Mesh Method (IMM) to create a matrix mesh that it can be combined with all the volumetric tow approximations to conduct an analysis. [13] This method does not create a fully compatible mesh between all tows and the matrix, but rather depends on a penalty method during analysis to impose displacement matching between the different meshes introduced. The results of these analyses have been validated against conventional finite element analyses with compatible meshes. However, there may be some inaccuracies that occur. [14] Also, neither VTMS or BSAM generate a compatible mesh between each geometric entity in the model (tows and matrix).

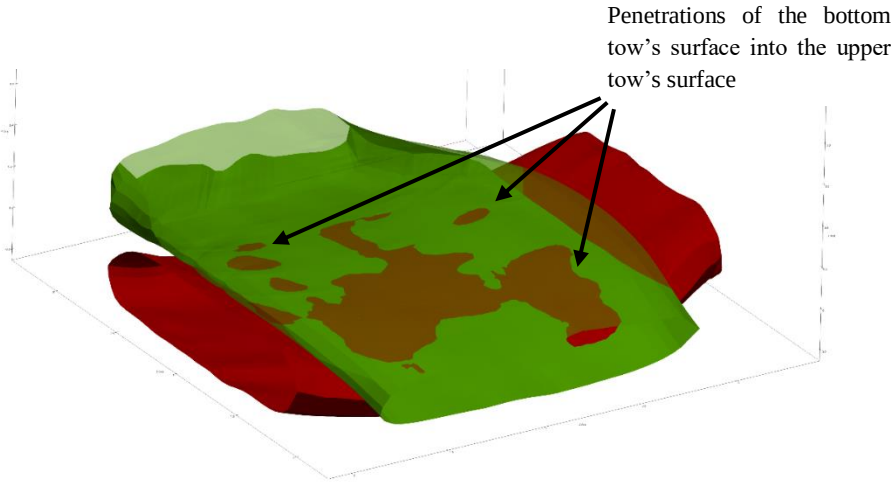
Non-interpenetrating domains are not a strict requirement for all finite element analysis tools. However, the finite element analysis tool used in this research (Beta), along with more simple tools, do required that domains be either completely separate or have perfect interfaces between domains in contact. Therefore, a method to create perfectly compatible domains (and the resulting domain meshes) could be usesful if more accurate modeling of the model geometries is required, either due to research topic or the analysis tool being used. Also, while a perfect contact interface between tows may not affect the overall predicted of a model on the macro level, the impact on

local responses in both the tows and the matrix may be signifcant. For these reasons, a method to create non-interpenetrating tow meshes is useful.

This thesis will develop a method to efficiently and accurately generate compatible tow and matrix meshes so that a finite element analysis can be conducted. Figure 1.2a shows two tow surfaces that are crossing over each other, which is a common area where interpenetrations occur. Figure 1.2b shows a cutaway view of 1.2a where the surface of the lower tow can be seen interpenetrating the lower region of the upper tow. Interpenetrations occur due to the digital element chain method and the approximation of the tow volume as a surface. These



a) Tows in close proximity



b) Transparent upper tow showing interpenetrations

Figure 1.2: Region of close tow geometries with interpenetrations

interpenetrations must be removed before a compatible mesh can be generated between all geometric entities that exist in the model. VTMS includes method to remove interpenetrations using a node-to-surface interpenetration detection algorithm. While this algorithm can remove most interpenetrations, it is susceptible to missing certain cases of interpenetrations and therfore is not a foolproof method.

Another solution has been developed by Keith Ballard that shrinks tow cross-sections in regions where interpenetrations occur before the surface mesh is created. After sufficiently shrinking certain regions along the required tow, a matrix mesh is generated using the tow surfaces that now do not have interpenetrations. This method is effective in eliminating interpenetrations, but can significantly alter the cross-sectional area in regions of the tow, thereby reducing tow volume. Tow volume reduction could alter the predicted response of the model, resulting in a loss of accuracy. Also, this method is incapable of creating contact patches between tows, which may be useful when attempting to model interactions between tows. This work will find another method of removing interpenetrations that will accurately identify interpenetration regions, remove them, and create compatible tow and matrix meshes that can be used with finite element methods.

This thesis will begin with a review of literature relevant to solving colliding and interpenetrating geometries of varying difficulty. The review will also cover any research or methods developed for a similar problem posed in this work. The chapter following the review will cover, in depth, the origin of surface interpenetrations in VTMS as well as address the shortcomings of the method implemented in VTMS to remove them. Then, the methods used in this work will be covered, including both the rationale and the process in which the method is applied. Comparative results from two methods, shrinking of tow cross-sections and the method introduced in this thesis will then presented. Also, two different mesh results created using the

method presented in this thesis will be shown to illustrate the effects of mesh refinement. Finally, a summar of conclusions, along with the direction of future work will be provided.

2. LITERATURE REVIEW

The major problem posed in this research is interpenetration of tow surfaces from VTMS. Interpenetration or collision detection of geometric entities is common to various industries from computer generated graphics to video gaming and even finite element software such as ABAQUS™. Therefore, a review of common methods applied in these industries will help to determine if one of these methods can be applied to tow surface interpenetrations. This chapter will present a review of various methods to accomplish the detection and removal of two or more volumes that occupy the same space. First, a general set of algorithms applied to generic polyhedrons and discrete (polygonal surfaces) will be discussed. Then, methods that can be applied to non-discrete (continuous) surfaces will be introduced. Finally, applications to problems similar to the one solved in this research will be covered.

2.1 Intersection and interpenetration detection of polyhedrons

The idea of solving surface penetrations is well documented in computer aided modelling of two- and three-dimensional shapes. [15] Various approaches have been used to detect whether two shapes, in both two and three dimensions, occupy the same space. Jimenez, Thomas, and Torras [16] asserted that intersection scenarios can be static or time dependent. For both cases, a static intersection step must be calculated. For a static intersection step, the assumptions made considering the polyhedrons affects the detection method.

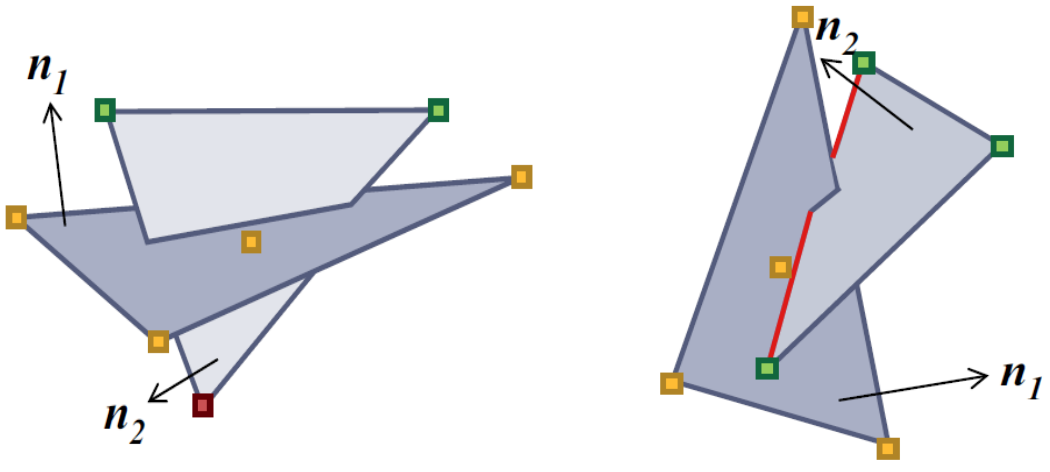
2.1.1 Intersection detection of convex polyhedrons

For surfaces that are polyhedral (defined as having multiple flat faces and can be open or closed), Dobkin and Kirkpatrick [17] state that a hierarchical representation of the polyhedral can be used. A hierarchical representation is described as a an outer-to-inner representation, where the

outer representation is a simplified, oversized approximation of the shape (like a bounding box). Each inner hierarchy that lives inside an outer hierarchy is a closer approximation of the actual geometry. As an algorithm moves inward in the hierarchy, more detail of the actual shape is achieved. If at any point, a hierarchy detects no intersections with another hierarchy, it follows that there is no intersection with the actual shape itself. In this manner, complex objects can be simplified to bounding boxes or spheres, which can be checked for intersection. This reduces the computation time required to detect an intersection. They also state that the representation of an intersection is embodied in the hierarchies of the two parent polyhedrons. In this method, the minimum distance between the shapes is calculated, and said to be null if the shapes intersect. This framework is useful because once a hierarchy is established for a shape, it can be used for every query involving the shape. Its limitation is that it requires the polyhedrals to be convex.

2.1.2 Intersection detection of polyhedrons with convex faces

Canny [18] discusses a method for the case of a polyhedron with convex faces. This is more general than a convex polyhedron as a polyhedron with convex faces can be concave. Rather, two faces on the polyhedron may face each other, but the polygon cannot have internal-facing surfaces.



a) Type A (node interpenetration) **b) Type B (edge-edge interpenetration)**

Figure 2.1: Two cases of interpenetrations between polygon surfaces. Reprinted from [35]

He states that two intersection cases exist between polyhedra, face-to-node contact (Type A, Figure 2.1a) and edge-edge (Type B, Figure 2.1b). By associating a comparison that is true or false for each case, for any set of points and edges, a series of tests can be run on the two polyhedrons. If any comparison remains true at the end of the tests, the shapes are said to have intersected. This method is useful because it requires simple vector math to run the tests. The method itself does not directly identify the case of containment (one shape lying completely in another). However, a simple ray intersection algorithm can determine if containment is occurring.

2.1.3 Intersection detection of non-convex polyhedrons

The most general polyhedral shape is non-convex, or rather a shape whose external faces can face each other. For this case, there are two strategies. The most common strategy is to subdivide the domain into convex sub-domains. Two popular methods are decomposing a volume into smaller convex polyhedron [19] and decomposing just the surface of a volume into convex surfaces, where each new, smaller surface has no part that faces itself [20]. Decomposing a volume into smaller convex polyhedra is a common technique because all volumes can be subdivided into smaller, convex polygons. However, the level of refinement to achieve all convex polyhedrons may require too many steps and/or result in a large number of intersection checks. After sub-division, the smaller, convex shapes can then use a multitude of intersection algorithms that apply to convex shapes. The main drawback of this sub-division method is the increase in number of operations and intersection checks required.

The more complex and less used method is a direct approach for calculating the intersection. This usually involves a two-step process to identify edge-face intersections [21] involving a ray-intersection algorithm to determine if edge end points lie on opposing sides of any face of a polyhedron. By counting the number of intersections an edge has with faces on the polyhedron, it

can be determined if the edge intersects with the polyhedron. Another method that does not require computing these intersection tests involves computing the signs of the determinants of the coeffcient matrix of a set of linear equations. A linear equation can be created to describe where a specific surface node is located in space. These equations can be set up so that they calculate the location of polyhedral surface nodes. Then, a set of comparisons, whose configuration set up is covered in [18] and [22], can be calculated. These comparisons check whether any point, surface, or edges intersect with any surface or edge on another polyhedron. They can be assembled in matrix form and by calculating the sign of the determinants, it can be determined if a particular type of intersection occurs. The method does not care about the convexity of the shape and can be applied to the most general of cases. The main drawback of this method is that it requires extensive setup of the shape vertex equations as well as the framework for solving the linear equations.

2.2 Intersection and interpenetration detection of non-faceted surfaces

Although detection algorithms for discretized surfaces are well documented, they are not the only method for detection. There are a variety of methods to translate discretized surfaces into continuous (non-discretized) descriptions of these surfaces. Two common types of continuous surfaces are implicit surfaces and parametric surfaces.

2.2.1 Implicit Surfaces

Implicit surfaces are in three-dimensional space and are defined by a function that, when the function is evaluated at a point on the surface in three-dimensional space, the function is equal to zero. If the function is a polynomial in *x*, *y*, and *z*, it is considered algebraic [23]. The most frequently used algebraic surfaces are quadratic, which are second degree polynomials in *x*, *y*, and *z*.

For implicit surfaces, the available algorithms are limited. Pentland and Williams [24] discuss the implementation of "inside-outside" functions that use the object's canonical frame (no rotation, centered on origin) and current location. Once the function is formed the surface to be tested has its points tested against another surfaces inside-outside functions. If a point is determined to be inside, it is intersecting. One main advantage of this algorithm over any polygon intersection detection algorithms is that it can obtain a good closed form solution that approximates the interpenetration region depth, area, and shape. This is very valuable when the shapes are static and more descriptive information of the penetration region is required. However, this method is only applicable to implicit functions and has drawbacks in terms of robustness as it relies on point sampling. Lin and Manocha [25] have discussed algorithms that extend their previously mentioned hierarchical representation algorithm that used curved models made of splines and algebraic surfaces, which work best on low degree curves.

2.2.2 Parametric Surfaces

The other typically used non-polygon surface is a parametric surface. These surfaces in three dimensions are described by functions that have two curvilinear coordinates. As a result, they are generally not closed but easier to discretize and render. A subset of parametric surfaces, Non-Uniform Rational B-Spline (NURBS), have gained traction in computer aided design software [26] and possess some ideal properties that make them easier to use. Parametric surfaces have a larger set of algorithms for intersection detection than implicit surfaces. There are four main methods: lattice, subdivision, tracing, and analytic methods. This review will cover the latter three as they are the most relevant to this research.

The subdivision method works in recursive steps. This method subdivides both surfaces at each computational step and tests the new subdivision for intersections. By recursively subdividing and

testing for intersections of the subdomains, the domain of the intersection region can be approximated. The intersected subdomains can be further subdivided to more accurately describe the intersection region [27]. A method very similar to this is used by Drach et al [28] to determine if surface nodes interpenetrate a surface. They then use another technique to remove interpenetrating surface nodes until they are all corrected. The drawback to this method is that higher refinement produces more accurate intersection detection. However, higher refinement also requires more computation time, and is therefore slow.

A second method used is tracing. This method starts by first finding a known point of intersection, for which there are multiple methods to choose from [29] [30] [31]. Then, the intersection curve is traced along by starting at the previously calculated point of intersection and moving along a determined vector by a set distance. The vector is found by intersecting the tangent planes of the two surfaces at this point and calculating the direction of the line that defines the intersection of these planes. The distance along this vector is specified and is the determining factor in the amount of "refinement" the curve has. One issue the method faces is determining if a curve has returned to its starting position. Determining if the start location has been reached is usually posed as a system of algebraic equations [32] or a differential equation problem [23]. This method can yield very good results when trying to identify a boundary curve for the interpenetration regions. The drawback for this method is the time it can take to compute the starting point and the time it can take to complete the marching steps. This method also does not give the actual region of surface that is penetrating, only the outline of the region. Therefore, each surface must be evaluated to determine which part of the surface lies within the boundary.

A third method is the analytic method. Generally, one parametric surface is converted into an implicit representation of the surface [33] and creates a scalar function in the two parametric

variables. The root locus of these functions in the parametric variable plane is the projection of the intersection curve [23] [34]. In other words, this method creates a series of algebraic equations that describe where one surface lies on another. In the case that they intersect, the equations can be solved and the result is a curve that defines the intersection of the two surfaces. This method can be difficult to implement as it requires knowledge of how to accomplish the parametric-implicit conversion as well as a framework for multiplying and solving complex polynomial equations.

2.3 Current implementations for similar problems

Drach et al [28] have used several of these techniques to solve a very similar problem to the one posed for this research. They have used a variety of software to produce realistic woven fabric geometries and have also encountered the tow interpenetration problem. Their first attempt was using a variation of the subdivision method where they create voxels (or bounding boxes) that collectively encompass the tow volume for the host tows. The tow being checked against the host is still in its faceted form and they check the host voxels against the surface nodes of the other tow. This allows them to quickly identify interpenetrating nodes. They then move the penetrating node outwards along a direction vector. This vector is calculated by averaging of all the normal vectors for the interpenetrating surface elements. When they detect no more interpenetrations, they consider them fixed. In a subsequent paper [35], they updated their method to also account for edge-edge intersections as well. This method accomplishes the task of fixing interpenetrations but results in the two tows not being in contact. During the removal of interpenetrating nodes, the nodes are moved until they are a minimum distance away from the host tow. This allows for small matrix pockets between tows that may not be present in actual computerized tomography (CT) scans.

Mazumder et al [36] have also developed a technique to facilitate the use of DFMA generated textile geometry. This method more specifically applies to creating a conformal matrix mesh, however they also attempt to resolve tow surface interpenetrations. Their method is very similar to that described in Appendix B, but still requires accurate point-to-surface interpenetration and detection, which may not work in all cases.

2.4 VTMS Interpenetration Detection Method

The surfaces in VTMS are discretized into convex faces, or surface elements, that make up the surface approximation. These surfaces qualify as non-convex polyhedrons. While not explicitly covered in the literature, VTMS comes with a method to remove the surface interpenetrations that occur in between surface approximations it produces. The method used is very similar to that discussed in [18][20][22], which evaluates whether a surface point penetrates into another polyhedron. The full method is explained in Appendix B. However, the VTMS algorithm does not account for edge-edge intersections, which still remain after the algorithm finishes. An update to this method was implemented (also covered in Appendix B), but the resulting surface meshes were still unsuitable for a conventional finite element method using a compatible mesh. Therefore, a new strategy will be preseneted in this research.

A review of the literature has shown that the implementation of a parametric surface intersection detection method would allow for a more accurate detection method than what is available for VTMS. There are various third-party libraries that support parametric surface intersection detection that can be implemented into pre-existing computational methods. Therefore, this thesis will use a parametric, specifically a Non-Uniform Rational B-Spline (NURBS), surface and curve library to detect interpenetrations between surfaces. Use of this library will require describing a VTMS surface so that the library can use the data to correctly

identify intersections between surfaces. Then, the intersection data will be used to identify regions of the surface approximations that need to be corrected. For this thesis, the correction made will result in a shared surface between the two surfaces where an interpenetration previously existed. The details of this method will be covered further in Chapter 4. Before the method is discussed in detail, Chapter 3 will explore the origin of interpenetrations to fully understand the problem. A brief theory section will cover the digital-element chain method used by VTMS. The following section in Chapter 3 will discuss the approximations made when using the digital-element chain and how those approximations result in surface interpenetrations.

3. THEORY

This chapter will cover an introduction to the digital-element chain method. Understanding this method is the key to knowing how and why surface interpenetrations occur. Then, the actual mechanics that occur to introduce surface interpenetrations will be discussed. The result will be a better understanding of where interpenetrations come from and where to start to fix them.

3.1 Digital-Element Chain Method

The digital-element chain method simulates the processes that manufacturers employ to create different fabrics and weave patterns. This method is implemented in a software suite called Digital Fabric Mechanics Analyzer (DFMA). [10] The process begins by simulating bundles of fibers as digital chains. These digital chains are frictionless pins (no transfer of moment) connected by uniaxial rods. (Figure 3.1a) Each pin, or node, has a sphere of influence that approximates the

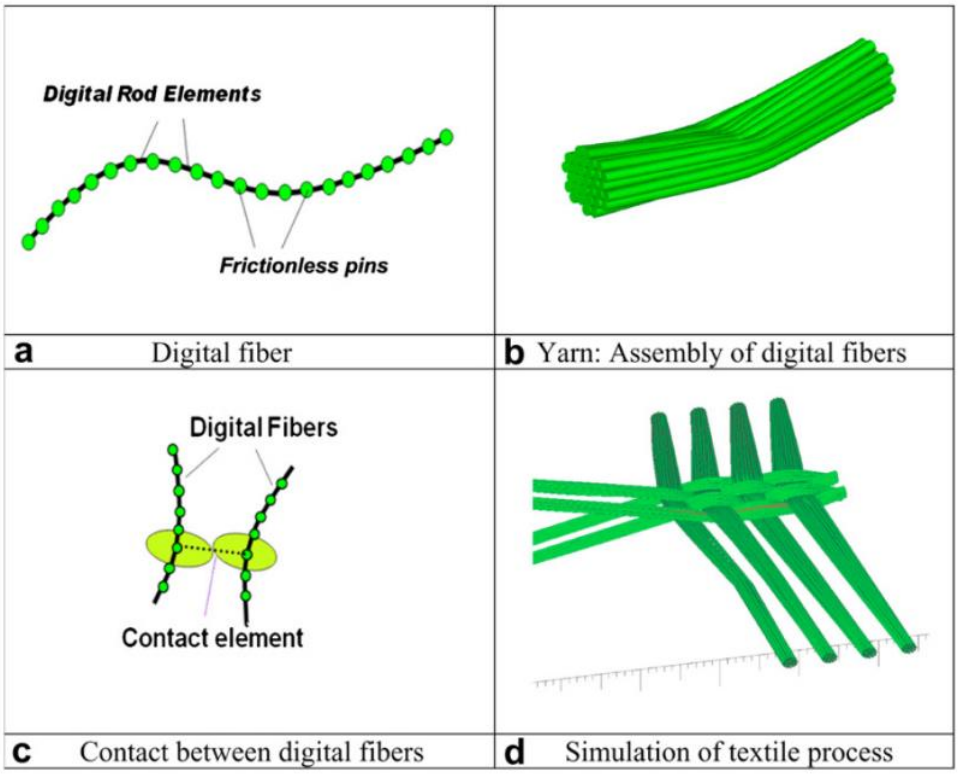
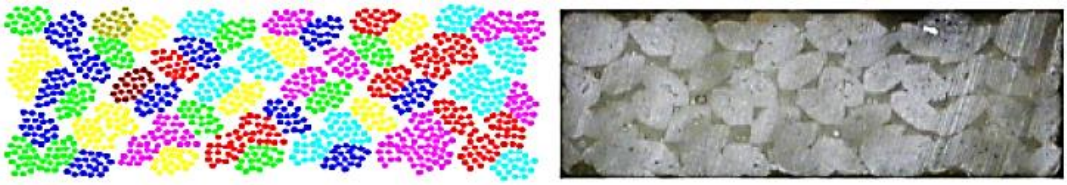
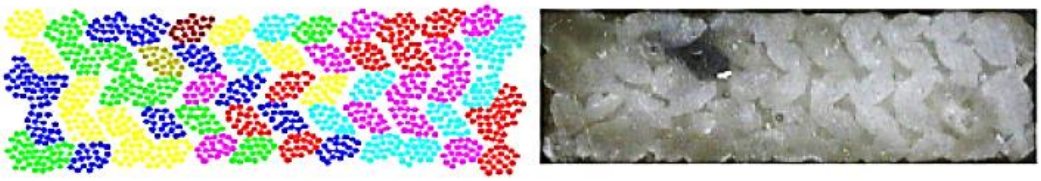


Figure 3.1: Digital chain simulation process. Reprinted from [11]

diameter of a bundle of fibers to simulate the chain having volume. In literature, the term “yarn” and “tow” are used interchangably to describe a volume that has a variable number of digital chains. Therfore, a tow can be descritized into many digital chains, as shown in Figure 3.1b. As the number of digital chains increase per tow, the resulting geometry better represents real tow geometry. Also, the shorter the axial rod length, the closer the predicted response will be to that of an actual tow. [10] An initial pre-stress is given to the chains that is used to pull the digital chains into contact with each other. Then, a contact problem is solved where spheres between two digital fibers can create forces between each other through a contact element (Figure 3.1.c) resulting in interactions between the digital chains. VTMS also simulates a set of plates that squeeze the chains in the out-of-plane direction with an applied force or displacement condition. This ensures a tightly compacted textile like those seen in experiments (Figure 3.1.d). The result is fiber bundle cross sections that are similar to micro-CT scans from actual woven specimens, shown in Figure 3.2, from [12].



(a) Cross-section 1

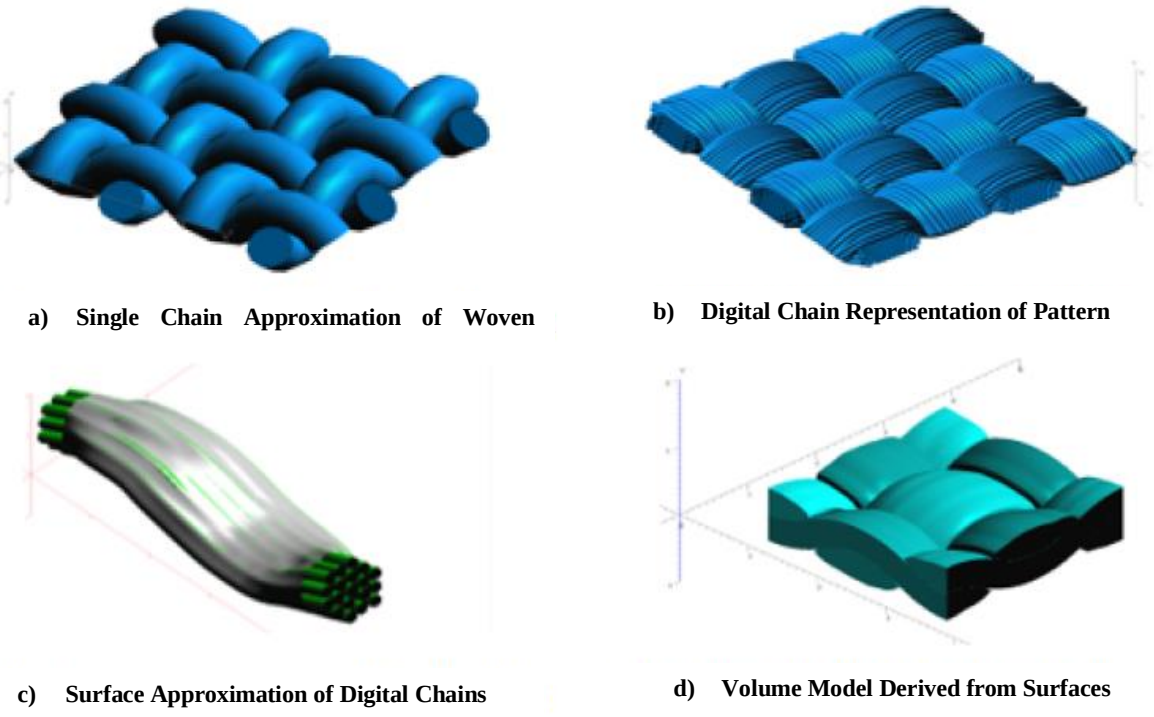


(b) Cross-section 2

Figure 3.2: Simulated vs. Actual Fiber Bundle Cross Sections. Reprinted from [12]

While there is possibly other software that use a similar method to the digital chain method, only two were explored. The digital chain method was introduced by Youqi Wong, Eric Zhou, and other researchers from Kansas State University. They developed the first iteration of DFMA which

is still being improved by this group. Eric Zhou has since moved on to AFRL and developed a similar software using a similar method, called Virtual Textile Morphology Suite (VTMS). The two software are related by their method of simulation but are being developed independently from each other. It is from VTMS that the woven textile model and surface data that is used in this study originates.



a) Single Chain Approximation of Woven

b) Digital Chain Representation of Pattern

c) Surface Approximation of Digital Chains

d) Volume Model Derived from Surfaces

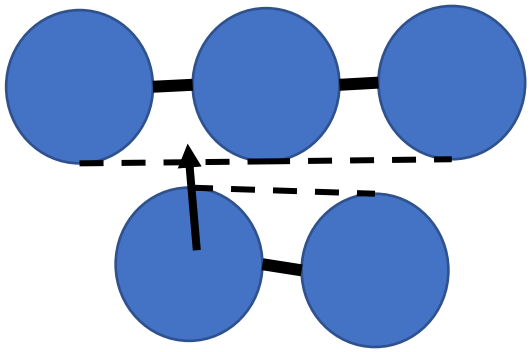
Figure 3.3: Evolution of Weave Textile Geometry. Adapted from [14]

Figure 3.3 shows steps that VTMS goes through to produce woven tow models. VTMS starts with an approximation of a woven pattern with a single digital chain to approximate each tow (Figure 3.3a). Using a user-defined refinement level, VTMS replaces each tow with multiple digital chains that approximate the tow fibers. These digital chains are still much larger in diameter than actual fibers but represent a bundle of fibers better than the initial tow approximation. These digital chains are then subjected to multiple compaction, tension, and relaxation steps to create a tightly woven set of tows (Figure 3.3b). The digital chain representation in theory could be used to create a model in which each tow is modeled by its individual fibers. However, even if such a

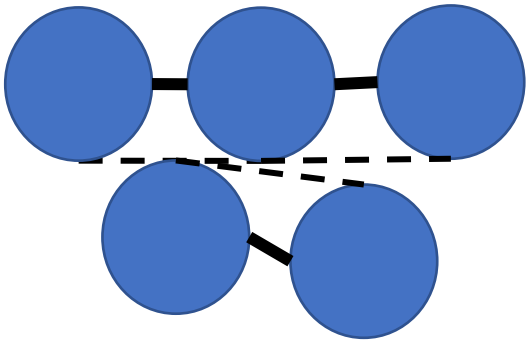
model could be created, the resources and computational time required to run such a model is too extreme. Instead, researchers use volumetric tow approximations to conduct an analysis of a woven textile model. Therefore, the bundles are approximated with a surface (Figure 3.3c), and then the surfaces are put together to create a volumetric model of all the tows (Figure 3.3d). AFRL uses these volumetric models in conjunction with BSAM to model and analyze various textile configurations.

3.2 Surface Interpenetrations

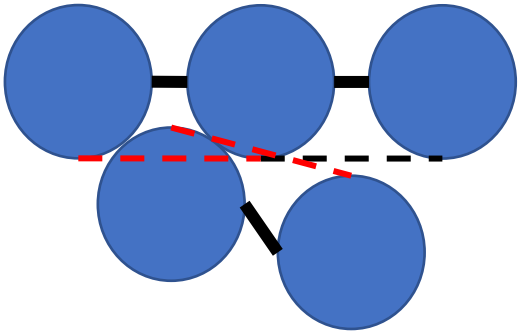
Any two surfaces that come into contact would create a contact region in the form of a shared surface. However, the surfaces in VTMS can have interpenetrations, or regions where two surfaces occupy the same volume in space. Interpenetrations can form because the contact between tows is modeled approximately with the digital chain method. Digital chains approximate a cylindrical volume with spheres connected by infinitesimally thin rods (Figure 3.4a). Each sphere has an associated diameter that sets a sphere of influence. The diameter is set so that the area of all the fibers being represented are contained in the cross-sectional area of the sphere. These volumetric spheres determine when contact elements are created between pins of two separate chains. If the distance between pins in different chains is less than the sum of the two radii of the spheres, a contact element is created that acts like a stiff spring between the pins. However, the distance calculation does not account for the distance to the connecting rods between pins. Therefore, the spheres of one chain can occupy the space between spheres of another chain (Figure 3.4c) where realistically the surfaces of the cylinders would contact each other (Figure 3.4b). The result is element chains that can have surface interpenetrations.



a) Initial digital chain configuration with dotted lines showing boundary of chain volume being approximated



b) Realistic result of contact model where digital chain sphere contacts edge of volume and stops



c) Actual result of digital chain method where sphere is stopped after penetrating volume boundary

Figure 3.4: Realistic vs. result of the digital chain contact method

Another approximation that contributes to interpenetrations is how the surface approximations of the digital fiber bundles are generated from the digital chains. The digital chain bundle is approximated using multiple cross-sections, defined by points, along the length of the bundle. First, the method creates a list of each digital chain's position (node) for the current cross-section. Next, a circle of equally spaced points is created by dividing the angular domain (2π) of the circle by the number of desired points, defined by the user, to acquire an angle $\boldsymbol{\theta}$ between each point. The circle radius **r** in figure 3.5a is calculated using equation 3.1:

$$r = d_{max} + \frac{(1+R)*D}{2} \tag{3.1}$$

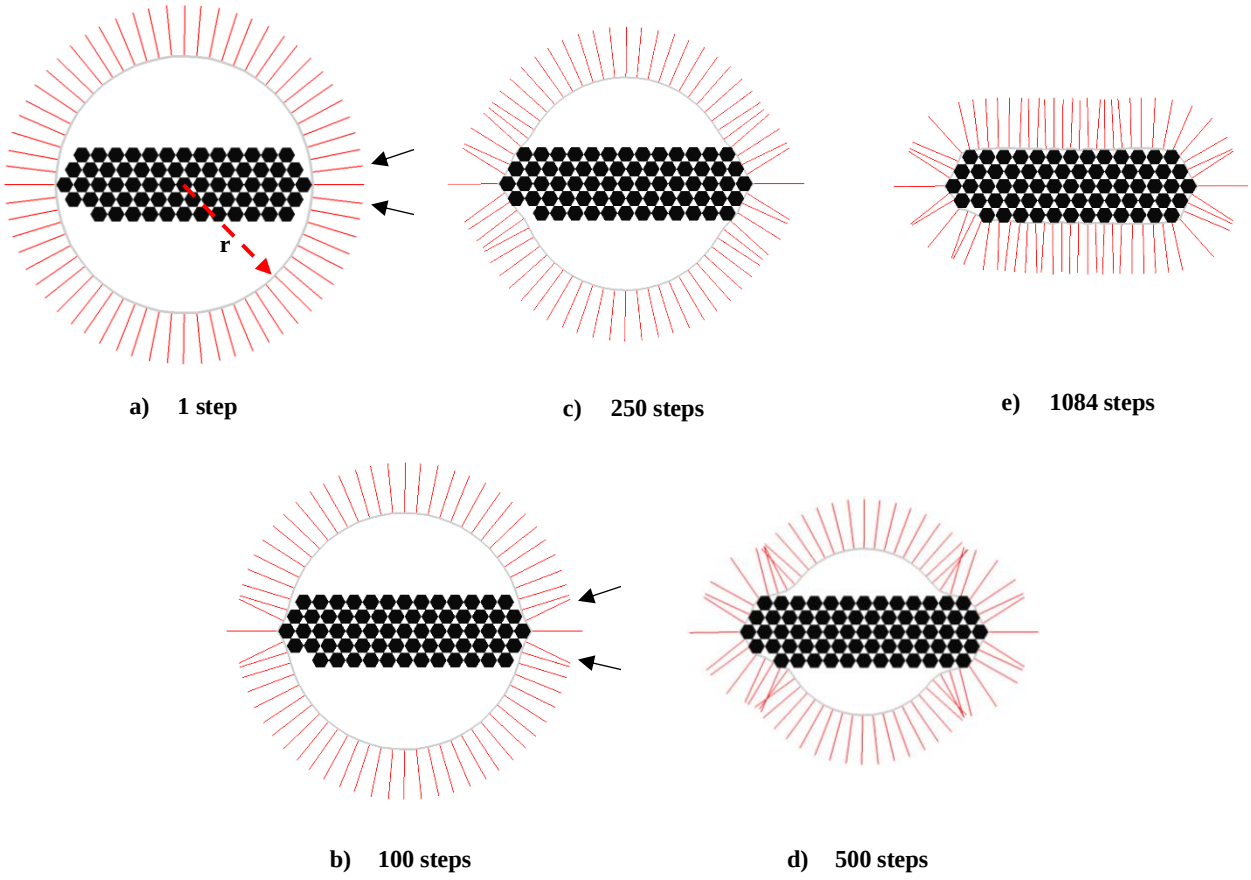
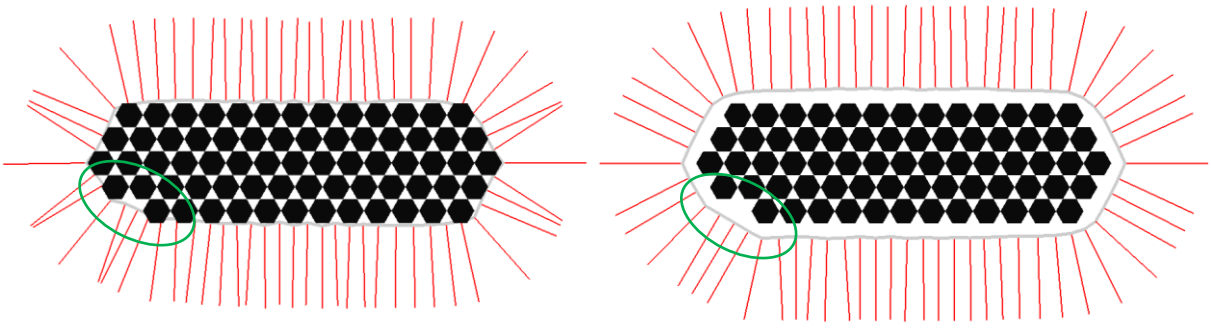


Figure 3.5: A slice of a digital chain bundle and its surface (with surface normal in red) at different iteration steps

where d_{max} is the distance to the furthest from center digital chain node, R is a tolerance factor, and D is the diameter of a a sphere in the digital chain. Figure 3.5a also shows the initial configuration of the surface and its points where the red lines indicate a surface point normal. Once this initial circle of points is created, an iterative process occurs that moves each cross-sectional point along its normal vector. The distanced moved per step is 1/100th of a digital chain radius so many iterations are required to move all cross-sectional points into contact with the digital chain bundle. Also, at each step the outward normal of a given point is recalculated by averaging tow vectors which point from the current cross-section point to the two points on either side. The effects of this calculation can be seen in figures 3.5a and b in the normal vectors indicated with the black arrows. These surface normals are consistently being changed as more iterations occur, as shown in figures 3.5b, 3.5c, and 3.5d. The process continues until each point comes within a "contact" distance (the radius of a digital chain multiplied by R) of any digital chain node. The result of this process is shown in figure 3.5e, which required 1084 iterations to complete. Figure 3.6 shows a comparison between two cases of R. Figure 3.6a is the same as figure 3.5e whereas figure 3.6b has a higher value of R. It is shown how a larger value of R increases the offset of the surface from the digital chains at the end of this step in the process. This increased R also smoothens some features



a) R = 0.1, 1084 iterations **b) R = 1.0, 1063 iterations**

Figure 3.6: Two VTMS cross-sections with varying values of R

of the surface shown by the circle in figure 3.6. There is more indention in the surface in this region with the lower *R* value in figure 3.6a compared to figure 3.6b. However, there is still one more step in the VTMS process before the surface is completed. In the final step, VTMS shrinks the surface one more time by the previously mentioned *R*-scaled distance that was previously used to determine if a surface point was contacting a digital chain node. During this step, every point is moved along its normal vector again by this set distance to eleminate the gap between digital chain bundle and surface. This process is repeated down the length of the tow until the entire tow is approximated with a surface. Then, the surfaces can be examined for interpenetrations (Appendix B).

However, the R value is not the only factor that affects how well the surface approximates the digital chains. The number of surface points also has an effect on the resulting surface. Figure 3.7 shows four different results for the same tow cross-section where the number of points for the

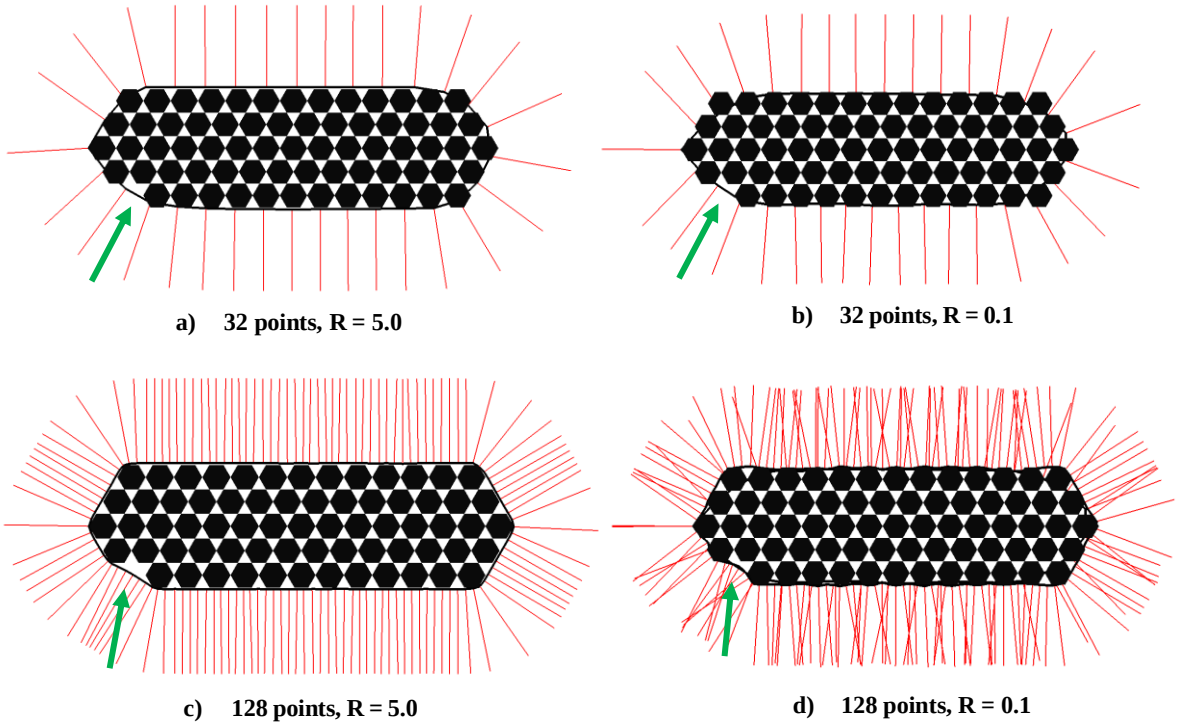


Figure 3.7: Four cases varying the two primary configuration parameters that affect VTMS surfaces

surface and the R value has been changed. It can be seen in figure 3.7a that a low number of points and a high R value approximates the surface better than in figure 3.7b, where the low R value causes the surface tomove too close to the digital chains and results in the surface disappearing into the chains. Also, neither surface in figures 3.7a or 3.7b show evidence of the intentionally removed digital chain in the region indicated with the arrows. If the number of points is quadrupled, as shown in figures 3.7c and 3.7d, the surface begins to conform to the missing digital chain. However, the lower R value in figure 3.7d helps the surface to better conform to the digital chains which can be seen in the variations of the surface normals both on the flatter tow regions and in the region where the digital chain is missing. However, none of the cases in figure 3.7 perfectly outline the digital chains and as a result there is still some extra volume being contained.

