Detection and Resolution of Interpenetrations of Woven Tows

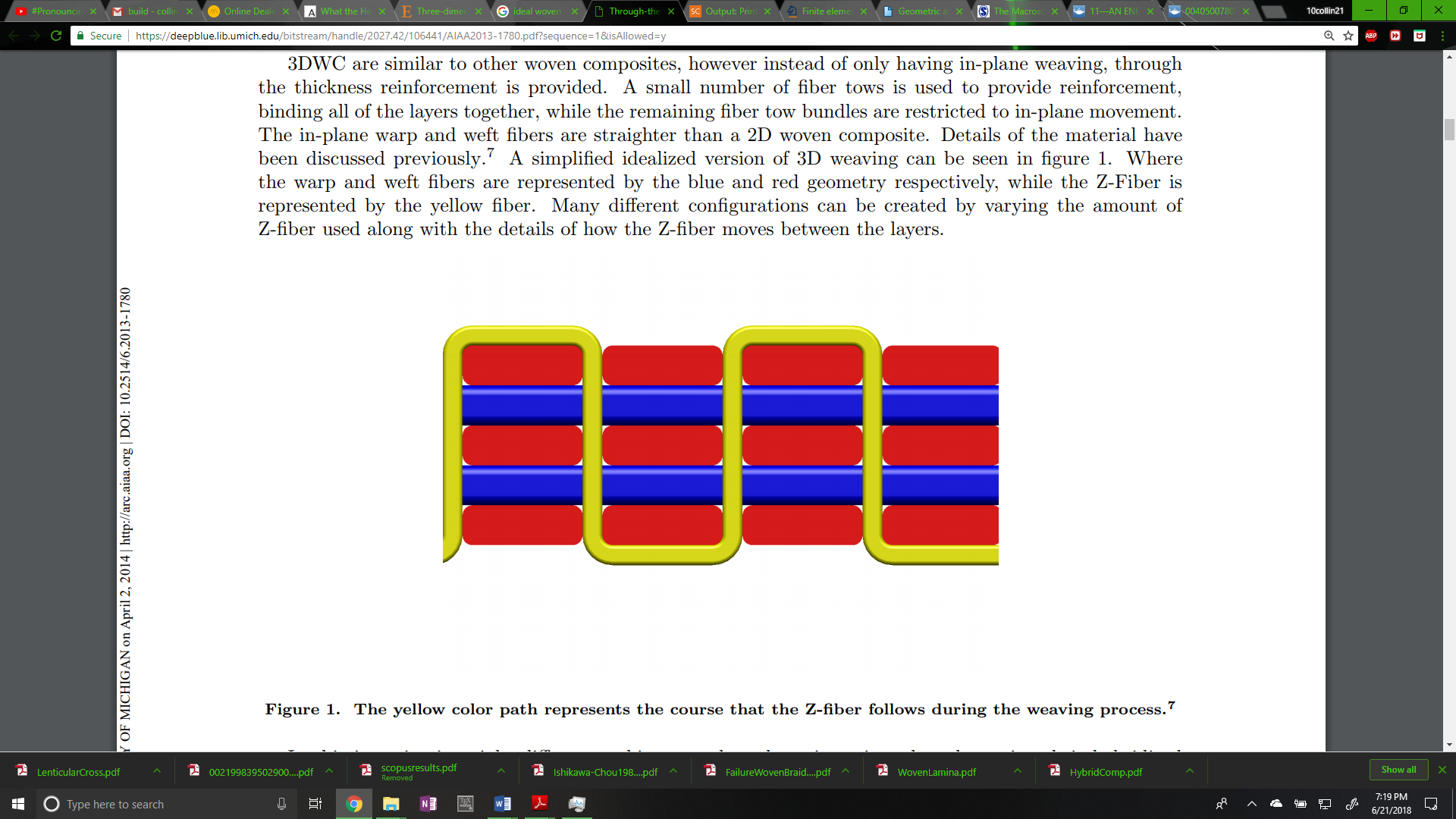
Collin Blake

# Introduction

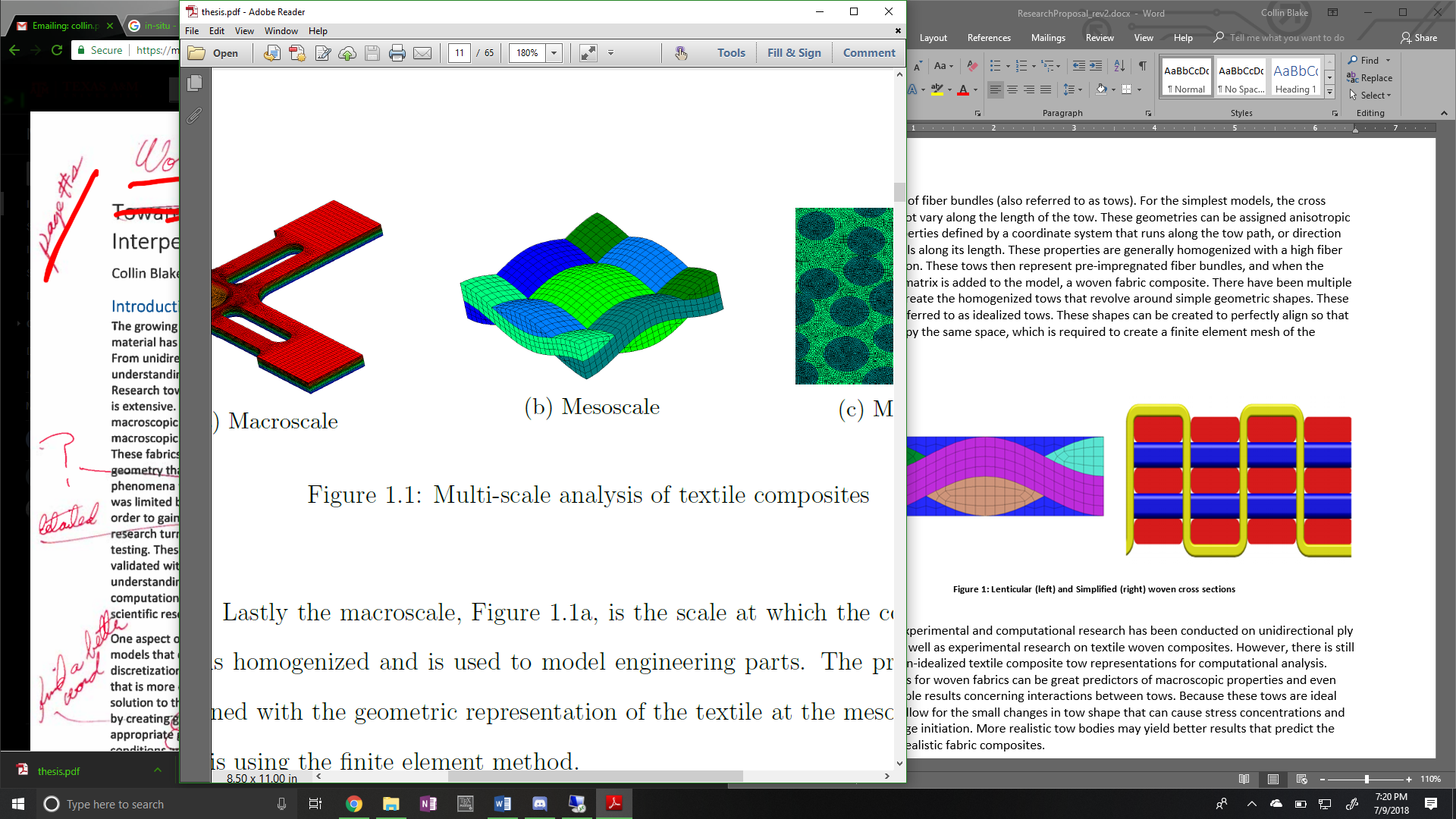
The growing use of fiber-matrix composite materials as both a functional and non-functional material has also increased the need to better understand these materials in all of their forms. From unidirectional ply composites to intricate woven textile composites, the need for understanding material properties and mechanics for these composites has never been higher. Experimental research towards mechanical properties of these geometries in a physical testing environment is extensive. [1][2][3] Many of these tests focused on identifying the different macroscopic properties and responses of composites. Further complexity was added with the introduction of woven fiber composites. These fabrics brought new challenges in the form of phenomena caused by tow interactions that had previously not been explored. During physical testing, equipment was used to measure various aspects, such as strain fields, energy dissipation, etc., of the materials. However, only a limited amount of insight was provided by these tests because of the technological limitations. To gain a more detailed understanding of the mechanical response for these composites, research turned towards computational modeling of woven fabrics. Equations such as Hooke’s Law and other theories were used in computational models and simulations that were validated with experimental data. [4][5] A well-known example is finite element analysis. As these models became more accurate, understanding of both mechanical response and damage initiation increased. The use of computational models and analysis is now a fundamental aspect of most engineering studies and scientific research.

One aspect of computational research for composite analysis is the use of finite element models that can simulate material responses to mechanical loads. Finite element analysis is the discretization of a large, complex problem into smaller, simpler pieces. The result is a problem that is more easily solved mathematically. This method 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 then given certain boundary conditions and a result can be computed. These results can be insightful to stress concentrations, deformation responses, energy absorption and other attributes that may be of interest. These results are, among other things, 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 can be achieved concerning deformation of the body.

