**AFRL Research Collaboration Program**

**Contract FA8650-13-C-5800**

**Effect of Constituents and Microstructure on Energy Dissipation Mechanisms During Damage Growth**

**University: Texas A&M University**

**REPORT COVERS PERIOD: 10-1-17 THRU 12-30-17**

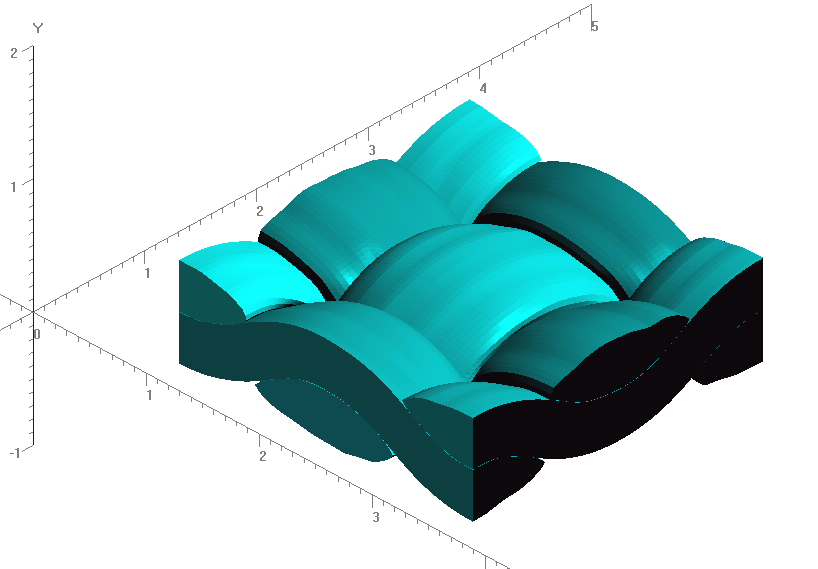
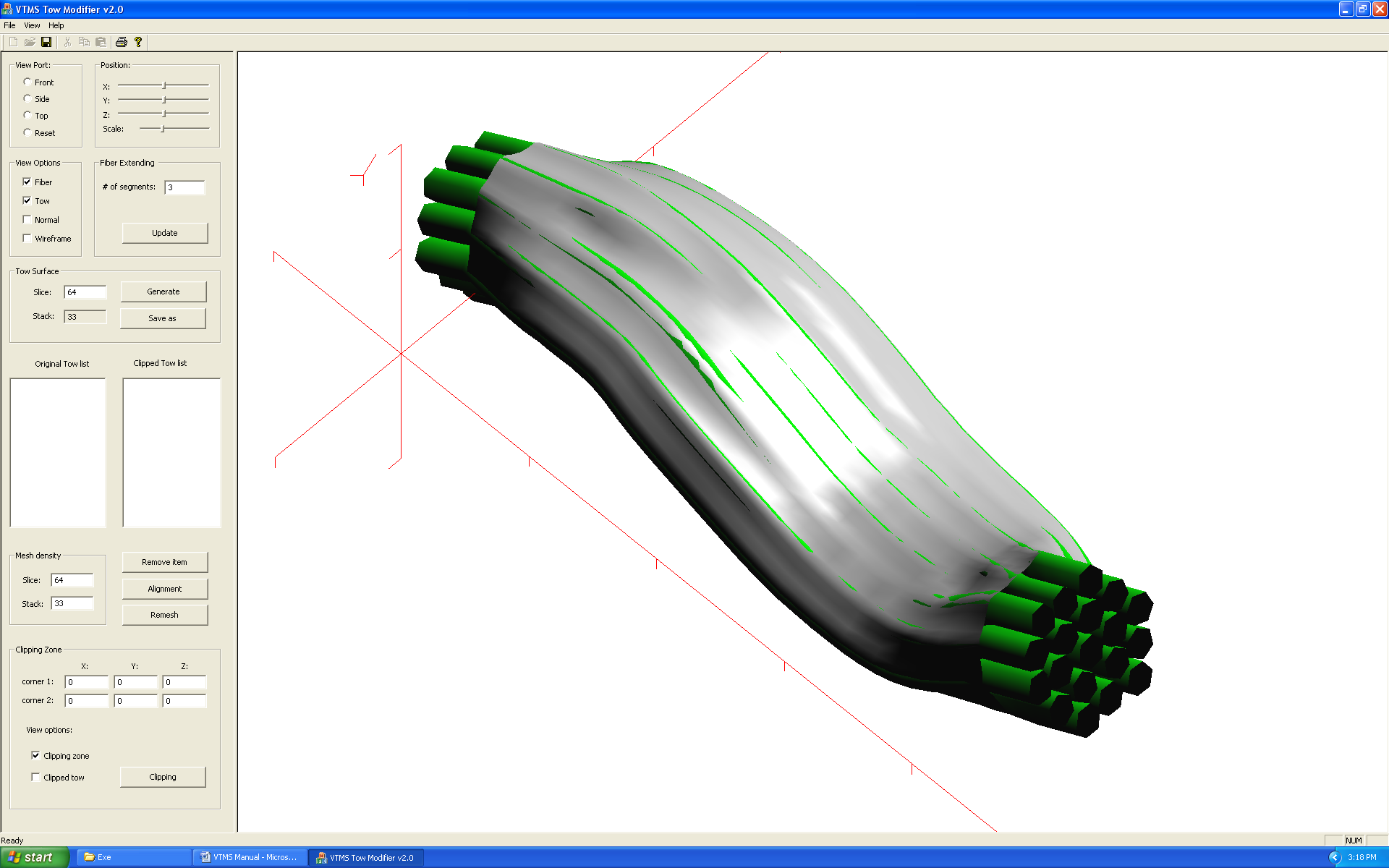
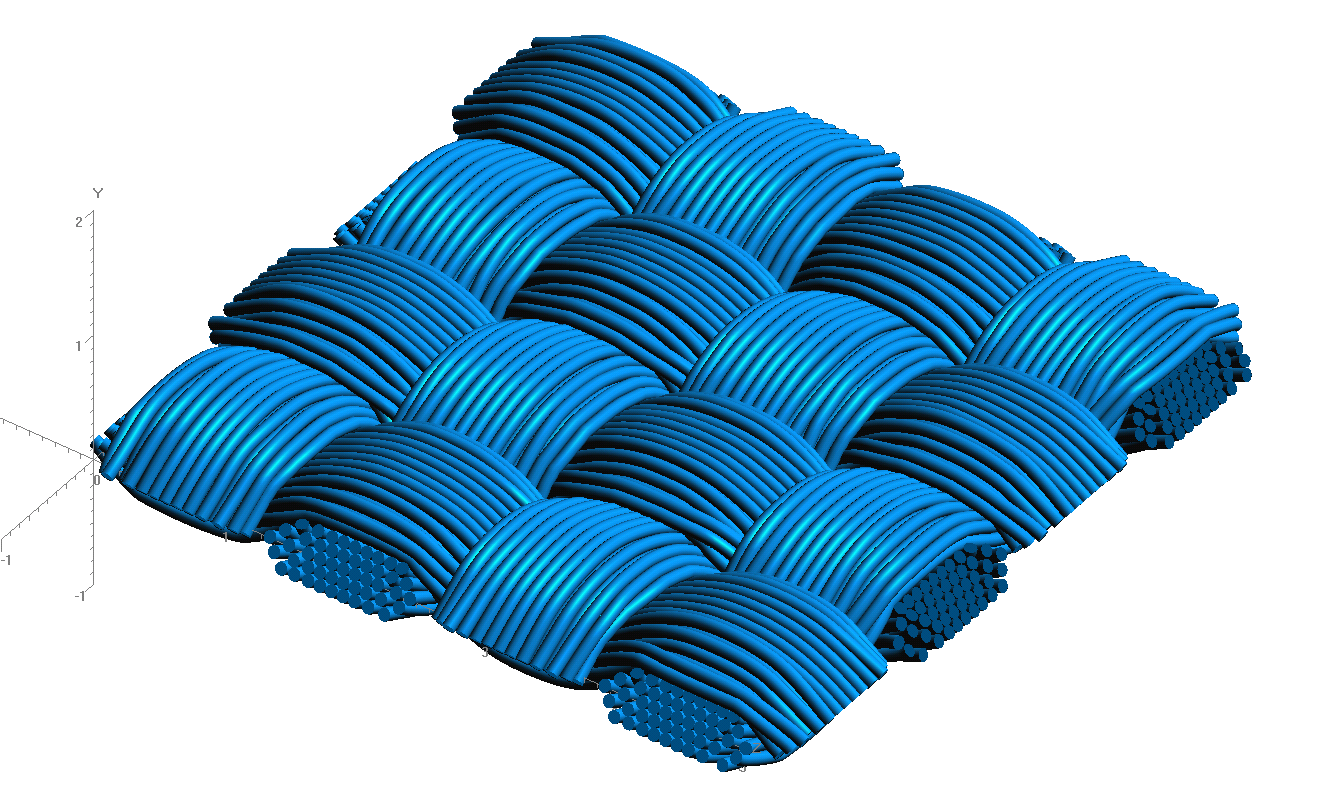
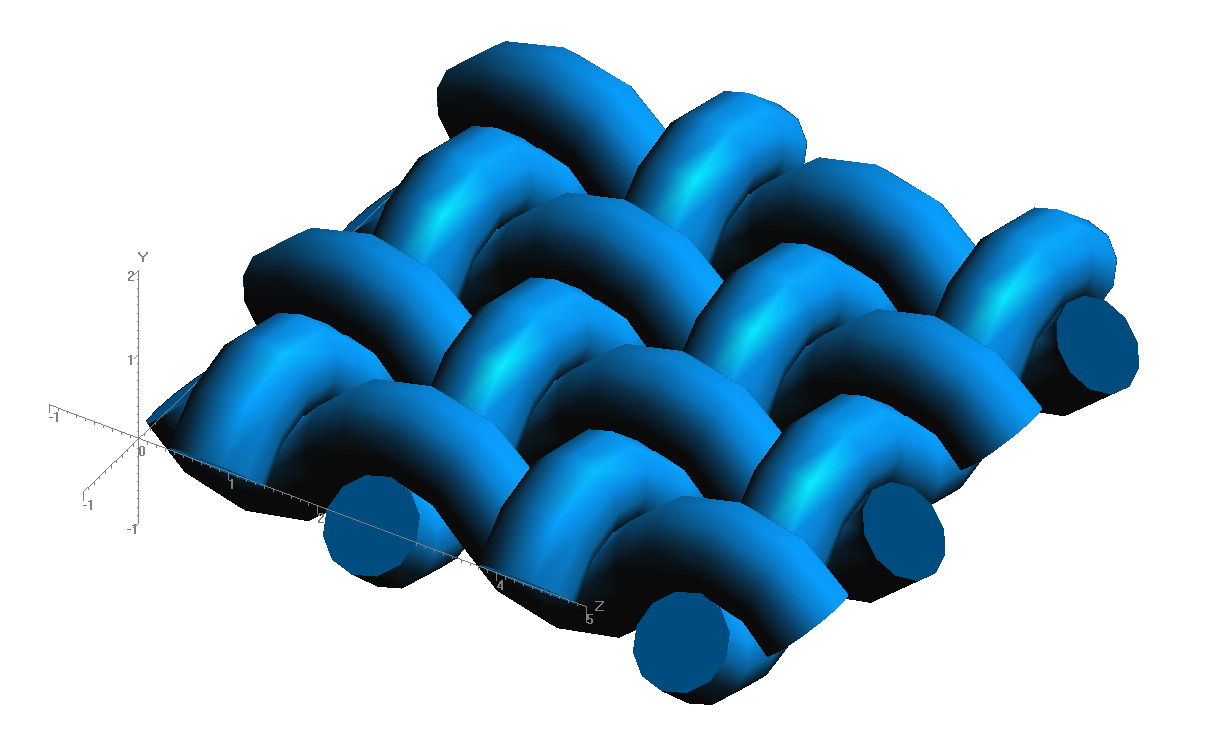
1. **PROJECT TEAM MEMBERS**
2. **LEAD UNIVERSITY POC:** John Whitcomb, 979-845-4006, jdw@tamu.edu
3. **PROJECT TEAM MEMBERS:** John Whitcomb, Collin Blake
4. **AFRL TECHNICAL POC:** Craig Przybyla
5. **TECHNICAL DISCUSSION**
6. **CURRENT WORK**Development of the infrastructure to perform mesoscale analysis of 3D textile composites.