Figure 3.8 shows two hypothetical (not generated in VTMS) digital chain bundles that have contacted each other during the VTMS compaction step. The pop-out shows a close up of the cross-sections with an approximated surface line drawn and how the surfaces might interpenetrate. Actual digital chains in a tow can interact with another tow in the same manner as figure 3.8, but it is the approximation of the surface in this case that introduces interpenetrations. With a higher

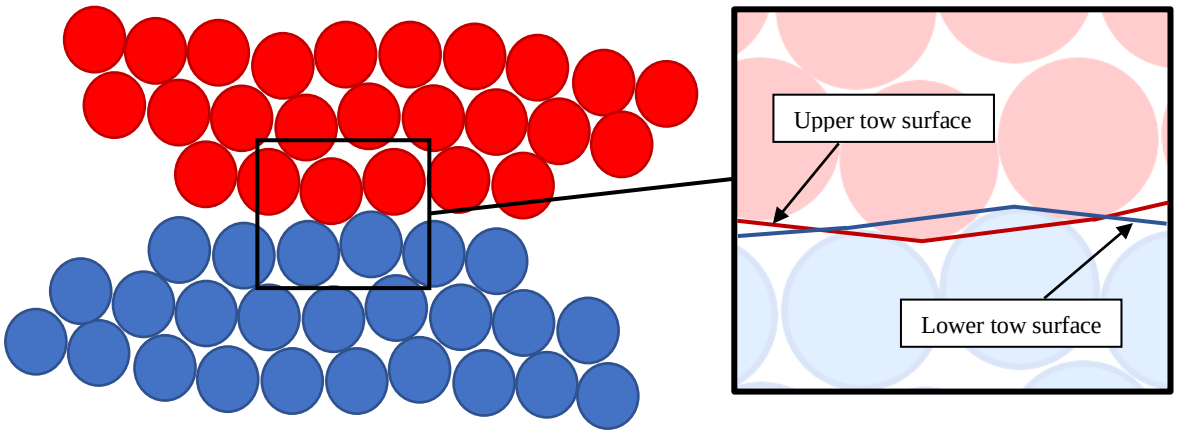


Figure 3.8: Tow bundles in close proximity. The pop-out shows lines that approximate the surface of the bundle and how they occupy the same volume.

number of surface points, as shown in figure 3.7c and 3.7d, some interpenetrations could be removed as the surface would better represent the boundaries of each exterior chain of the digital chain bundle. When less points are used, as shown in figure 3.7a and 3.7b, volume between chains on the exterior is overestimated and the approximations of the volumes can interpenetrate. If the surface better follows the boundaries of the individual chains, these overapproximations are reduced and may not interpenetrate. However, there still exists some surface interpenetrations due to the digital chains themselves. Improvements could be made to VTMS to better model the chain-to-chain interactions to eliminate the scenario shown in Figure 3.4c (interpenetration of volumetric chains). However, modifying VTMS is outside of the scope of this thesis. Instead, it will focus on fixing the surface geometry provided by VTMS for use with conventional finite element methods. Conventional FEA requires that two objects cannot occupy the same space and must have compatible meshes along any boundaries that they share.

Chapter 4 will discuss in detail the methods developed during this research to identify and correct interpenetrations. Other methods were developed that used discrete surfaces to identify interpenetrations. However, these methods were inadequate and a new detection method was required. A software library called SISL was used that can identify interpenetrations between Non-Uniform Rational B-Spline (NURBS) curves and surfaces, as well as having functions to approximate discrete surface data (like VTMSs surfaces) as a NURBS representation. The identification of this library drove the development aspect of this research and is the main factor in choosing NURBS over another surface representation type. Chapter 4 will further introduce both this library and the method developed around it to remove interpenetrations between the tows.

4. METHODS

This chapter will cover, in detail, the method used to identify and fix surface interpenetrations between tow surface approximations provided by VTMS. The SISL libaray and its importance to this research will be discussed. Next, using this library with VTMS data with this library will be covered. Surface intersection curves will be created, which will outline the interpenetration regions. These curves will be used to create new surface meshes for the tows and will also allow mesh compatibility to be enforced along where the surfaces intersect. Then, a chosen set of surface elements from one of the surfaces will be used to create a compatible mesh for in the interpenetration region. Once the tow surface meshes have been created, another method will generate a matrix mesh to create a unit cell for stress analysis. The results will be covered in Chapter 5.

4.1 Non-Uniform Rational B-Splines (NURBS) and the SISL library

During an investigation into intersection detection libraries, the SISL library from the Department of Applied Mathematics at SINTEF ICT was identified. The SISL library is a NURBS library designed for the "modeling and interrogation of curves and surfaces." [37] A NURBS surface is a type of parametric surface, which uses a curvilinear coordinate system (usually using coordinates $\boldsymbol{u}$ and $\boldsymbol{v}$) and a set of piecewise functions (called basis functions) to describe a three-dimensional surface. Hence there are only two curvilinear coordinates. For Cartesian coordinates, each of the three coordinates is calculated using an evaluation rule in $\boldsymbol{u}$ and $\boldsymbol{v}$ that uses the basis functions to evaluate the surface and return the three dimensional coordinate of a given $\boldsymbol{u}$ and $\boldsymbol{v}$, as shown in equations 4.1-3. Equation 4.4 is the equation for the position vector $\boldsymbol{r}$ of any point using curvilinear coordinates and the basis functions.

$$x = \bar{x}(u, v) \tag{4.1}$$

$$y = \bar{y}(u, v) \tag{4.2}$$

$$z = \bar{z}(u, v) \tag{4.3}$$

$$\boldsymbol{r}(u, v) = < \bar{x}(u, v),\ \bar{y}(u, v),\ \bar{z}(u, v) > \tag{4.4}$$

Figure 4.1 further shows that Cartesian coordinates for point $\boldsymbol{p}$ are related to the curvilinear coordinates via the basis function $\boldsymbol{B}$. The equation or set of equations that define $\boldsymbol{B}$ depend on the type of parametric representation used. The equations that define a NURBS surface basis function are, as previously discussed, complex. Coveniently, the SISL library handles all of the complexities without requiring a detailed knowledge of how the NURBS basis functions are created and applied.

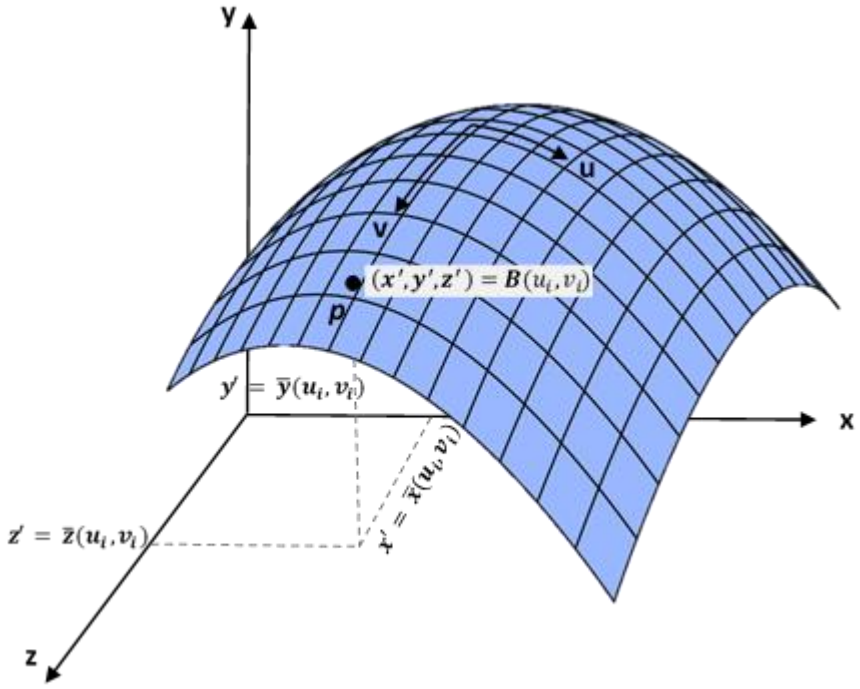


Figure 4.1: A three-dimensional surface with point p and how a NURBS basis function relates the parametric and Cartesian coordinate systems.

SISL handles the creation and implementation of the complex NURBS basis functions so that the user only needs to provide adequate surface data to use the full functionality of the library. To describe a discrete surface as a NURBS surface, SISL requires a list of points that become the control points of the NURBS surface and the number of control points in the two curvilinear directions. SISL was used for fitting a NURBS surface to the discretized VTMS surface geometry and the detection of the intersections between two NURBS surfaces. The data from VTMS is converted from the format provided by VTMS to the required input for the SISL library. The library fits a NURBS surface to each VTMS surface and calculates all the intersections between the two surfaces. The intersections are described using B-Spline curves. The B-spline intersection curves returned from the library are not always closed and may have curves that describe some or all of another intersection curve. A later section will discuss the requirement of unique and closed intersection curves as well as how the issue was handled.

Before SISL could be used properly, visualization techniques were needed to display the surfaces and study the impact of the methods developed. The next section will cover these techniques.

4.2 NURBS visualization techniques

Most visualization software suites use a collection of polygons to approximate the surface of complex shapes to render them for viewing. SISL comes with a visualization tool, but the tool lacks critical functionality to investigate the problem of interpenetrations fully. Therefore, Paraview was selected as the primary visualization software. Paraview has a large amount of functionality such as surface clipping, data filtering, and other functions that help to visualize surface interpenetrations. However, in order to use Paraview, the NURBS surfaces and B-spline curves must be discretized. After discretization, the resulting meshes can be stored for later use.

This research uses BetaMesh, which is a meshing software suite that has various classes and functions to create, store, and modify finite element meshes. BetaMesh was developed and used in-house at Texas A&M and is currently maintained by Dr. John Whitcomb, Dr. Keith Ballard, and other students under Dr. Whitcomb's direction.

The SISL library does not actively store any three-dimensional data for the surface as the library leverages the basis functions for many of its internal functions. Instead, SISL provides access and interrogation functions to return any three-dimensional the user may need. One of these functions returns the Cartesian coordinates of the surface at a chosen set of parameter values. Therefore, a grid of parameter values can be created to evenly interrogate, or query, the basis functions and define the surface in Cartesian space. The parameter grid is created by evenly dividing the range of the u parameter by the original number of cross-sections from the VTMS surface data and the range of the v parameter by the original number of points per cross-section. This results in a refinement level that is similar to the original VTMS surface. Once the parameter ranges have been evenly divided, the surface is evaluated at individual parameter pairs and the Cartesian coordinates of the surface are obtained. The interrogation function is not creating new points on the surface. Instead, the SISL interrogation function is parameter pairs to calculate the Cartesian coordinates of the surface at that parameter pair using the defined SISL basis functions discussed earlier. The entire method is shown in the flowchart in figure 4.2.

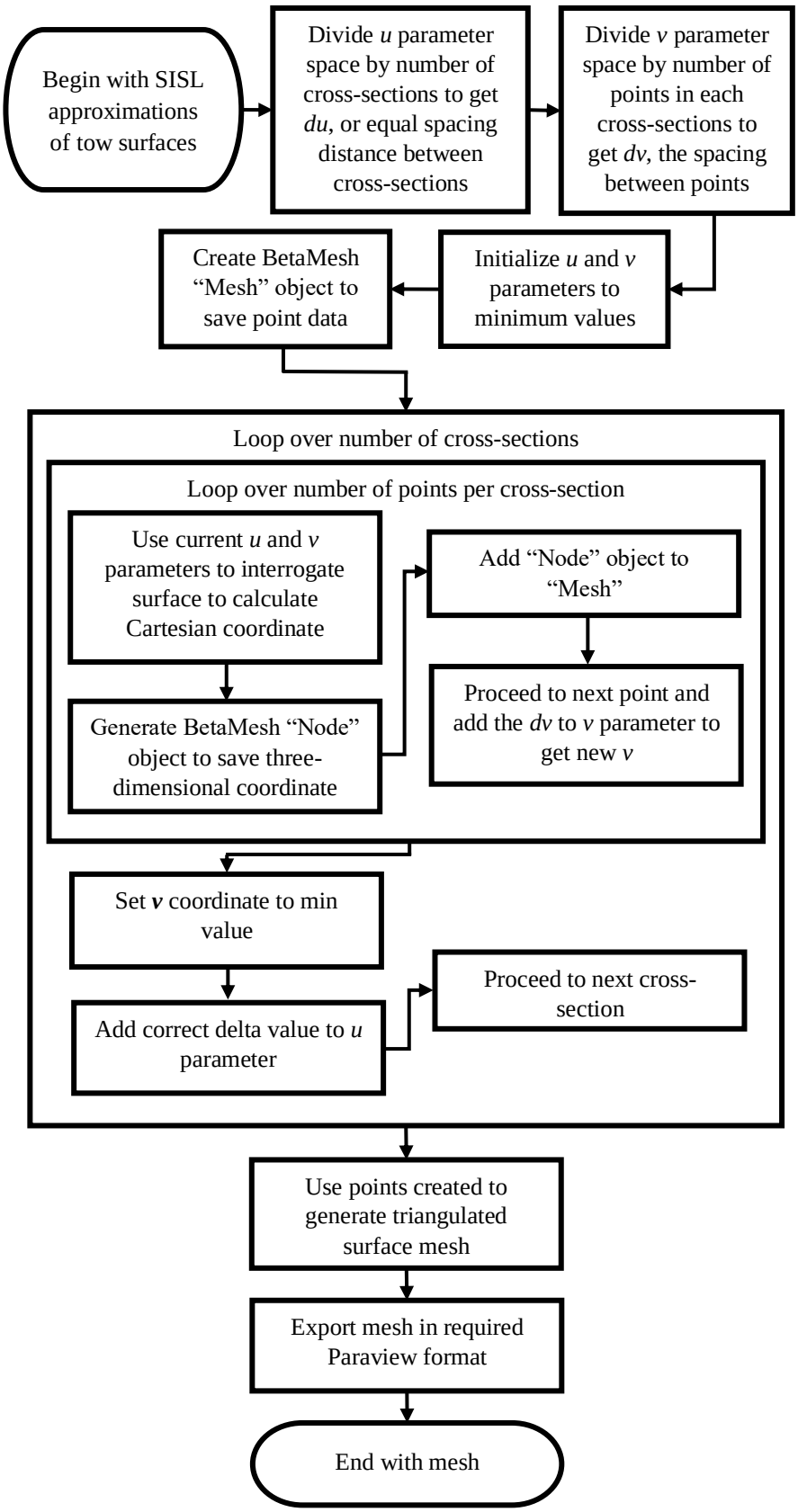


Figure 4.2: Flow chart that shows the process to create a discrete surface from a SISL NURBS surface

Figure 4.3: Curvilinear coordinate system shown for a NURBS surface in Paraview

A NURBS surface displayed in Paraview using this method is shown in Figure 4.3 with the curvilinear coordinate system shown in three-dimensions.The coordinates calculated using the surface interrogation function are stored in a format similar to the VTMS .stw format (see Appendix A). This allows for one mesh generation method to be created to display either NURBS or VTMS surfaces regardless of the origin of the point data. Once the surface is interrogated, according to the refinement chosen, a surface mesh can be created using the collection of coordinates.

Figure 4.4 shows how points can be connected to form surface elements for visualizing the surface. In Figure 4.4, the nodes labeled $< i, j >$, $< i + 1, j >$, and $< i + 2, j >$ are all the j^{th} node on their corresponding cross-sections. Every j^{th} node can be connected to the j^{th} node on the proceeding cross-section due to the circumference of the cross-section being evenly divided and the initial cross-section point being in the same position for each cross-section. This relationship

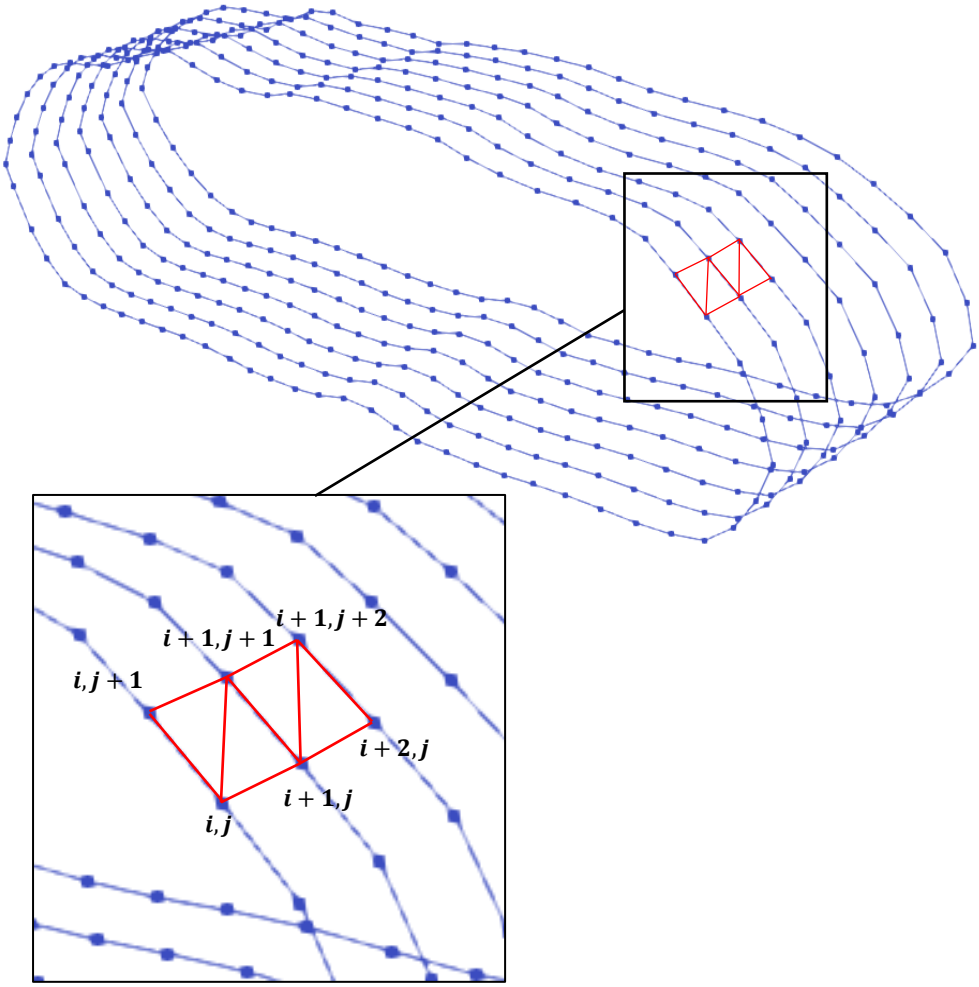


Figure 4.4: Cross-section polygons with connecting surface elements

was assumed and then visually verified. Then, the j^{th} and the $j^{th} + 1$ node on each cross-section can be connected to create quadrilateral elements formed by nodes $< i, j >$, $< i + 1, j >$, $< i + 1, j + 1 >$, and $< i, j + 1 >$. Two opposite corners of the quadrilateral are connected to create two triangles, as shown in Figure 4.4. These triangles are stored as triangular elements. These elements create a surface mesh that can then be exported by BetaMesh to the file format that is used in Paraview.

The SISL library also returns the intersection curves as B-splines, which must be discretized to be visualized using Paraview. The same function that interrogates the NURBS

surface is used to calculate a Cartesian coordinate at a given parameter. The B-spline curves only have one parameter, in contrast to the NURBS surface, which has two. Therefore, discretizing a B-spline curve is easier than a surface. As before, the range of the one parameter is evenly divided. The curve is then evaluated at a high refinement so that the linear segmented curve closely matches the B-spline curve. Once the Cartesian coordinates of the points on the B-spline curve have been collected, all the points are connected by line elements to create a piecewise linear approximation of the B-spline curve. Figure 4.5 shows an interpenetration boundary curve and an example of the refinement level of the boundary curve. Figure 4.5 also illustrates that the intersection curve approximation is at a much higher refinement than the surface approximation due to the refinement at which the intersection curve was interrogated. This refinement can be reduced and directly affects the refinement of the surface mesh around the intersection curves.

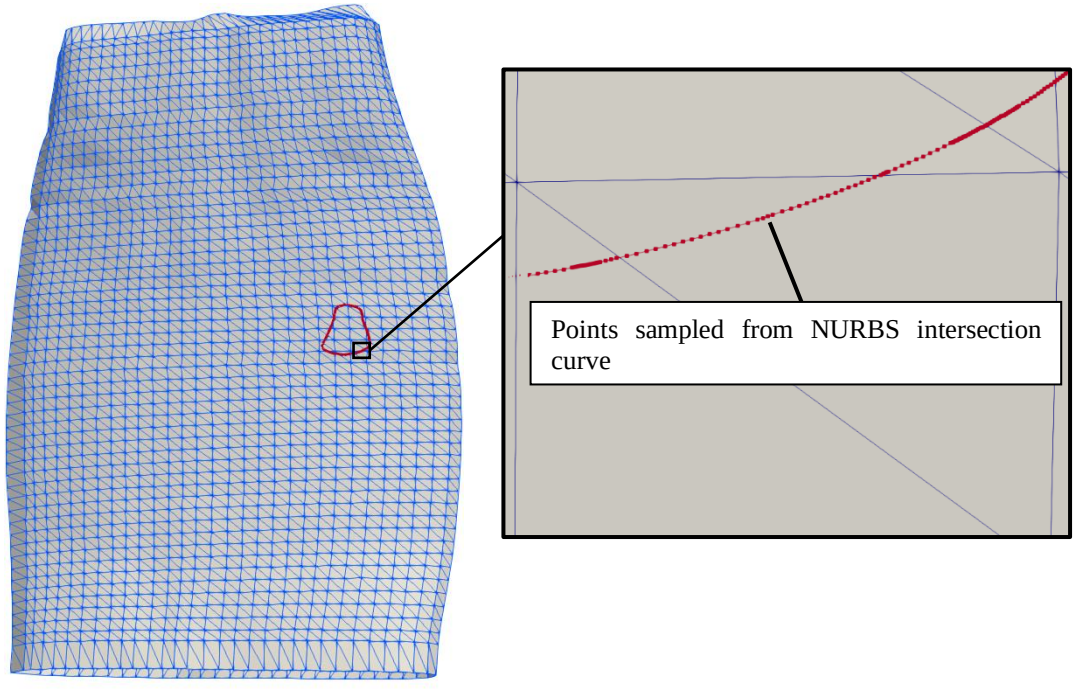


Figure 4.5: Surface mesh of NURBS surface with boundary curve and relative refinement of curve compared to the refinement of the surface.

Once a tool to visualize the SISL NURBS was developed, the VTMS surfaces can now be approximated and intersections can be calculated. The next section will discuss the process of taking VTMS data and using it with the SISL library.

4.3 Using VTMS data with the SISL library

The first step in using the SISL library is to create control points using the VTMS surface data that is exported by VTMS and format them according to SISL requirements. SISL requires a set ($\boldsymbol{S}$) of all surface control point coordinates ($\boldsymbol{p}$) in a specific order of the form

$$\boldsymbol{S} = \{\boldsymbol{p}_{i,j}, \boldsymbol{p}_{i,j+1}, \boldsymbol{p}_{i,j+2}, \dots, \boldsymbol{p}_{i,n}, \boldsymbol{p}_{i+1,j}, \boldsymbol{p}_{i+1,j+1}, \dots \boldsymbol{p}_{m,n}\}$$

$$i = 1 \dots m, \qquad j = 1 \dots n$$

where m is the number of tow surface cross-sections and n is the number of points per cross-section. The indexes i and j refer to the j^{th} point on i^{th} cross-section. Each component a coordinate are saved consecutively, resulting in a set that has $3 * i * j$ number of values. SISL also requires the number of cross-sections and the number of points per cross-section which are used to define the range of the curvilinear coordinates. From this data, the SISL library will generate a surface that is fit to the control points.

It is important to note that the resulting surface approximates the VTMS data. Generally, a B-spline curve does not intersect the control points that define it as opposed to other types of splines that do. Therefore, the NURBS surface that approximates the original VTMS surface do not run through the original VTMS surface points because now the original points are the control points for the NURBS surface. Figure 4.6 shows the original VTMS points of a single cross-section compared to the resulting NURBS curve. The figure shows that in the region where the surface curvature rapidly changes, the points that are used to control the surface do not lie on the curve. When the curvature of the surface is more constant, the points lie close to or on the curve. By

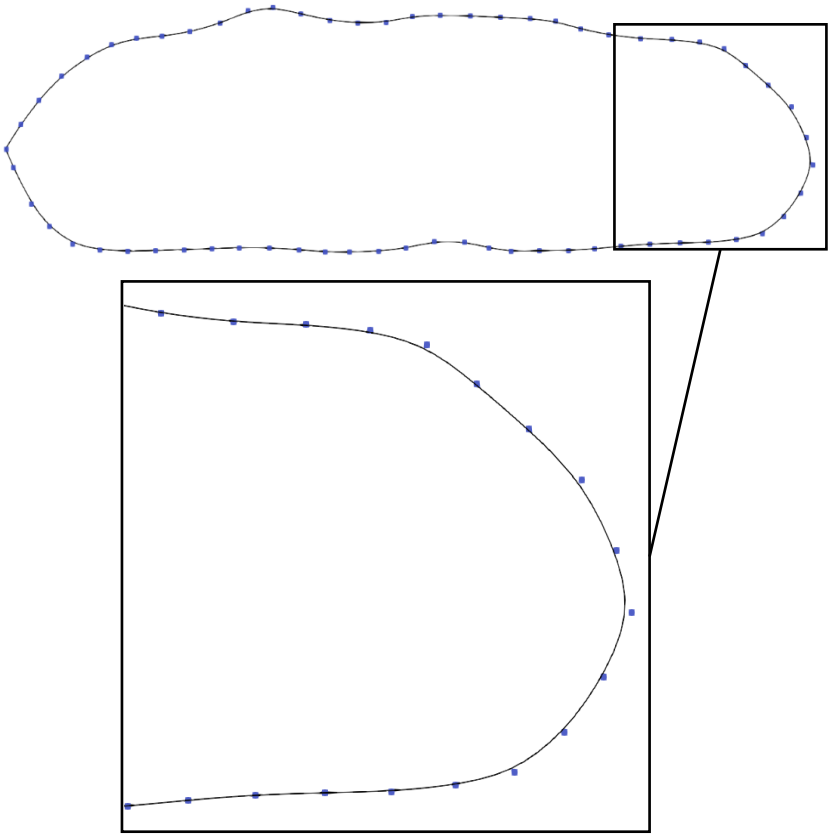


Figure 4.6: NURBS approximation of tow cross section with original VTMS data as control points

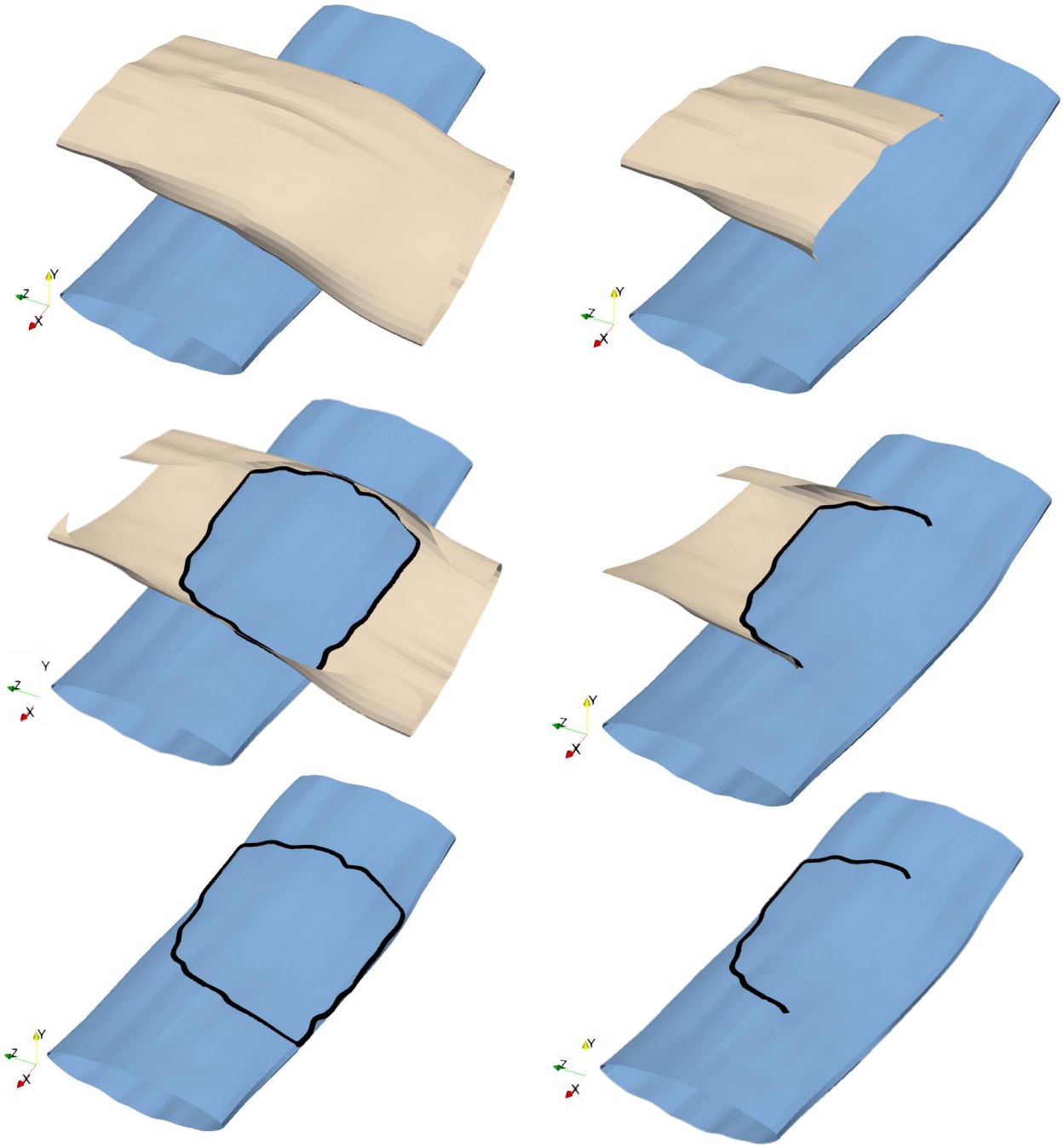
observation, the SISL library captures nearly all of the tow cross-section area as well as keeps many of the topological features (peaks and valleys of the surface) that are important for detecting surface interpenetrations. The change in tow volume during this approximation will be discussed in the results section. Therefore, the NURBS surface approximations are considered to be sufficiently accurate approximation of the VTMS surfaces and will result in accurate detection of the interpenetration regions.

Once the NURBS surfaces have been created by the SISL library, they are then used by the library to detect the interpenetration regions and return intersection curves that outline the regions where two surfaces intersect. These intersection curves between surfaces outline the interpenetration region and therefore are also considered boundary curves for the interpenetrations. The library returns these interpenetration boundary curves as B-splines which are then linearly

approximated using the methods in section 4.2. However, these linear approximations must be subjected to a validation process before they can be used to ensure that they are closed and unique. The next section will discuss why this validation must occur and how the validation process works

4.4 Consolidation of intersection curve approximations

When considering the region where two surfaces intersect, it is not safe to assume that the intersection curve between the surfaces is closed. However, if the surfaces are continuous through the region where they intersect, the intersection curve between the surfaces must also be continuous, or closed. Furthermore, there can only be one unique intersection curve for each intersection between surfaces. Figure 4.7 shows two cases of a pair of surfaces interpenetrating. In Figure 4.7a, the upper tow crosses over the top of a lower tow. The top of the upper tow is cut away to show the interpenetration region where the two tow surfaces encompass the same volume and the curve of intersection between the tows. The third picture in Figure 4.7a shows that the intersection curve is completely closed. A closed intersection curve indicates that both surfaces are continuous through the region of interpenetration. However, Figure 4.7b shows the scenario where the upper tow is terminated in the region of interpenetration. The third picture in Figure 4.7b shows an intersection curve that is not closed. Open intersection curves are due to the surfaces being hollow, and having open ends. Physically, the volume that the surface represents would have closed ends. However, modeling close-ended surface with NURBS surfaces would be difficult to implement and even more difficult for the SISL intersection detection method to accurately calculate. Instead, the textile model from VTMS is clipped, or cut, in a way such that each individual tow is ensured to not interpenetrate another where the surface terminates. If clipped properly, all interpenetration regions that exist in the model should have unique and closed intersection curves.



a.) Closed intersection curve due to both surfaces being continuous through interpenetration region

b.) Open intersection curve due to upper surface terminating in interpenetration region

Figure 4.7: Two variations of surface intersection curves and how they are formed

The main drawback of using the SISL library is that the intersection curves returned from the methods are not guaranteed to be unique or closed even if the surfaces are continuous through the interpenetration region (Figure 4.7a). Therefore, the curves need to be corrected before they are used to generate the surface mesh. Figure 4.8 shows an example of two surface intersection curves

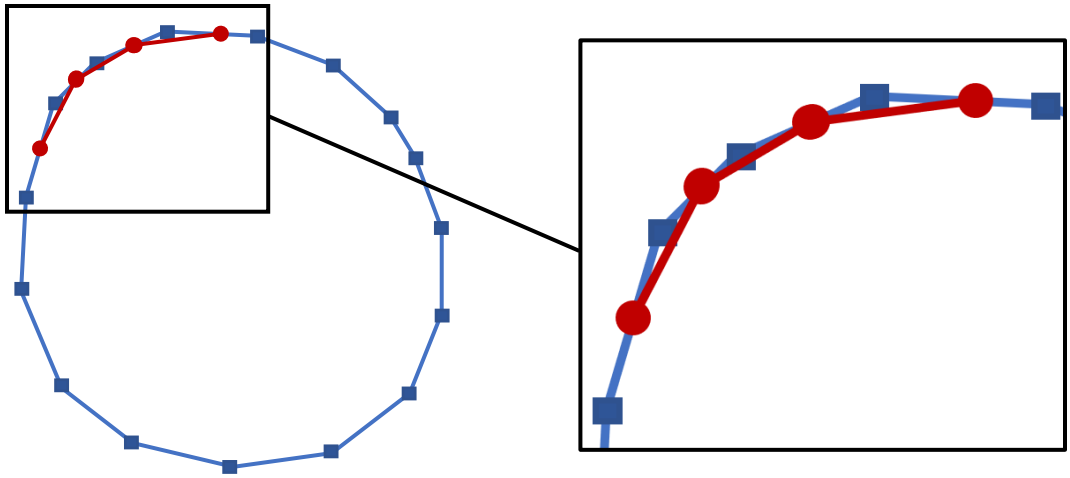


Figure 4.8: Non-unique meshed boundary curves with duplicated curve data

that were returned by SISL and converted into a line element mesh. The squares and circles indicate points that lie on the surface intersection curves calculated by SISL. These curves are described as non-unique because the smaller, open curve (in red with circles) duplicates a region of the larger, closed curve (blue with squares). There are multiple scenarios where two curves either overlap and duplicate data, or connect to create a closed boundary curve. In both scenarios, the curve data needs to be processed to create unique, closed curves so that each individual interpenetration region has one curved that describes it. In the event that a closed curve cannot be found for an interpenetration region, the method fails.

4.4.1 Systematic assembly of linearly approximated intersection curves

When the B-Spline intersection curves are linearly approximated for visualization purposes, they are also classified as being open or closed depending on whether the beginning and end points

are the same. However, even if the curve is closed, the beginning and end points are preserved so that every other point in the curve (apart from the beginning and end points) are connected to two line segments. There exists no line segement between the beginning and end points because they are in the same location. This allows for open and closed curves to be evaluated the same way when checking if a curve is unique or not.

The method to evaluate uniqueness begins by starting at the beginning or end of a curve, both of which are ideal starting points for piecing together the curve approximations from line segments. The directionality with which the curves are pieced together (beginning to end or end to beginning) does not matter. However, it is easier to start at one end of a curve than to piece the curve together starting in the middle of the curve. The base logic used to determine uniqueness of intersection curves is a check between points and line segments. Any curve that has a start or end point that lies on a line segment of another curve shares either part or all of that curve. Each intersection curve calculated by the SISL library can be checked in this manner. A method compiles all of the curve meshes into a list and performs a dual-loop iteration to determine if any of the curves are duplicates. A curve known to be closed is chosen first. Then, every other curve in the list is tested against the closed curve to determine if either the beginning or endpoint of the curve lies on any line segment on the closed curve. This test uses a dot-product and cross-product calculation to determine where the point being tested is in relation to the endpoints of each line segment of the closed curve. Two vectors are created, one that connects the beginning and endpoints of the line segment on the closed curve and a second from the beginning point on the line segment to the point being tested. The dot product "$\boldsymbol{l}$"of the two vectors will indicate if the test point lies between the two points (Figure 4.9a). If the dot product is less than zero or greater than $|v_1|$, the point does not lie between the two points along the line segment. The cross-

product of these two vectors determines the orthogonal distance "**d**" between the point and the line segment (Figure 4.9b). If the orthogonal distance is greater than a user-specified tolerance, the point is also determined to not lie on the line segment. If both of these conditions are passed, then the point and the curve it belongs to are not unique, and the curve is discarded. If the curve does not lie on the closed curve, it saved to be tested against other curves. These checks are also used to check open curves against each other to determine if two open curves complete a fully

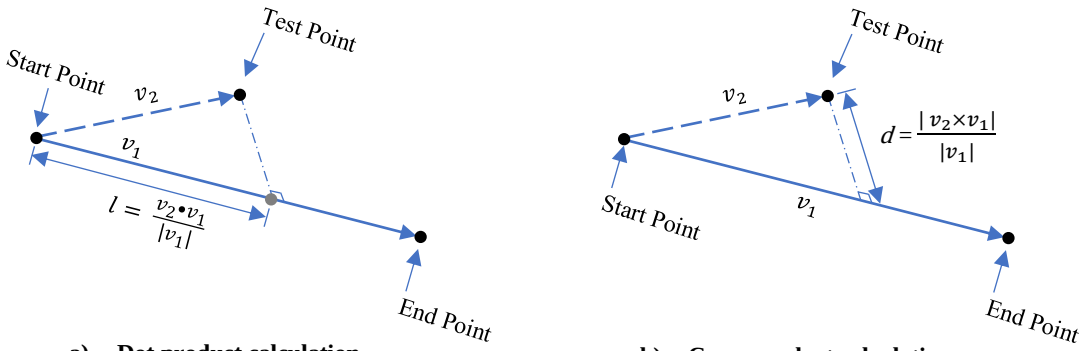


Figure 4.9*:* Show the dot and cross product calculations used to determine if a point lies on a line segment

closed curve or even a larger, open curve. If so, the curve is saved and retested against the other curves as well. The result of this method is a set of unique, closed curve meshes that are used to identify which surface elements need to be modified and removed to eliminate interpenetrations between surfaces. If at the end of the method there still exists an open curve, the method will fail and the tow surfaces will need to be evaluated to ensure that all surfaces are continuous in the interpenetration regions.

An initial method (see Appendix C) was developed that used the intersections curves to modify a pre-existing surface mesh in an attempt to identify and remove interpenetrating mesh elements. However, this method proved to be unstable. Instead, it was decided that including the intersection curves during surface mesh generation would ensure a mesh that would conform to the intersection

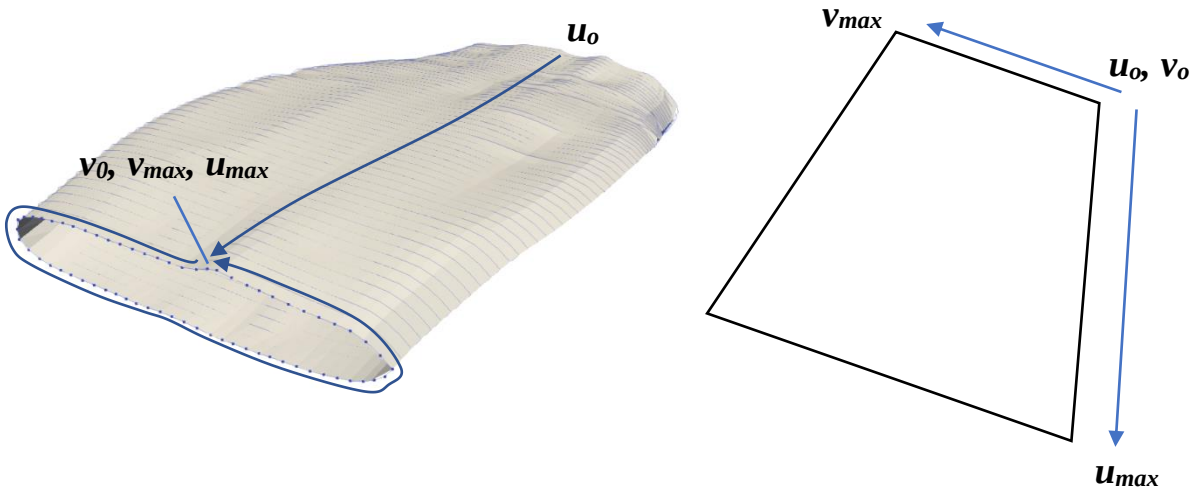
curves. Then, any elements within the curves that penetrate another surface can be removed with ease. The following sections will discuss this process in detail.

4.5 Intersection curve inclusion during mesh generation

In order to create a tow mesh that conforms to mulitple intersections curves, both the surface and the curves must be described in way that a meshing software can include both geometries during the mesh generation step. The Triangle meshing library [38], which has been added as an extension to the BetaMesh software, has the ability to create two-dimensional triangle meshes from a given set of line segments that describe both the bounds of the mesh and any internal features as well, such as holes or other artifacts. However, Triangle can *only* generate two dimensional meshes, not three dimensional meshes. Fortunately, the intersection curves and tow surfaces, which exist in three-dimensional space, can also be described in the curvilinear coordinate system, which lies in two dimensions. Therefore, this curvilinear coordinate system can be leveraged with Triangle to generate meshes in the curvilinear space. An added benefit of creating a mesh in two-dimensions is that detecting interpenetrating elements inside of the intersection curve becomes easier in two-dimensions because intersection curve segments no longer have to be projected, as in Appendix C. For these two reasons, the surfaces and intersections curves will be meshed in two-dimensions first and then converted back into three dimensions, with their interpenetrating elements having been identified.

4.5.1 Representing the Original Surface in Two Dimensions

A NURBS surface uses a series of piecewise functions to describe a three-dimensional surface or shape using only two parametric variables, as previously discussed (Figure 4.10a). These functions can be “reversed” to return the parametric coordinates of any three-dimensional point on the surface. These parametric coordinates are used to create a flattened, or two-dimensional,



a) **Parametric coordinate space of surface in three dimensions**

b) **Parametric coordinate space of surface in two dimensions**

Figure 4.10: Curvilinear coordinate system for tow input data to the SISL library

representation of the surface (Figure 4.10b). A good analogy to describe this transformation is to take a paper tube, cut it along its length, and then flatten it on a table. This "cut" is made where the parametric coordinate system of the NURBS surface begins and ends.

The meshing library Triangle requires boundary segments that enclose the region it is meshing. Therefore, the boundary of the surface in its parametric space needs to be bounded in the same manner. The SISL data structure that maintains the description of the NURBS surface in the B-spline format also contains the limits of the parametric space of the surfaces. These ranges are used as the values of u_0, u_{max}, v_0, and v_{max}. The parametric range in both the **u** and **v** directions (Figure 4.10b) is divided up by the same refinement that the original VTMS data was described (number of stacks in **u** direction, points per stack in **v** direction). A bounding rectangle is created by linking linear segments along the bounds of if the parametric space, starting from $<u_0, v_0>$, to $<u_{max}, v_0>$, to $<u_{max}, v_{max}>$, to $<u_0, v_{max}>$, then finally to $<u_0, v_0>$. The result is a rectangle made of linear segments that traces the bounds of the parametric space of the surface. These segments could then be used with the Triangle meshing library to create a Delaunay triangle mesh of this rectangle.

However, the intersection curves have still not been included in the representation so a mesh is not generated at this point.