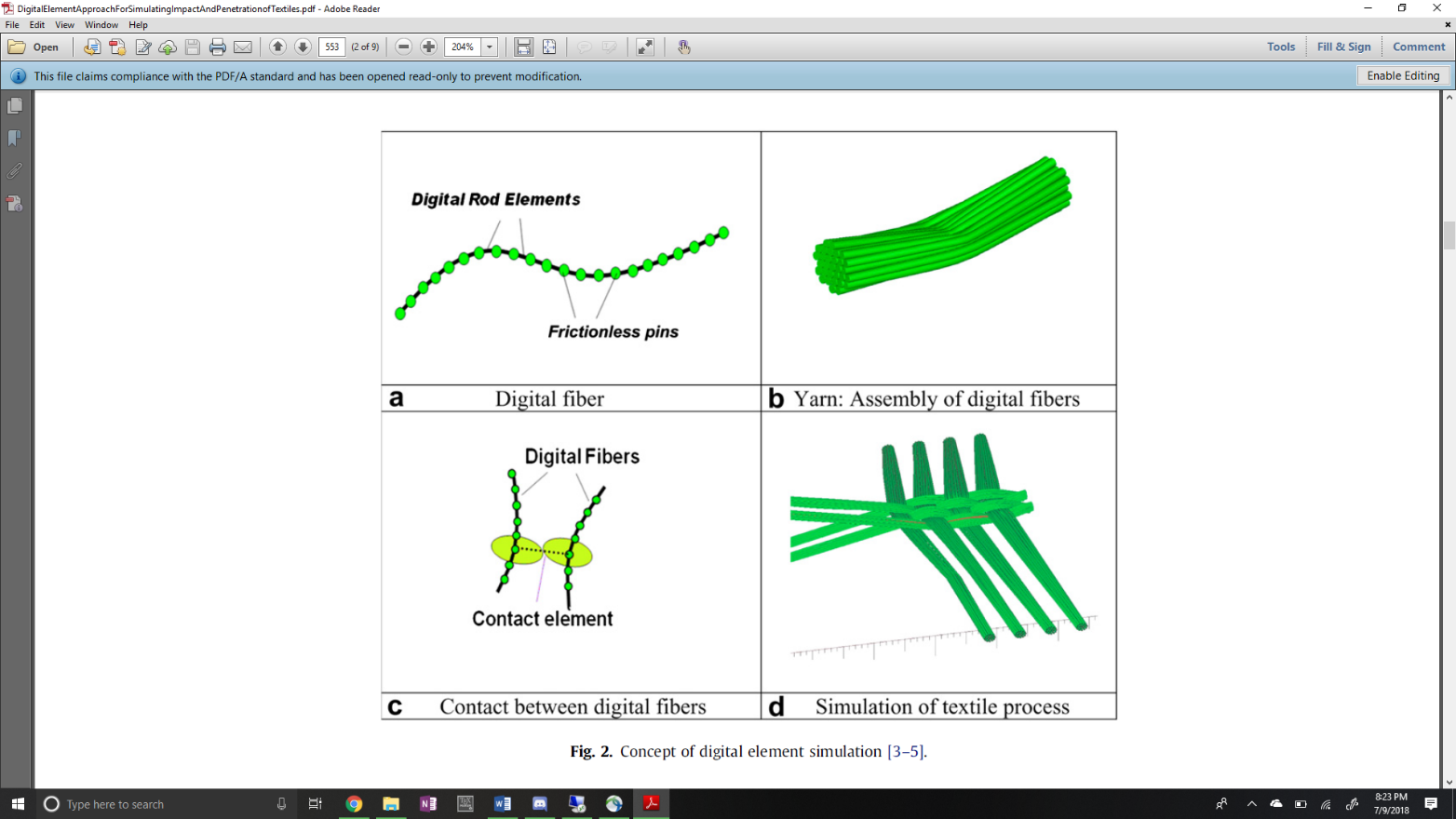
At first, modeling of textile composites using finite element made simplified geometric models in place of the more complex body. These models consisted rounded rectangles and lenticular cross-sections (Figure 1) to define the cross-section of fiber bundles (also referred to as tows). For the simplest models, the cross sections do not vary along the length of the tow. These geometries can be assigned anisotropic material properties defined by a coordinate system that runs along the tow path, which is defined by the centroid of individual cross-sections taken along the length of the tow. These properties are generally homogenized with a high fiber volume fraction. These tows then represent pre-impregnated fiber bundles, and when the surrounding matrix is added to the model, a woven fabric composite. There have been multiple methods to create the homogenized tows that revolve around simple geometric shapes. These are usually referred to as idealized tows. These shapes can be created to perfectly align so that no tows occupy the same space, which is required to create a finite element mesh of the shapes.



**Figure 1: Lenticular (left) and Simplified (right) woven cross sections**

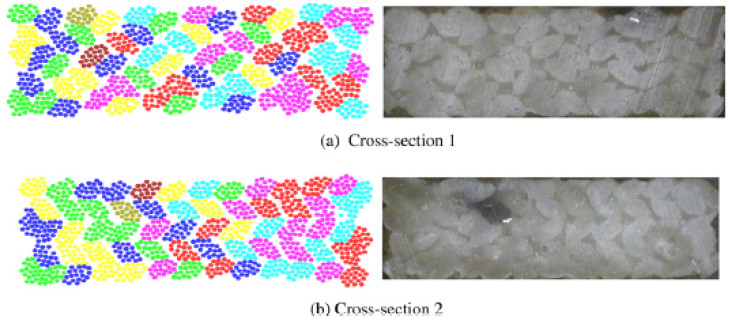


Substantial experimental and computational research has been conducted on unidirectional ply composite as well as experimental research on textile woven composites. However, there is still a need for non-idealized textile composite tow representations for computational analysis. Idealized tows for woven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



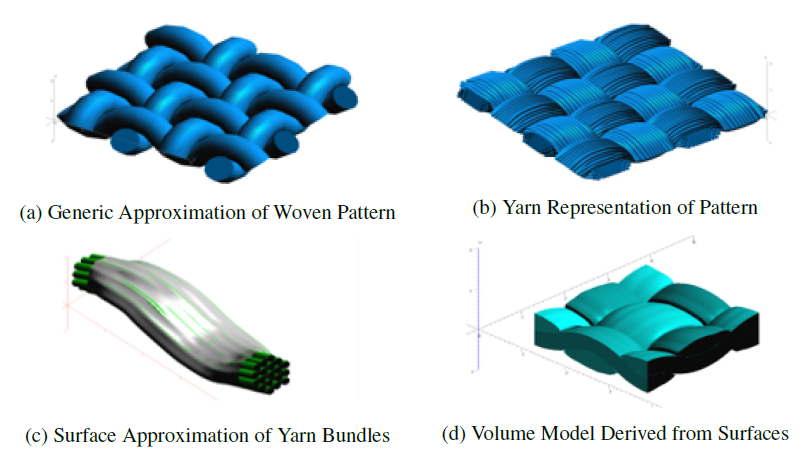
**Figure 2: Digital chain simulation process [6]**