4.5.2 Intersection Curves in a Surface's Parametric Space

As is with the surfaces in SISL, the intersection curves also have their own parametric coordinate system. While the curves curvilinear coordinate system only use one coordinate, the curves also have a parametric representation of the intersection curve in each surface's parametric reference frame. Therefore, for each intersection curve there will be four representations: the three-dimensional representation, the representation in the curve's parametric space, the representation in the parametric space of surface 1, and the representation in the parametric space of surface 2. The purpose of having the parametric representations for each surface is so the intersection curve can be represented in the parametric space of each surface. This feature will be used so that an intersection curve can be meshed with each surface. Triangle also requires that the curve be described as segments, so the parametric version of the intersection curve in discretized into linear segments. Now, the curve's linear segments can be added as boundary segments to the list of segments for Triangle to mesh. This is also an opportunity to modify the curve data, such as changing the refinement. The refinement of the curve will affect the refinement of the resulting surface mesh in the region around the intersection curves. Two levels of refinement were chosen for analysis to determine the effects of refinement on the predicted stress reponse in Chapter 5.

This method works well if the entirety of an intersection curve never crosses the bounds of the parameter space of the surface. This is not always the case. The surfaces are continuous along the ***v*** direction in three-dimensions, but not in the parameter space. There is a discontinuity where the parametric space ends and begins again. The curves can cross this discontinuity, so the algorithm must account for that. Referring to Figure 4.11, as an intersection curve (dashed line,

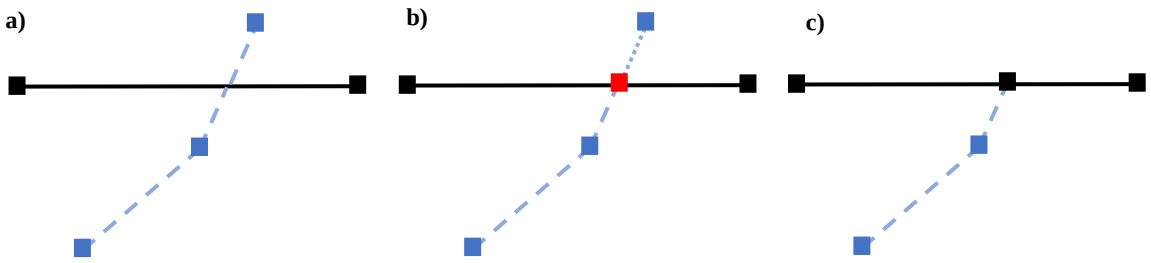


Figure 4.11: An illustration of how intersections between surface intersection curves and the boundary segments of the parametric space of a surface are resolved.

Figure 4.11a) crosses the surfaces parametric bound (solid line), it creates a point of intersection of line segments (red marker, Figure 4.11b). The endpoint (or start point, depending on direction) of the intersection curve is moved to this point (Figure 4.11c). The SISL library does this when it creates the parametric representation of the intersection curves in each surface. Therefore, any curve that intersects this discontinuity in the parametric space is divided into two, open curves. When the curves are being added to the surface boundary segments, each curve segment point is checked to see if it lies on a surface boundary segment. If so, the surface boundary segment is subdivided at this point. This ensures that mesh compatibility is maintained for the future surface mesh in three dimensions (when the surface becomes continuous). In two dimensions, adding the

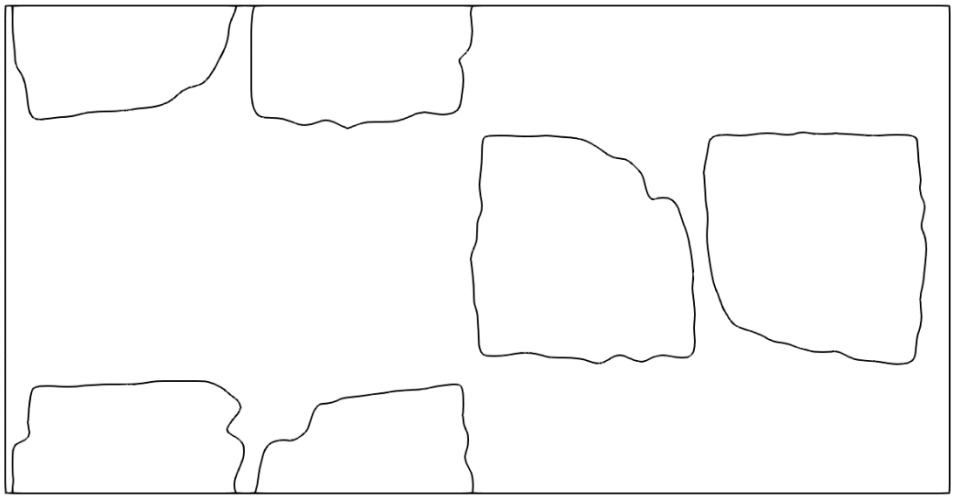


Figure 4.12: Result of combining the boundary segments of a surface's parametric boundary and the segments of multiple intersection curves' parametric descriptions for that surface

intersection curves looks like Figure 4.12. Figure 4.12 shows intersection curves that lie entirely inside the parameter range of the surface and curves that cross the limits of the parameter space.

4.5.3 Meshing the Boundary Segments Using Triangle

Once all the appropriate boundary segments have been collected, these segments are provided to the Triangle library so that a triangle mesh can be created. The resulting mesh refinement away from the intersection curves can be controlled by setting a maximum triangle area and minimum angle value. The maximum area is set by simply calculating an area based off of the surface boundary segment length. If this value is not set, Triangle will create large surface elements in regions where there are no boundary segments and there will be large variations in surface element size. This would cause potential issues when trying to approximate the three-dimensional representation.

The result of these triangle settings can be seen in Figure 4.13. Figure 4.13 shows how the triangles are uniform in size, except for the triangles surrounding the intersection curves. The

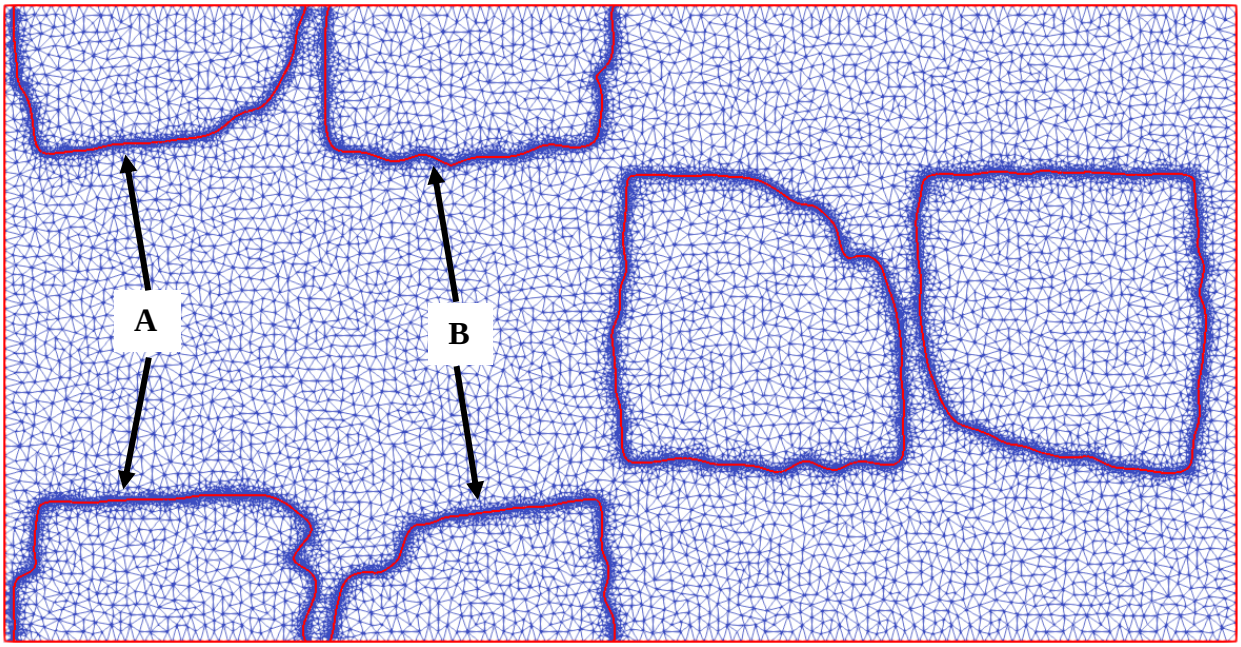


Figure 4.13: Result of Triangle meshing library when including the surface bounds and surface intersection curves.

higher refinement is due to the refinement of the intersection curves. If a lower level of refinement is desired, the linear approximation of the curves can be adjusted. The results of a high and low refinement for the intersection curves will be discussed in the results section.

4.5.4 Parametric Space into Three Dimensions

Now that the surface parametric space has been discretized into a triangular mesh, a transformation must be made from the parametric space to the three-dimensional space. However, it is important to understand in a general sense what will occur during this step. The parametric space can be represented as a two dimensional flat plane where the length and width are the range of the parametric space in each direction. This rectangle is shown in Figure 4.14a as a two-dimensional rectangle in three-dimesional space to aid in the visualization of the two-dimensional to three-dimensional transformation. The actual domain of the parametric space is perfectly retangular and discontinuous in both directions. However, the three-dimensional surface being

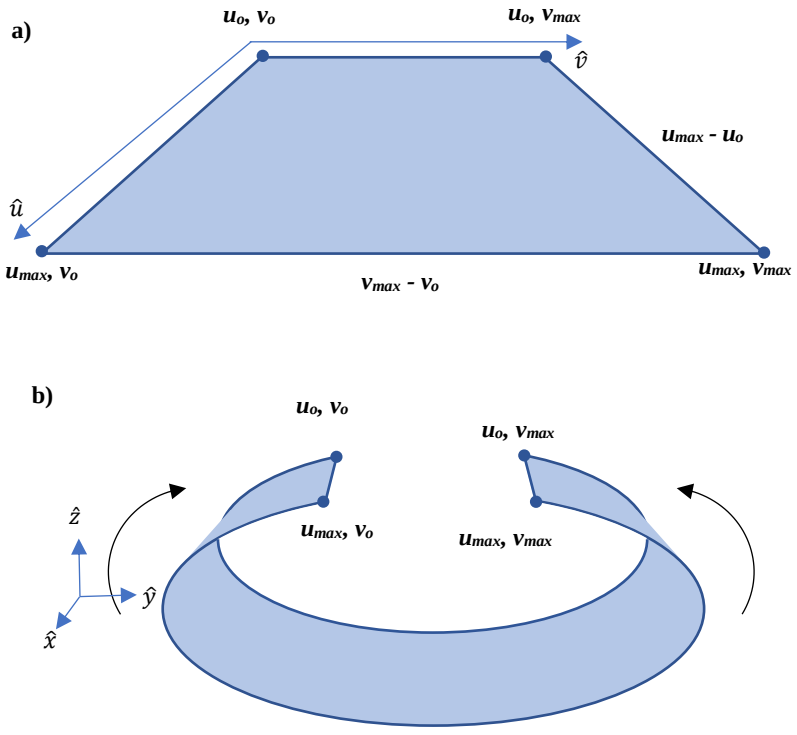


Figure 4.14: Transformation of the parametric coordinate system (a) to the Cartesian space (b) for a tow surface

described is still continuous in the circumferential direction. Therefore, as the mesh is converted from the parametric space to the three-dimensional space, the discontinuity of the parametric space will appear in the mesh. Figure 4.14b shows a physical representation of parametric discontinuity becoming continuous in three-dimensions as the flat plane in Figure 4.14a is being rolled into a tube such that the tube will be continuous in three-dimensions. As the edges of the mesh come together, continuity of the surface and compatibility of the mesh is enforced.

To accomplish this transformation of the mesh, each node from the two-dimensional mesh in Figure 4.13 has a set of coordinates that describes that nodes location in the parametric space. The SISL library has the functionality to calculate a point on the surface in three dimensions using any parametric coordinate pair. These nodes coordinates are used to find their corresponding location on the surface in three dimensions. The same triangular elements are used but their node coordinates in the parametric space (two dimensional) are replaced with three-dimensional coordinates that lie on the surface. The result, shown in Figure 4.15, is a discretized surface representation with the intersection curve linear approximations included in the surface mesh. Figure 4.15 is the three-dimensional result of Figure 4.13. The intersection curves A and B shown in Figure 4.15a directly correspond to the curves A and B in Figure 4.13. This is confirmed if the topology of the curves is visually compared. Figure 4.15 also has a mesh artifact on the top of the surface (indicated by the arrows in Figure 4.16) where mesh element edges line up down the length of the tow. This artifact is the result of the process shown in Figure 4.14. Each tow mesh in the model will have a similar artifact because each tow has a two-dimensional mesh that is used to create it's three-dimensional mesh. Although these artifacts exist down the entire length of the tow mesh, they had no noticeable effects on the predicted stress response of the model.

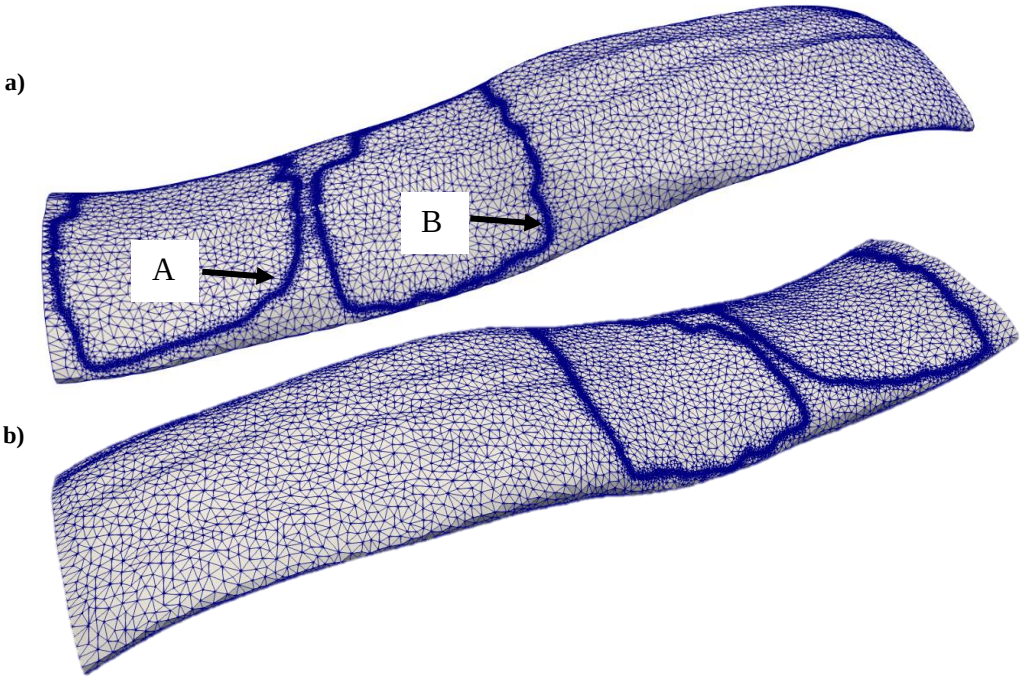


Figure 4.15: Three-dimensional mesh as a result of converting parametric surface mesh coordinates to Cartesian coordinates

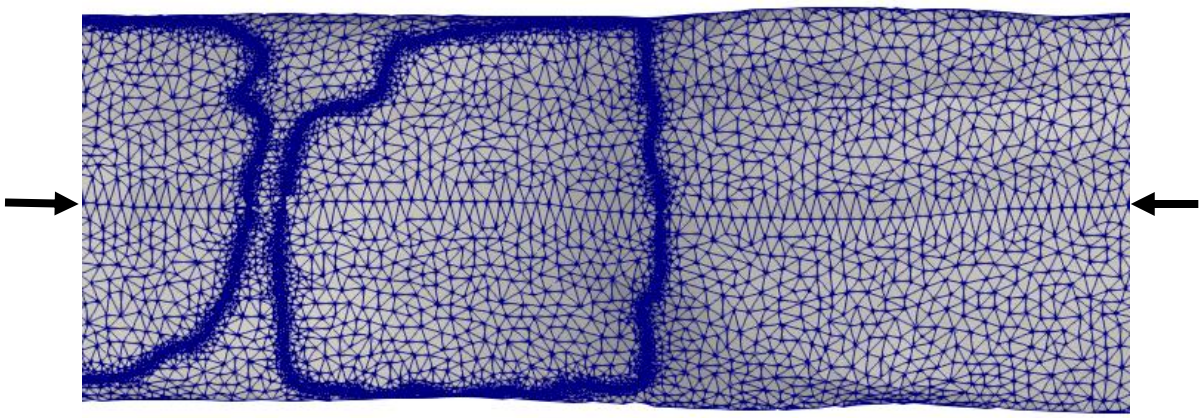


Figure 4.16: Arrows indicate ends of mesh artifact due to beginning/end of parametric space in three-dimensions

Figure 4.17 shows a region with two intersection curves and the resulting mesh in closer detail. It can be seen that there are no sliver (extremely narrow) surface elements and the mesh is fairly uniform in this region. Therefore, it can be seen that even high refinement intersection curves do not prohibit quality mesh generation. This shows that including the intersection curves in the meshing process can create useful meshes without any problematic surface elements.

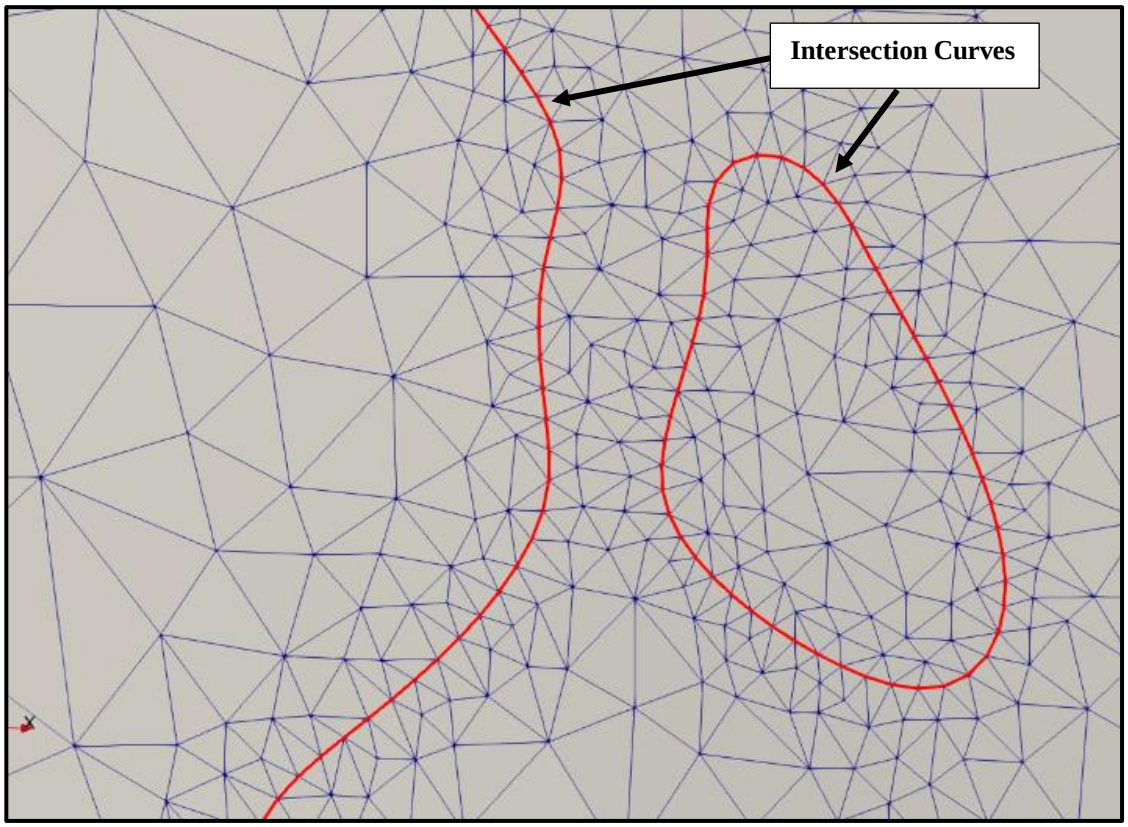


Figure 4.17: Two intersection curves of two separate interpenetration regions in close proximity.

4.6 Restructuring the Surface Meshes

Now that the surface meshes have been generated, the regions of each surface mesh that interpenetrate need to be removed. The best way to remove these interpenetrations is to identify which surface elements lie within intersection curves and remove them. In the original method implemented, a complicated algorithm that used the three-dimensional surface meshes was

prototyped to identify possible interpenetrating elements near the intersection curve and test them individually. This algorithm was prone to false positive element detections due to the three-dimensional nature of the surfaces themselves. Therefore, a different method was implemented that uses the two-dimensional representation of the surface meshes in the parametric space and a ray casting algorithm. In the parametric space, the curves will not need to be projected onto the surface mesh because both the curves and the surface lie in the same plane. The lack of any projections will greatly increase the accuracy and robustness of the method. Once any elements that are identified as being contained inside of an intersection curve in the parametric space, they can be removed in the three-dimensional space.

4.6.1 Ray casting method and identifying elements

In order to determine if a surface element lies within the intersection curve, a ray cast method was implemented. While there are many implementations of the ray casting method, the premise is simple. Given a point and a polygon in two dimensions, how can one determine if the point lies inside, or outside, of the polygon? Consider Figure 4.18:

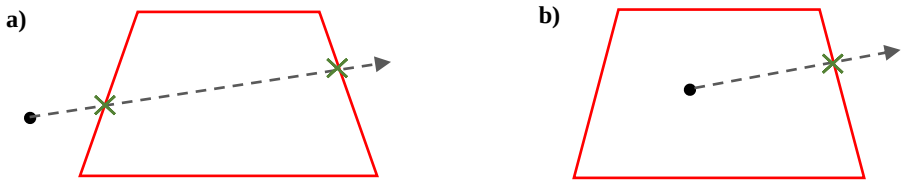


Figure 4.18: Test cases for ray cast method: a) Exterior point, even number of intersections and b) interior point, odd number of intersections

In Figure 4.18, there are two scenarios. In both scenarios, a ray is extended in a chosen direction and the number of times the ray intersects an edge of the polygon is calculated. Per the algorithm, containment of the point in the polygon can be determined by the number of intersections. If the number is even, the point is not contained in the polygon (Figure 4.18a). If odd, then the point is

contained (Figure 4.18b). Figure 4.18 shows two simple examples with one point and a four-sided polygon. However, the premise of the ray cast method can be easily upscaled to fit the needs of this research. The main difference between the base idea of this method and the implementation for this research is that instead of a single point, an entire element needs to be checked for containment. Therefore, another test case will be created and discussed.

Figure 4.19 shows a new scenario with multiple elements and a boundary curve. Point **p** represents the average position of element **E**'s vertices: **a**, **b**, and **c**. After the average position has been calculated, the same ray cast method can be used to determine if the point **p** is inside of the intersection curve by creating a ray that points in a randomly chosen direction and calculating the

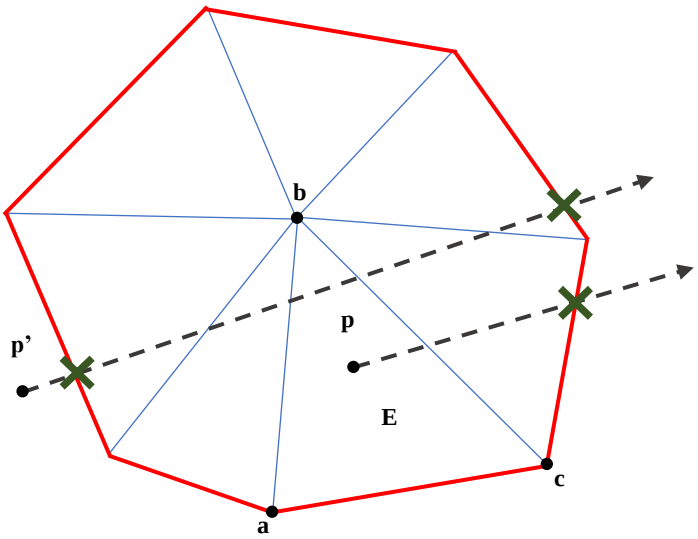


Figure 4.19: Elements inside of an intersection curve being tested for containment via the ray cast algorithm

number of intersections with the intersection curve. In the case of point **p**, there is only one intersection point for a ray in any given direction. Point **p'** illustrates an alternate case where a point may lie outside of the intersection curve. In that scenario, the algorithm calculates two intersection points. Another result for point **p'** is when the ray direction is reversed and there are no intersections. Therefore, the ray cast method treats zero intersections and an even number of

intersections the same. The average of the element's vertices can be used because each surface element is a triangle which is guaranteed to be convex and therefore the average of the vertices will lie within the bounds of the triangle. The Triangle library ensures that all intersection curve line segments coincide with an element edge and that no element edges can cross the intersection line segments. Therfore, there are no elements that can lie partially inside of an intersection curve. It then follows that any average of an element's vertex positions can be used to determine if an element itself lies inside of the intersection curve. By repeating these calculations for every element in the surface mesh, the elements that lie inside of the perimeter of the intersection curve can determined.

4.6.2 Removing and replacing elements

After each surface's set of interpenetrating elements have been identified in the parametric space, the same elements in three-dimensions can be identified. Figure 4.20 shows a parametric and three-dimensional representation of a surface with the interpenetrating elements identified. Once the elements have been identified, they can be removed from both the parametric and three

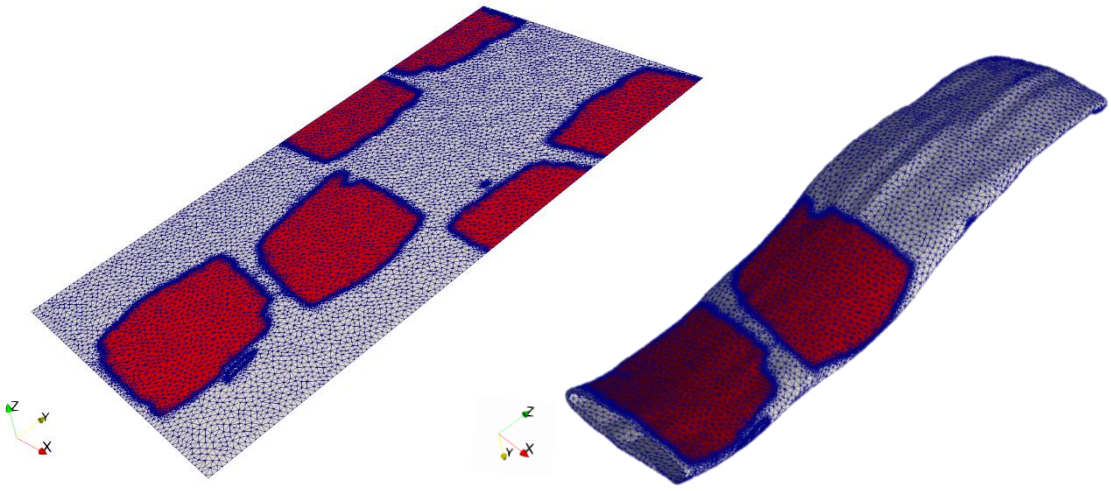
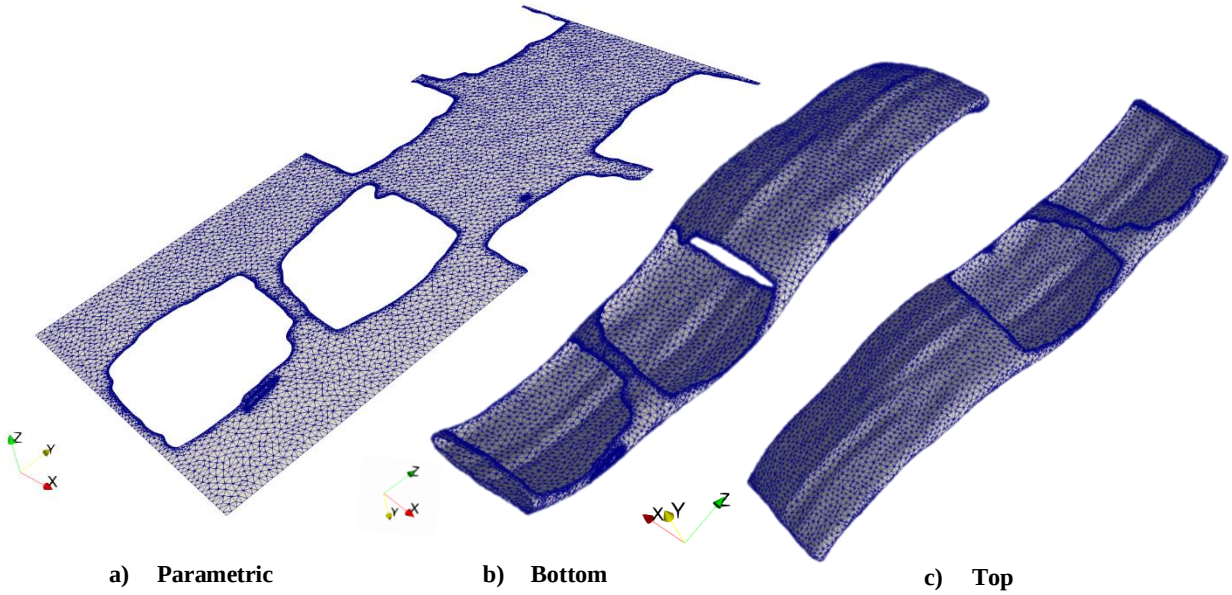


Figure 4.20: Parametric and three-dimensional representation of a surface mesh with interpenetrating elements identified



a) Parametric b) Bottom c) Top

Figure 4.21: Parametric and three-dimensional representation of a surface mesh with interpenetrating elements removed

dimensional representations. Figure 4.21 shows both representations with their elements removed. The removal of the three-dimensional surface elements can be verified by the white showing through from the elements on the bottom of the surface as well.

For each surface pair that intersect each other, there is a primary tow and secondary tow assigned and the secondary tow has its elements removed. Then, the interpenetrating elements

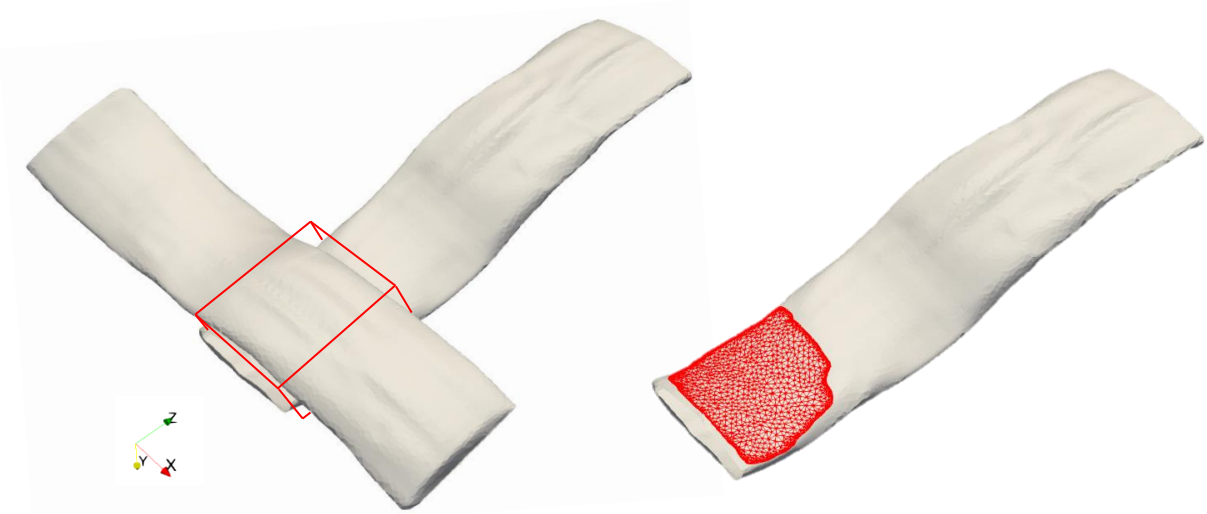


Figure 4.22: Two tows that interpenetrate in the region specified. The lower surface is shown with its interpenetrating elements.

from the primary surface are inserted into the secondary mesh. Consider the intersection between the two tows in Figure 4.21 in the marked region and the interpenetrating elements shown for the bottom (in this case secondary) surface. If the interpenetrating elements from the upper (blue) and lower (red) surface are laid upon one another, as in Figure 4.22, it can be seen that the pattern of the elements are similar, but different due to the random nature of the Triangle mesh generation away from the set boundary segments. If they were an exact match, only one color could be seen. The region enclosed by the two surface regions in Figure 4.23 is the exact volume of interpenetrations between the two tow surfaces in Figure 4.22. The pop out in Figure 4.23 shows that along the boundary of the interpenetration region, each element set share the same points and edges. This is the result of using the linear approximation of the intersection curve in each surface. Finally, the set of elements from the secondary surface (red) can be replaced with the primary surface elements (blue) and still maintain mesh compatibility for that surface mesh (Figure 4.24).

Now that the elements have been inserted into the secondary mesh, both surfaces share the exact same element configuration which creates a surface where the interpenetration occured. This

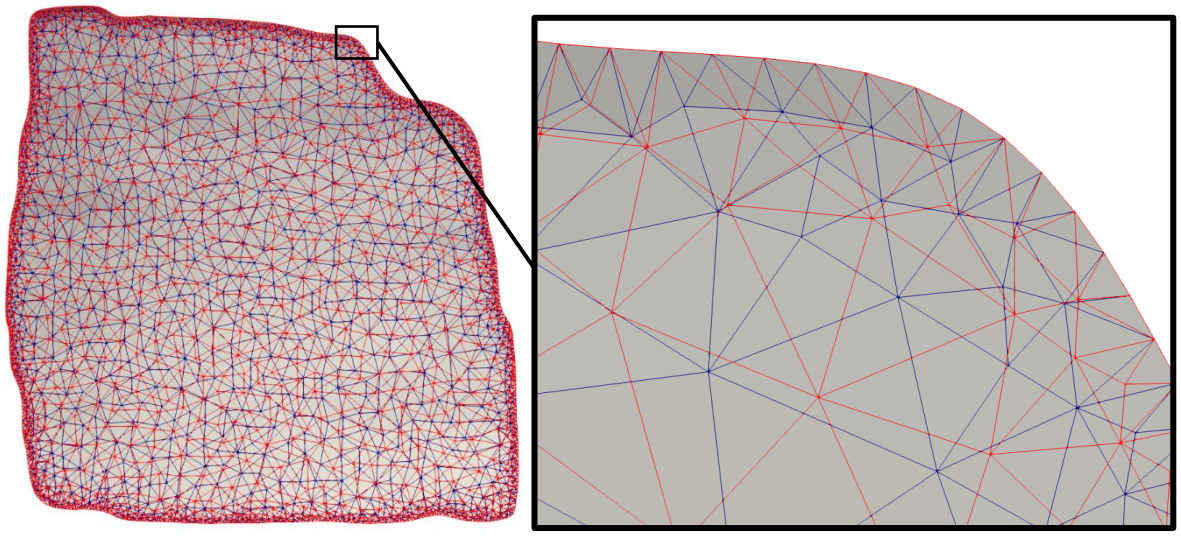


Figure 4.23: Elements identified as interpenetrating from primary (upper, blue) and secondary (lower, red) surfaces in Figure 4.22

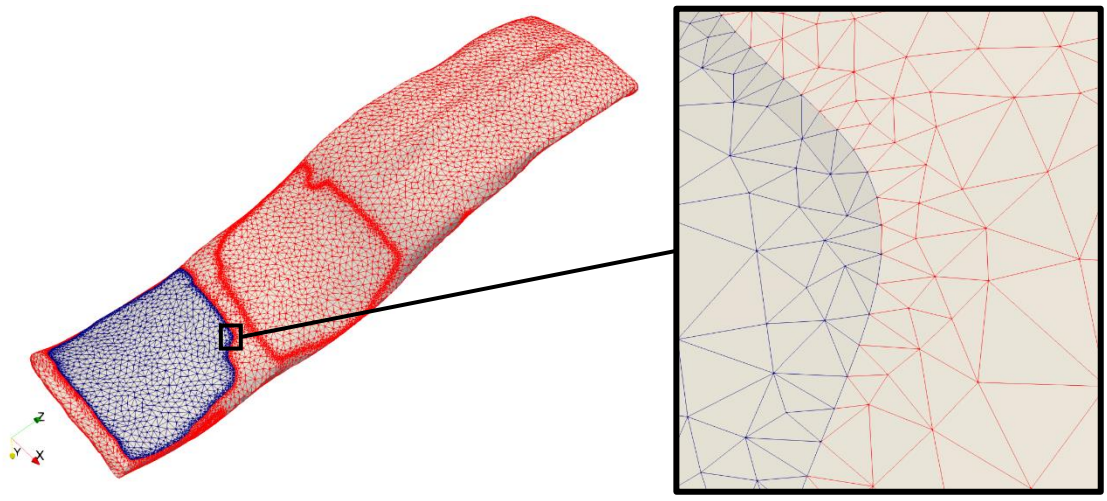


Figure 4.24: Secondary (lower) surface from Figure 4.22 with primary surface (upper) elements inserted

will be refered to as the connected tow mesh region. In this region, both primary and secondary tows have mesh compatibility and the two tows share this set of nodes. If this constraint is not desired, the compatiblity between surfaces allows for any model, such as an elastic bond or cohesive zone model, to be applied to these nodes during an analysis. From here, the volume meshes for the tows and surrounding matrix needs to be created and the model will be really for an analyis. For the purposes of this research, the software BetaMesh is used to generate a compatible matrix mesh as well as the volumetric meshes for the tows. BetaMesh generates tetrahedral volume meshes for both the tows and the matrix so that the matrix will fully encompass

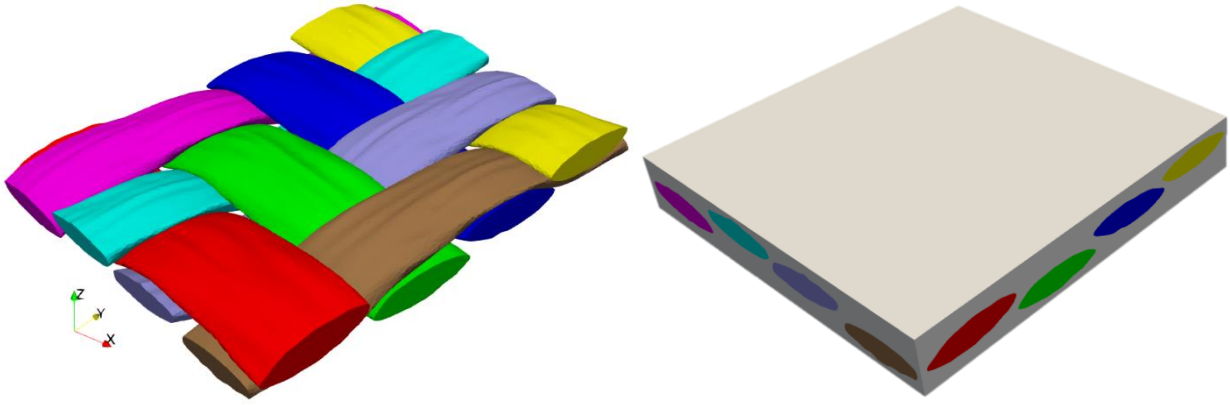


Figure 4.25: Tow volume meshes only and with the matrix included

the tows plus a small percentage extra in the out-of-plane direction. The resulting meshes are shown in Figure 4.25. It is shown that the tows extend to the edge of the matrix in the x- and y-directions but not the z-direction. The matrix has been sized in this manner to avoid the tows being too close to the z-direction edges in the matrix which could result in bad mesh quality in the matrix mesh. Now that all of the relevant meshes have been generated and compatiblity between them has been enforced, an analysis can be conducted using any traditional finite element software.

The next chapter will discuss, in detail, the results of three different analyses. Two of the analyses use meshes generated by the NURBS method presented in this research, but have different level of refinements for the intersection curve. The third analysis will use the shrunken tow method that has been a proven method to remove interpenetrations to generate surface meshes. The purpose of the different models is to observe the similarities and differences between the NURBS removal method and the shrunken tow method. Also, the effects of intersection curve refinement will be explored. Comparisons between the model geometry and predicted stresses will be made to better understand how each method affects the predicted response of the model. The results presented are not to be considered a comprehensive exploration of the response of a twill weave model.

5. RESULTS

The results presented in this chapter use three variations of a twill weave configuration. The differences between these configuration will be discussed in two separate sections. First, the geometrical differences between each variation of the twill weave model will be covered. Next, the model configuration of each analysis will be discussed, including the material properties, boundary conditions, and meshes used. Then, the results of a conventional finite element analaysis for each model will be compared to understand the effects of the differences in geometry. The goal of these results is to understand the effects of the geometrical differences on the predicted stress response of the composite.

5.1 Geometry

The tested section is four tows by four tows. The four-by-four section of tows has been clipped from the center of a larger VTMS generated twill weave that was eight-by-eight tows (Figure 5.1) to eliminate any effects due to the edge of the VTMS model. The region is also clipped in such a

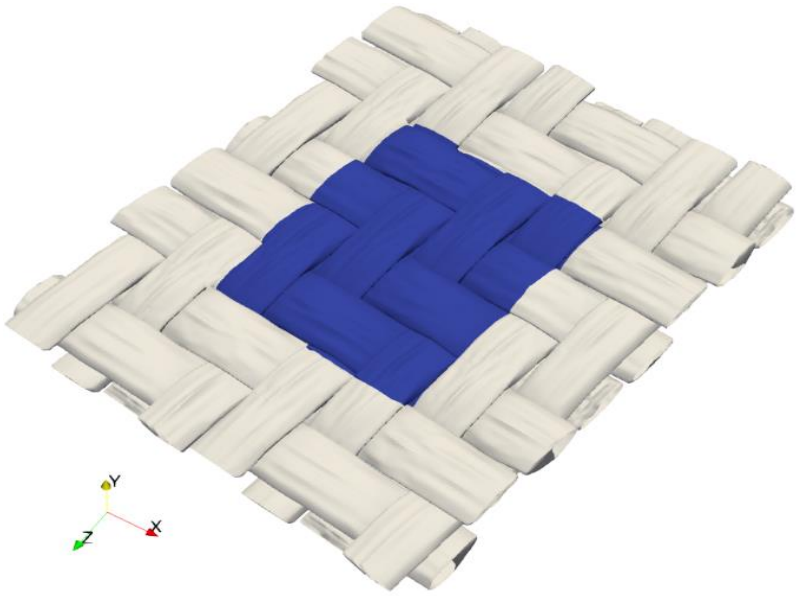


Figure 5.1: Large 8x8 twill weave with clipped region used in analysis shown

way that it could be used as a unit cell for this twill weave configuration. The initial VTMS data is then modified by one of two methods:

- The shrunken tow method, which shrinks tow cross-sections where interpenetrations are identified to remove said interpenetrations
- The NURBS method, which has been implemented in this thesis

Each method starts with the exact same surface data to reduce any potential variations due to different initial geometry. A third model is generated using the NURBS method but uses intersection curves that have a much lower refinement than the first NURBS model. The purpose of the lower refinement intersection curve is to understand the effect of mesh refinement in regions that may experience high stress concentrations. Figure 5.2 shows the dimensions of the models and tow numbering scheme used in the following sections.

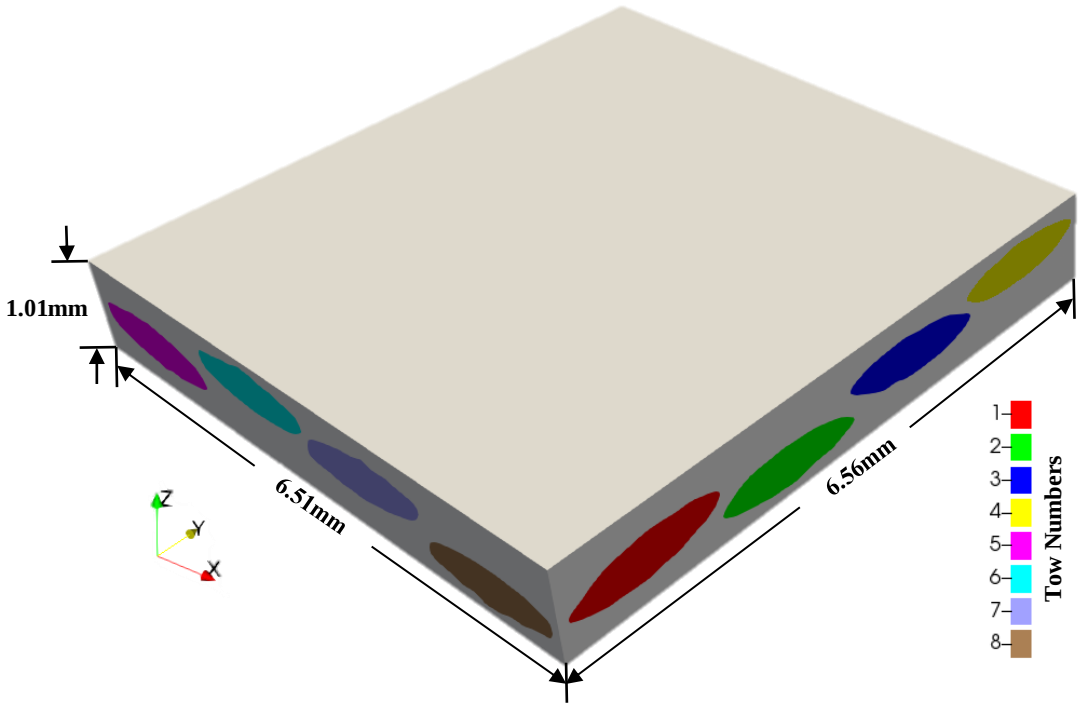


Figure 5.2: Analysis region with dimensions

Table 5-1: Tow volume and percent difference between methods compared to original

Method	Tow 1	Tow 2	Tow 3	Tow 4	Tow 5	Tow 6	Tow 7	Tow 8	Total
Original (mm^3)	1.4343	1.4199	1.4221	1.4238	1.4376	1.4292	1.4381	1.4256	11.4304
Shrunk (mm^3)	1.3729	1.3578	1.3601	1.3614	1.3763	1.3681	1.3770	1.3648	10.9392
NURBS (mm^3)	1.4275	1.4137	1.4157	1.4174	1.3567	1.3490	1.3583	1.3462	11.0845
NURB to Shrunk	3.83%	3.95%	3.86%	3.95%	-1.45%	-1.42%	-1.37%	-1.38%	1.33%
NURB to Original	-0.47%	-0.44%	-0.46%	-0.45%	-5.62%	-5.61%	-5.55%	-5.57%	-3.03%
Shrunk to Original	-4.28%	-3.95%	-4.30%	-4.38%	-4.26%	-4.27%	-4.25%	-4.26%	-4.30%

Each method modifies the initial VTMS surfaces and therefore the overall volume. The change in overall tow volume is an indicator of the level of modification done by each method and which tows are modified. Table 5.1 shows the change in volume for both the shrunken and NURBS method compared to the original volume of each tow in the model. The difference in volume between high and low refinement cases of the NURBS surfaces is negligible.

Table 5.1 shows that both methods significantly modify tows 5-8. The NURBS method has a larger reduction of volume in these tows because they are secondary tows, from which the interpenetrating volume of both tows is removed. The primary tows are unmodified. The shrunken method reduces both the primary and secondary tow volume. This relationship between primary and secondary tows in each method is further shown by the volume of tows 1-4. In the NURBS method, the volume is only slightly reduced due to the approximation of the discrete VTMS surfaces as NURBS surfaces. However, the shrunken method reduces the volume in these tows in a similar amount to tows 5-8 because it shrinks the cross-sections in both tows. This leads to a more uniform volume reduction in all tows, but a higher total volume reduction compared to the NURBS method. Table 5.2 further shows these relationships for major interpenetration regions (four per tow). Positive percentages indicate more volume for the NURBS method in that region.

Table 5-2: Tow volume difference between NURBS and Shrunk methods in interpenetration regions

Tow	Region 1	Region 2	Region 3	Region 4
1 (x-direction)	4.1285%	3.9518%	3.7012%	3.6000%
2 (x-direction)	3.7966%	4.0357%	4.1027%	4.1004%
3 (x-direction)	3.8176%	3.7556%	3.8735%	4.0406%
4 (x-direction)	3.8931%	4.0728%	3.9683%	4.0872%
5 (y-direction)	-1.5037%	-1.5239%	-1.4778%	-1.7923%
6 (y-direction)	-1.6128%	-1.3856%	-1.3410%	-1.5926%
7 (y-direction)	-1.4644%	-1.6995%	-1.5928%	-1.4442%
8 (y-direction)	-1.1035%	-1.4670%	-1.4266%	-1.5403%

To better understand how each method affects the tow geometry, Figure 5.3 shows how the cross-sectional area varies along the length of a tow for a primary tow (a) and a secondary tow (b) for both methods compare to the original data. It can be seen how for primary tows, the NURBS method closely follows the original data with a slight cross-sectional decrease at every point along

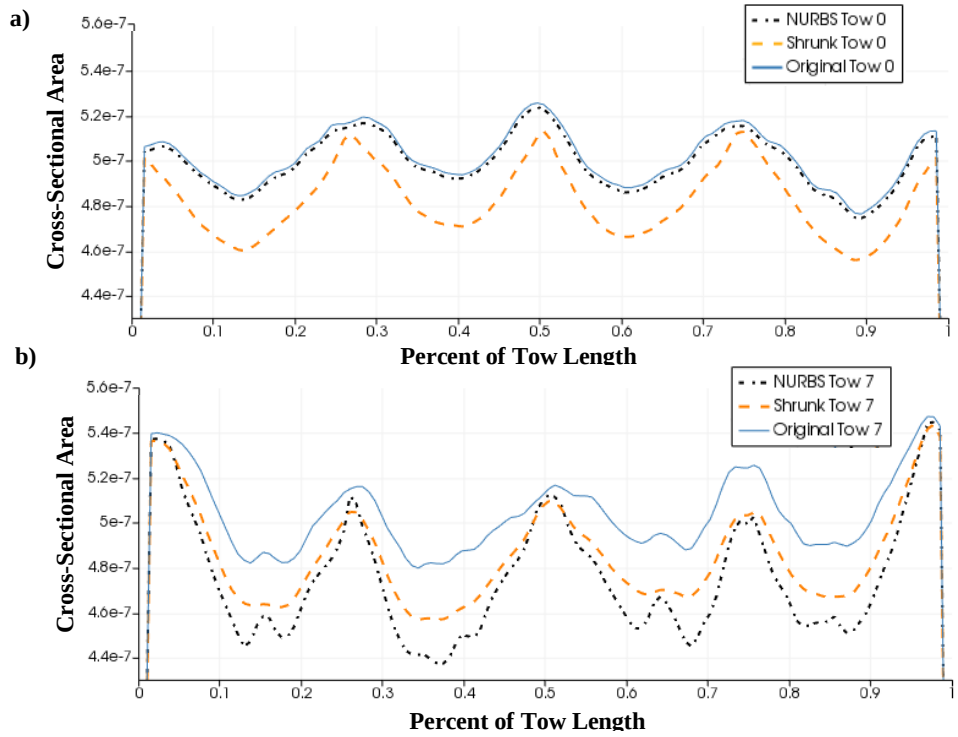


Figure 5.3: Graph comparing cross-sectional area along a tow length between original VTMS tow (Original), the shrunk tow method (Shrunk) and the NURBS method (NURBS) for a primary tow (a) and a secondary tow (b)

the tow path. This reduction is due to the slight under-approximation of the surface as a NURBS approximation. The shrunken method has a higher percentage of cross-sectional reduction along the tow path due reductions in both primary and secondary tows. Also, it is important to note that the original tow cross-sectional area does have reductions in the regions where crossing tows are in proximity. These dips in cross-sectional area are due to the way that the geometry is created and is common in many models generated in VTMS.

In Figure 5.3b, the reduction of cross-sectional area between the two methods is much more comparable, with the NURBS method reducing the area more in certain areas. The peaks where both methods come close to the original area value are regions where interpenetrations do not occur. In Figure 5.3b at approximately 0.75 along the tow path, both methods reduce the cross-sectional area where otherwise the area would return to almost the original value. This is due to an overlap of interpenetration regions (one on the upper surface, one on the lower surface). The outlines of these regions for the give tow are shown in Figure 5.4. The outlines are shown from the z-direction.

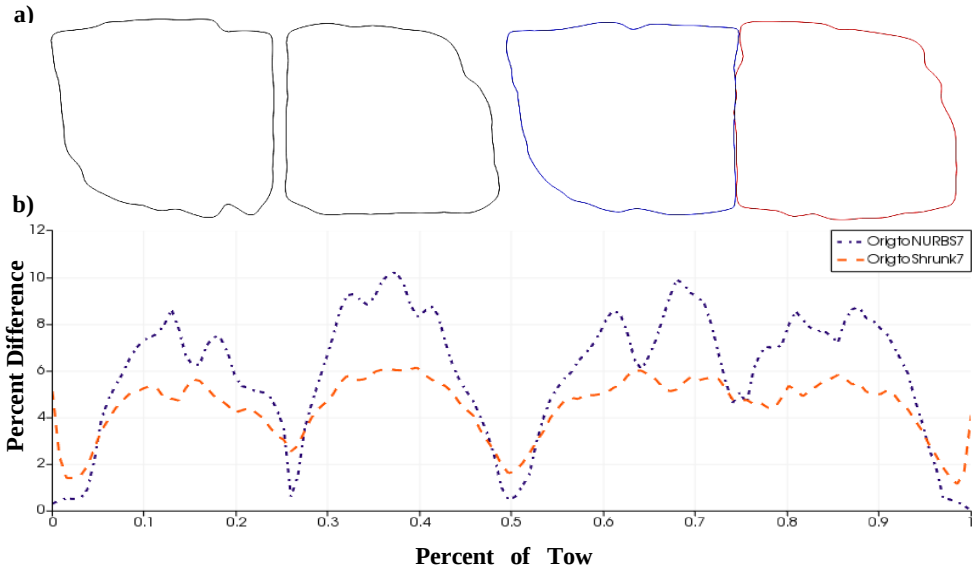


Figure 5.4: Graph showing percent difference between each method and the original tow number 7 as a function of tow path percentage

Figure 5.4b shows the percent difference between the original geometry and each method's cross-sectional area along the tow path. As the lines in Figure 5.4b trend towards zero, the cross-sectional area of the tow for that method is approaching the original value, specifically in the regions where there are not interpenetrations. Near 0.75 along the tow path, both methods correct the interpenetration regions on both sides of the tow, resulting in a reduction of cross-sectional area and therefore a higher percent difference where one might expect a closer return to zero percent difference like at 0.25 and 0.5 along the tow path. Figure 5.5 shows primary tows (number 1-4) in (a) and secondary tows (number 5-8) in (b) for both methods. Figure 5.5 further illustrates

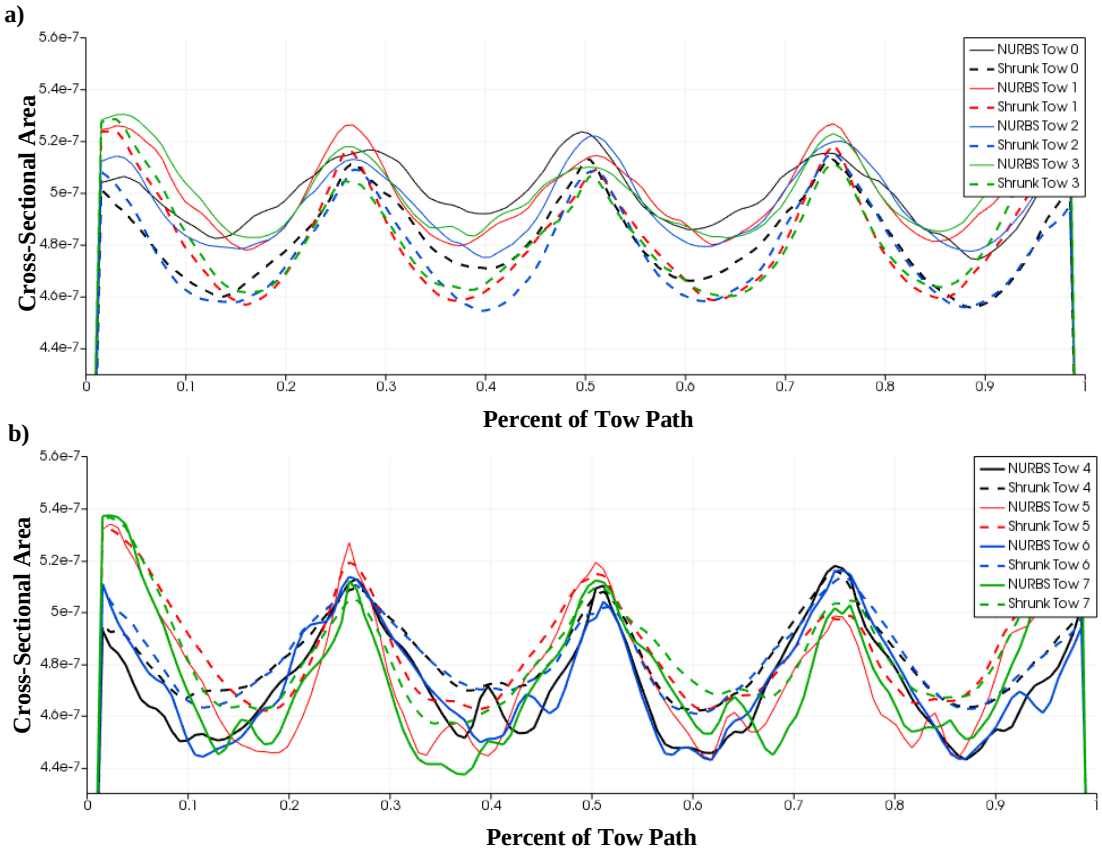
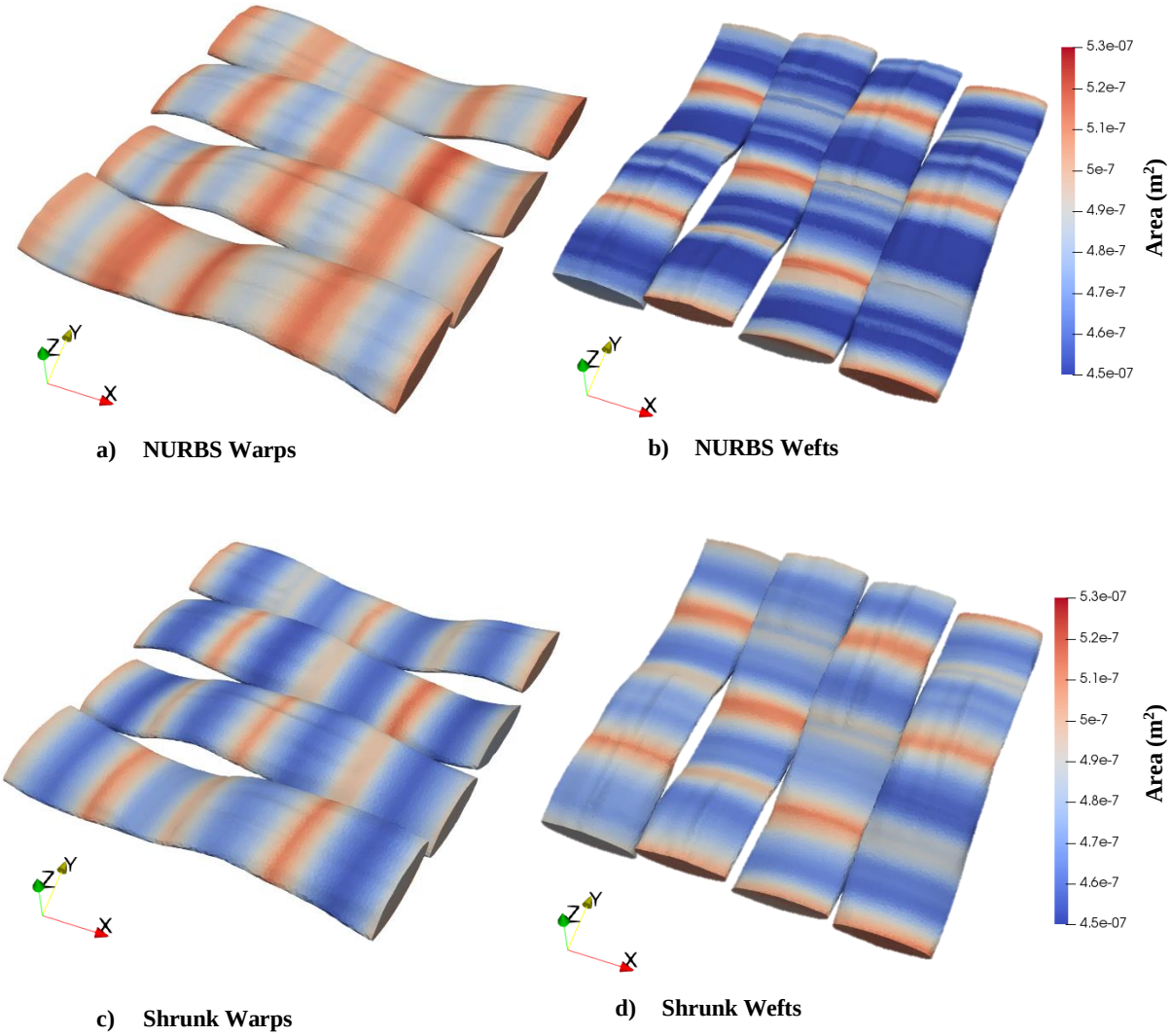


Figure 5.5: Comparisons between primary (a) and secondary (b) tows for the NURBS and shrunken methods

the point that the NURBS method does not reduce the cross-sectional area in primary tows as much

as the shrunken method does (Figure 5.5a), but affects tow cross-section area more in secondary tows (Figure 5.5b).



a) NURBS Warps **b) NURBS Wefts**

c) Shrunk Warps **d) Shrunk Wefts**

Figure 5.6: Tow cross-sectional area of warp and weft tows for each analysis

Figure 5.6 shows the variation of the tow cross-sectional area for the shrunken tow and NURBS methods on a three-dimensional model. There are some common characteristics between both methods. In each method, cross-sectional area is reduced where an orthogonal tow crosses either over or under another tow and where the interpenetrations have been removed, especially in Figure 5.6b. Also, the cross-sectional area for a tow increases in regions where there is no orthogonal tow to squeeze the tow during the compaction process in VTMS. However, there is some variability

between the two models. Figure 5.6a shows the warp tows from the NURBS method, which have been previously shown to closely match the original tow geometry. When compared to Figure 5.6c, the shrunk method's warp tows, it can be seen that the shrunken method has more tow volume at lower cross-sectional area. This is expected due to the method shrinking the tows to remove interpenetrations from both warp and weft tows. However, when comparing Figure 5.6b and 5.6d, the significant reduction of cross-sectional area due to the NURBS method becomes more apparent. Figure 5.6b has large regions of significant cross-sectional area reduction (less than 4.6E-07 m^2) compared to both Figure 5.6c and 5.6d, which have also been subjected to tow volume removal. These large regions of reduced area are due to the one-sided interpenetration removal technique currently used in the NURBS method. Therefore, there is a noticeable difference between the NURBS warp and weft tows cross-sectional areas (Figure 5.6a and 5.6b) compared to the shrunken method's warp and weft tows (Figure 5.6c and 5.6d), which are more uniform. The effects of these differences in cross-sectional area, along with using a compatible surface between interpenetrating tows, on the predicted stress results will be explored later.

5.2 Analysis Configurations

The configuration of each analysis will be discussed the following sections. The material properties and boundary conditions discussed will be consistent across all three analyses. The variability of each analysis is directly due to the configuration of the mesh. Each mesh will be shown prior to the analysis results.

5.2.1 Material Properties

Table 5.3 shows the material properties used for the tows and matrix during the analyses. The tows are assumed to have a 70% fiber volume fraction. The effective tow properties were generated using an assortment of randomized fiber/matrix unit cells, described in [39].

Table 5-3: Material Properties

IM7/5220 Tows	
E_1 (GPa)	194
E_2 (GPa)	12.6
G_{12} (GPa)	7.14
G_{23} (GPa)	4.03
v_{12}	0.30
v_{23}	0.56

5220 Epoxy Matrix	
E (GPa)	3.45
G (GPa)	1.28
v	0.35

5.2.2 Boundary Conditions

Each model is subject to the same boundary conditions. Typically, periodic boundary conditions would be applied to simulate a unit cell. However, the geometry in this study, although clipped to simulate a unit cell from a larger textile, does not accurately represent a perfect unit cell. Therefore, simple boundary conditions are applied to the model..

The negative x-, y-, and z-faces of the model are assumed to be planes of symmetry. The negative x-face is constrained in the x-direction, but allowed to displace in the y- and z-directions.

The negative y-face is constrained in the y-direction, but allowed to move in the x- and z-directions. The negative z-face is constrained in the z-direction, but allowed to move in the x- and y-direction.These boundary conditions above can be displayed as shown below.

$$u(0, y, z) = 0 \tag{5.1}$$

$$v(x, 0, z) = 0 \tag{5.2}$$

$$w(x, y, 0) = 0 \tag{5.3}$$

The loading boundary condition applied is a uniaxial tension in the x-direction. The results shown are the result of a 1% volume average strain in the x-direction. This boundary condition for all models is below.

$$u(6.51\ mm, y, z) = 0.0651\ mm \tag{5.4}$$

The material properties and boundary conditions in the two above sections are used for each model. The next section will discuss each model's finite element mesh and how they differ.

5.2.3 Meshes

Each model has a different mesh. The differences in the meshes themselves will be discussed later. One metric to understand model size is the node count of each model. Table 5.3 shows the node count of each model.

Table 5-4: Node count of each model

Node Count	
NURBS Low-Refinement	1,356,768
NURBS High-Refinement	4,921,505
Shrunk Tow	1,301,552

Generally, as the number of nodes in the model increases, so does the accuracy of the results. However, the analysis is more costly in terms of time or computer resources. Therefore, a trade-off occurs between accuracy and resources. The accuracy of the high and low refinement models

will be part of the comparison in the stress results. For each model in the following section, the same tow (tow number 2) will be used so that the meshes compared are similar. Any differences will be due to the method chosen and not differences introduced by VTMS.

Shrunken Tow Mesh

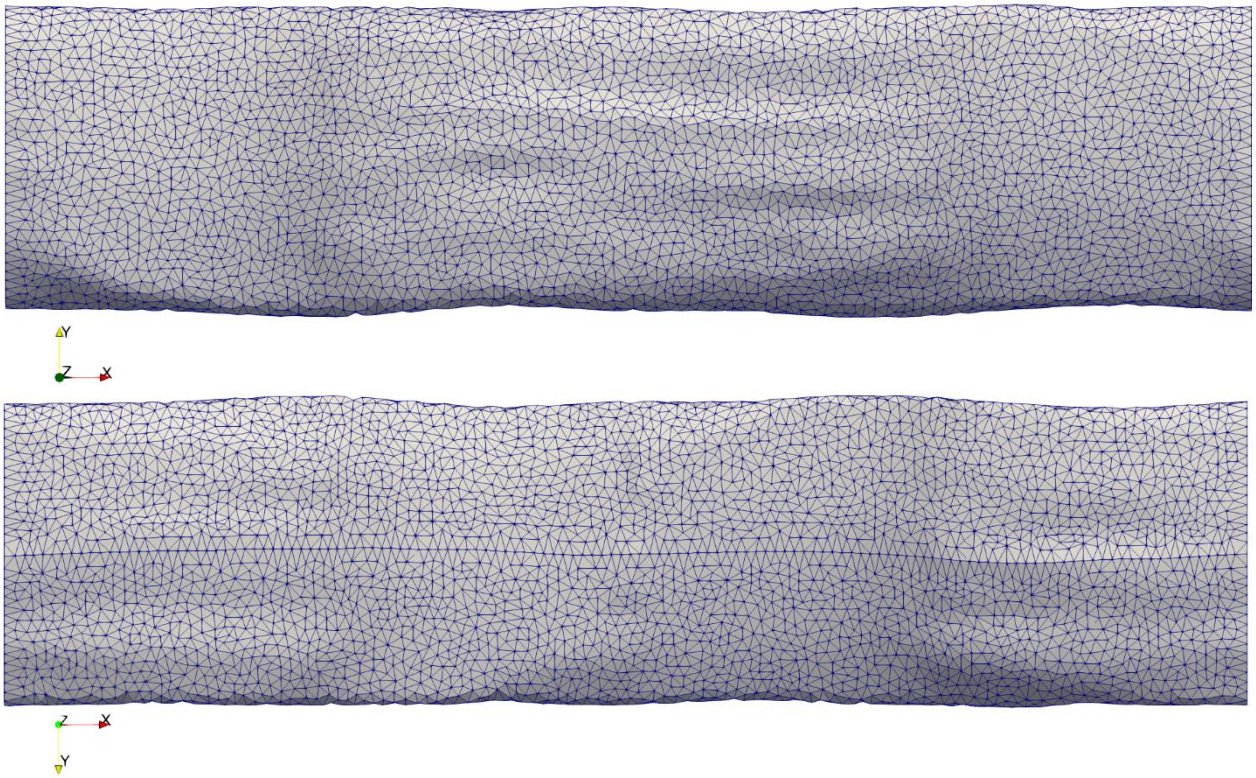


Figure 5.7: Positive and negative z face of shrunken tow method tow number 2

Figure 5.7 shows positive and negative z side of the number 2 tow from the shrunken method. It can be seen that all the elements are of relatively the same size and shape. The size of the elements is set by the smallest edge length in the model before element generation. In the shrunken model, this edge length is the shortest segment length on the first polygonal cross-section of a tow generated in VTMS. The uniform distribution of the mesh is due to the lack of an intersection curve to constrain any of the element edges and therefore the size. A uniform distribution is ideal

as it is less likely to introduce any preferential concentrations of stress due to refinement size and convergence.

NURBS High-refinement

Figure 5.8 shows the positive and negative z face of tow number 2 produced by the NURBS method with high-refinement intersection curves. It is apparent that there are much smaller elements present in the mesh than those in Figure 5.7. The dark lines that trace out the intersection curves on the surface are very small elements whose close proximity to each other create a dense field of elements. At the length scale shown, these elements appear as a solid line. The high refinement around the intersection curves can allow higher stress gradients to occur where two tows are connected and will be more sensitive to very high stress values in these regions. Therefore, a lower refinment mesh was generated to serve as a small mesh sensitivity study to determine where stress singularities occur. The low refinement mesh will still show regions of high

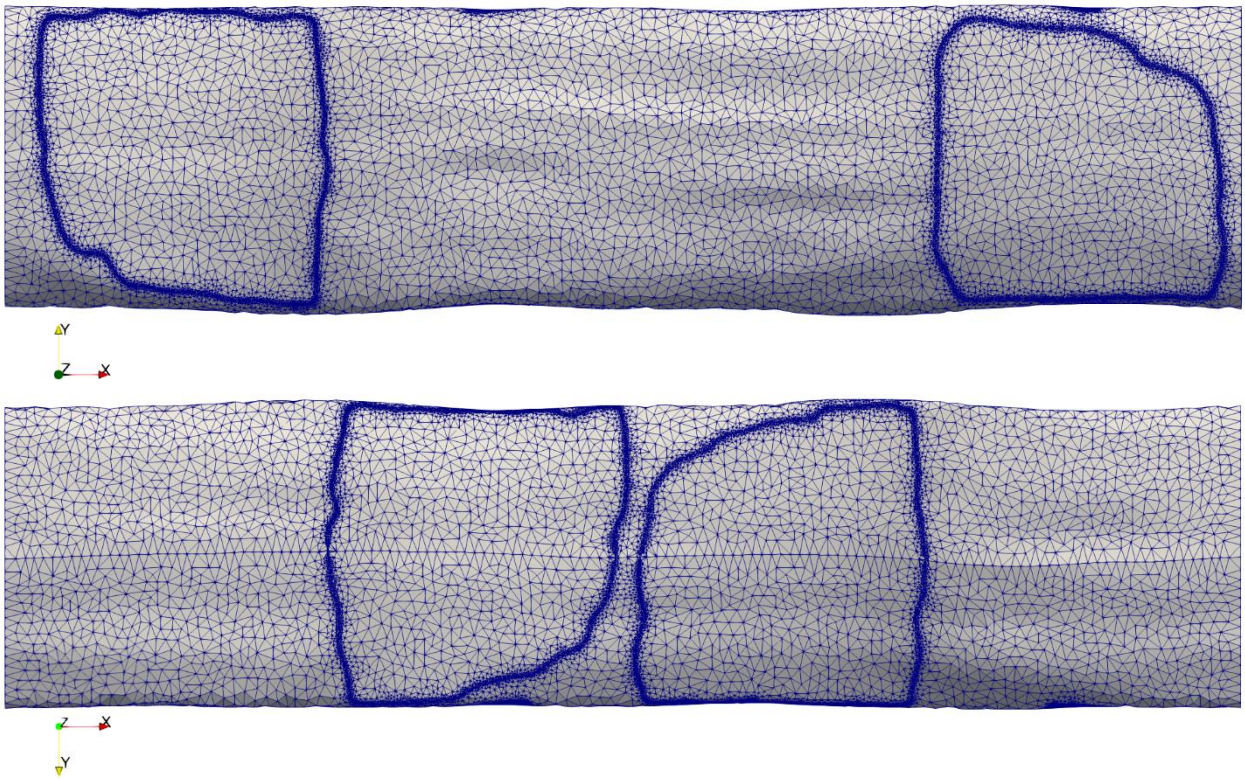


Figure 5.8: Positive and negative z face of the number 2 tow from the high-refinement NURBS method

stress while not allowing the same level of stress convergence that the high refinement model does. In this manner, areas with possible stress singularities will be shown.

NURBS Low-refinement

The two NURBS methods will exhibit similar element topograhpy around the intersection curves, but the refinement level around the intersection curves is much closer to the shrunken tow method. Figure 5.9 shows the low-refinement NURBS surface. Both Figure 5.8 and 5.9 show the same mesh artifact on the negative z face as Figure 5.7. Therefore, the results of their analyses may exhibit similar concentrations near this artifact that would not be present without the artifact. The purpose of this third model is to check for convergence of stresses due to mesh refinement and also to observe any reduction of accuracy due to lower refinement around the intersection curves.

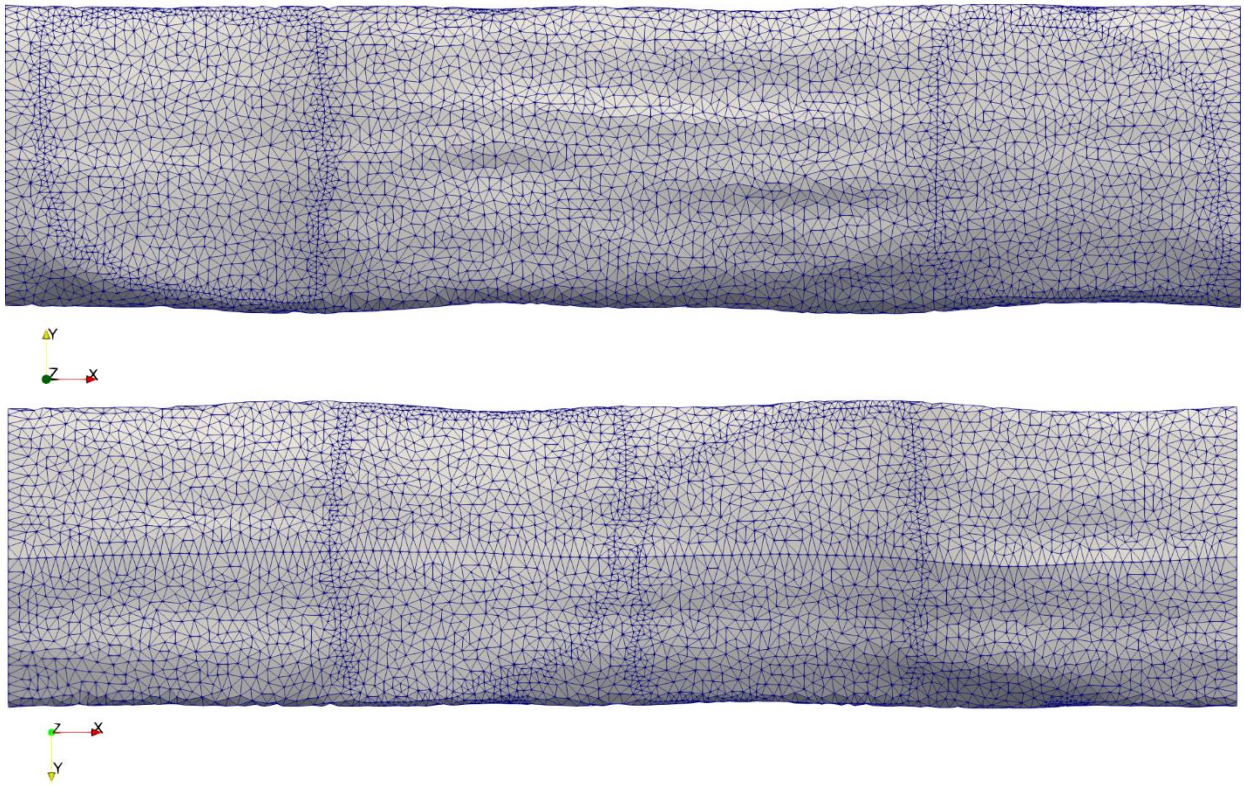
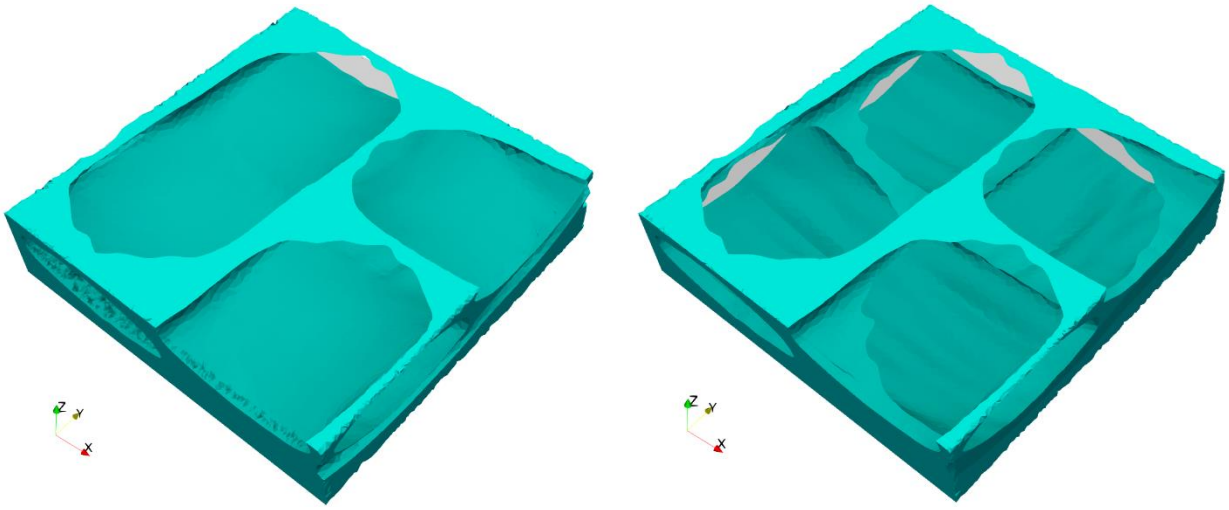


Figure 5.9: Positive and negative z face of the number 2 tow of the NURBS low-refinement NURBS method

Matrix Meshes

An example of each tow surface mesh has been shown. Figure 5.10 shows a cut away of each method's matrix mesh. The individual elements have not been shown. However, it can be seen how the shrunken method matrix mesh (Figure 5.10a) has regions of matrix that the NURBS method (Figure 5.10b) does not. That is because there is a layer of matrix between tows in the shrunken method that is removed in the NURBS method due to the shared compatible surface in



a) Shrunken tow method matrix **b) NURBS method matrix**

Figure 5.10: Clipped region of matrix mesh for both shrunken tow (a) and NURBS (b) methods

the NURBS method. The edge of the matrix in these open regions may also be susceptible to stress concentrations and will discussed in the analysis results.

The next section will discuss the effects of the mesh artifacts and features shown in this section. The analysis section will also show and discuss the predicted stress reponse in regions away from the intersection curves to determine how each method affects the results. The purpose of the next section is to highlight the advantages and disadvantages of the method shown in this research and whether it is a viable method to remove interpenetrations between tow surfaces.

5.3 Stress Analysis

A conventional finite element analysis was conducted for each of the three variations of the model. A compartive approach will be used to explore the predicted stress response of each model. Many of the Figures shown will consist of side-by-side comparisons between the different models to show their similarities and their differences in the predicted stresses. Some components of stress have been excluded from these results as the differences in predicted response between methods is minimal. The stress components discussed will be primarily in the tows local coordinate system denoted with a prime such as σ'_{ij}. Any stresses discussed in the global direction will be denoted without a prime.

5.3.1 Warp tows

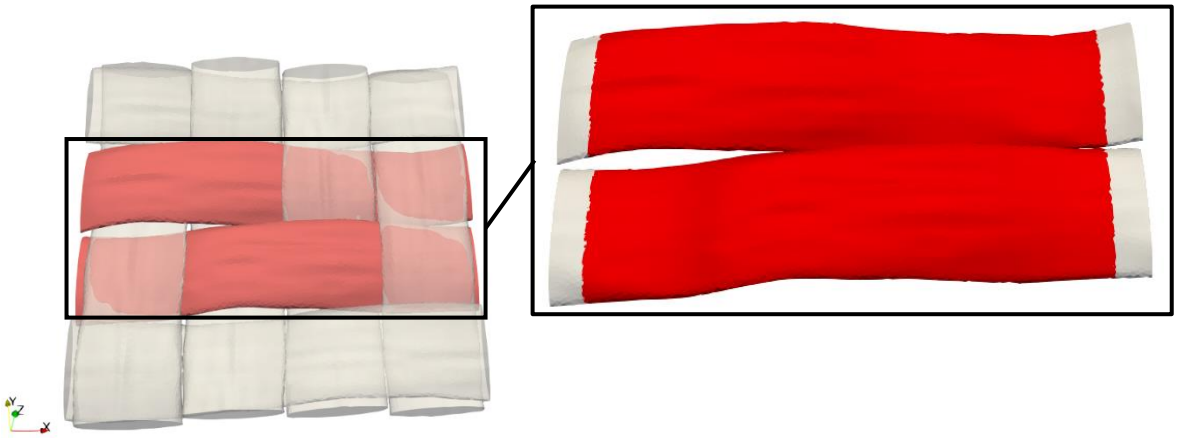


Figure 5.11: Highlights warp tows (tow numbers 2 and 3) that will be closely observed

Figure 5.11 shows the middle two warp (x-direction) tows and the clipped region which will be used over the next few Figures. The ends have been clipped to remove edge effects due to how the boundary conditions were placed on the model.

Figure 5.12 shows σ'_{xx} on the warp (x-direction) tows 2 and 3 from the negative z-direction. The overall distribution of stress is similar for all three analyses except in the regions between y-direction tows and along the boundary of the connected regions where the NURBS method tows

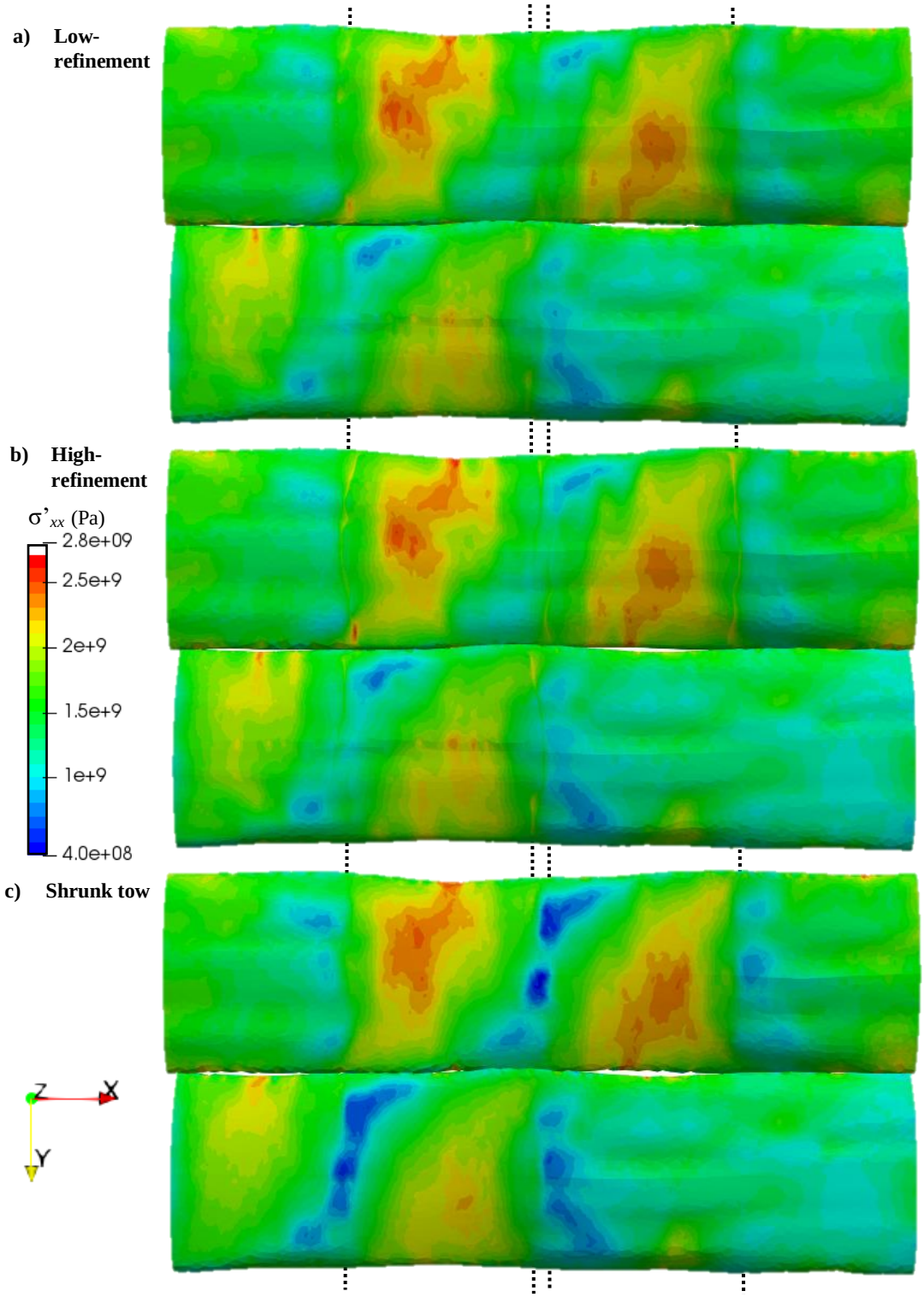


Figure 5.12: Stress in the warp tows local XX direction on the negative z face for each analysis (a.) low-refinement, b.) high-refinement, c.) shrunken tow).