One method for creating more realistic woven geometries is to simulate the process that manufacturers employ to create the fabrics. [6] The process begins by simulating bundles of fibers as digital chains. These digital chains are spheres that act as frictionless pins (no transfer of moment) connected by uniaxial rods. (Figure 2.a) The more spheres along the length of the digital chain, the closer its simulated response will be to that of an actual fiber. [7] 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 2.c) resulting in realistic interactions between the digital chains. This ensures a tightly woven textile like those seen in experiments. The pre-stress given to the chains can result in contraction of the digital chains that reduces the length of the uniaxial connecting rods. This can result in spheres of the same chain becoming too close together. If the spheres are too close, the physics of the simulation will be affected. If the chain is corrected during the process, the simulation works as intended. The result is fiber bundle cross sections that are similar to micro-CT scans from actual woven specimens, shown in Figure 3, from [7].



**Figure 3: Simulated vs. Actual Fiber Bundle Cross Sections**

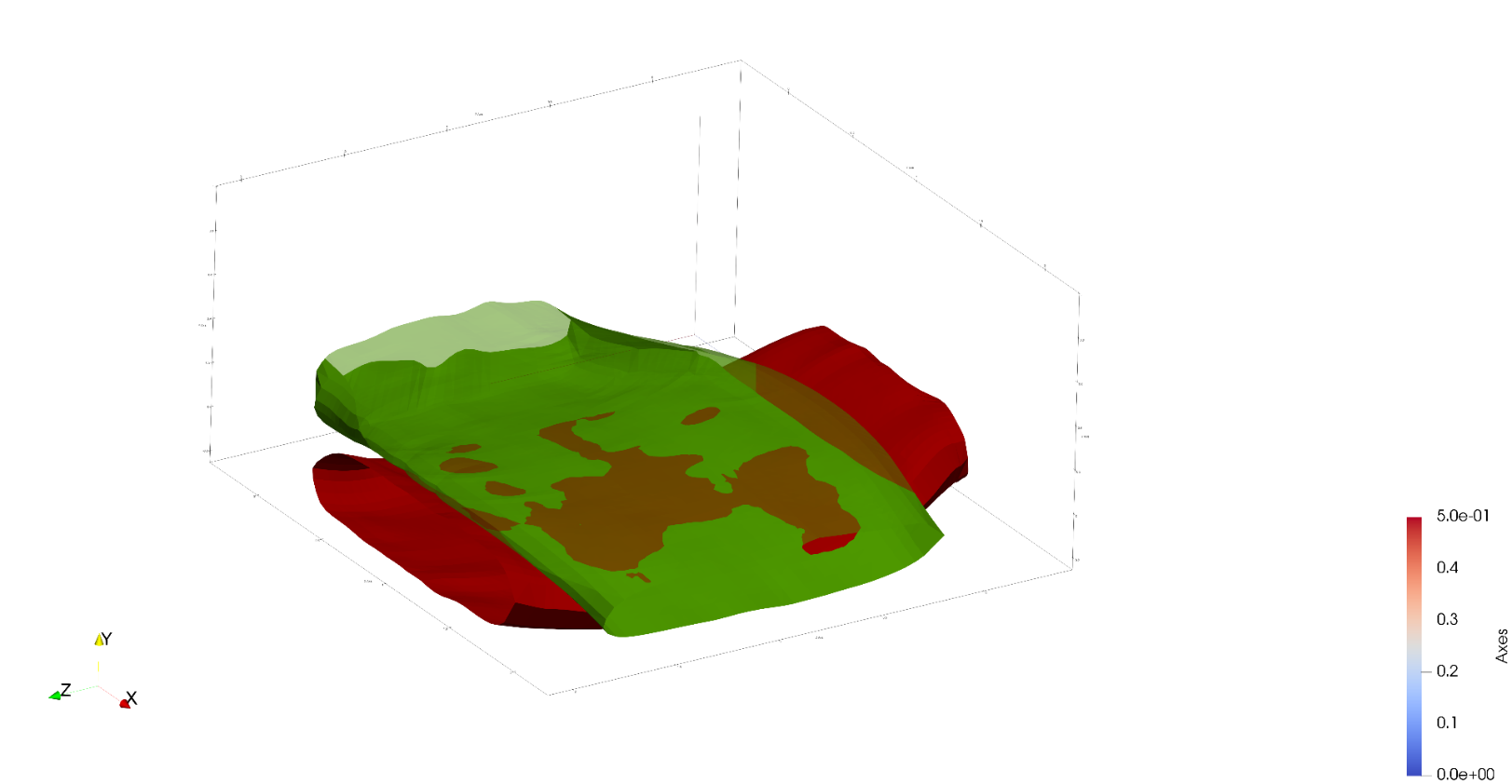
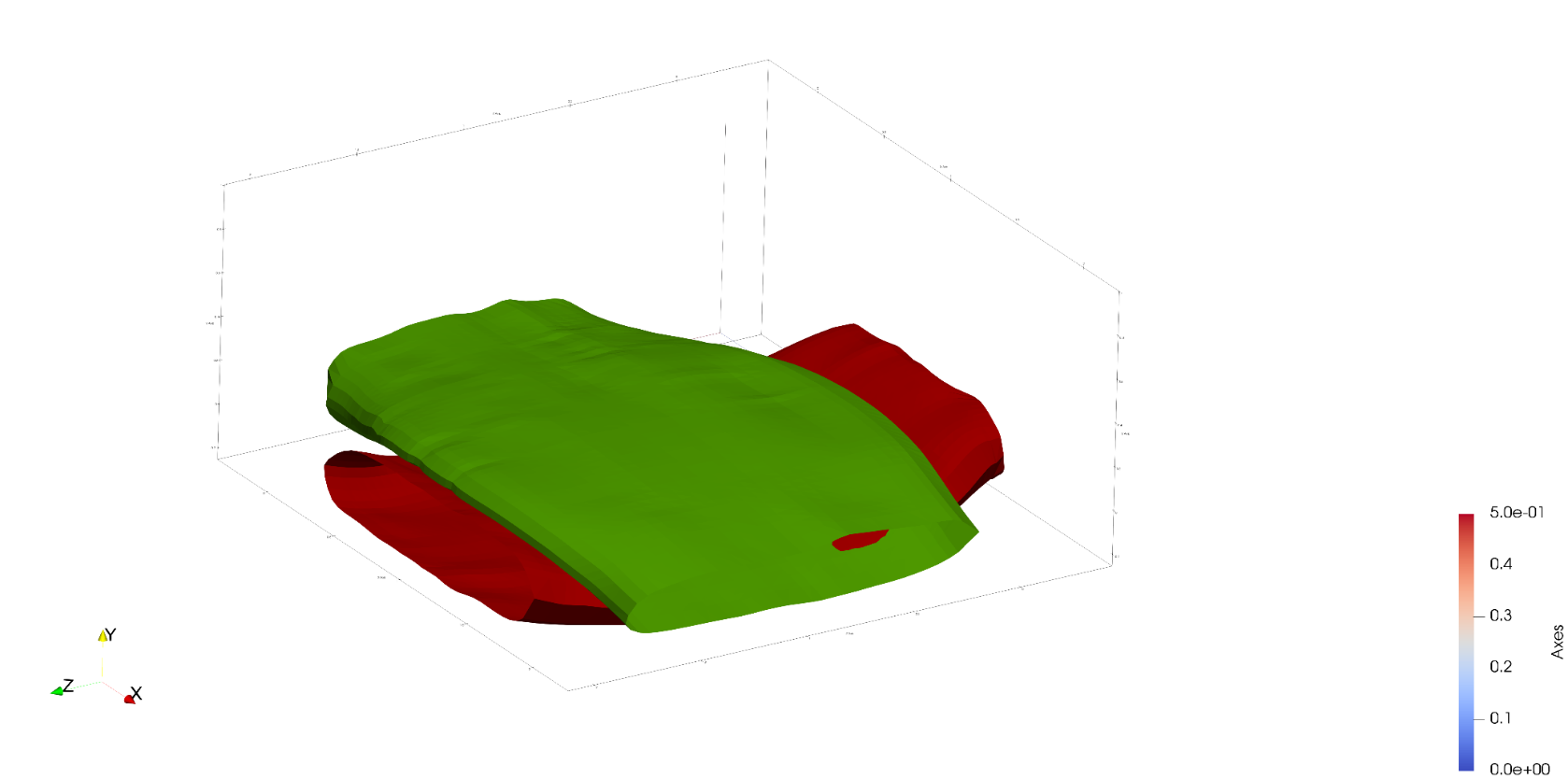
While there is possibly other software that use a similar method to the digital chain method, only two were explored. The digital chain method was introduced by Youqi Wong, Eric Zhou, and other researchers from Kansas State University. They developed the first iteration of the Digital Fabric Mechanics Analyzer (DFMA) which is still being improve 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 base geometry and surface mesh that is used in this study originates. Figure 4 shows the process visually. The surface inter-penetrations come as a result from the geometries shown in Figure 4.d.