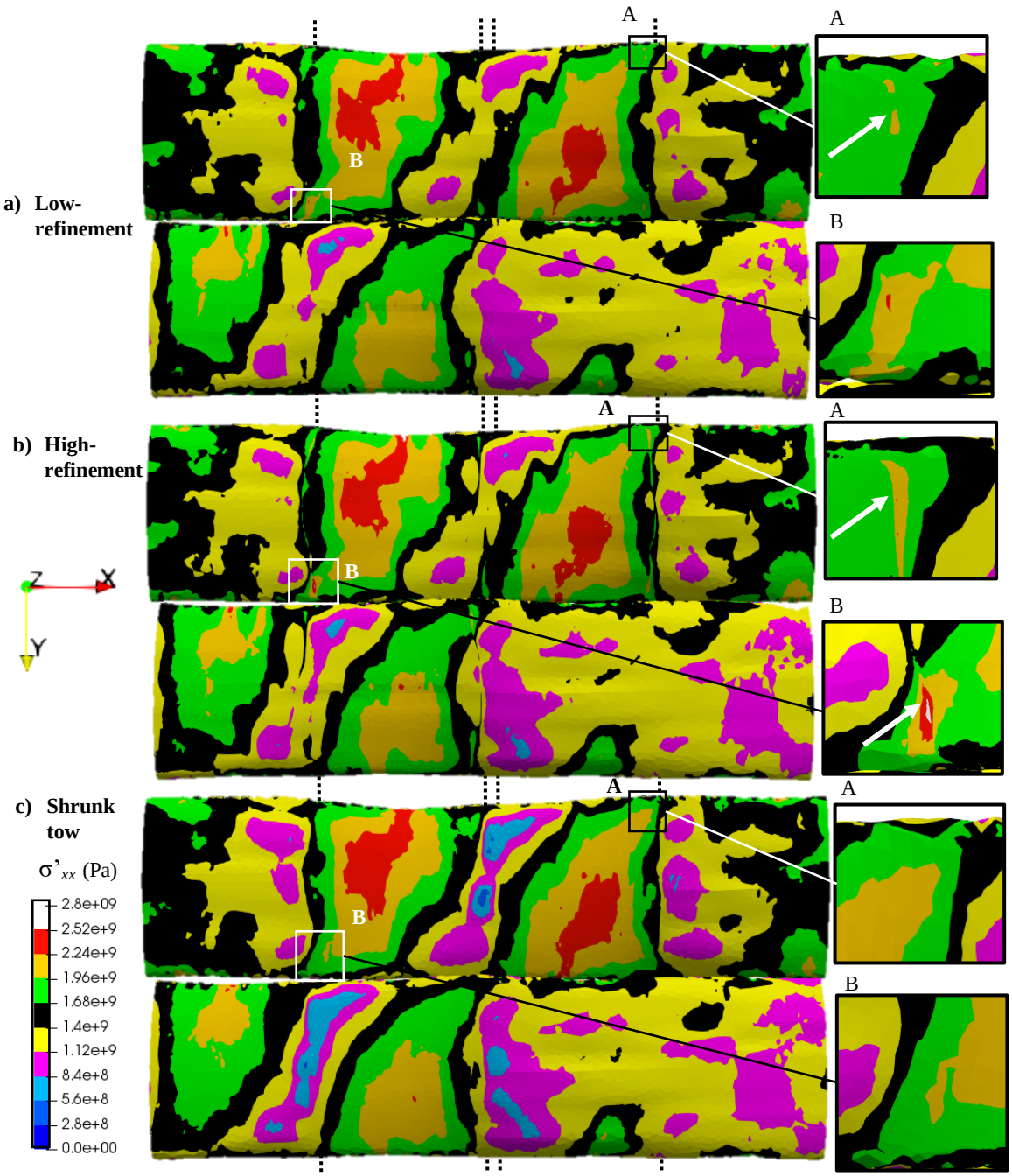


Figure 5.13: Stress in the warp tows local XX direction on the negative z side for each analysis (a.) low-refinement, b.) high-refinement, c.) shrunken tow). Dotted lines indicate boundaries of weft tows.

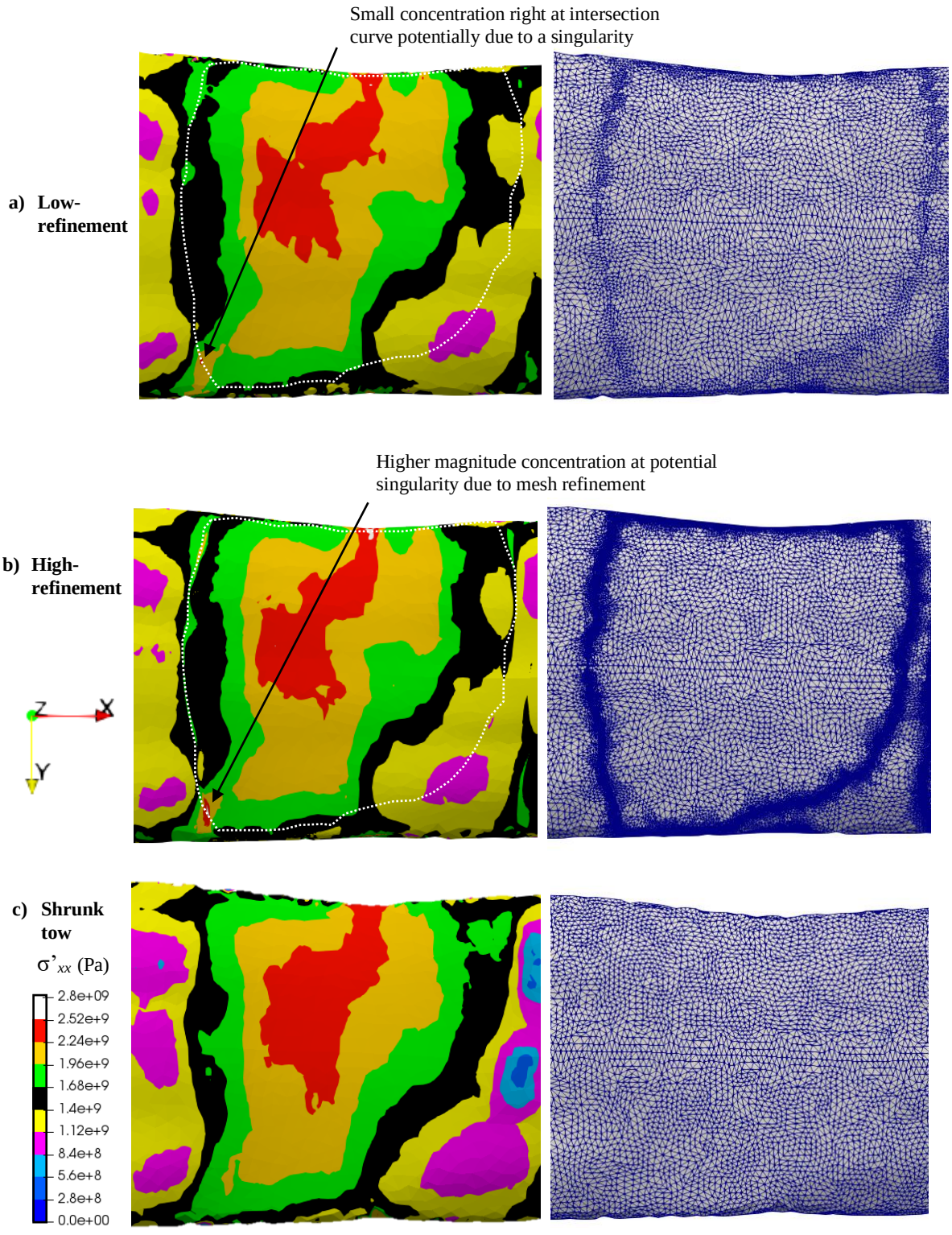
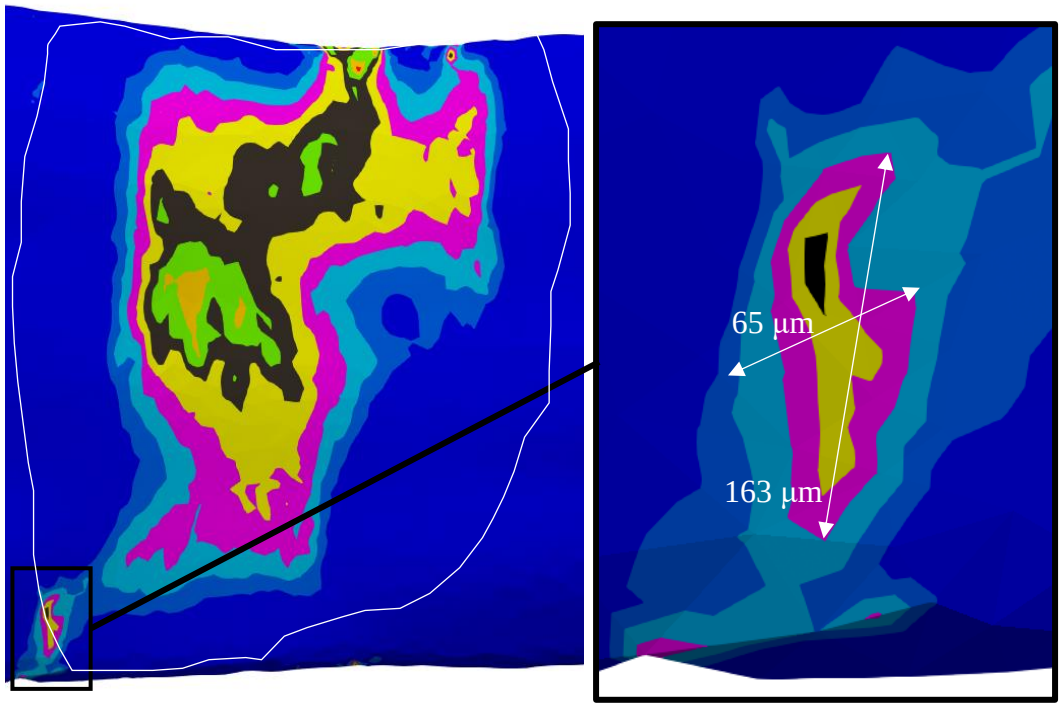


Figure 5.14: High stress regions paired with mesh refinement in the stress region being observed

experience higher stresses. Figure 5.13 is the same stress component and tows, but the contours have been changed to help highlight the highest stress regions in all three analyses. When comparing against the cross-sectional areas (Figure 5.6), it is logical that regions of reduced cross-sections would show a higher stress than regions of higher cross-sectional area. Region "A" in Figure 5.13 shows that the shrunken tow method predicts higher stress in region adjacent to the intersection curves compared to the NURBS method analyses due to the reduction in cross-sectional area in the shrunk tow analyses. However, at the intersection curve, the NURBS analyses both predict a similar stress magnitude to the shrunken tow method (indicated with white arrows in pop-outs), with the high refinement model having a larger stress magnitude. This indicates that some of the load is being distributed along the boundary of the region where the tows are connected. Region "B" in Figure 5.13 shows a region where the high refinement analysis has a much larger stress magnitude than either the low refinement or the shrunken analysis. In fact, both Figure 5.13a and 5.13c predict similar stresses in this region, which indicates that there may exist a singularity in this location. In order to determine if a singularity indeed exists, a mesh convergence study would have to be conducted.

Figure 5.14 shows a closer view where region "B" from Figure 5.13 is located. The intersection curve from the NURBS method is shown in Figures 5.14a and 5.14b as a dotted line. Along with the stress results are also the element distribution in this region. It can be seen that in the interior region (where the interpenetrations occurred), all three methods predict similar stress in terms of magnitude and gradient. However, Figures 5.14a and 5.14b show that the existence of an intersection curve can introduce stress concentrations. These figures also show how refinement can affect the magnitude of the concentrations and that the increased refinement can converge to a higher stress magnitude where a singularity may exist. However, it can be seen that away from

a) Low refinement



b) High refinement

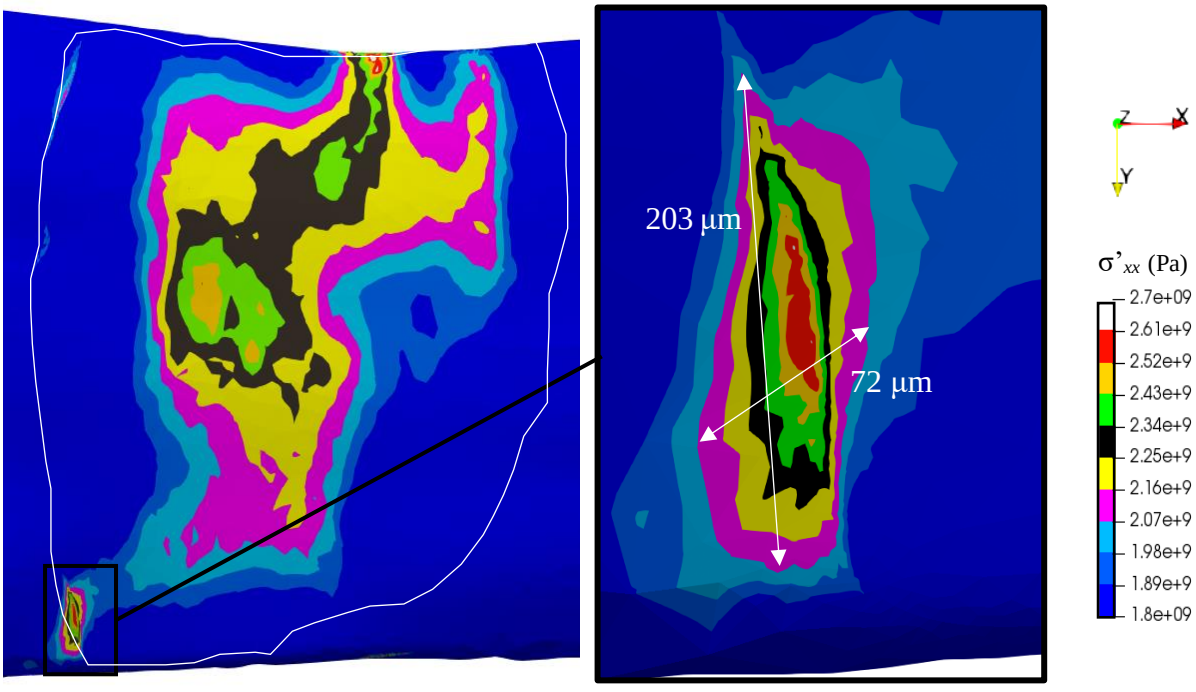


Figure 5.15: Measurements of high stress concentration (XX-stress) for both NURBS method analyses.

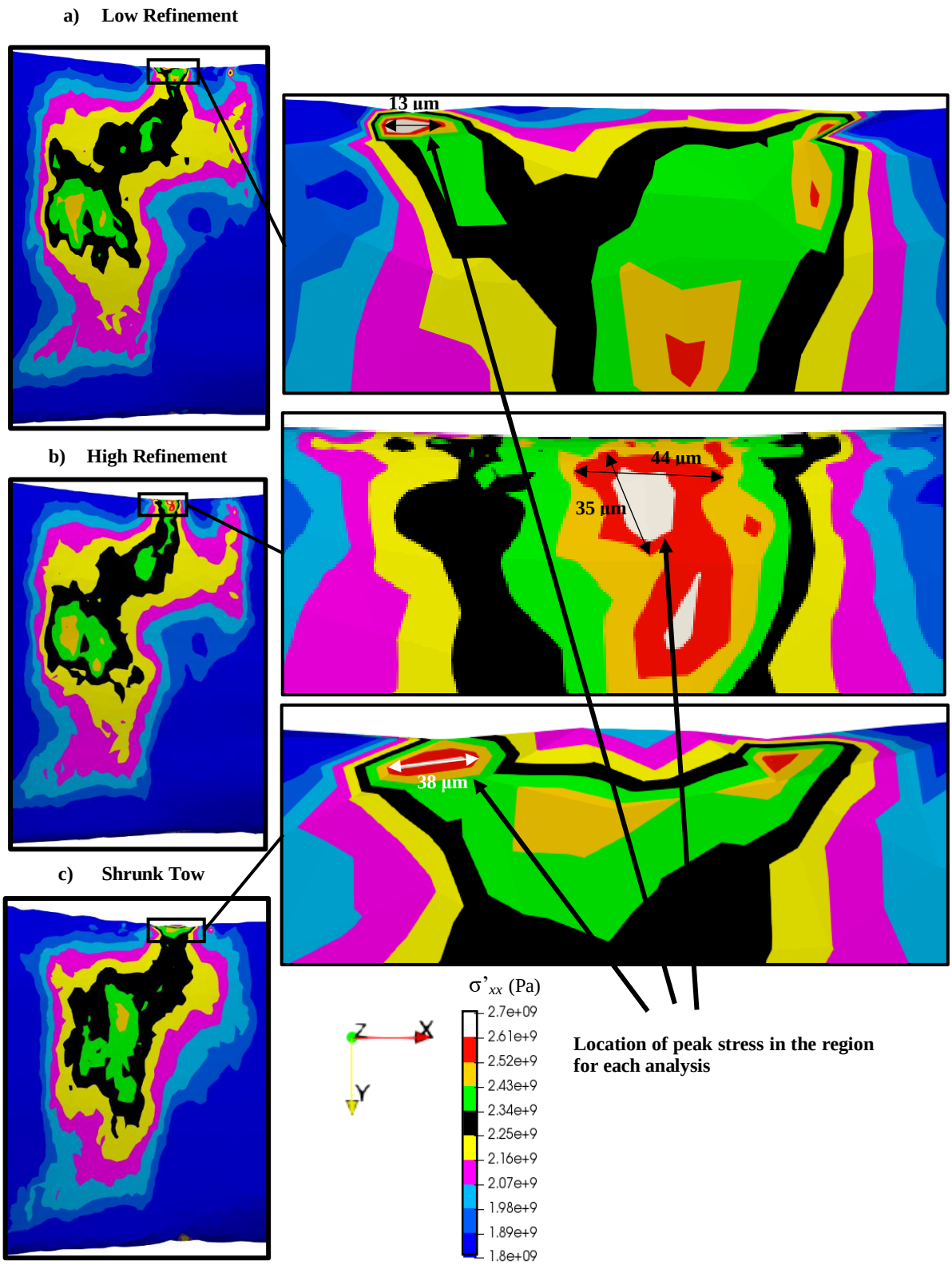


Figure 5.16: Common localization of stress concentration across all analyses with peak stress indicated with arrows

the intersection curve the refinement of the intersection curve does not affect the predicted stress.

Figure 5.15 focuses on the difference between the high and low refinement NURBS analyses. In this figure, the stress scale has been narrowed to highlight the size and gradient of the small stress concentrations unique to the NURBS method. When discussing concentrations on a small scale, it is useful to introduce a measure of length comparable to that scale. For this discussion, measurements will be made with respect to the diameter of an average IM7 carbon fiber, which is 5 microns or 0.005 mm. In Figure 5.15a, the sizing of the unique concentration is approximately 13 fiber diameters wide by 39 fiber diameters wide. The concentration in Figure 5.15b is of similar size, 13 diameters by 33 diameters, but has a significantly higher maximum stress magnitude (2.7E09 Pa compared to 2.3E09 Pa) for σ'_{xx}. For both concentrations, the depth into the tow volume is 1-2 fiber diameters. The stress gradient does indicate that the higher refinement of the mesh causes the higher peak stress in this region, as the lower refinement mesh also has the same localization of stress, but at a lower magnitude. The length scale (on the same scale as a fiber diameter) indicates that these concentrations, unique to the NURBS method, would not have a significant impact on the response of the textile at the macroscale. However, the refinment of intersection curve and the resulting concentration of elements around these regions can predict higher magnitude stress concentrations at the mesoscale than a mesh that is less refined in these regions, as shown in Figure 5.15. Therefore, the location of the peak stress could potentially be mischaracterized to lie along these boundaries when they may occur elsewhere. For models that depend on stress concentration location (e.g. damage initiation), a mesh refinement study would need to be conducted. The advantage to the NURBS method is that if the predicted location shown in Figure 5.15 is consistent with physical results, the shrunken tow model may not predict the same response.

Figure 5.16 shows the same region as Figure 5.15 for all three analyses. The pop-outs highlight a smaller subregion where all three analyses share a stress concentration location. Figure 5.16 shows that some stress concentrations are common across all the analyses due to the geometry of the tows. At the shown length scale, the location of peak stress can be dictated by mesh element position and alignment. However, the concentration along the edge of the tow indicate that all methods predict some concentration in the region regardless of interpenetration removal m ethod. It is also important to note that the size of these concentrations are smaller than in Figure 5.15 and may be insignificant if the response of the model as a whole is considered.

Figure 5.17 shows σ'_{yy} for tows number 2 and 3 for all three analyses. The compression forces in this direction are minimal compared to the tensile stresses. Therefore, the scale has be modified to show only tensile stresses, which allows for a better understanding of the concentration regions. The arrows labeled “A” in Figures 5.17a and 5.17b show concentration regions unique to the NURBS analyses. Figure 5.17c also has a concentration in this region, but the magnitude of this concentration in 5.17a and 5.17b is significantly higher (greater than 1.8E+8 compared to 8.0E+7). The higher stress magnitude is due to the edge of the region where the tows are connected for the URBS analyses shown in Figures 5.17a and 5.17b. Along this boundary, the weft tows resist the warp tows transverse shrinkage in the global y-direction causing a local tensile stress in the warp tows. The “B” arrows in Figures 5.17a, 5.17b, and 5.17c show regions where the stress concentration and magnitude are similar across all analyses. These concentrations are due to the tows attempting to shrink laterally away from each other but the close proximity of the tows causes a concentration to form where the tows are closest.

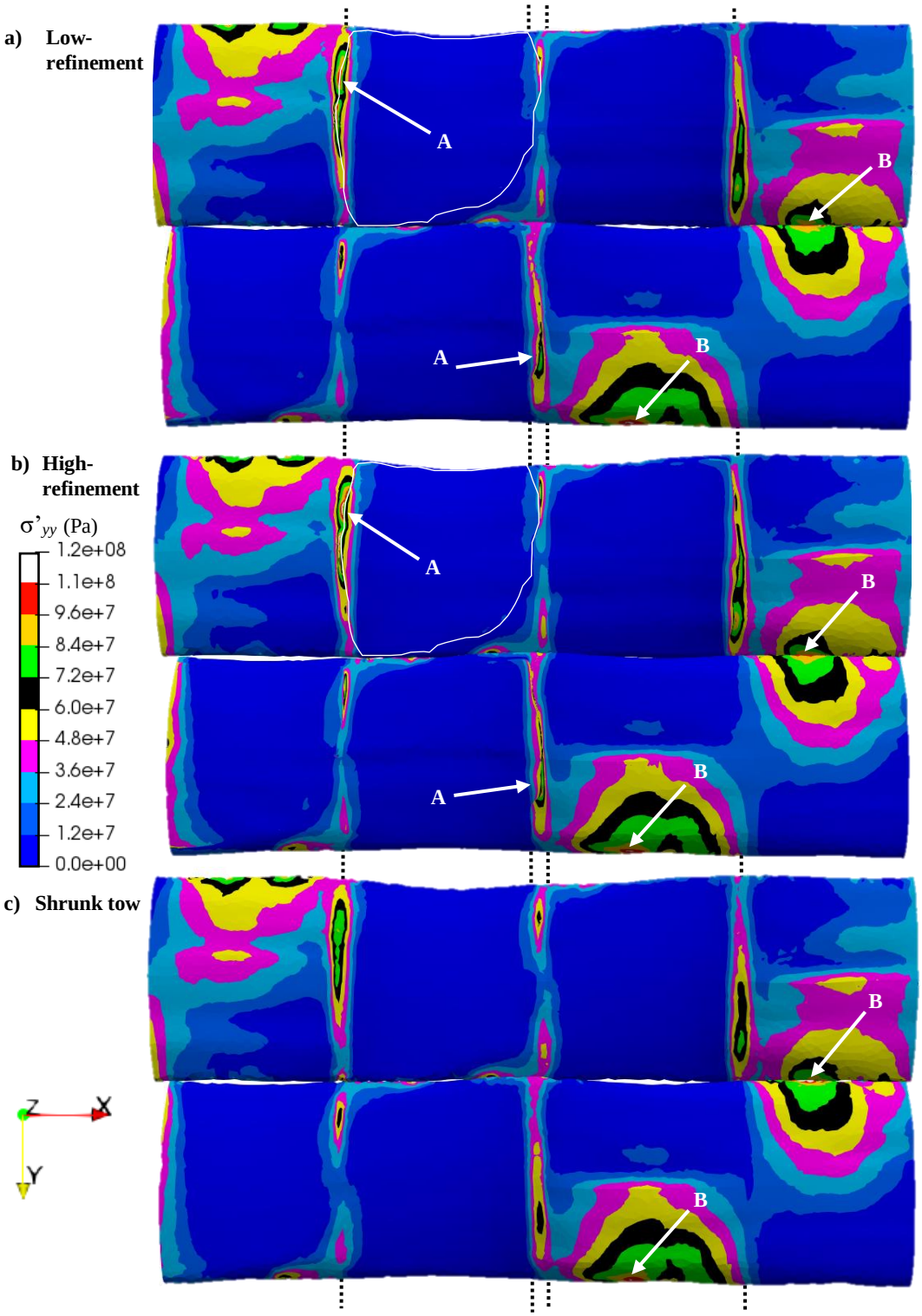


Figure 5.17: Stress in the warp tows local YY direction on the negative z face for each analysis (a.) low-refinement, b.) high-refinement, c.) shrunken tow). Dotted lines indicate boundaries of weft tows. "A" arrows indicate max stress concentrations specific to the NURBS analyses and "B" arrows indicate max stress concentrations similar across all analyses.

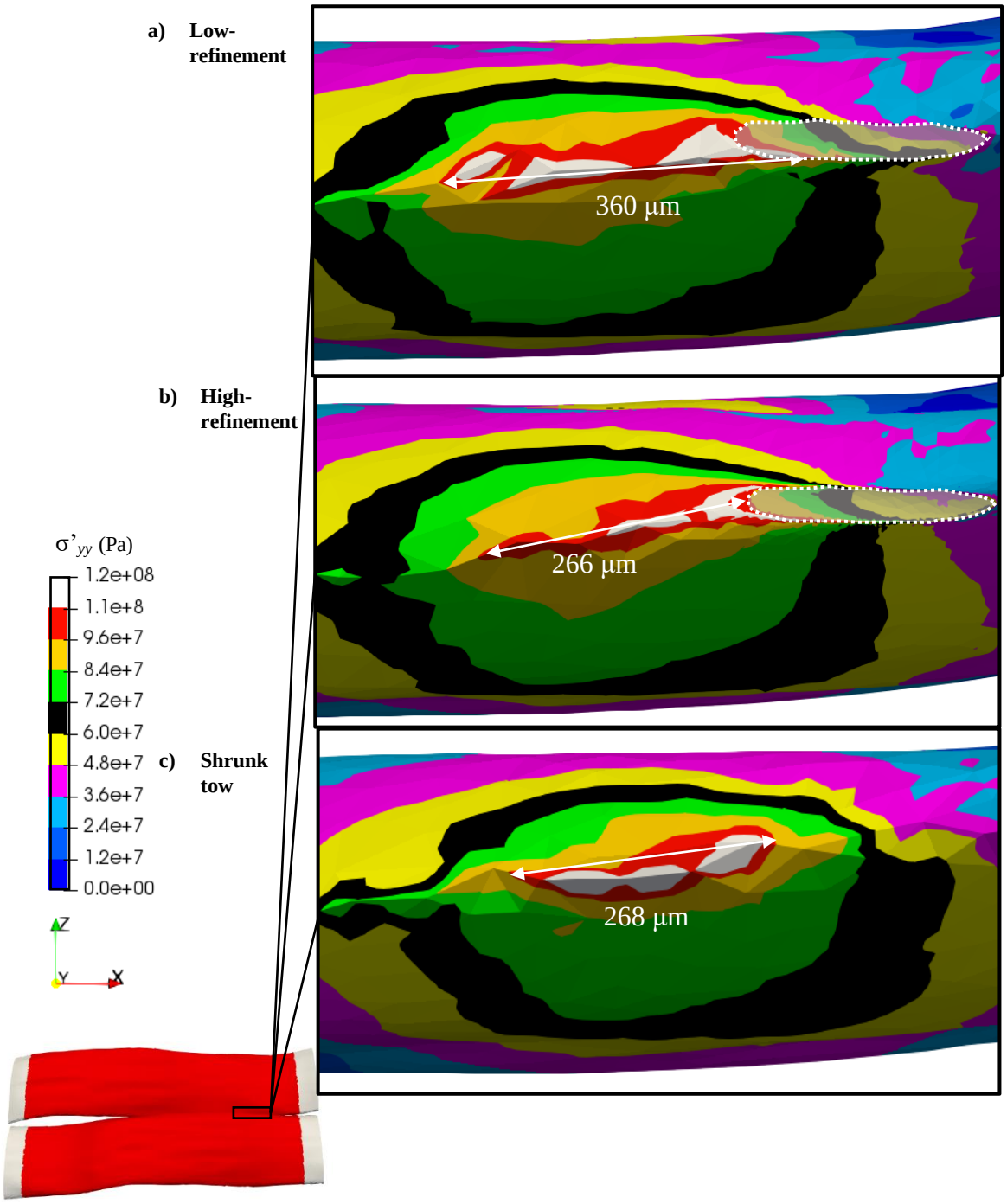


Figure 5.18: Edge-on view of tow number 3 where neighboring tow comes in close contact (or shares a surface). The sizing of the concentrations at the widest points are marked.

Figure 5.18 further shows that all three analyses predict the same level of stress in certain regions due to the configuration of the textile. Figures 5.18a and 5.18b both have a surface that is shared with the neighboring tow. Although the stress concentration does conform to the boundary of the surface region (indicated by the white dotted line), it is not the source of the concentration as all three analyses have the same magnitude of stress (greater that 1.1E+08) at the same location. All three analyses in Figure 5.18 have the highest stress concentrations where the edges of the tows are the closest with little or no matrix between them. The NURBS method analyses have tows directly connected to each other in these regions, but the concetration of stress is not directly attributed to the connected region as previously discussed. Instead, the configuration of the weave itself and the close proximity of the tows are the causes of the concentrations. This indicates that textile configuration can affect the concentration location more than the existance of a connecting region between tows for σ'_{yy}. There were no significant differences between the analyses for σ'_{zz}.

Figure 5.19 shows one of the out-of-plane shear stresses, σ'_{yz}, for all three analyses. The "A" arrows in Figures 5.19a and 5.19b show where the boundary of the connecting region between tows significantly increases the magnitude of the shear stress (greater than 5.04E+07). This concentration does exist in the shrunken tow analyses (magnitude of stress is less than 3.78E+07), however the edge of the connected region increases the magnitude and the refinement of the high refinement case causes an even higher increase in stress magnitude. The gradient of this stress indicates that the discontinuity betweeen three material properties may be creating a sinularity. The effect of these singularities can be reduced with a less refined mesh in these regions, as shown in Figure 5.19a. The "B" arrows in Figure 5.19 indicate common regions of stress concentrations away from the boundary of the connected regions between tows. The magnitude of stress in these regions are similar, and in some cases higher for the shrunken tow method. Furthermore, the

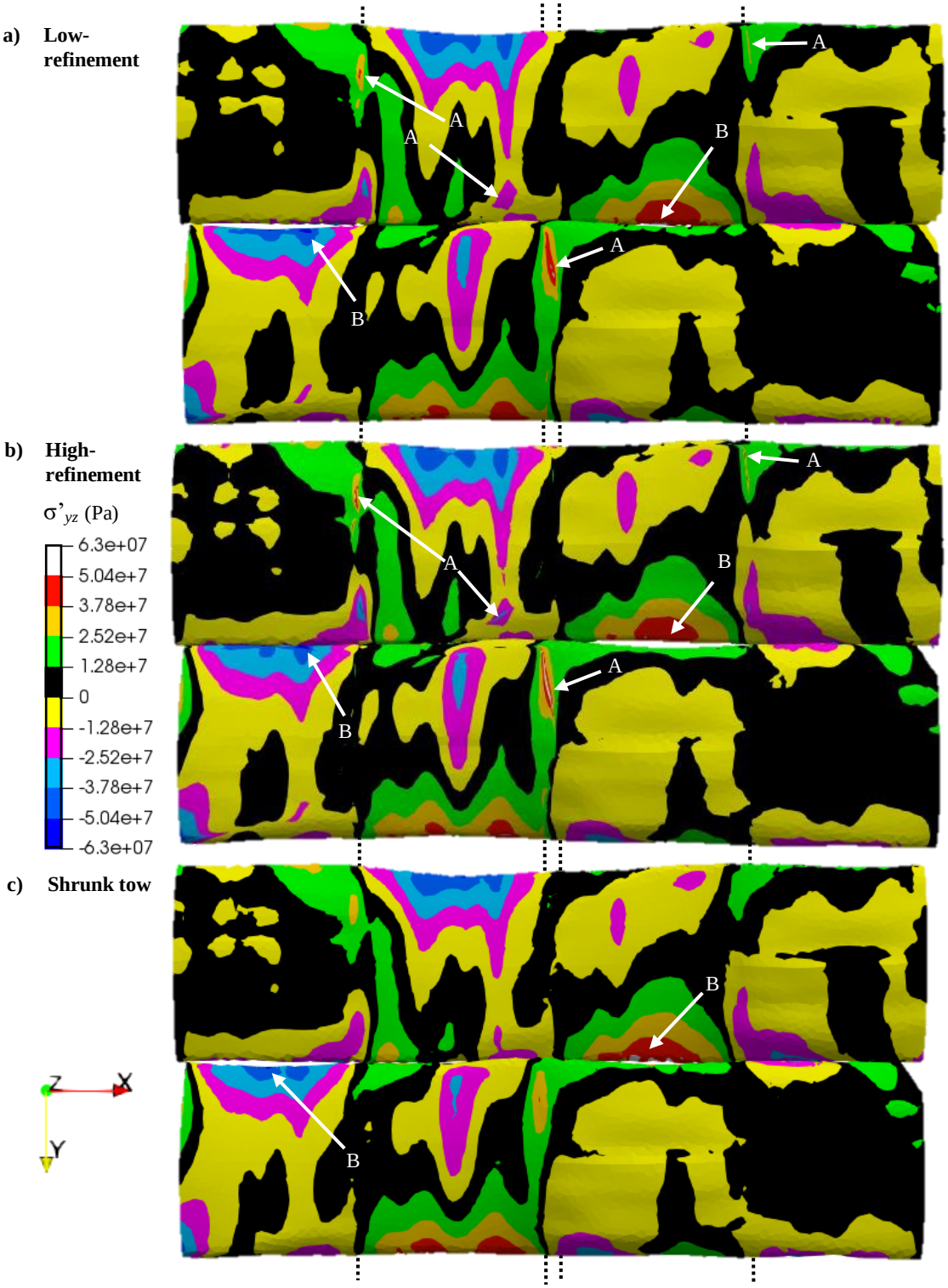


Figure 5.19: Shear stress in the tows local YZ-direction on the negative z face for each analysis (a.) low-refinement, b.) high-refinement, c.) shrunken tow). Dotted lines indicate boundaries of orthogonal tows. "A" arrows point to peak stresses unique to NURBS method. "B" arrows point to peak stresses common to all analyses.

pattern and magnitude of stress in the connected tow regions themselves is very similar across all three analyses. This shows that away from the localized concentrations due to the boundary of the connected regions, all three analyses predict a similar response for σ'_{yz}. This is also the case with σ'_{xy} and σ'_{xz}, which have no significant differences between the three analyses.

In the next section, the predicted response of the weft tows will be discussed. Careful observation of the global coordinate system each figure will be vital as the local coordinate systems of the tows are not closely aligned with the global coordinate system. Also, any relevant relationship between the local and global coordinate systems will be discussed in detail.

5.3.2 Weft Tows

Figure 5.20: Weft tows 6 and 7 highlighted as the tows of interest

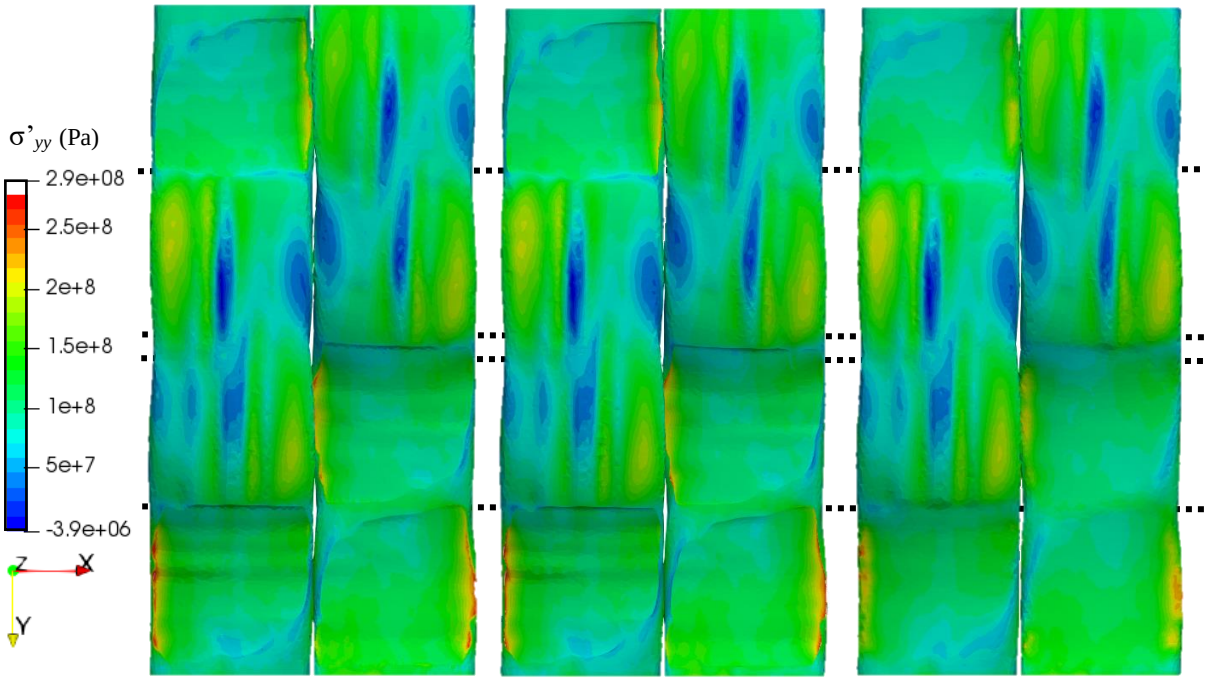
This section will discuss the predicted response of the weft tows (aligned in the y-direction) for all three analyses. Figure 5.20 shows the middle two weft tows (number 6 and 7) that will be

used for the duration of the discussion. It is important to note that the primary direction of loading is perpendicular to the weft tows and therefore aligned with the local y-direction of each tow.

Figures 5.21 and 5.22 shows σ'_{yy} and σ'_{xx}, respectively, in weft tows for each analysis. These figures server to orient the reader with a more traditional stress contour. The white arrows in Figure 5.23 indicate a region of high stress across all three analyses. The identified region is also where an interpenetration has been removed using the discussed methods. There is a larger (in dimension) stress concentration predicted in the NURBS method analyses (40-80 fiber diameters in length) compared to the shrunk tow analysis and a high stress gradient located just before the end of where the warp and weft tows are connected. The shrunken method also predicts a similar peak stress in this region but the gradient is much more focused on the small region of peak stress.

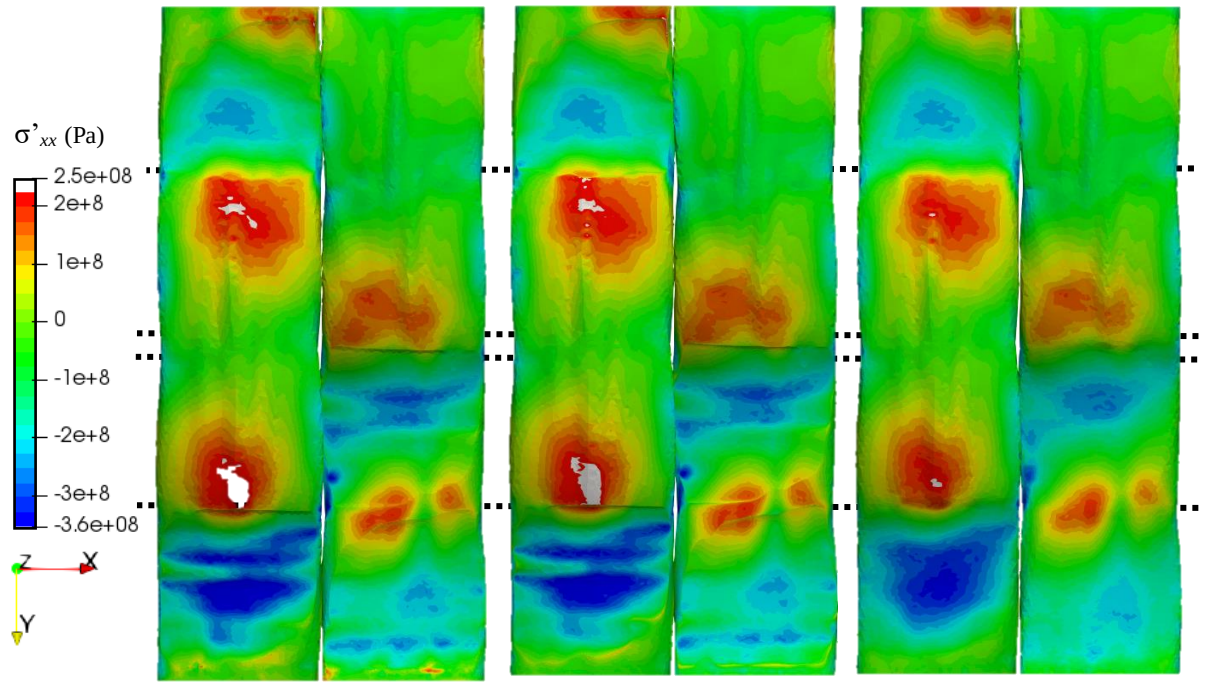
Figure 5.24 further shows these concentrations where the gradient and magnitude is easier to observe. All three analyses do reach a similar peak stress (≈2.73E+08), however the NURBS method analyses predict a different shape of concentration that conforms to the boundary of the region where the warp and weft tows are connected. Figures 5.24b shows how the higher refinement mesh in this region allows the concentration to converge to the edge of the connected region and that there is likely a singularity causing the large magnitude of stress. It also appears in Figure 5.24c that the shrunken tow method may also have a singularity as much of the stress in this region is lower than the two NURBS method analyses and the gradient to the peak stress is quite high. A similar gradient is shown in Figure 5.24a and 5.24b which is already suspected to have a singularity. Away from these peak stress regions, the NURBS method analysis is at a consistently higher stress value than the shrunken tow method analysis at a given point on the tow.

Localized concentrations at the boundary of the connected region between tows have been shown in the warp tows. However, the effect is compounded for the NURBS method analyses in



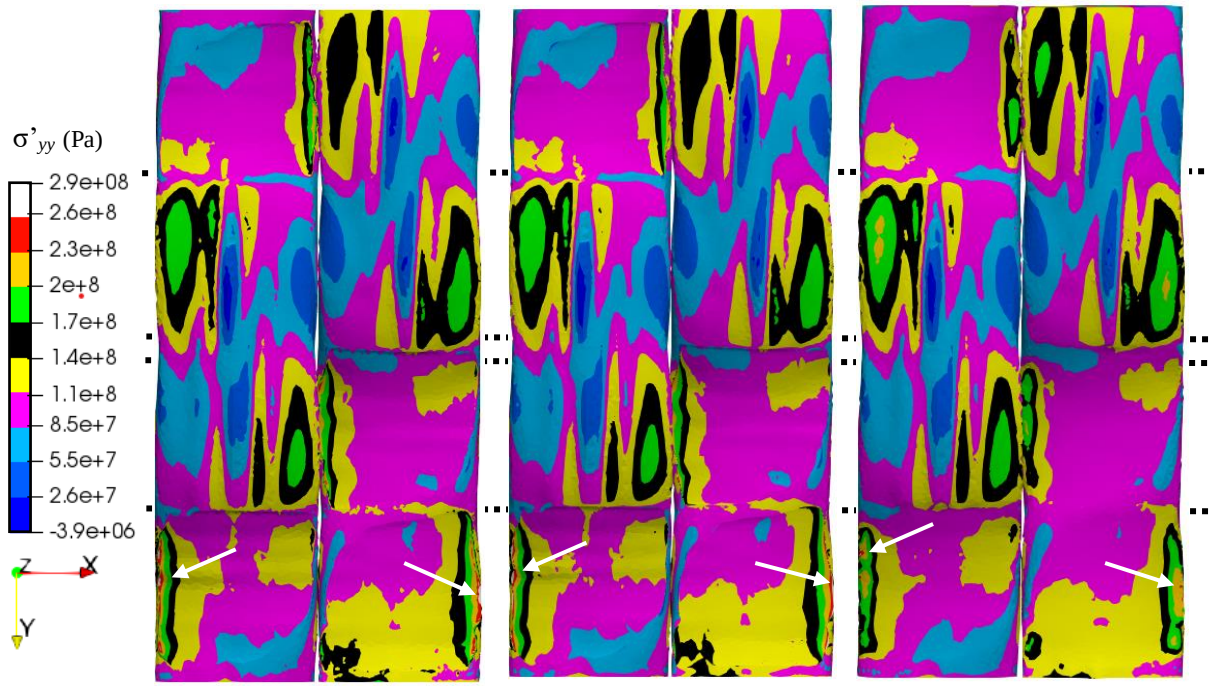
a) Low-Refinement **b) High-Refinement** **c) Shrunken Tow**

Figure 5.22: Stress in the weft tows local YY direction on the positive z face for each analysis (a.) low-refinement, b.) high-refinement, c.) shrunken tow).



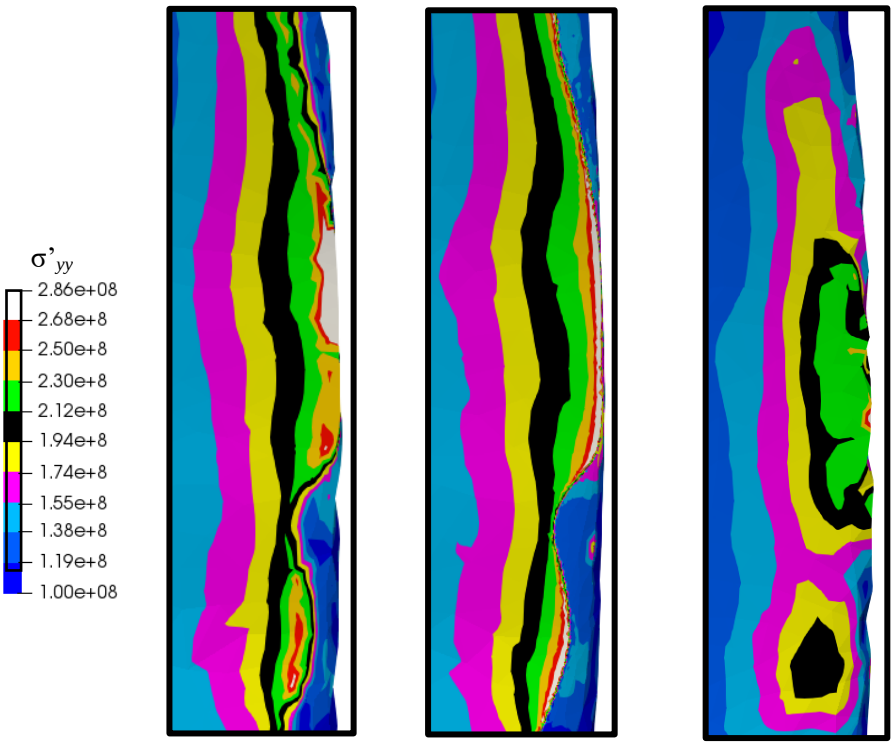
a) Low-Refinement **b) High-Refinement** **c) Shrunken Tow**

Figure 5.21: Local XX-direction stress in the weft tows for each analysis.



a) Low-Refinement b) High-Refinement c) Shrunken Tow

Figure 5.24: Stress in the weft tows local YY direction on the positive z face for each analysis (a.) low-refinement, b.) high-refinement, c.) shrunken tow). Dotted lines indicate boundaries of warp tows. Arrows indicate location of peak stresses common across all analyses.



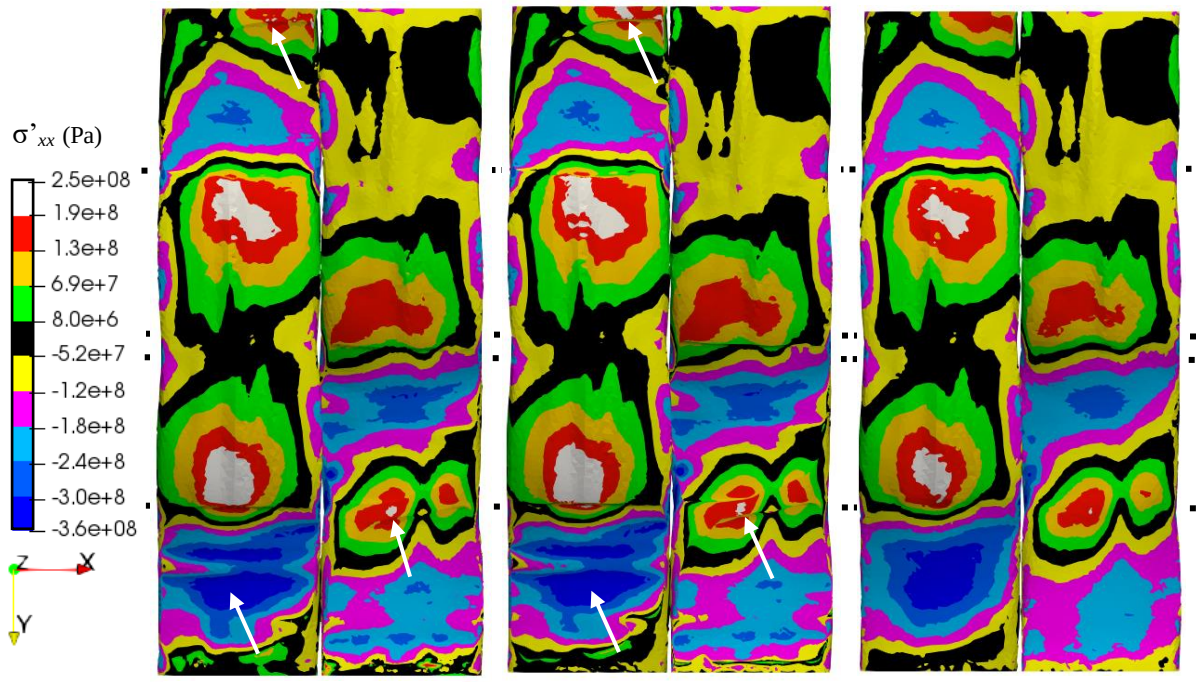
a) Low-Refinement b) High-Refinement c) Shrunken Tow

Figure 5.23: Close up view of high local YY-direction stress near edge of weft tows for each analysis

the weft tows because the warp tow is transferring a significant amount of load into the weft tows along these boundaries. This results in a high stress where the weft tow is less stiff in the loading direction than the warp tow, but must deform the same amount. This transference of load happens across the entire portion of the boundary between the tows, but is more severe in regions where the boundary is perpendicular to the loading direction.

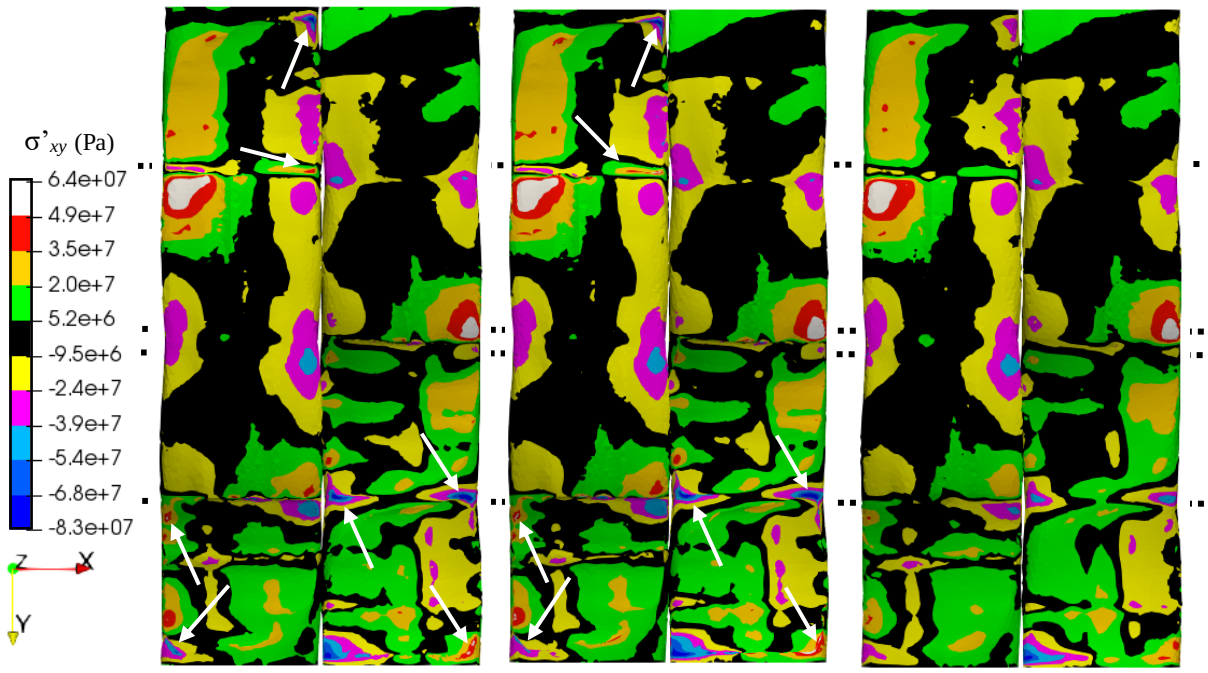
The last factor contributing to these large stress concentrations is the geometry of the textile itself and how the warp tow is weaving from the negative z-side of the textile to the positive z-side. In regions where the tows are connectec but the warp tow does not transfer sides, there are no observable large tensile stress concentrations in σ'_{yy}. This is because the warp tow takes more of the x-direction load as it is better aligned in the x-direction. The tow's local x-direction is less aligned in the loading direction when transferring sides causing more load to be distributed into the weft tow. This behavior is why a stress concentration is seen in the same location in the shrunken tow analysis. This is also why not all surface boundaries aligned perpendicular to the loading direction has a tensile stress concentration.

Figure 5.25 shows σ'_{xx} for the weft tows. The stress contours for each analysis are similar apart from certain areas marked with white arrows. These variations are due the volume removed by the NURBS method which cause there to be less material to distribute the load amongst in the interpenetration regions. The variations are most noticeable where the surface of the tow is close to the edge of the bounds of the model along the y- and z-face of the model. Also, the stress contours in the region where tows are connected are not as smooth as the shrunken tow analysis. This is due to the waviness of the surface in the shared regions which varies the thickness of the tow enough to influnce the stress carried in the region. The waviness is introduced during the surface replacement step where the surface of the warp tow is used to cut away the surface of the



a) Low-Refinement b) High-Refinement c) Shrunken Tow

Figure 5.26: Local XX-direction stress in the weft tows for each analysis. Arrows indicate regions where the contours of the peak stresses are affected by the shape of the NURBS method tows.



a) Low-Refinement b) High-Refinement c) Shrunken Tow

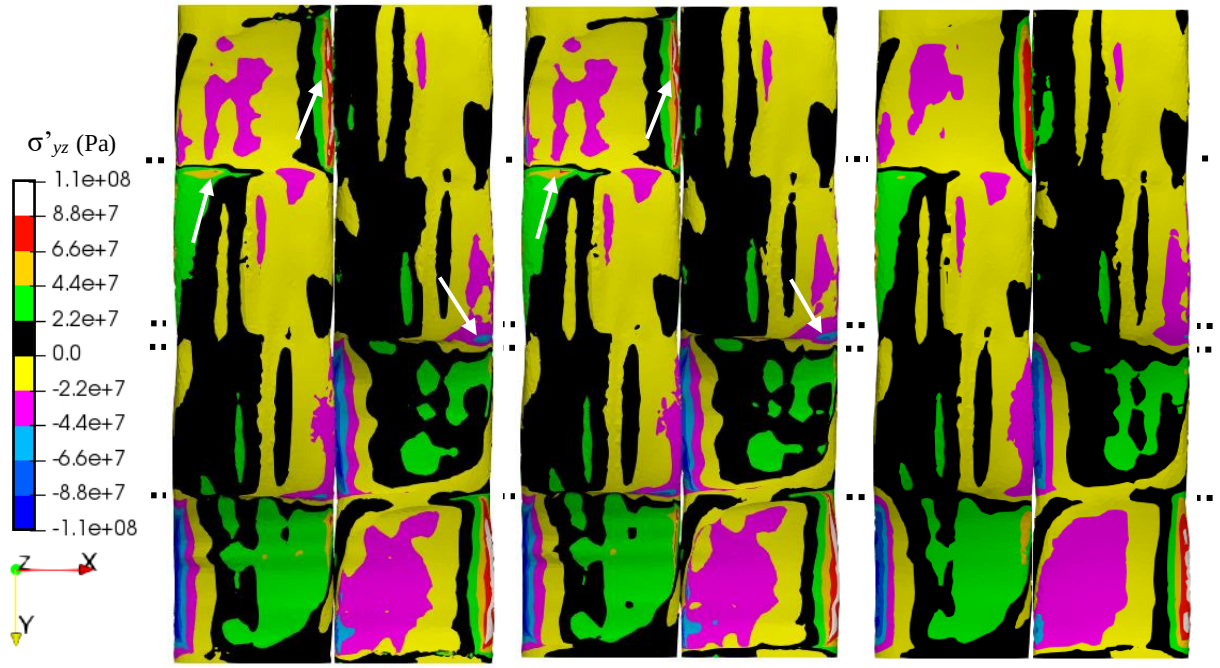
Figure 5.25: Local XY-direction shear stress in the weft tows for each analysis. The arrows indicate elevated concentration magnitude due to edge of connected region between warp and weft tows.

weft tow. The warp tow surface has undulations in it due to the VTMS method. There are no major differences in σ'_{zz}.

Figure 5.26 shows σ'_{xy} for the weft tows. In Figures 5.26a and 5.26b there are multiple arrows to indicate stress concentration areas that have increased magnitude (compared to Figure 5.26c) due to the warp and weft tows being connected. The in-plane shear stresses are increased along in these regions due to the large σ'_{yy} stresses being transferred from the x-facing side of the tow, around the edge of the connected region, and into the thicker part of the tow. The NURBS method tows have higher magnitudes in these regions because the warp tow is directly connected to the weft tow, which imparts a direct shear stress on the weft tow to transfer some of the load. The shrunken tow method transfers this load through the small amount of matrix between the tows which softens the transfer of load. For the NURBS method tows, this load is transferred to the regions of the weft tow where the cross-sectional area between interpenetration regions was unaffected. These regions leave a small amount of weft tow between two warp tows that can take some of this load where the shrunken tow method does not have any weft tow volume and has to transfer this load through the matrix. Therefore, the concentration magnitude is higher.

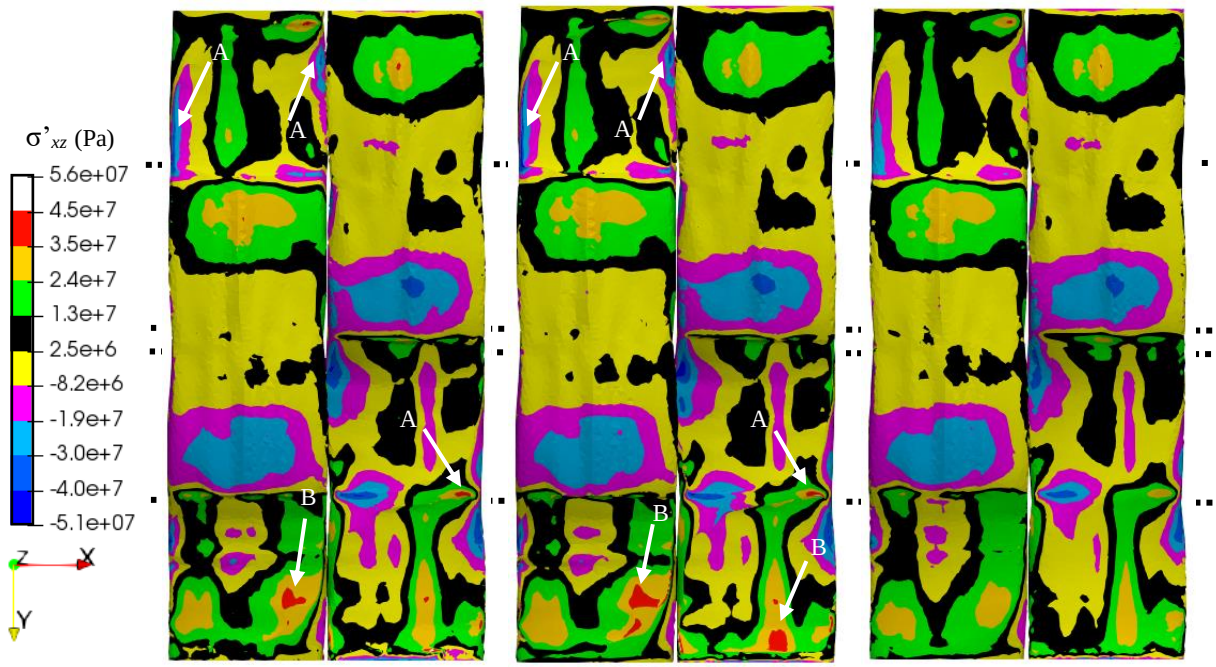
Figures 5.27 and 5.28 show σ'_{yz} and σ'_{xz}, respectively. In Figure 5.27, there exists only a few regions where the NURBS method analyses predict higher magnitude stress concentrations versus the shrunk tow method. These regions are only where the connecting regions between warp and weft tows end and the existance of this region is the cause of the elevated magnitude of stress, but not the cause of a concentration itself. The shrunken town method also exhibits concentrations in these regions, but has a smaller magnitude for the stress.

Similar to σ'_{yz}, Figure 5.28 shows how there are certain regions where the stress magnitude is increased around the edges of a connecting region (indicated by “A” arrows). These concentrations



a) Low-Refinement **b) High-Refinement** **c) Shrunken Tow**

Figure 5.27: Local YZ-direction shear stress in the weft tows for each analysis. Arrows indicate increased stress concentrations unique to the NURBS method analyses.



a) Low-Refinement **b) High-Refinement** **c) Shrunken Tow**

Figure 5.28: Local XZ-direction shear stress in the weft tows for each analysis. The "A" arrows indicate concentrations due to the boundary of the connected region between warp and weft tows. The "B" arrows indicated concentrations due to the warp tow surface replacing the weft tow surface and creating large variations the tow cross-section

exist in the shrunken tow analysis as well (less than 3.0E+07), but the increased magnitude (greater than 3.0E+07) is attributed to the connection between warp and weft tows and the discontinuity of material properties. However, Figure 5.28 also shows certain areas ("B" arrows) that have elevated shear stress magnitudes that are larger in size and not directly located on the boundary of the connected region. There is some elevated stress in these regions for the shrunken tow method, but the larger magnitude for Figures 5.28a and 5.28b is due to the reduced volume in these areas from the interpenetration removal method and the undulations in the surface. The undulations cause small variances in the cross-sections and as the warp tow attempts to pull away from the weft tow the load is transferred out-of-plane and through the thickness of the weft tow, causing a stress concentration. This same interaction occurs in the shrunk tow analysis but the larger amount of tow volume in this regions reduces the stress magnitude.

The weft tows have exhibited similar behave to the warps tows concerning incresed stress concentration magnitude around the edge of the tow connection regions. However, the weft tows also exhibit other concentrations directly due to the tow volume not being equally removed from warp and weft tows. This shows that there is still some work needed to resolve this inequality.

5.3.3 Interaction between tows

In the previous sections, the interaction between warp and weft tows was presented and discussed in detail. However, a graphical representation of these interactions was not provided. In this section, the interaction between warp and weft tows will be shown. A detailed discussion will be had on how each method affects this interaction and the benefits and drawbacks to each.

Figure 5.29 shows an out-of-plane shear stress component that best illustrates an interaction between a warp and weft tow. The top of Figure 5.29 shows a warp tow (number 3) from the high refinement case with σ'_{yz} contour plot shown. The white arrow indicates a concentration that is

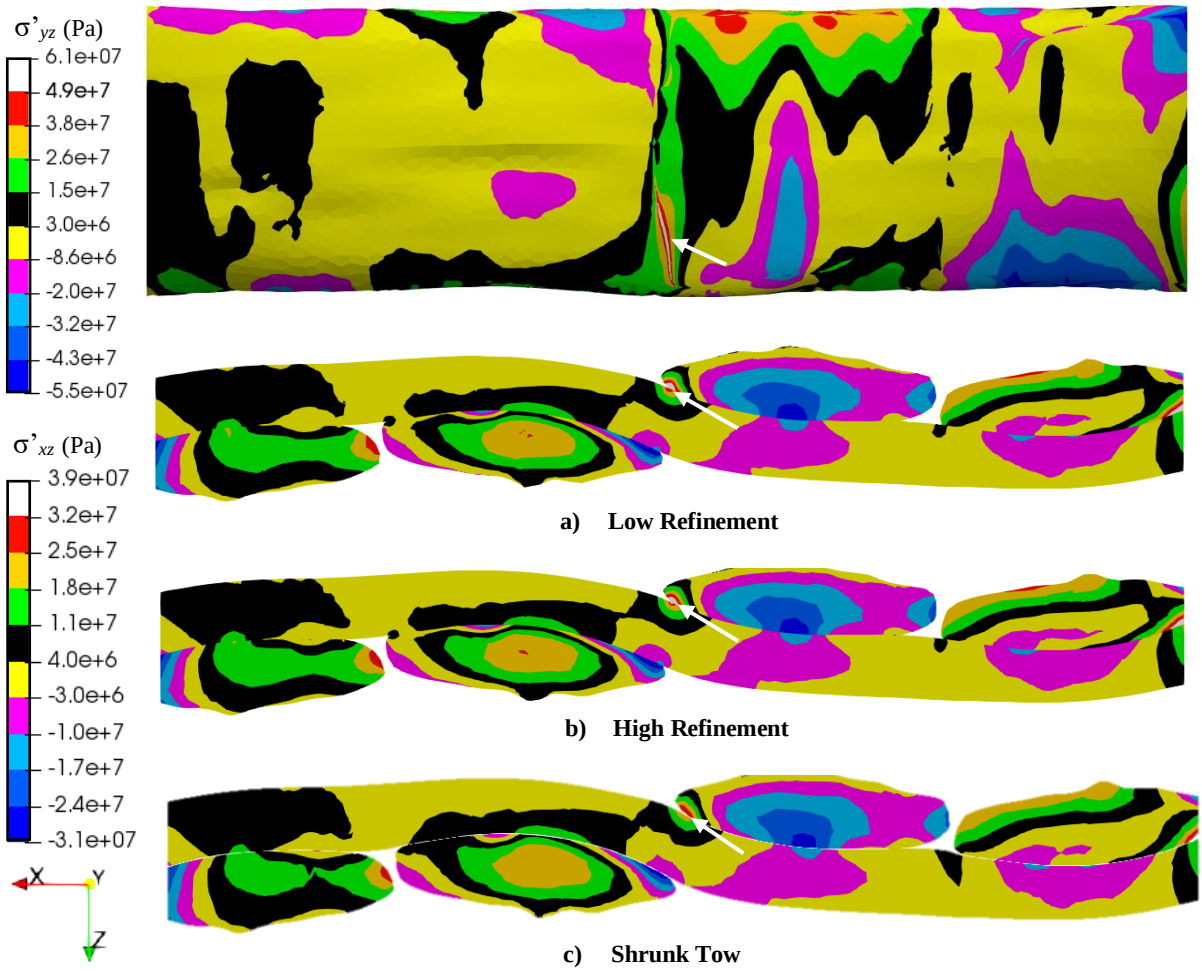


Figure 5.30: Comparison of shear stress transfer for out-of-plane stresses. Warp tow shows local YZ stress and weft tows show local XZ stresses. Arrows show concentration of interest on warp tow and the corresponding region on the slice view.

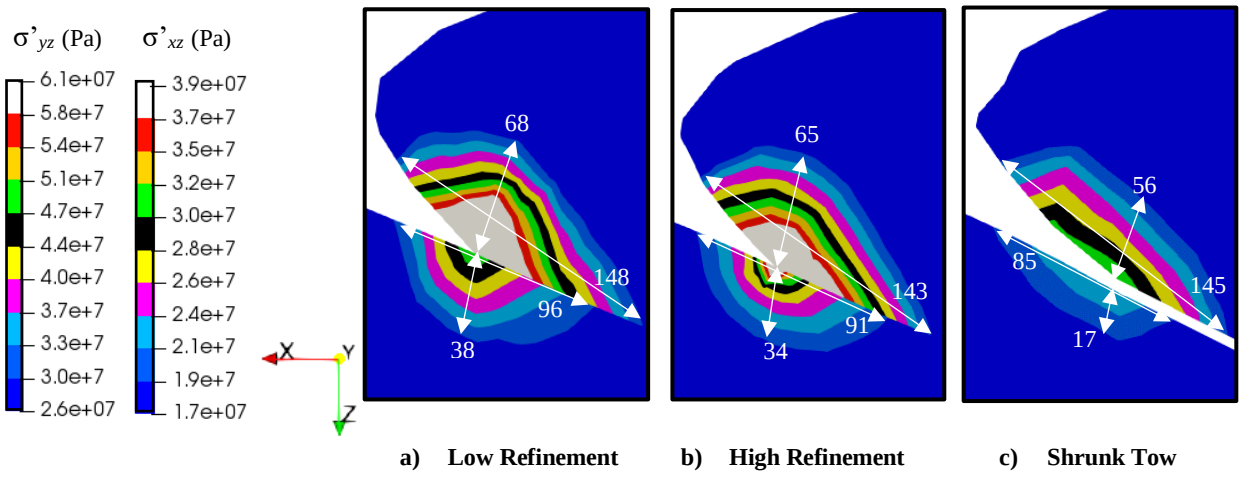


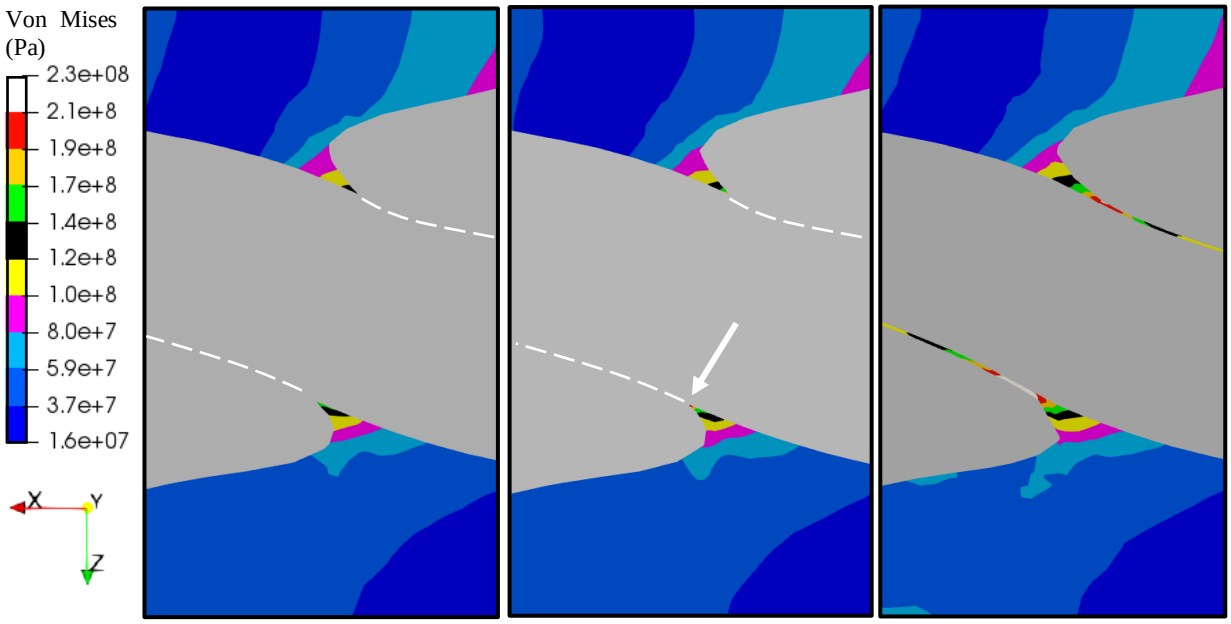
Figure 5.29: Close up comparison of shear stress transfer for out-of-plane shear. Warp tow shows local YZ stress and weft tows show local XZ stresses. Measurements are in microns.

high in magnitude for all three analyses but the NURBS method analyses have a higher peak magnitude. The NURBS method Figures 5.29a, 5.29b, and 5.29c show how this shear stress is transferred between the warp and weft tows for each analysis. The weft tows are displayed with σ'_{xz} so that the shear stress components are aligned in the global coordinate frame σ_{YZ}. It can be seen how the shear stress between the warp and weft tows is related and how the stresses reach a maximum at the edge of the connecting regions.

Figure 5.30 examines this region is close detail and the gradient of the high stress regions can be seen. The gradient to the high stress in the weft tow is approximately 65 by 143 microns (9 by 20 fiber diameters) in the high refinement case (Figure 5.27b), 68 by 148 microns (9 by 21 diameters) in the low refinement case, and 56 by 145 microns (11 by 21 diameters) in the shrunk tow case. The gradient in the warp tow for the high refinement case is 34 by 91 microns, the low refinement case is 38 by 96 microns, and the shrunk tow case is 17 by 85 microns. It is shown that the gradient size to each concentration is similar in size, however the magnitude of the concentration is significantly higher in the NURBS method analyses. In the region shown, the higher concentration can be definitively proven to be in direct response to the tows being connnected to one another and the singularity caused by the different orthotropic material properties. This is enforced by the lack of connection between the tows in Figure 5.30c where space can be seen between the tows. This space is occupied by matrix which acts to soften the load transferred between the two tows. In the next section, the matrix of each analysis will be compared.

5.3.4 Matrix

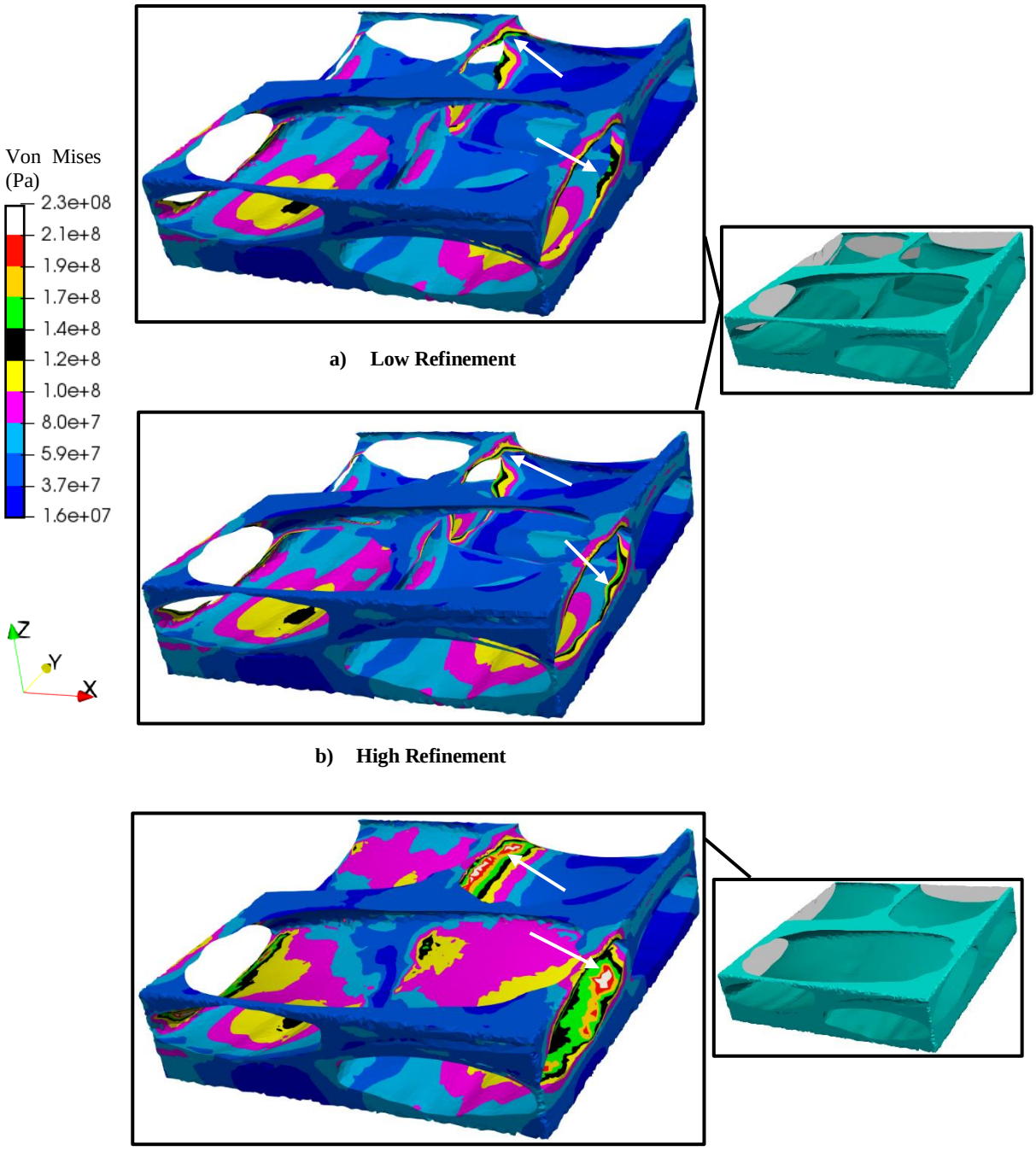
The previous sections provided an overview of the stress contours for warp and weft tows for all three analyses and then explored how load is transferred between orthogonal tows. This section will explore the role of the matrix and how the matrix facilitates the transfer of load between tows.



a) Low Refinement **b) High Refinement** **c) Shrunk Tow**

Figure 5.31: Von Mises stress in matrix surrounding interpenetration regions of orthogonal tows for each analysis

The focus will be on the regions around previous areas of interpenetrations as the matrix away from these regions are very similar in the predicted response. First, the region of matrix around Figures 5.29 and 5.30 is shown in Figure 5.31. The Von Mises stress is shown as it provides a general understanding of overall stress in these regions for an isotropic material. It is shown that the NURBS method analyses predict a lower magnitude stress concentration in the region where the tow compatiblity region ends (white dotted lines indicate boundary of tows). The shrunken tow method predicts a higher magnitude stress in this region (greater than 2.1E+08 Pa) primarily in the region of matrix missing from the NURBS method analyses. This corresponds to previous observations of the NURBS method analyses carrying more load in the tows in this region compared to the shrunken tow analysis. Figure 5.31b has a region at the tip of the matrix (white arrow) that reaches a similar stress magnitude to the shrunken tow analyses, but is much smaller in volume. Figure 5.31c shows that the matrix is transferring the load between tows and therefore



a) Low Refinement

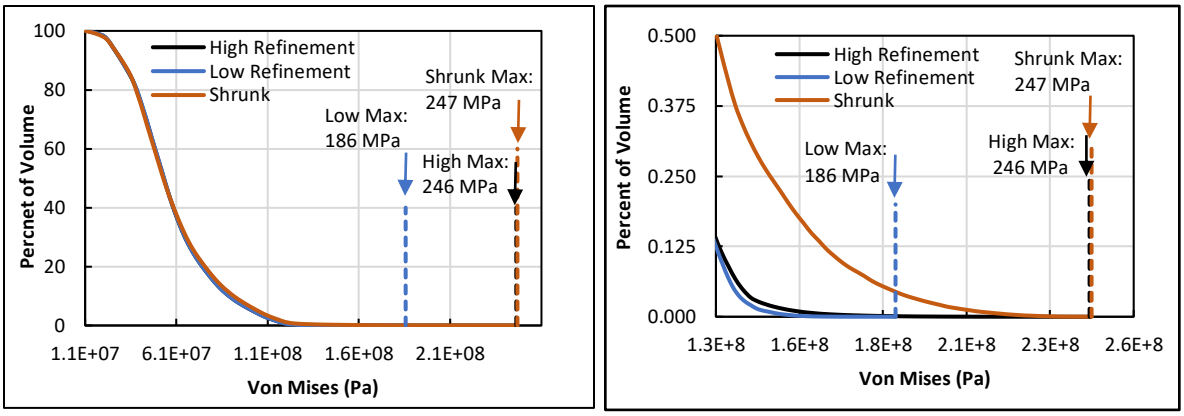
b) High Refinement

c) Shrunk Tow

Figure 5.32: Von Mises stress in complex tow regions for all three analyses

is subjected to a higher stress. It is also shown that the stress contours in the matrix are nearly identical where the matrix exists in all three analyses.

Figure 5.32 shows a larger cutout region of matrix for each analysis where there are a few tows in close proximity to each other. Using the pop outs to the right in Figure 5.32, it can be seen where the two method's matrix differ and how there exists some matrix between tow in the shrunken method that has been removed in the NURBS method. There are concentrations (indicated by white arrows) in the thin (or edge) regions of matrix. These regions either surround or replace the tow interpenetration regions and therefore strictly exist because of the chosen interpenetration removal method. These regions are of significant interest as this is where damage in the matrix commonly initiates in physical testing. The stress coutours away from these regions are very similar across all three models. However, the regions with the highest stresses (white arrows) have some differences. The shrunken tow method has a significantly higher (greater than 2.3E+08 Pa) von Mises stress than the low refinement case (less than 1.9E+08 Pa). The high refinement case achieves a similar concentration magnitude as the shrunken tow case but in a smaller amount of matrix right the edge. The gradient from the highest concentration magnitude to the lower stress regions is very similar across all three analyses. This indicates that each analysis predict similar concentration location and gradient. The shrunken tow method predicts higher concentration magnitude and volume in these regions as it is the only method of load transfer between tows, whereas the NURBS method analyses transfer more load through the tow-to-tow connection. The increased concentration magnitude in the high refinement case is due to the higher number of elements around this region and convergence to a stress singularity between the three different materials.



a) Full volume region

b) High stress Volume

Figure 5.33: Distribution of Von Mises stress through volume show in Figure 5.32 for all three analyses

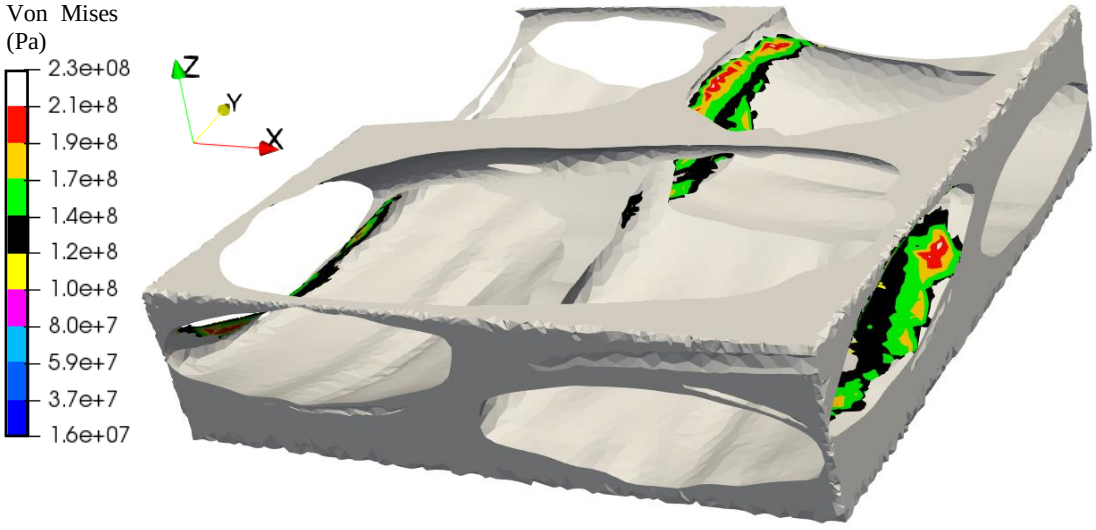


Figure 5.34: Matrix of shrunken tow analysis (contoured) at high stress that does not exist in NURBS method analysis (white matrix)

Figure 5.33 shows the distribution of Von Mises stress through the region in Figure 5.32 for each analysis. It is shown how each analysis predicts a similar amount of volume at a given stress for most of the volume. The difference between analyses is shown at the higher stresses where the low refinement analysis predicts a lower peak stress (1.86E+08 Pa) compared to the high refinement (2.46E+08 Pa) and the shrunken tow analysis (2.47E+08 Pa). Figure 5.33 also shows that at stress values above 1.3E+08 Pa, the shrunken tow method has a larger amount of volume (0.5% of the total volume) at this stress than the low and high refinement (0.12% and 0.125%, respectively) analysis. However, these numbers may be misleading because much of the matrix in the shrunken tow analyses above this stress value does not exist in the other two analyses. Figure 5.34 shows the high Von Mises stress matrix from the shrunken tow analysis that does not exist in the analyses where the tows are connected to each other. The contoured matrix shown corresponds to the matrix volume shown in Figure 5.33a. These two figures show that while the shrunken tow method may predict more volume at higher stress, it does not indicate that the matrix of the shrunken tow analysis is more susceptible to higher stress. Instead, it can be seen that the shrunken tow method has more volume in the high stress regions and therefore more volume to be subjected to high stresses.