**Figure 4: Evolution of Weave Textile Geometry**

The reasoning behind creating surface and volume approximations (Figure 4.c and d) is that the computational cost of analyzing many bundles that represent woven fibers is very high. Researchers instead can use volumetric tow approximations and apply homogenized material properties to conduct an analysis of a woven textile model.

**Figure 5: Region of close tow geometries with interpenetrations**



1. **Tows in close proximity**
2. **Transparent upper tow showing interpenetrations**

Once a surface representation is created, surfaces in close proximity have the ability to penetrate into each other, as shown in Figure 5. Physically, the two surfaces would come into contact and create a contact region in the form of a shared surface. Interpenetrations for the surfaces from VTMS form because the contact is modeled with the digital chain method, which is an approximation of the true physical contact. Digital element chains can still interpenetrate because they are spheres chained together rather than a full, volumetric fiber representation. These digital chains are then further approximated by creating a surface that wraps the digital chains. The surface is then smoothened which further approximates the fiber bundles. This chaining of approximations leads to errors in the form of interpenetrations between two tow surfaces. Traditional finite element software requires that two geometries cannot occupy the same space and must have compatible meshes along any boundaries that they may share. These regions must be fixed if a traditional FEA is to be conducted.

# Literature Review

The idea of solving the penetration (also know as intersection) issue is one documented well in computer aided modelling. Various approaches have been used to detect whether two shapes, in both two and three dimensions, occupy the same space at any given point. Jimenez, Thomas, and Torras [8] asserted that intersection scenarios can be static and time dependent. For both cases, a static intersection step must be calculated. For surfaces that are polyhedral (defined as having multiple flat faces and can be open or closed), Dobkin and Kirkpatrick [9] state that a hierarchical representation of the polyhedral can be used. This representation 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 polyhedra. During 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.

Canny [10] discusses in his book a method for the more general case of a polyhedral with convex faces. He states that two intersection cases exist for polyhedrals, face-to-node contact (Type-A) and edge-edge (Type-B). By associating a predicate that is true or false for each case, a series of tests can be run on the two polyhedrals. If either predicate 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.

The most general case for a polyhedral shape is a non-convex shape. For this case there are two trains of thought. The most popular response is to subdivide the domain into convex sub-domains. Two popular methods are decomposition into smaller convex polyhedral [11] and decomposition of only the surface into convex surfaces [12]. After sub-division, the smaller, convex shapes can then use a multitude of intersection algorithms that apply to convex shapes. The main drawback to 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 to calculating the intersection. This usually involves a two-step process to identify edge-face intersections [13] 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 a set of linear equations. Suppose there exists a linear equation that determines where a surface node lies in space. These equations can be set up so that they calculate the location of certain polyhedron surface nodes. The equations are set up to quantitatively calculate the predicates mentioned previously [10] [14]. They can be assembled in matrix form and by calculating the sign of certain determinants, it can be determined if a certain case 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.

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 polyhedral surfaces into non-polyhedral descriptions of these surfaces. With these surfaces there are also methods to detect intersections between surfaces of similar description type. Two common types of non-polygon surfaces are implicit surfaces and parametric 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 [15]. The most frequently used algebraic surfaces are quadric, which are second degree polynomials in *x*, *y*, and *z*. The other typically used non-polygon surface is a parametric surface. These surfaces in three dimensions are described by functions that have two input parameters. As a result, they are generally not closed but easier to polygonalize and render. A subset of parametric surfaces, Non-Uniform Rational B-Spline (NURBS), have gained traction in computer aided design software [16] and possess some ideal properties that make them easier to use. For each of these non-polygon surfaces, there are algorithms for detecting intersections.

For implicit surfaces, the available algorithms are limited. Pentland and Williams [17] 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 interpenetration region depth, area, and shape. This is very valuable when the shapes are static and simply detecting intersections is not enough. 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 [18] 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.

Parametric surfaces have a larger set of algorithms for intersection detection. 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 by subdividing both surfaces at each computational step. 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 [19]. A method very similar to this is used by Drach et al [20] to determine if surface nodes interpenetrate a surface. They then use another technique to remove interpenetrating surface nodes until they are all corrected. The main drawbacks to this approach is that the desired level of refinement of the intersection region negatively affects the computation time. As the level of refinement increases, so does the computation time.

A second method used is tracing. This method starts by first finding a known point of intersection, of which there are multiple methods to choose from [21] [22] [23]. Then, the intersection curve is traced along by starting at the previously calculated point of intersection and a 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 distances along this vector is predetermined and is the determining factor in the amount of “refinement” the curve has. One issue the method faces is determining if a curve has reached its starting position. This is usually posed as a system of algebraic equations [24] or a differential equation problem [15]. This method can yield very good results when trying to identify a boundary curve for the interpenetration regions.

A third method is the analytic method. Generally, one parametric surface is converted into an implicit representation of the surface [25] and creates a scalar function in the two parametric variables. The root locus of these functions in the parametric variable plane is the preimages of the intersection curve [15] [26]. 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 where and how the two surfaces intersect. This method can be difficult to implement as it requires knowledge of how to accomplish the parametric-implicit conversion as well as the framework for multiplying polynomials and solving multi-basis functions.

Drach et al [20] have used a couple 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 polygon 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 in the mean normal direction of all the interpenetrating surface elements inside the host. When they detect no more interpenetrations they consider them fixed. In paper published shortly after [27], 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 are not present in actual CT. This can cause minor yet important inaccuracies when observing the interaction between tows. It is the goal of this research to further reduce these inaccuracies.

There are a number of possible methods for detecting interpenetrations between polyhedral surface representations. Many use the same polygon representation that is similar to the standard output from VTMS while others depend on mathematical (parametric and implicit) representations. Using a method that uses the polygon form will be quicker but less accurate than its mathematical counterparts. Both will be explored for their potential in solving this problem.