6. CONCLUSIONS

The purpose of this research was to develop and implement a method (NURBS method) to resolve interpenetrations between tow surfaces created by VTMS. The results of this method as well as the predicted stress response were compared to a pre-existing method (shrunken tow method) used to remove interpenetrations to determine the benefits and drawbacks to each method. This was accomplished by starting with the same textile configuration for each method and applying the same matrix generation algorithm to create a finite element mesh with strict node-to-node compatibility. Each model was subjected to the same boundary and loading conditions to further reduce any potential variances due to boundary conditions and/or textile configuration. Each model was subjected to a detailed analysis to understand how each method affected the predicted stress response in both tow and matrix volumes.

It was shown that two major factors contributed to the major differences between the two methods. One, the method presented in this research creates a region between tows where there exists compatibility between the surfaces of each tow with no matrix material in between. This compatibilty forces a direct tranfer of stress between tow volumes, which can cause elevated stress concentration magnitudes, particularly along the boundary of these regions. This effect is most promenent for in-plane normal stresses and out-of-plane shear stresses for the tows. The elevated stresses in these regions are common across both methods. However the localization and higher magnitudes seen in the NURBS method analyses are directly related to the creation of the compatible surface regions and the singularity in the model caused by the convergence of three different material properties on the same point (warp tow, weft tow, and matrix). It is also shown that this effect can be reduced by lowering the refinement of the mesh around the boundary of

these regions. The lower refinement of the mesh does not allow for the high gradients and high magnitude of stress at the singularities around the boundary of where warp and weft tows are connected. Outside of these localizations, the predicted response in the tows is similar in most cases. In the situations where they do not match, a second factor comes into play.

The second factor is how each method removes volume in both tows and matrix. The method shown in this research removes all interpenetration volume exclusively from the weft tows. This is due to how the tows are processed (weft tows consistently identified as secondary tows in the method) and the use of primary tow surfaces being used to replace secondary tow surfaces that had been removed. It was shown that warp tows in the NURBS method consistently had more volume than the shrunken tow method's warp tows, while the inverse was true for weft tows. It was also shown that the warp tows in the NURBS method closely matched the original volume of the VTMS warp tows, but had more volume removed from the weft tows than the shrunken tow method. The effect of this is that normal stress (σ'_{yy}) aligned in the loading direction in weft tows is lower in interpenetration regions for the NURBS method but elevated for stress perpendicular to the loading direction (σ'_{xx}). The lower volume in warp tows between analyses does not affect the predicted results with the boundary conditions used.

The NURBS method also removes the matrix that exists between tows in the interpenetration regions. Parts of the very thin matrix between tows in the shrunken tow method are shown to be at high von Mises stress which results in more of the matrix volume appearing to be at these high stresses. However, it is shown that the matrix for the NURBS analysis can also reach these high magnitudes with sufficient mesh refinement and the location of these concentrations are not unique to the shrunken tow method. Therefore, models that predict damage may see similar initiation locations across both analyses, provided the mesh refinment of the matrix is high enough. If the

mesh refinment of the matrix for the NURBS method is not high enough, the stress magnitude will not converge to the magnitudes predicted in the shrunken tow analysis.

The most important benefits of the NURBS method are the complete and detailed descriptions of the interpenetration regions between tows and the ability to directly identify surface elements that are in contact with another tow. The regions of interpenetration have been explicity identified and defined so that they may be corrected by whatever means the user choose. This is an improvement over other methods that identify interpenetrations through point-to-surface penetration detection algorithms and may require multiple iterations to ensure interpenetration removal. These detection methods can also overlook complex interpenetration cases that occur in discrete surface meshes, such as element edge-edge intersection. The method shown in this research also allows for the specific tow-to-tow surface elements to be identified and used for any type of contact or cohesion modeling if desired.

The drawbacks of the NURBS method is the artificial concentrations that can occur around the perimeter of the tow connection regions. The magnitude of these concentrations can be artificially increased due to the mesh compatiblity between three different materials in one location. It is shown that reducing the mesh refinement can reduce these effects, but may also underpredict stress concentration magnitude in the matrix. Also, the NURBS method can significantly reduce the volume in secondary tows (specifically weft tows in the shown configuration) compared to the primary tows. The result of this may be more prominent in more complex loading cases than the one shown in this research.

This research has explored a new method to remove interpenetrations between tow surfaces generated by VTMS. Future work on this method should include a better method for removing volume from both primary and secondary tows uniformly. Also, a study on various contraint

models for the connected regions between tows should be conducted as the strict node-to-node displacement constraint between tow meshes may artificially raise stress concentration magintudes along the boundary of where tows are directly connected to one another.

REFERENCES

[1] Cox, B.N.; Dadkhah, M.S.; Morris, W.L. “On the tensile failure of 3D woven composites.” *Compos. A Appl. Sci. Manuf.* 1996, *27*, 447–458

[2] Chen, X.; Spola, M. “Experimental studies on the structure and mechanical properties of multi-layer and angle-interlock woven structures.” *J. Text. Inst*. 1999, *90*(1), 91–99

[3] Quinn, J.P.; McIlhagger, A.T. “Examination of the failure of 3D woven composites.” *Compos. A Appl. Sci. Manuf.* 2008, *39*, 273–283

[4] Naik, N.K.; Shembekar, P.S. “Elastic behavior of woven fabric composites: I-lamina analysis.” *J. of Comp. Matls.* 1992, 26, 2196-2225.

[5] Ishikawa, T.; Chou, T.W. “Stiffness and strength behavior of woven fabric composites” *J. Matls. Sci.* 1982, 17, 3211-3220

[6] Whitcomb, J. D. “Three-Dimensional Stress Analysis of Plain Weave Composites” *NASA Tech Memo*. 1989

[7] Naik, R. A. “Analysis of Woven and Braided Fabric Reinforced Composites” *NASA Tech Memo*. 1994

[8] Raju, I. S.; Foye, R. L.; Avva, V. S. “A Review of the Analytical Methods for Fabric and Textile Composites” *Composite Structures: Testing, Analysis and Design.* 1992, 273-293

[9] Naik, N.K.; Shembekar, P.S. “Elastic behavior of woven fabric composites: II-lamina analysis.” *J. of Comp. Matls.* 1992, 26, 2226-2246.

[10] Wang, Y.; Sun, X. “Digital-element simulation of textile processes.” *Comp. Sci. and Tech.* 2000, 61, 311-319.

[11] Wang, Y.; Miao, Y.; Swenson, D.; Cheeseman, B. A.; Yen, C.; LaMattina, B. “Digital element approach for simulating impact and penetration of textiles.” *Intl. J. of Impact Eng*., 2010, 37, 552-560.

[12] Maio, Y.; Zhou, E.; Wang, Y.; Cheeseman, B. A. “Mechanics of textile composites: Micro-geometry.” *Comp. Sci. and Tech.* 2008, 68, 1671-1678

[13] Iarve, E. V.; Mollenhauer, D. H.; Zhou, E.G.; Breitzman, T.; and Whitney, T. J. “Independent mesh method-based prediction of local and volume average fields in textile composites.” *Composites: Part A*, 40:1880-1890, 2009

[14] McQuien, J. S. “Evaluation of Independent Mesh Modeling for Textile Composites” *Masters Thesis*. 2017

[15] C. Fares, Y. Hamam. “Collision Detection for Rigid Bodies: A State of the Art Review.” *GrapiCon.* 2005

[16] Jiménez, P.; Thomas, F.; Torras, C. “3D collision detection: a survey.” *Computers & Graphics.* 2001, 25, 269-285.

[17] Dobkin, D.; Kirkpatrick, D. “Determining the separation of preprocessed polyhedral – A unified approach.” *Lecture Notes in Comp. Sci.* 1990, 443, 400-413.

[18] Canny, J. “The complexity of robot motion planning.” *MIT Press, Cambridge Mass.* 1987.

[19] Chazelle, B. “Convex partitions of polyhedral: a lower bound and a worst-case optimal algorithm.” *J. Comput.* 1984, 13, 488-507.

[20] Chazelle, B.; Dobkin, D.; Shouraboura, N.; Tal, A. “Strategies for polyhedral surface decomposition: an experimental study.” *Computational Geometry: Theory and Applications.* 1997, 7(4-5), 327-342.

[21] O’Rourke, J. “Computational geometry in C.” *Cambridge University Press, Cambridge, Mass.* 1998

[22] Thomas, F.; Torras, C. “Interference detection between non-convex polyhedral revisited with a practical aim.” *Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, CA.* 1994, 587-594

[23] Kriezis, Ga.; Prakash, P.V.; Patrikalakis, N.M. “Method for intersecting algebraic surfaces with rational polynomial patches.” *Computer-Aided Design.* 1990, 22, 645-654.

[24] Pentland, A.; Williams, J. “Good vibrations: modal dynamics for graphics and animation.” *In Computer Graphics.* 1989, 23, 215-222

[25] Lin, M.C.; Manocha, D. “Fast interference detection between geometric models.” *The Visual Computer.* 1995, 11, 542-561.

[26] Lane, J.M.; Riesenfeld, R.F.; “A theoretical development for the computer generation and display of piecewise polynomial surfaces.” *IEEE transactions on Pattern Analysis and Machine Intelligence.* 1980, 2(1), 150-159

[27] Snyder, J. “Interval arithmetic for computer graphics.” In *Proceedings of ACM Siggraph.* 1992, 121-130.

[28] Drach, A.; Drach, B.; Tsukrov, I. “Processing of fiber architecture data for finite element modeling of 3D woven composites.” *Advances in Engineering Software.* 2013.

[29] Barnhill, R.; Farin, G.; Jordan, M.; Piper, B. “Surface/surface intersection.” *Computer Aided Geometric Design.* 1987, 4, 3-16.

[30] Bajaj, C.L.; Hoffmann, C.M.; Hopcroft, J.E.H.; Lynch, R.E. “Tracing surface intersections.” *Computer Aided Geometric Design*. 1988, 5, 286-307.

[31] Hohmeyer, M.E.” A surface intersection algorithm based on loop detection.” *International Journal of Computational Geometry and Applications.* 1991, 1, 473-490.

[32] Manocha, D.; Canny, J.F. “A new approach for surface intersection.” *International Journal of Computational Geometry and Applications.* 1991, 1, 491-516

[33] Sederberg, T.W.; Anderson, D.C.; Goldman, R.N. “Implicit representation of parametric curves and surfaces.” *Computer Vision, Graphics, and Image Processing.* 1984, 28, 72-84.

[34] Krishnan, S.; Manocha, D. “An efficient surface intersection algorithm based on the lower dimensional formulation.” *ACM Transactions on Graphics.* 1997, 16, 74-106.

[35] Drach, A.; Drach, B.; Tsukrov, I.; Bayraktar, H.; Goering, J. “Realistic FEA modeling of 3D woven composites on mesoscale.” *19th International Conference on Composite Materials.*

[36] Mazumder, A.; Wang, Y.; Yen, C. “A structured method to generate conformal FE mesh for realistic textile composite micro-geometry.*” Composite Structures,* 2020, 239, 112032

[37] “SISL - The SINTEF Spline Library.” SINTEF, 009, www.sintef.no/projectweb/geometry-toolkits/sisl/.

[38] Jonathan Richard Shewchuk, “Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator.” *Applied Computational Geometry: Towards Geometric Engineering*, vol. 1148 of Lecture Notes in Computer Science, 203-222, Springer-Verlag, 1996.

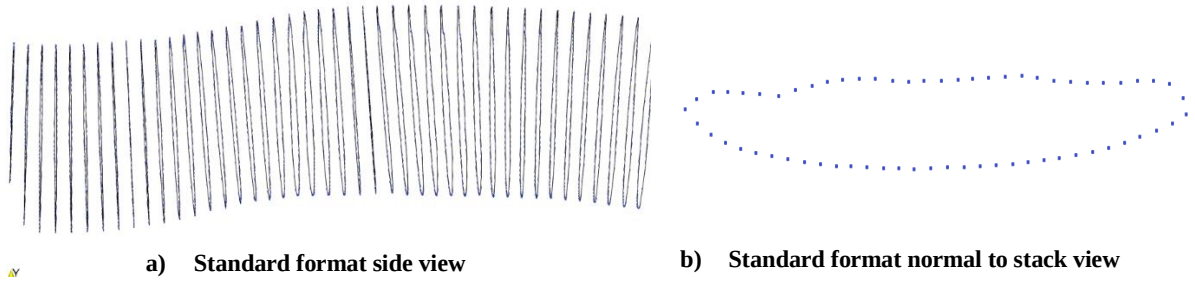
[39] M. K. Ballard, W. R. McLendon and J. D. Whitcomb, "The influence of microstructure randomness on prediction of fiber properties in composites," *Journal of Composite Materials,* vol. 48, no. 29, 3605-3620, 2014.

[40] Chong, K. S. "Collision Detection Using the Separating Axis Theorem." *Game Development Envato Tuts+*. Envato Tuts, 2012.

APPENDIX A DESCRIPTION OF DATA TYPES

The method of how VTMS produces tow surfaces was covered in the main text. However, once the surfaces are created, VTMS exports the surface data. Currently there are two forms of exported surface data. Each format will be discussed in detail.

A.1 Standard Tow Format



a) Standard format side view **b) Standard format normal to stack view**

Figure A.1: Views of VTMS standard tow format

One form of the exported surface data is the standard tow format. The standard tow format is exported after the relaxation steps that form the non-idealistic tow shapes. This format is made of stacks (cross sections) of the tow that are outlined by nodes. These nodes are the result of the surface roller method previously discussed and the number of nodes describing the cross-section outline is user defined. Each node has its coordinates stored in the file. The format of the file is by stack where each stack is listed in order along the length of the tow. In each stack, the nodes that make up the outline of the stack are listed with their coordinate. Figure A.1a shows the tow with the stacks normal (defined as the normal of the plane that all stack nodes lie on) in the plane of the page. Figure A.1b shows one stack from the tow with its normal perpendicular to the page. One important feature of this format is that the tow surfaces are open at the ends, resulting in a tube-like surface. Figure A.2 shows the numbering scheme for the standard format with consecutive

cross-sections labeled starting at ***i*** and node numbers starting at **1**. It is shown how each cross-sections initial node is highlighted with a red circle.

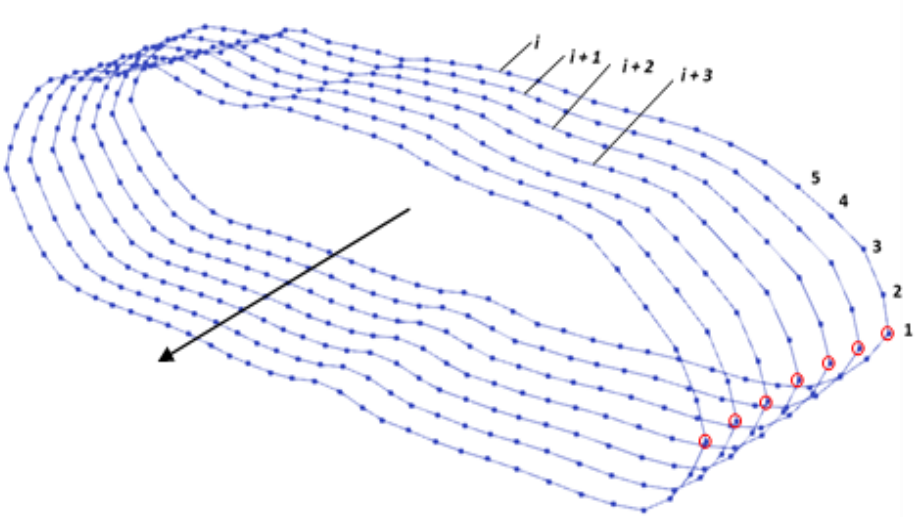


Figure A.2: Cross-section and node number scheme used in the Standard Tow (.stw) format in VTMS

VTMS also has a method to visualize this data in its own viewer. It creates surface triangles that connect surface nodes to create a faceted surface. However, it does not output this information to its standard tow format. VTMS uses this format to save and load surfaces after all relaxation steps have been completed so that the surfaces may be recovered by VTMS. They can be loaded into the tow modify module of VTMS where they can then be clipped to a user-defined size. This clipping results in the other VTMS format.

A.2 Clipped Tow Format

Once a standard tow is clipped in VTMS, the clipped surfaces can be exported in the clipped tow format. This surface format is very similar in layout to a finite element mesh in that the surface is defined by surface elements. The file exported stores each node and its coordinates followed by each element and its list of nodes that make up the corners of the element. The surface itself is primarily made up of triangular shapes but can be made up of general polygons as well. It is important to note that the clipped tow surfaces are not mesh ready as they can have surface shapes that have more than 4 sides. Therefore, some preprocessing is required to make the surfaces finite element ready. One feature is that once a clipped tow is created, VTMS closes the ends of the surface creating a closed surface representation of the volume occupied by the tow. Figure A.3 shows a clipped tow displayed in VTMS.

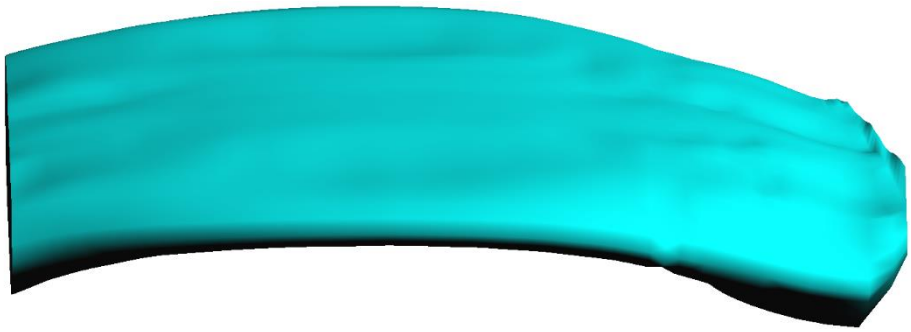


Figure A.3: VTMS clipped tow

The points that define the surface are the same points as the standard tow. When looking at the two surfaces side by side one can see that the stacks line up. This means that both surfaces are the same surface described in two formats and choosing one over the other does not change the surface being described.

APPENDIX B

This appendix covers research done during the early stages of this research. Originally, the goal was to improve the methods inside of VTMS using the VTMS architecture. The limitation of this architecture is that it only uses discrete representations for the tow surfaces and therefore is susceptible to the same shortcomings of methods that use faceted surfaces. However, this work has been included here to show other attempted solutions to the interpenetration problem. The result of the following methods led to the pursuit of new way to indentify and solve tow surface interpenetration.

B.1 Detection of Interpenetrations

A method was already in place inside of VTMS to detect interpenetrations. The default routine for VTMS uses a method that pulls known interpenetrating nodes along its outward normal back toward the tows center by a set fraction of the distance between neighboring cross-sections. The fractional value is set in the software. Nodes are found to be interpenetrating by calculating a vector between the node and a point known to be outside of the tow. This vector is compared against all surface triangles to see if this ray intersects. If this ray intersects then the point is contained in the tow. The method works for a majority of interpenetration cases but is not fool proof. Therefore, a more robust interpenetration routine was implemented.

This routine uses a ray-surface intersection algorithm between nodes and surface triangles created in VTMS using the standard tow format of the surface. First, a point and a surface are needed to compute a minimum distance. The point to test is from one tow and the surface needs to be chosen from the opposing tow. Suppose there is a node **N** that needs to be tested against tow **T**. Any planar surface needs three points to define it. Therefore, three nodes from tow **T** need to be

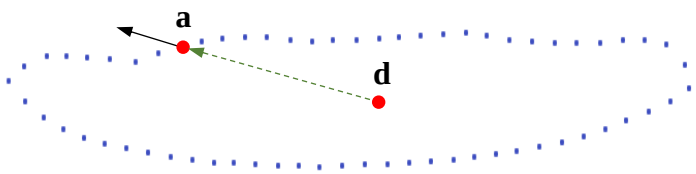


Figure B.1: Node outward vector using stack centroid

found to create a planar surface to test node **N** against. The algorithm loops over all the nodes in tow **T** and records which three nodes are the closest to node **N**. Once three nodes are found (nodes **a, b,** and **c**), a surface element is created from these points. There are two methods for doing this and the method depends on which surface format is being used.

For a standard tow, the stack to which the node (see node **a** in Figure B.1) belongs is identified. An outward vector for this node from the tow surface is created. To do this, the stack centroid (node **d** in Figure B.1) is calculated from the average location of all of the nodes of the stack. Then a vector is drawn from this centroid to the node **a**. This vector defines an outward direction relative to the tow surface at this node. This vector is not perpendicular to the tow surface. This is done for the three captured nodes (**a**, **b**, and **c**). The average vector of all outward vectors is taken and is projected onto the normal vector of the surface containing all three of the closest nodes to node **N**. This results in an outward facing vector **x** (outward relative to the inside of the tow surface) that

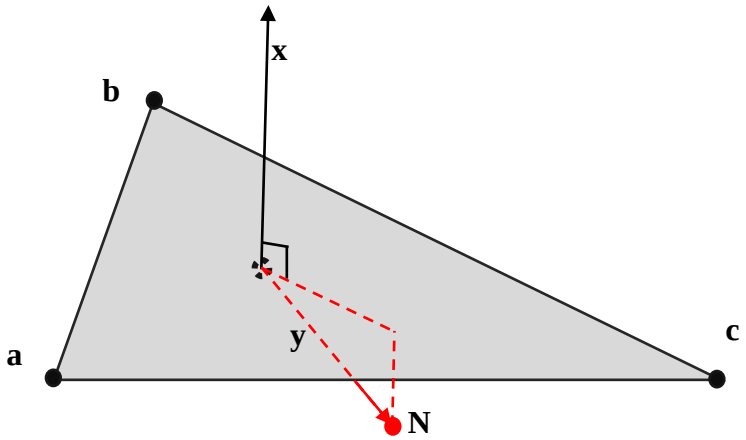


Figure B.2: Surface element with outward normal vector and relative vector to node N

is normal to the surface that contains the three nodes. If the artificial element normal is used, there is no guaranteed that it is outward facing. Using the three nodes along with an outward normal, a surface element is created. For a clipped tow, the algorithm is less involved. Because VTMS stores surface elements for a clipped tow, the element that uses all three nodes can be found and its outward normal can be used.

Once a surface element is created, a vector (vector **y** in Figure B.2) is created that runs from the average of the three captured nodes (**a**, **b**, and **c**) to the node of interest (node **N**). This gives the relative location of the node to the surface element. The dot product is calculated between vectors **x** and **y**. This dot product will result in a positive number if node **N** is outside of the tow surface and negative if it is inside. If the node is inside of the tow surface, the node is marked as being an interpenetrating node.This method is used for every node on both tows to collect all the nodes that are inside of an opposing tow.

The algorithm for detecting interpenetrating nodes for standard and clipped tows is very similar. The main difference is that a reference element has to be created for the standard tow format as there are no surface elements. Once all of the interpenetrating nodes have been found, an algorithm was developed to fix the interpenetrations for both formats.

B.2 Eliminate Interpenetration regions

Once the interpenetrated nodes were identified for each tow, an attempted solution was implemented to solve the interpenetrations. The goal of the solution is to remove the interpenetrations and create a compatible surface where the interpenetrations occurred using only the faceted VTMS surfaces.

B.2.1 Method 1: Artificial contact surface

The goal of this algorithm was to remove surface interpenetrations and establish a surface that is shared by both tows that will be referred to as a "contact surface". A flat surface was chosen as it is the simplest surface to implement into the algorithm. When two tows come in close proximity there can be multiple regions of interpenetrations that are disconnected (Figure B.3). Therefore, the algorithm must account for this possibility. This is accomplished by implementing a sphere detection algorithm that creates a spherical detection region around a known interpenetration node. Any interpenetrating nodes that lie within this detection region are collected. These collected nodes are given a spherical detection region to identify other interpenetrating nodes. This continues until there are no more interpenetrating nodes being detected for this specific node group. This creates node groups that are part of the same interpenetration region but excludes other nodes that are not part of the same region. The result is distinct node groups for each region of interpenetration. This is important because if we were to use all interpenetrating nodes at the same time with a flat surface, there could be large distortions in the tow surfaces. Once there are defined node groups, the algorithm continues below for each individual region.

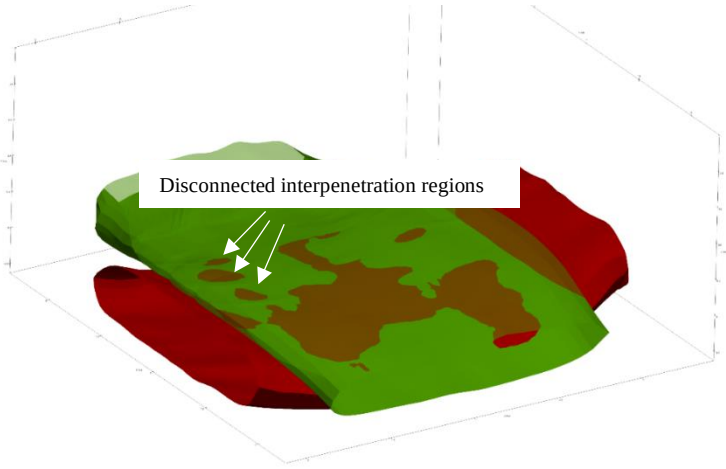


Figure B.3: Tow surface interpenetrations with emphasis on disconnected regions

The algorithm begins by calculating the mean location of all the nodes using all of the collected interpenetrating nodes from the two tows. This acts as the central point of the interpenetration.The algorithm then iterates over all of the interpenetrating nodes of one tow. For each node, the stack that it belongs to is saved in a list to reference later. If the stack already is in the list, then the algorithm continues to the next node in the list. We verify that the stack is not in a list using a simple find-in-list function. The result is a list that has every tow stack that has an interpenetrating node. A list is created for each tow surface so that a record of which stacks have interpenetrating nodes is kept.Then, for each stack that has interpenetrating nodes, the centroid (or mean) of all the nodes that make up the stack is calculated. The purpose is to give a node that roughly determines a central point for the stack. We then order the stacks in their list so that they are in the same order as they are listed in the tow surface data. This ensures that as the list is iterated over, the stacks are in order. This important when approximating the tow path.

Next, the centroids of the interpenetrating stacks in the tow surface are recorded. We order the centroids in their own list such that the centroid and the stack it belongs to can be referenced at the same index for their respective list. Suppose in Figure B.4 VTMS outputs the stacks as [**A**, **B**, **C**, **D**]. We ensure that the order of the saved centroids is in the order [**a**, **b**, **c**, **d**]. This ensures that when iterating over the centroid list we know that the next centroid corresponds to the next stack in the tow surface.

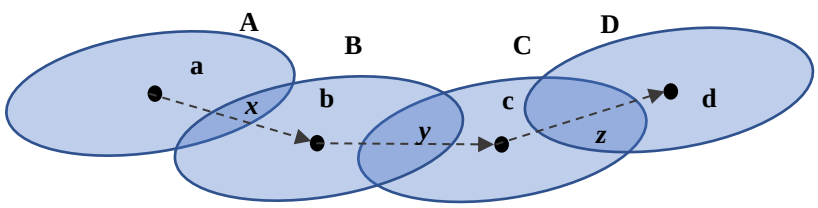
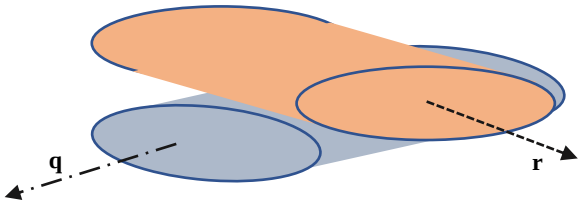
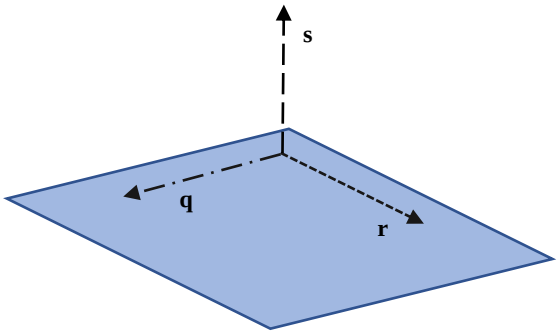


Figure B.4: Simplified stack representations with centroids and connecting vectors

Now that all the information concerning the interpenetration region has been prepared, a surface needs to be created for the tows surfaces to share. A plane was chosen as an initial surface shape as it is the easiest to implement. The normal vector to the plane needs to be established to create the surface. To create this normal, we start by calculated the vector between centroid in the interpenetrating stack centroid list and the centroid that immediately comes after it. This is done for every centroid except the last in the list. This results in a collection of vectors [***x***, ***y***, ***z***] that describe the direction the tow moves along it length. This is what we refer to as the tow path. For each tow, we average the vectors that are created in this step to give the general axial direction of the tow in the region where the interpenetrations occur (Figure B.5a, vectors **q** and **r**). We can use these vectors to determine how the two tows are oriented relative to each other in the region where they interpenetrate. We take the cross-product of these two vectors to find a vector that is perpendicular to the two tows in the region (Figure B.5b, vector **s**). Because the average tow path vectors are the average of the tow path along the entire interpenetration region, the result of the



c) Two tows with tow path vectors *q* and *r*



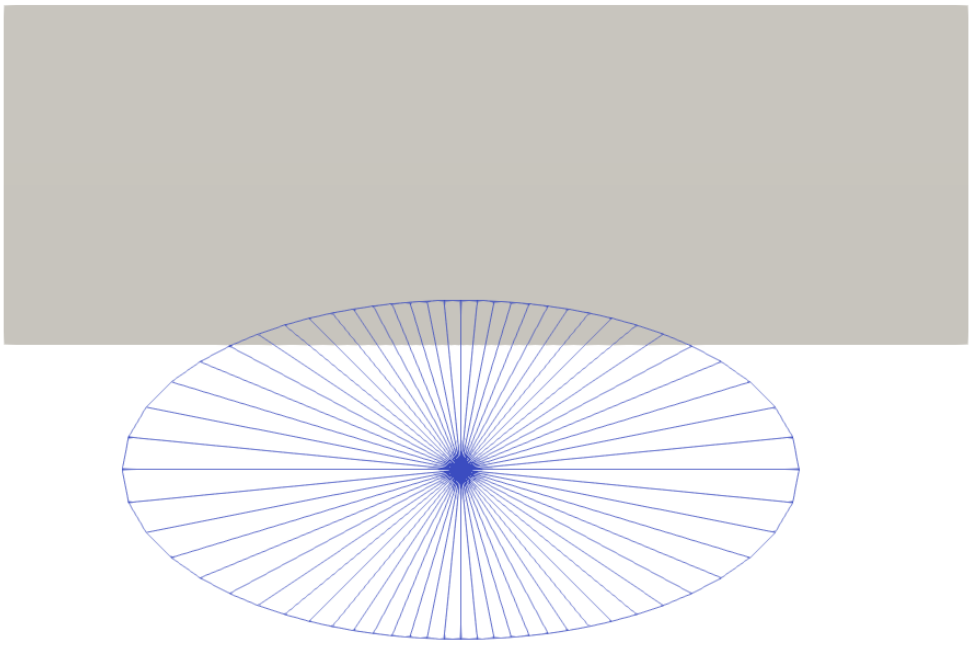
a) Cross product of *q* and *r* resulting in *s* that defines a plane

Figure B.5: Visualization of plane normal vector calculation

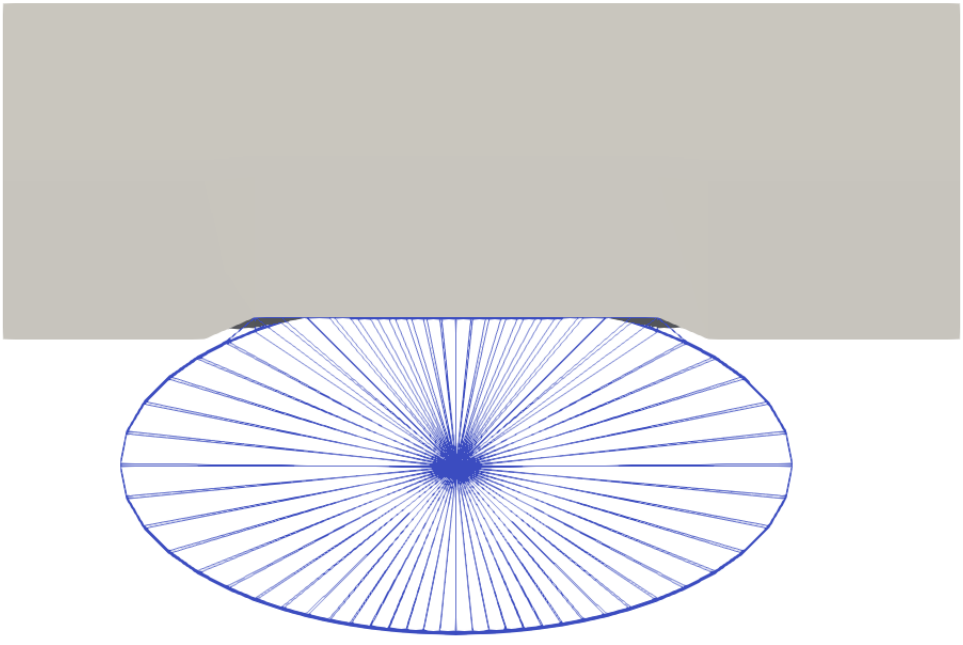
cross product is not perpendicular to both tows along the entire region. However, as long as the tows are not excessively curved along the interpenetration region, the perpendicular vector is relatively orthogonal to both tows.This perpendicular vector, along with the average, or centroid, of the interpenetration nodes, is used to create a flat that defines a plane along which the tows will share a surface. This plane is not parallel to both tow paths everywhere due to the curvature of the tow surfaces. However, it is a reasonable approximation for the interpenetration region.

Once the surface is defined, a vector from the centroid of the interpenetrating nodes to each interpenetration point is calculated. This gives the relative location of any point to the reference point (centroid) on the shared surface plane. This vector is then projected onto the normal vector of surface. The resulting vector projection (often referred to as the minimum translation vector, or MTV) accurately describes the exact translation the interpenetrating node must undergo to be on the plane. The node is then move along this vector to the plane.After all the nodes have been moved, a check is made to ensure that all of the interpenetrating nodes are no longer interpenetrating. The surfaces are then saved and exported so that they can be used later.

There exists another case where an interpenetrating node does not detect any other interpenetrating nodes in its detection sphere. When this occurs, the node finds the three closest nodes on the opposing tow and creates a surface element. The outward normal of the surface element is calculated using the same method as when it is calculated during the detection phase. This normal is what the interpenetrating node uses to project onto to find its minimal translation vector. Then the interpenetrating node is moved to the surface.



a) Two surfaces



b) Interpenetrations fixed with common surface

Figure B.6: Interpenetration and resolution of simple tows

Figure B.6 shows a simplifed case of the result of this method. Figure B.6a shows a slide where the simplified upper tow crosses through the lower tow coming out of the page. In Figure B.6b, the tows have been modified so that the tows share a flat surface between them and no longer interpenetrate.

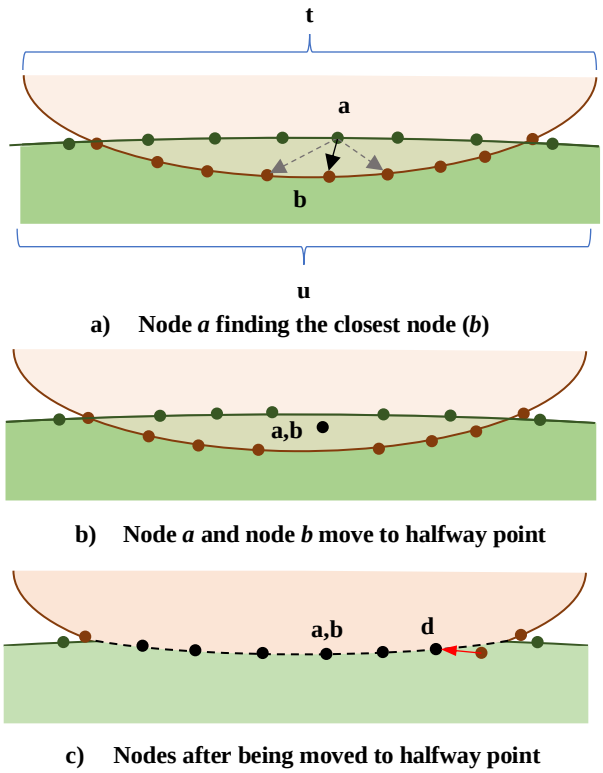
This method works great for multiple penetration regions between tows. The limitations are that this algorithm currently only creates flat surfaces between tows and does not ensure compatibility of tow meshes along contact region. This methods merit is that when the interpenetrations are fixed, the surface nodes are not simply moved beyond the boundary of the surface they penetrate. Many methods are satisfied with moving the interpenetrating nodes outside of the surface by a minimum distance. This method forces the nodes to lie on a surface so that the tows are in contact where their surfaces were interpenetrating. The result is a more realistic interaction between tows where they are in close proximity. This surface can then have some form of contact interaction imposed along this surface.

This method does have faults. When the interpenetration region is large or the tows are curved, the planar surface can change the topology of the tows excessively. Also, because the method uses a sphere-based detection algorithm for finding interpenetrating nodes, disconnected regions of interpenetrations can be grouped together unintentionally if the detection sphere is too large. This can lead to misalignment of the shared surfaces and can significantly modify the tow surfaces. We considered creating a master surface and a slave surface between two tows as it would eliminate the need for defining a surface between the two tows. However, we became concerned that if one tow in a fabric was set as a slave for every interaction, the tow volume could be affected greatly from the original volume. This would have a worse effect on the material properties of the fabric during analysis than if the surface approximation between the tows is slightly misaligned.

Enforcing compatibility is another issue with this method. It is difficult to find a node from each tow to pair and make a compatible mesh because of the stack description style. The stacks of two tows that are orthogonal rarely have nodes in close proximity. Also, moving surface nodes can result in high-aspect ratio elements and collapsed surface elements. For all these reasons, a different method was developed using the clipped tow format.

B.2.2 Method 2: Node-to-Node half distance compatibility solution

This method came from the need to solve the issue of the lack of compatibility between two tows along a shared surface. The method for detecting the interpenetrations between two polygon surfaces is the same as for the standard tow. To solve the interpenetrations, a separate method is implemented for the clipped tow format. In Figure B.7a, n interpenetrating node **a** is chosen from one surface (tow **t**). The starting node is arbitrary. Then, every interpenetrating node from the



a) Node *a* finding the closest node (*b*)

b) Node *a* and node *b* move to halfway point

c) Nodes after being moved to halfway point

Figure B.7: Stages of clipped tow interpenetration resolution

opposing tow (tow **u**) is iterated over, recording the closest node (node **b**) to node **a**. A vector is created that runs from **a** to **b**. This vector gives both the direction and distance that node **b** is from node **a**. This vector is used to determine where the two nodes will be moved to so that they coincide. Using the vector that connects **a** and **b**, we calculate the distance between the two nodes and divide it by two. This gives the halfway distance between the two nodes. To ensure that we do not change the topology of the surface drastically, we move both nodes to the halfway point between the two (node **a**,**b** in Figure B.7b). This point lies on the vector that connects **a** and **b** and is halfway between the two nodes. This accomplishes both the task of removing interpenetrations and enforcing compatibility for the two nodes. Once compatibility is enforced for **a** and **b**, they are removed from their respective interpenetrating node lists. This is to ensure that they are not detected and moved again. The method is used until one interpenetrating node list has had all of its nodes removed. Because the interpenetrating node list for the two tows are not usually the same size, the algorithm must handle the remaining nodes in the list that is larger than the other. If the remaining interpenetrating nodes are from tow **u**, the algorithm identifies the closest node (node **d**) on the opposing tow **t** to a node from tow **u**. The found node from tow **t** does not have to be interpenetrating since all interpenetrating nodes from tow **u** have already been resolved. The interpenetrating node **c** is then moved to node **d**. This is done for all the remaining interpenetrating nodes from tow **u**. The result is no interpenetrating nodes from either tow, as shown in Figure 9.7c.

The result is a non-planar contact region between the two tows. The merit of this method is that when it removes the interpenetrations it simultaneously enforces compatibility. It uses a point searching algorithm and simple vector calculation. The method is also unpolished as can be seen below where some nodes are moving when they are not required to, resulting in very jagged edges. One reason the nodes may be moving is that they have been improperly identified as

interpenetrating and would require improvement on how the interpenetrations are detected. There is no protection against collapsing elements as the nodes move to each other. Therefore, this method is not a full solution but rather the initial implementation of the method. Figure B.8 shows the result of this method on non-idealized tow surfaces. First, there are “spikes” along the edges of the tow where the method as mis-identified interpenetrating elements and have moved them to the surface of the bottom tow, and vice versa. Second, even if the node are mis-identified, the method has reversed the direction the points must move and had pushed into the opposing surface instead of moving it away from the surface. Multiple iterations of development work consistently revealed new problems with this method as other were fixed. This attempt at a solution was a significant factor in deciding that another method for both identifying and resolving interpenetrations was necessary.