# Research Problems

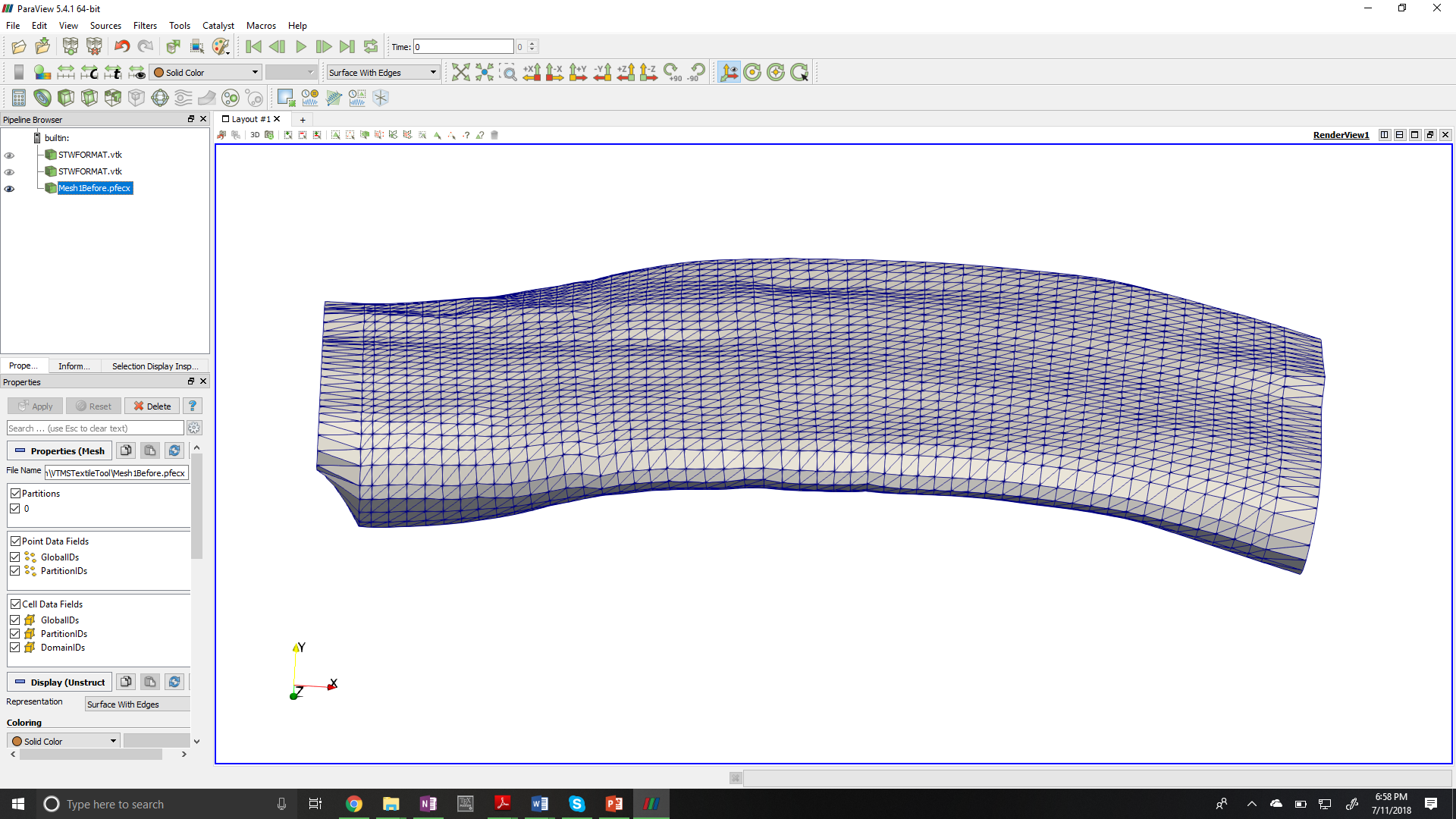
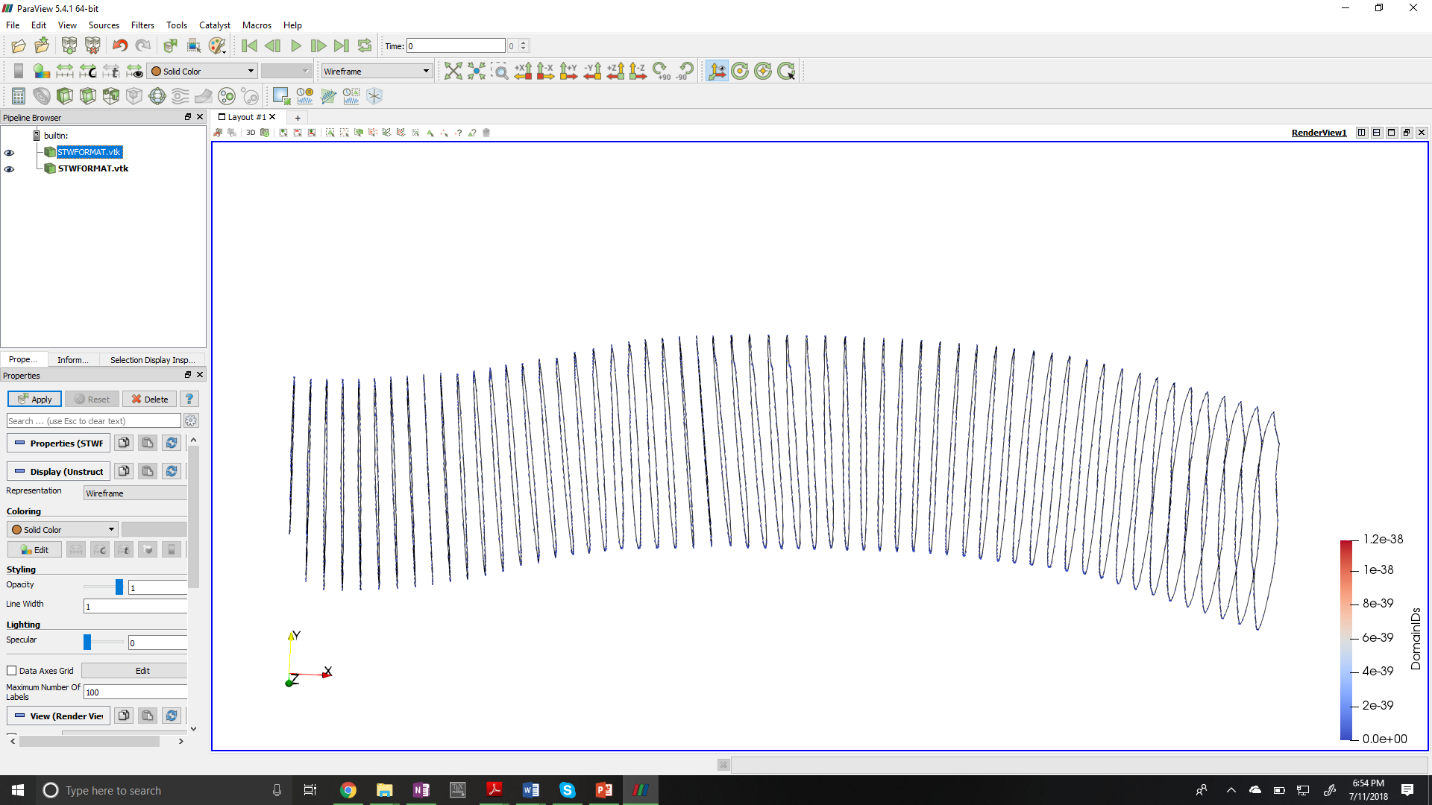
This research can be summarized into three main goals. The goals of this research are:

1. Determine if the VTMS data can be directly used to solve the interpenetrations or if converting to another surface data type (such as NURBS or parametric surfaces) is more useful.
2. Develop methods for each representation type that will accurately identify interpenetration regions.
3. Develop methods that resolve interpenetrations for the chosen surface representation type.

To understand the reasoning for these goals, they will be discussed individually.

## Surface Representation Data Types

The first objective will be to determine if the default form of the surface data from VTMS can be used to solve the interpenetration regions. Currently, VTMS has two data forms of the same surface. One format, the clipped tow format (named by VTMS documentation) is a polygonal surface representation that has polygon surface elements (Figure 6.a). The other form is the standard tow format and is described as multiple cross-sections taken from along the length of the tow surface (Figure 6.b). These cross-sections are called stacks and are individually outlined by nodes that form the outline of the cross-section when connected. The user can choose how many stacks are taken from the tow and how many nodes make up the stack.



1. **Standard tow format**
2. **Clipped tow format**

**Figure 6: VTMS formats for tow surfaces**

Another surface representation type that shows promise in solving interpenetrations is NURBS. There is documentation showing successful resolution of interpenetrations using NURBS for intersection problems. There are also libraries that can be used to help with NURBS related functions such as surface fitting that will be useful so that development time can be spent elsewhere. Other formats will be explored if NURBS are not as simple to use as originally thought.

## Identification of Interpenetration Regions

The second objective will be to determine an accurate way to identify interpenetration regions for the representation types chosen in the first objective. There are ideas and methods presented in literature that can be adapted to the problem of tow surface representations. The results of completing this objective will give information such as boundary curves, element sets, node sets, and surface subsets to used in resolving interpenetration regions.

## Resolution of Interpenetration Regions

The third objective is to develop a method that will resolve the interpenetrations. This method will result in a compatible mesh along a surface or set of surfaces that is shared between the two surface meshes. This method will also export the interpenetration region data to the user. The user may wish to implement a solution that is more complex that the provided result (i.e. they may wish to conduct a contact analysis with the known interpenetrating element sets). The returned tow meshes can then be used by the user to create a matrix mesh that is compatible.

# Research Plan

A short summary of methods and expected results are provided in Table 1.

Table 1. Research Plan Outline

|  |  |  |
| --- | --- | --- |
| **#** | **Task** | **Description** |
| 1 | Determine if the default form of VTMS surface data is the easiest form to use or if a conversion is more useful | Two surface representation types are exported by VTMS. These two forms of the surface may not be the best forms to identify and solve interpenetrations. Therefore, other surface types will be explored for their ability to be used in detecting and resolving interpenetrations. The most promising alternative form is a NURBS representation type as it has third party library support and multiple documented identification methods. The expected results are one or more representation types that show promise in solving interpenetrations. |
| 2 | Identify the region of interpenetrations for the representations chosen | In this task, we will develop methods that can accurately identify the regions of interpenetration between the tow geometries. The methods will not only identify interpenetration regions but also return surface data that will be useful for solving the interpenetrations by a user’s choice method. This includes boundary curves, element sets, node sets, and other data that will be useful in solving the interpenetrations. |
| 3 | Resolve the interpenetrations | This task will use the representations and methods previously identified to correct interpenetrations. Once a method is chosen that identifies the interpenetration region, a resolution will be developed. The method will result in a compatible (in terms of meshing) contact surface. This surface could then have a cohesive zone (or other scheme for modelling the contact properties at the surface) to model the effect of the surfaces being in contact with each other. The method will select one interpenetration region as the master to create a compatible surface between the two surfaces. This method will also export the interpenetration regions from each tow (as element subsets) that can be used in a future user’s own resolution method. This allows for the user to either use the provided resolution or implement their own. |

# 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] Youqi, W.; Sun, X. “Digital-element simulation of textile processes.” *Comp. Sci. and Tech.* 2000, 61, 311-319.

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

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

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

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

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

[12] 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.

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

[14] 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

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

[16] 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

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

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

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

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

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

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

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

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

[25] 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.

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

[27] 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.*