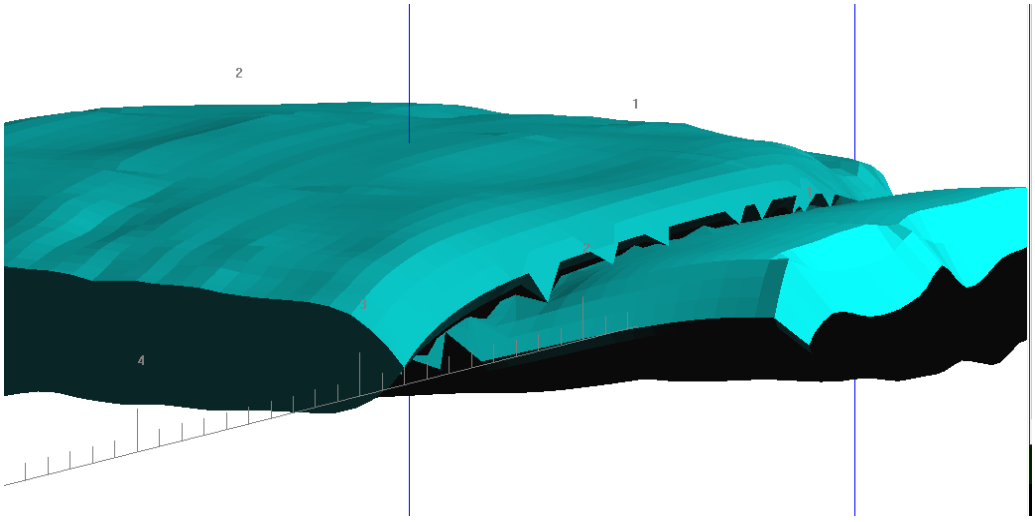


Figure B.8: Compatibility method result that shows several incorrect node movements

APPENDIX C

Prior to the method implemented in the main text, a similar but separate method was implemented that attempted to modify an exsiting surface mesh using the intersection curves produced by the SISL library. The intersection curves, which are also the boundaries for the interpenetration regions, are used to remove any surface mesh elements that lie within the perimeter of the intersection curves. Figure C.1 shows two scenarios between an intersection curve and surface elements. Elements marked with an **o** that lie entirely inside the curve are removed. Elements marked with an **x** are divided where the intersection curve intersects that element. These elements will need to be subdivided and remeshed so that the resulting submesh elements are entirely inside or outside the intersection curve.

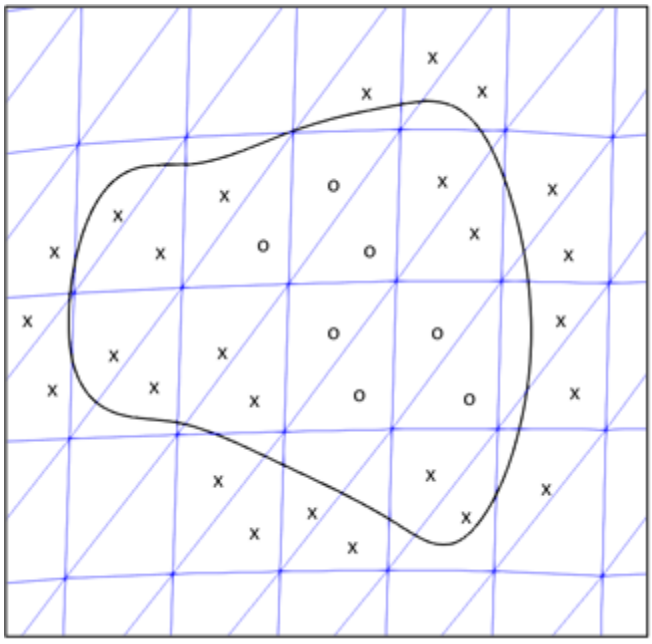


Figure C.1: Surface intersection curve on a tow surface

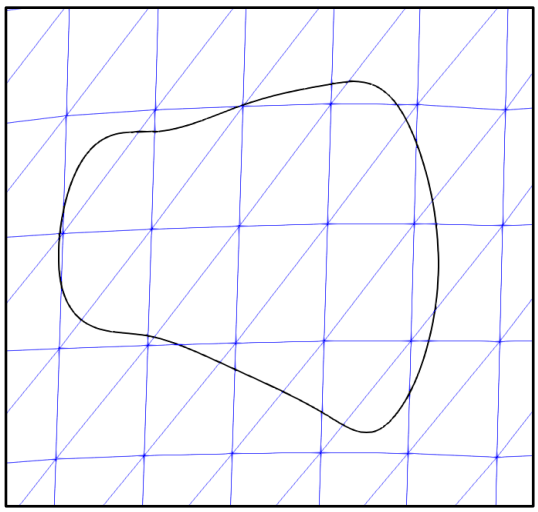
Figure C.1 is only considering one surface mesh. However, both surfaces need to have their interpenetrating elements removed using the same intersection curve. Removing elements from both surfaces using the same curve ensures that neither surface has elements inside of the

interpenetration region bounded by the intersection curve. A later section discusses the process of removing the elements from both surface meshes in a systematic manner. Once both surface meshes have their interpenetrating elements removed, a new mesh is created using the intersection curve as the boundary. The new mesh is inserted into both surfaces to replace the deleted interpenetrating elements.

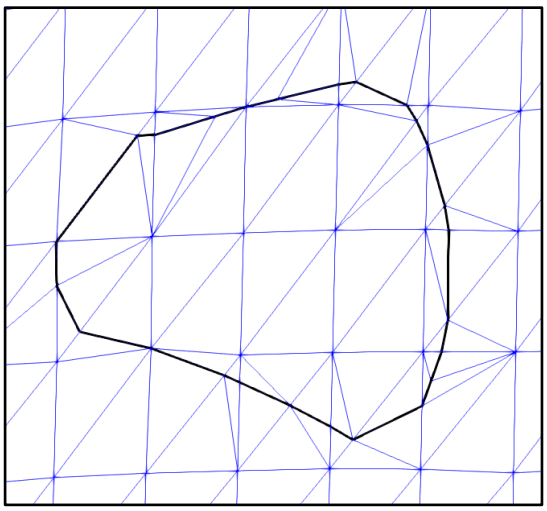
Now, the elements for both surfaces match perfectly inside of the intersection curve, resulting in a perfectly compatible mesh between the tows within the curve. The method used to remove the interpenetrating elements will be presented using an intersection curve and one surface. In reality, the method is performed on both surfaces to remove the interpenetrating elements on each surface. However, discussing the method using only a single surface first allows for a clear description of each step of the method.

C.1 Anticipated result of method

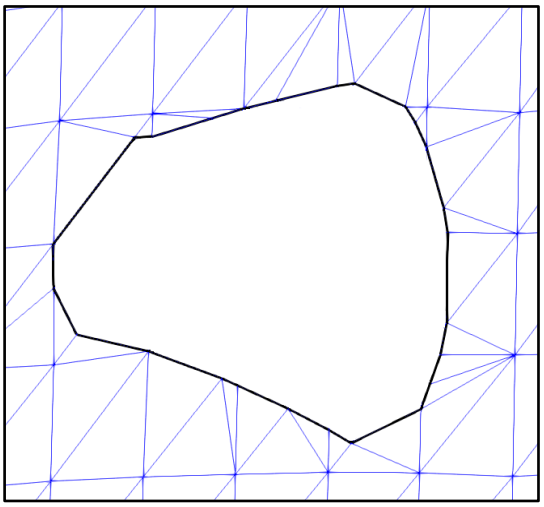
Figure C.2 shows an example of an intersection curve that is used to remove the interpenetrating elements. Figure C.2a show the initial linear approximation of the intersection curve overlaid on a section of the example surface mesh. Figure C.2b shows a further approximation of the intersection curve when only the intersection points between the curve and the surface elements are used to define the intersection curve. This curve does not represent the original curve very well, but a later discussion will discuss how the fit is expected to be improved once a second surface is included in the method. Figure C.2b also shows how the surface mesh is modified so that no surface elements lie partially in the area enclosed by the intersection curve. Elements that lie partially inside the intersection curve must be sub-divided into smaller elements so that the region inside of the intersection curve can be removed. Figure C.2c shows the surface



a) Initial intersection curve and surface mesh



b) Intersection curve with divided surface elements



c) Intersection curve with interpenetrating elements removed

Figure C.2: Boundary curve and element removal result

mesh after the interpenetrating elements have been removed. The curve must be represented in a form so that the interpenetrating elements from the surfaces can be removed. The chosen solution is to divide the surface elements where they are intersected by the intersection curve into smaller elements and remove the interpenetrating elements from the surface mesh. Once the curve is used to remove interpenetrating elements from both surface meshes, it becomes the common interface along which the two surfaces will be compatible.

C.2 Moving surface mesh nodes

The first step is to move any surface node that is near the intersection curve to a node on the curve itself. The surface nodes are moved to avoid the situation where the intersection curve contains a very small part of a surface element. Figure C.3 shows a section of the upper part of the curve in Figure C.1 that is very close to a surface node on the surface mesh. In Figure C.3 there are very small regions of two surface elements that lie within the intersection region. In order to remove this small region, the element must be divided into smaller elements such that the new elements lie either entirely inside or outside of the curve. The size of these new elements is very small compared to the original element size. Enforcing compatibility between these small elements and the surrounding surface mesh elements would require a large amount of work and could negatively affect any finite element analysis conducted using the mesh.

Instead of attempting to re-mesh the surface elements to capture the very small regions that lie inside of the intersection curve, any surface nodes that are within a certain distance (determined by the user) of the intersection curve are moved to the curve, as in Figure C.4. Because the relative refinement of the intersection curve is very high compared to the refinement of the number of surface elements it intersects (usually two orders of magnitude higher in the number of elements), a surface node is directly moved the closest intersection curve node. A k-d tree searching algorithm is implemented to find the nearest intersection curve node to a surface mesh node. The distance between the two points is calculated and compared against an established tolerance. A larger tolerance allows for surface nodes farther away from the boundary curve to be adjusted to lie on the curve. However, a larger tolerance will affect the surface mesh more than a small tolerance that does not allow the surface nodes to move as much. The result is a surface mesh whose surface

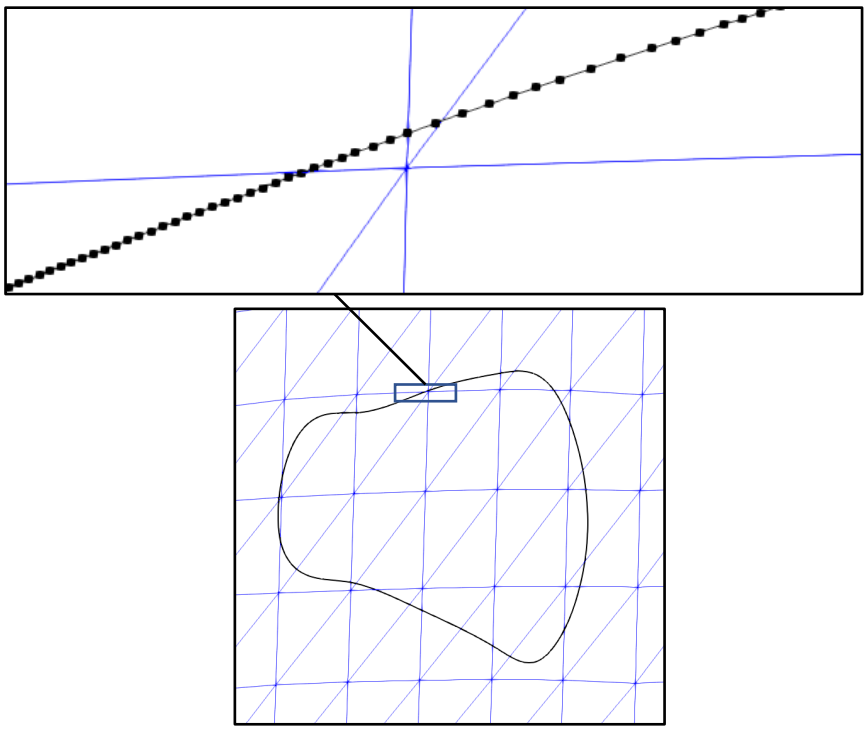


Figure C.4: Intersection curve in close proximity to a surface mesh node

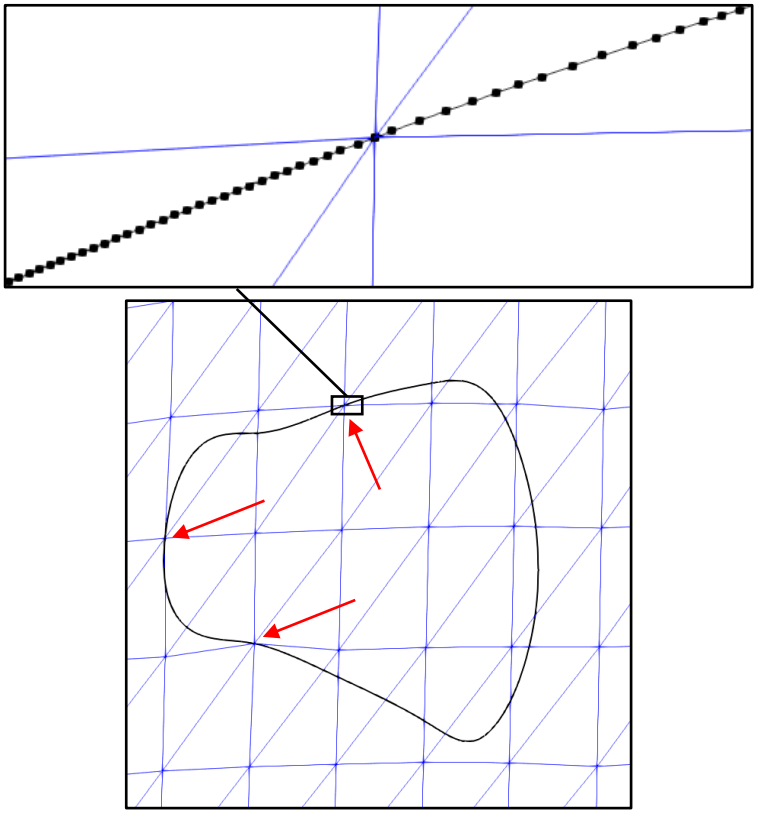


Figure C.3: Boundary curve with surface mesh nodes moved to the boundary curve

nodes that lie close to the intersection curve (Figure C.3) have been moved to the curve (Figure C.4). Moving surface nodes to the intersection curve eliminates the localized high refinement that is caused by a small corner of the element remaining is the intersection curve.

Once the surface nodes have been moved, the intersection points between the boundary curve and the surface mesh are calculated. The Separating Axis Theorem (SAT) is implemented to detect if an intersection curve segment intersects with a surface element. The SAT is discussed in detail in Appendix D. First, the purpose of calculating these intersection points will be discussed

C.3 The purpose and use of intersection points

The next two sections will discuss the purpose of identifying the intersection points between the boundary curve (surface intersection curve) and the surface mesh.

C.3.1 Compatibility between surfaces

The main purpose of dividing the intersection curves at each element edge intersection point is to establish a basis for compatibility between the two surface meshes. The algorithm calculates the intersection points for both tow surfaces whose interpenetrations are bounded by the curve. It is known that along this curve both surfaces have elements with edges that run directly through a

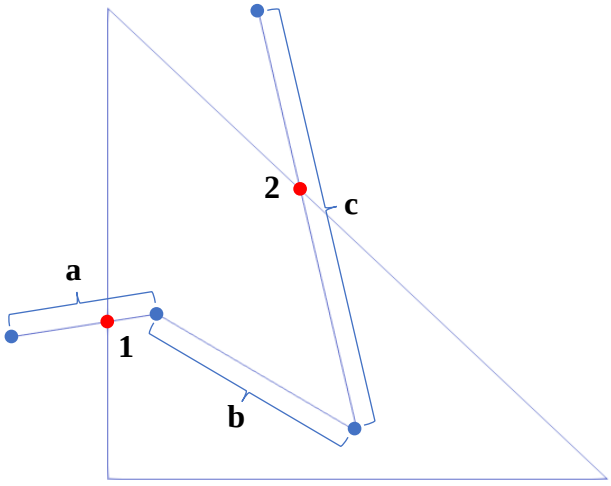


Figure C.5: Intersection curve with marked edge intersection points with a surface element

point on the curve. Therefore, elements on each surface intersected by the curve will have a set of points on the curve that their edges will line up with, which is required in traditional finite elements. However, the intersection points are calculated for only one surface in this section for simplicity. The method for calculating the intersection points is the same for both surfaces.

The intersection points are calculated by first finding which curve segments intersect the edge of an element. In Figure C.5, curve segments **a** and **c** intersect the triangular surface element's edges. Then, the segment is divided at the intersection points (points **1** and **2**), and the points are inserted into the curve. Every calculated element intersection point is added to the points that define the linearly approximated intersection curve. Now, the intersection curve is a base curve for compatibility between the tow surfaces. Once this compatibility is created along the intersection curve, the curve can be used to divide the surface mesh elements that the curve intersects.

C.3.2 Reducing the refinement of the boundary curve

The element intersection points also serve a secondary purpose. The surface mesh refinement around the intersection curve would be much higher than the existing surface refinement if all of the points in the intersection curve were used to re-mesh the intersected surface elements. The intersection points identify the path the boundary curve takes through each element. By connecting the intersection points with line elements, the curve refinement can be reduced. If the curve refinement is not reduced, the algorithm that re-meshes intersected surface elements will produce high-refinement meshes to replace the original surface elements. Large changes in the refinement of a mesh can have negative consequences on a finite element analysis. Therefore, the relative refinement of any surface element that requires a new mesh should match the same level of refinement of the surrounding elements as much as possible.

The high refinement is caused by the number of intersection curve segments that lie inside any element that is meshed. Reducing the refinement of the intersection curve will also reduce the relative refinement of the elements that are intersected by the intersection curve. Also, because the intersection points are calculated before the boundary curve refinement is reduced, there is no loss of accuracy when removing interpenetrating nodes and elements. An iterative loop is run to remove

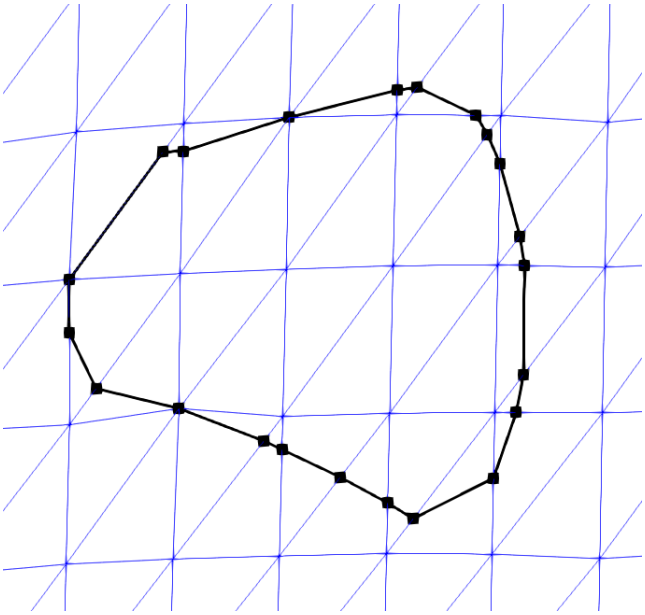


Figure C.6: Reduced refinement boundary curve with marked intersection points

any curve points between two consecutive intersection points to reduce the intersection curve's refinement. The result of the method can be seen in Figure C.6.

Figure C.6 shows that all of the surface nodes inside the perimeter of the original intersection curve from Figure C.1a are still in the interior of the curve. The shape of the curve is affected, but the majority of the interpenetration region remains. Figure C.6 is the result of capturing the interpenetration points of just one surface for illustrative purposes. However, once the intersection points from both curves have been added to the intersection curve, there will be intersection points that line within the elements in Figure C.6 as well, as shown in Figure C.5.

C.4 Detecting line segment and element edge intersections

Before the SAT can be used, the intersection curve segment and the surface element must lie in the same plane. An assumption is made that any segment that intersects a surface element is nearly planar with the element because the intersection curve lies on the intersection of both

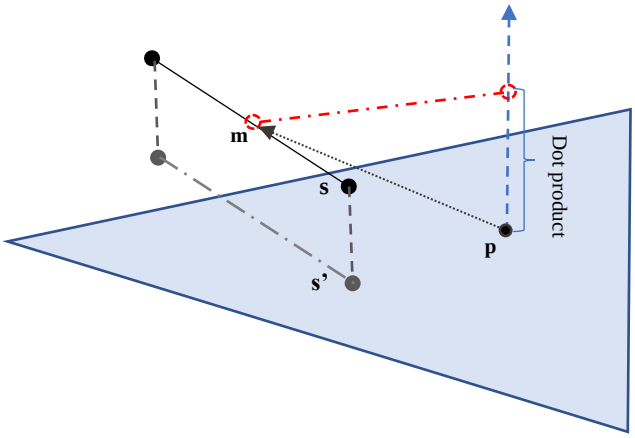


Figure C.7: Illustration of proximity check algorithm

surfaces. Therefore, if an intersection curve segment intersects an element edge or lies within the element edges, the intersection curve segment should lie on the same plane as the element. However, this is not guaranteed because of the approximation of the NURBS surface as a faceted surface mesh, which changes the intersection point of the meshes slightly. Therefore, when evaluating if part of the intersection curve intersects a surface element, the curve segment is verified to be in the proximity of the element and then projected onto the surface element. Figure C.7 shows an example of how a linear segment **s** is projected onto an element, resulting in segment **s'**. Intersection curve proximity is verified by creating a vector between a point (**p**) in the plane of the element and the mid-point (**m**) of the curve segment (**s**) being checked, as in Figure C.7. If the result is within a user defined tolerance, the segment midpoint is verified to be in or near the plane of the element. Another check is made to verify that the segment is within a certain vertical distance of the element surface by taking the dot product of the previously mentioned vectors and verifying

the result is within a set tolerance (**Dot product** in Figure C.7). The last check projects the segment (**s'**) onto the element plane and conducts the SAT. Projecting the segment onto the element plane is accomplished by determining the distance to the plane for each segment node and then moving the node along the element plane normal vector by the calculated distance to the plane of the surface element. The projection of the curve segment is shown in Figure C.7 as **s'**.

The SAT is used to determine if the segment, when projected onto the surface element plane, intersects or lies within the element. Once the intersected elements are identified, the points where the intersection curve segments intersect the edges of the surface element can be calculated. The result is clearly defined intersection points where the interpenetration boundary curve intersects surface element edges, shown in Figure C.8. During this process the elements that are intersected by the intersection curves are recorded for later use in the routines that divide and re-mesh the intersected elements.

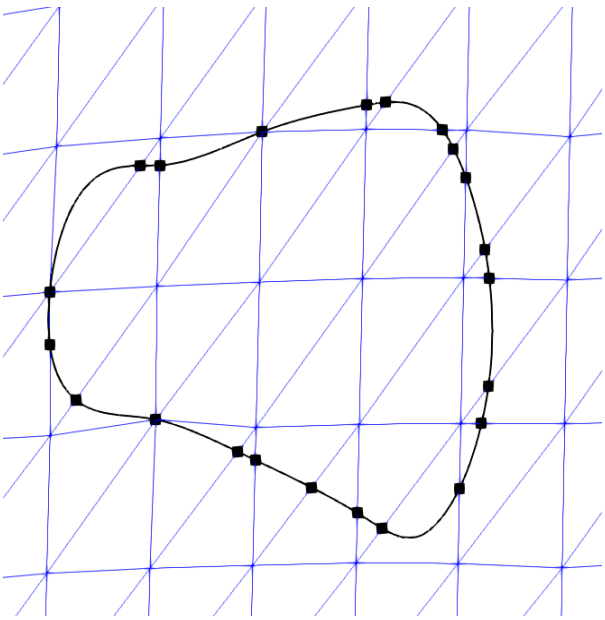


Figure C.8: Boundary curve with marked surface element intersection points

C.5 Division of surface mesh elements

After the intersection points between the surface elements and the intersection curve of the surfaces have been used to reduce the curve's refinement, the surface meshes element are divided where the curve intersects the element. The interpenetrating elements that lie partially inside the intersection curve are divided along the curve. The two parts of the original element are re-meshed using any intersection points that lie in the element or on the edges of the element. The new tow surface mesh created by individually adding each intersected element's divided mesh back into the original surface mesh. The elements intersected by the intersection curve are iterated over and divided individually. The same intersection algorithm involving the Separating Axis Theorem is used to collect intersection curve segments that lie within the current surface element being divided. After the segments have been collected, they are checked against the edges of the element to verify which intersection curve segments have endpoints on the element's edge. Figure 8 shows only one segment per element, but once the intersection points from both surfaces are added to the intersection curve, there could be more than one segment per element. Figure C.9 illustrates this point with an example element with multiple intersection curve segments. Figure C.9 shows two intersection curve segments with endpoints lying on the surface element edges and one curve

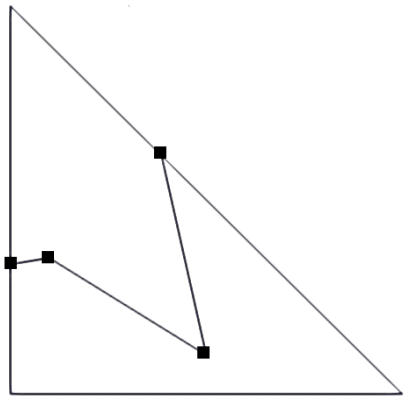


Figure C.9: Example surface element with three boundary curve segments intersecting

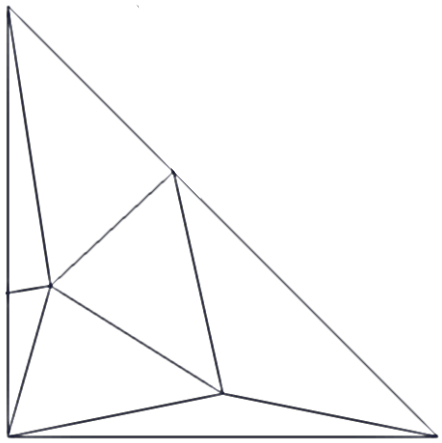


Figure C.10: Sub-mesh of surface element with boundary curve

segment completely contained in the element. A mesh generation library called Triangle is used to create a new mesh of the surface element in Figure C.9 that includes the intersection curve segments. The library requires that all line segments, referred to as boundary segments by the library, that define the required boundaries be included in the list of segments be given to the library. These segments include all surface element edges and the intersection curve segments that lie within the element. Once the curve and element segments are given to the library, a mesh is returned, as shown in Figure C.10. The newly meshed element is then be added back into the surface mesh, replacing the original element. Figure C.11 shows the result of the algorithm for the test case shown. Figure C.11 shows both the relative refinement of the newly meshed elements as well as the reduced refinement of the intersection curve. The refinement is comparable between the sub-meshed surface elements and the untouched elements.

Figure C.12 shows the original boundary curve compared to the new curve used to re-mesh the surface. Figure C.12 also shows how the overall shape of the intersection curve does not fit the original intersection curve well. Although no surface nodes or elements are left out from the reduced refinement intersection curve, a better fitment of the original curve data is desired. Some sections of the curve are not captured when the curve has been coarsened, but it is expected that

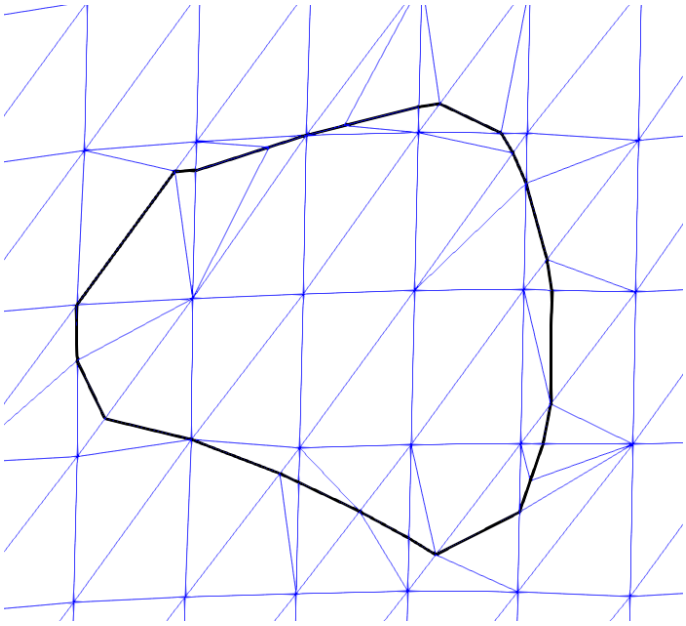


Figure C.11: Reduced refinement intersection curve and re-meshed surface elements

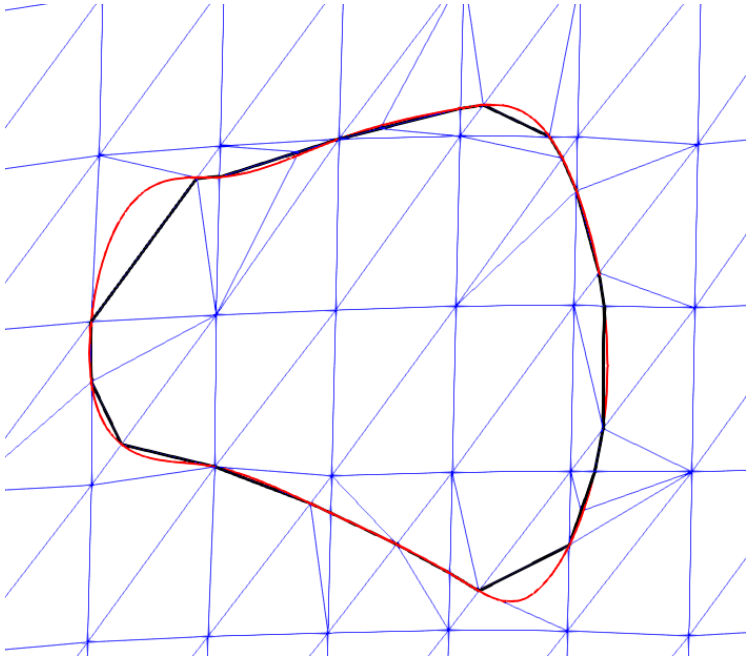


Figure C.12: Original (red) vs reduced refinement boundary curve with new surface mesh

the linear approximation will improve in accuracy to the original curve with the inclusion of the intersection points from the other tow surface.

The result is a mesh that includes the curve along which the surface mesh will be cut to remove the interpenetrating region. When the curve has been used to remove interpenetrating elements from both surfaces, the surfaces will then share a curve along which there is compatibility. The compatible intersection curve is the most important feature of the methods developed during this research. Previously, there has not been a method that will ensure a compatible region between any two tow surfaces. Now, the intersection curve between the two tow surfaces is also where the two surfaces are connected in a compatible manner. Using the intersection curve, a compatible mesh can be created in the interior of the intersection curve that will be used to replace both surfaces elements that have been removed. The result is a compatible mesh that both surfaces share, creating a connecting surface between the tow meshes.

C.6 Issues and Drawbacks of the Projection Method

The biggest issue with the projection method and how it was implemented is the extensive use of tolerances. Some of these tolerances are co-dependent. This means that adjusting one affects the effectiveness of another tolerance and results in failure of the method. Failure occurs specifically with the tolerance for moving a surface node to a preexisting node on an intersection curve, and the tolerance for creating a new point on an intersection curve where a surface element edge intersects the intersection curve.

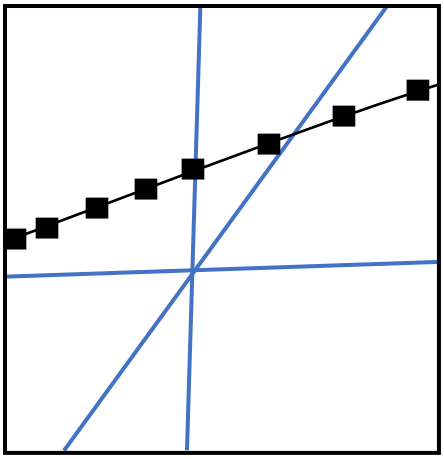


Figure C.13: Intersection curve in close proximity to surface mesh node (intersection of non-marked lines)

Figure C.13 shows an example where these tolerances would all potentially come into play. First, there exists a surface node (intersection of surface element edges) that is close to an intersection curve (line with markers). If the tolerance for moving surface nodes to intersection curves is large, then the surface node will move to the nearest intersection curve node. However, there is a limit to this tolerance as too large will move the surface node locations too much, and too small will result in tiny surface elements, compared to the original size. The smaller elements present issues during a future analysis so these should be avoided as well. In the case that a surface node is not moved, but still lies close to an intersection curve, the intersection points between the

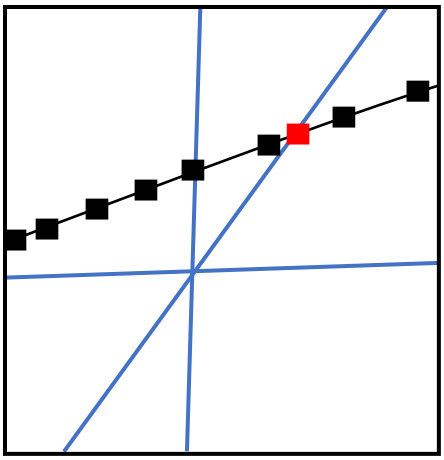


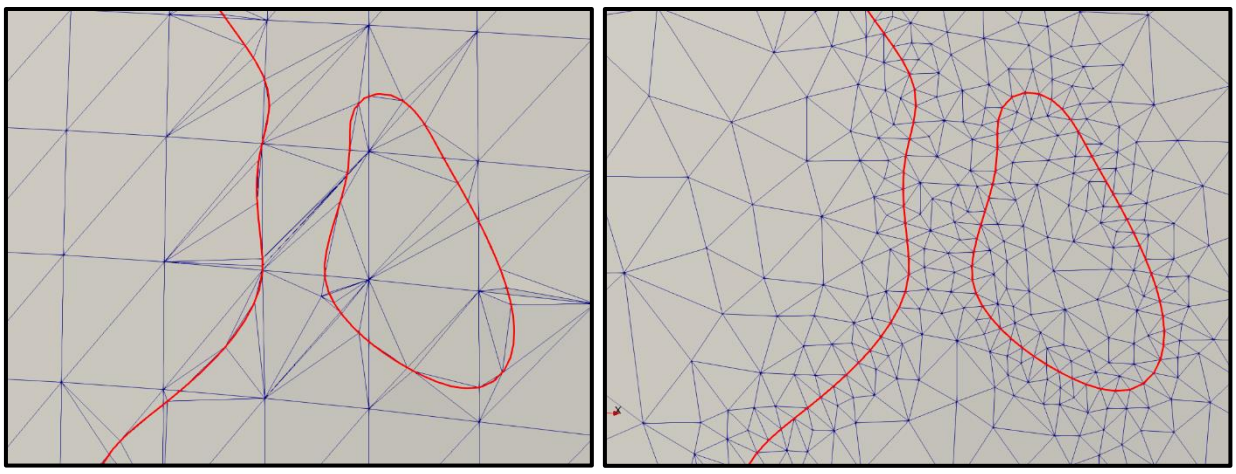
Figure C.14: Intersection curve in close proximity to surface mesh node (intersection of non-marked lines)

edges of the surface element and the intersection curve are found (red marker, Figure C.14). There is a second tolerance that determines if an existing intersection curve point exists close to this new point. If so, the intersection curve point moves to the new point. If not close enough (tolerance small), the new point is inserted into the intersection curve. Again, an issue arises if the chosen tolerance is too low, the resulting meshes will have tiny surface elements that may introduce stress artifacts during an analysis. If the tolerance is too large, multiple points will move to this intersection point, and refinement of the intersection curve can be lost. The meshing algorithm used later depends on using properly sized intersection curve segment lengths. If the tolerance is small enough, the meshing algorithm will assume there is no element to mesh in the region, and gaps will form in the surface, resulting in a failed algorithm. Using too many tolerances just for determining surface node placement results in a delicate and difficult-to-conFigure method that is not user friendly.

Using three-dimensional curves to remove interpenetrations requires the surface to also be in three dimensions. This requirement leads to difficulty in determining tow surface elements that were in proximity to the intersection curve but not contained by the curve. The difficulty is due to the curve enclosing an area instead of a volume; and identifying elements becomes tedious because the curve is not planar. Therefore, the curve has to be projected onto the surface element being tested for containment (or intersection). For a curve that is not relatively flat, it becomes nearly impossible to avoid false positive surface element detections as any elements that lie within the three-dimensional bounds of the curve are treated as “contained”, per the current implemented method. The solution to this problem is either an extensive and complex identification algorithm or a methodical way to include the intersection curves during the mesh creation step. Triangle, a two-dimensional mesh generating library, fulfills this need as it requires bounding segments that

define the geometry to be meshed. The intersection curves have already been approximated as linear segments which can be directly translated to boundary segments. The remaining issue is converting all representations into two-dimensions, a requirement of Triangle.

A second issue with the previous method is the number of computational tolerances required during the algorithm. These tolerances add error to the surface representations themselves and the intersection curves. These small errors can compound and cause the algorithm to fail, resulting in small gaps in the surface representations after the surface elements have been modified. Tightening these tolerances only result in failures elsewhere in the method. This results in a fragile and difficult to use method. Figure C.15a shows an example of when these tolerances are adjusted perfectly and a mesh can be created. It is shown that even if the method does succeed, the mesh is not ideal and contains many high-aspect ratio elements that are long and thin. Their refinement is significantly different that the surrounding elements. Compare Figure C.15a to C.15b which is the result of the method presented in the main text. The elements in Figure C.15b have a much better aspect ratio and although the are smaller that the elements away from the intersection curve,



a) **Result of projecting intersection curves on pre-existing surface meshes**

b) **Result of generating surface meshes with the inclusion of linearly approximated intersection curves**

Figure C.15: A comparison of surface meshes from using each method

the change is relatively gradual. Overall, the results in Figure C.15b are much more favorable and much more reliable in its implementation.

APPENDIX D SEPARATING AXIS THEOREM

The Separating Axis Theorem (SAT) [40] is implemented to identify which elements are intersected by a intersection curve and to refine the boundary curve where it intersects surface elements. The SAT starts by projecting a shape onto pre-determined axes. The projection can be thought of as the shadow of the shape on an axis. The axes are created by taking the normal direction of an shapes edge and creating an imaginary infinite line in the same direction. Figure D.1 shows a projection of a triangle onto the red axis (RA) that is determined by the red edge (RE) and is parallel to the red edge normal (REN). The line segment along RA (labeled "Triangle Projection") is the projection of the triangle onto the axis RA in Figure D.1. Three axes are identified corresponding to the normal of each side of the triangle (REN axis, GEN axis, and BEN axis).

Next, the projections are tested to see if they overlap. If there is any axis on which the projections do not overlap, then the polygons do not intersect. If the projections overlap on every axis, then the polygons do intersect. A reference picture is shown in Figure D.2.

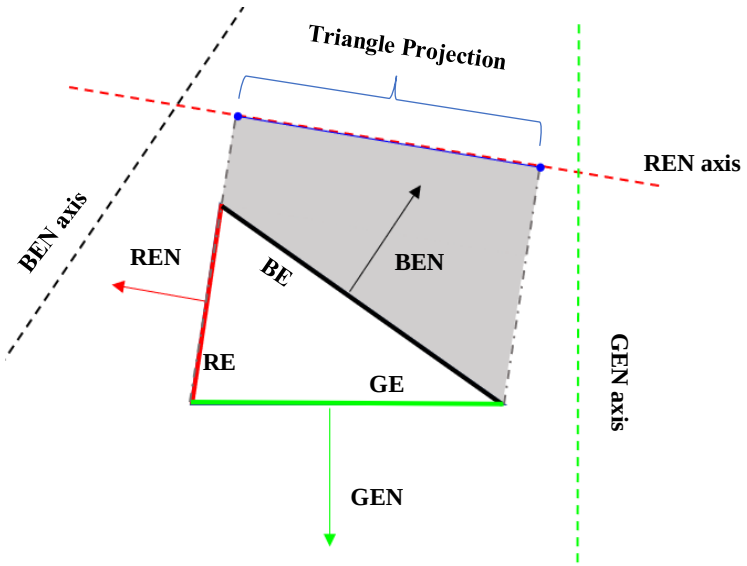
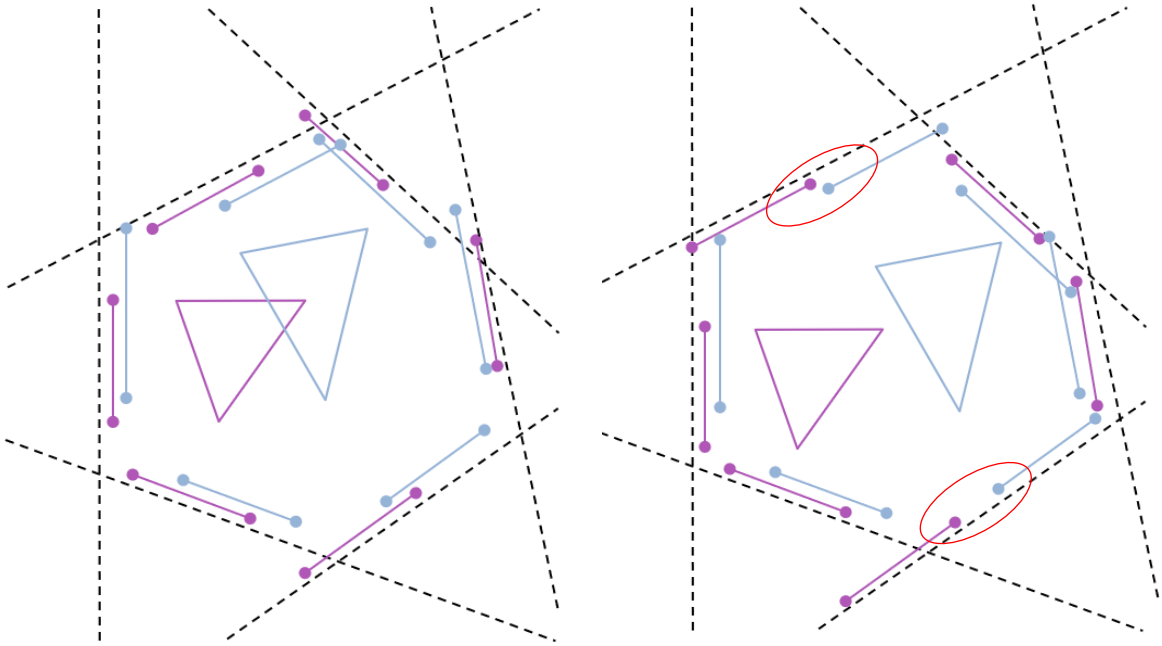


Figure D.1: Projection of a triangle on an axis

In Figure D.2a, two triangles are shown that intersect. This can be verified by looking at each dotted line that represents a projection axis. Along each axis the bounds of the triangles are shown. There is no axis in which the bounds do not overlap. Figure D.2b shows the case when the two shapes do not intersect. The axes (dotted lines) are the same in Figures D.2a and D.2b, since the orientations of the two triangles are the same, only the positioning is different. Circled are shape bounds that do not overlap in Figure D.2b and therefore verify that the triangles do not intersect. This method is adapted so that the second shape is simply a line segment from a curve.



a) SAT in which triangles **b) SAT in which triangles do not**

Figure D.2: Two cases for testing the Separating Axis Theorem