**Background**

In the past couple of years, Collin has been working with AFRL. This past summer, he recently started pursuing the problem of solving inter-penetrations of surface representations of composite tows. These representations come from the process discussed previously and shown briefly in Figure 1. Once the surface representations of the tows are made, regions of penetrations occur between the two representations. Initially, we pursued a solution using volume meshes and polyhedron detection algorithms. This provided some resolution but also had many issues to be dealt with. This pushed us to examine other ways to solve the penetrations. Once solved, Keith Ballard has developed techniques to use the geometry calculated using VTMS and other modeling tools to develop a standard finite element model for a 3D textile composite. I should point out that Keith is not supported by this contract, but does provide mentoring for Collin.

**Approach**

VTMS conducts a four-step process to generate geometric textile models. The results of this process are shown in Figure 1.



**b) Simulated Filament Representation of Pattern**

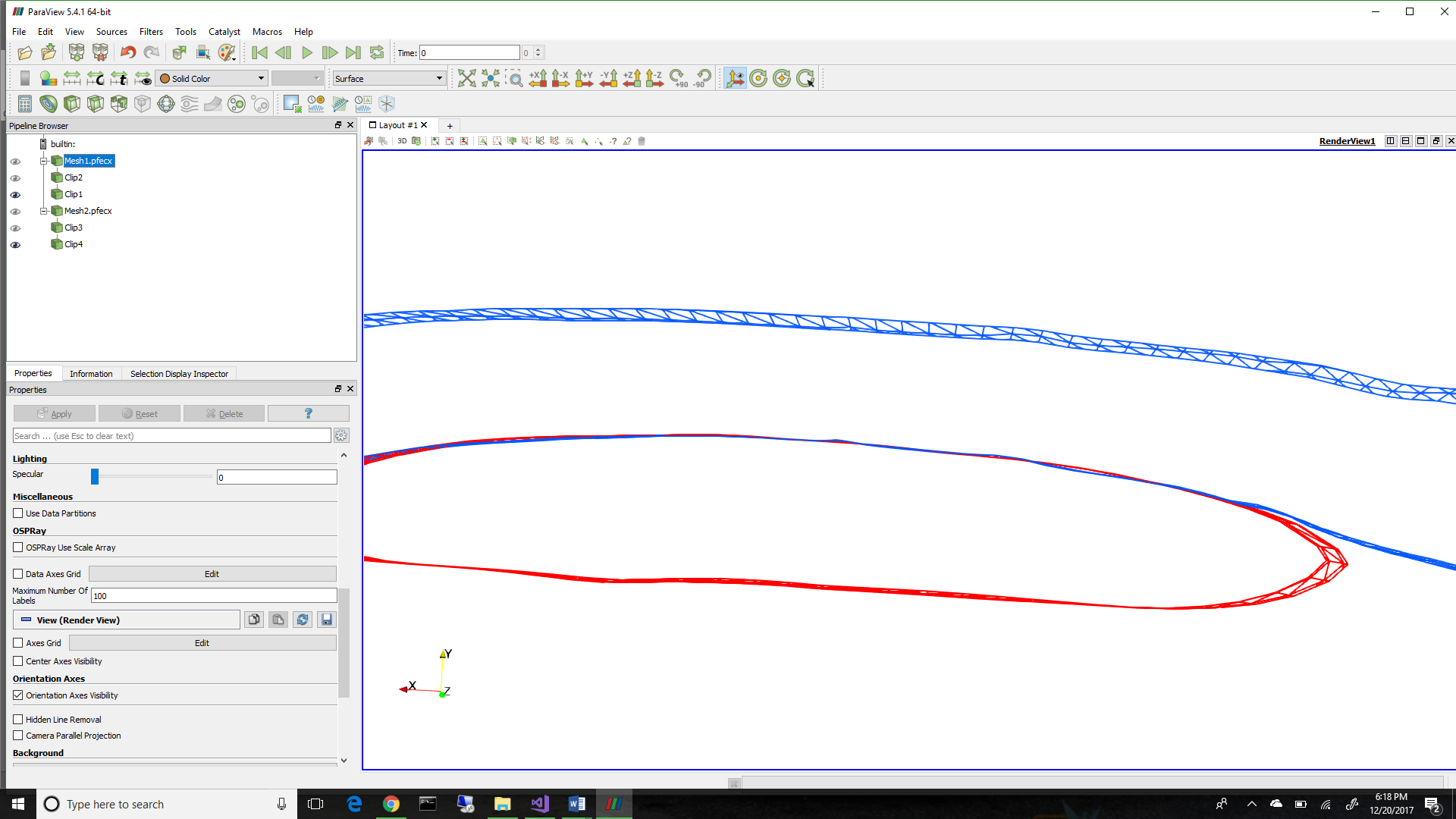
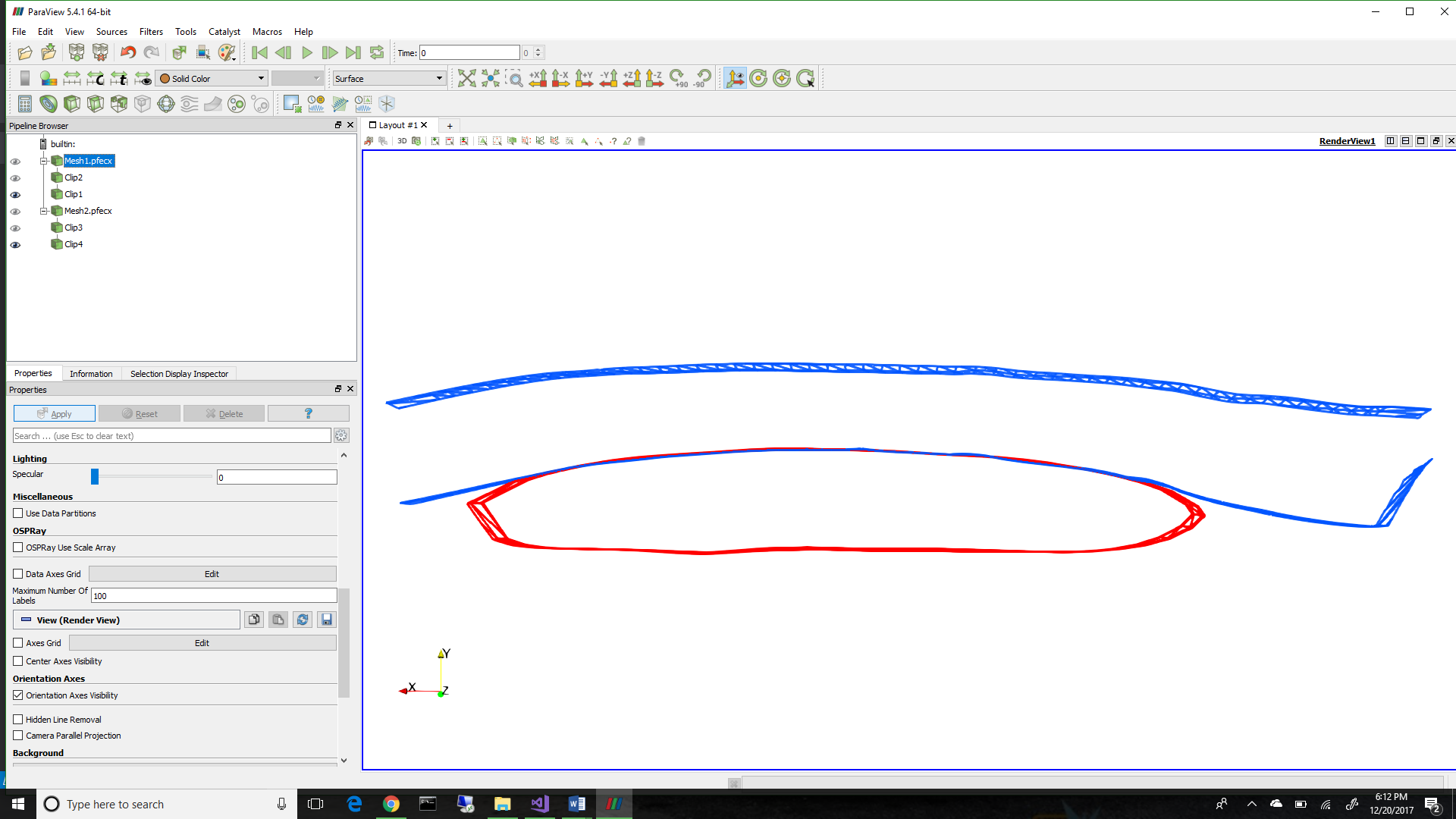
**a) Generic Approximation of Woven pattern**

**d) Volume Model derived from Surfaces**

**c) Surface Approximation of Filament Bundles**

**Figure 1: Evolution of Weave Tow Geometry**

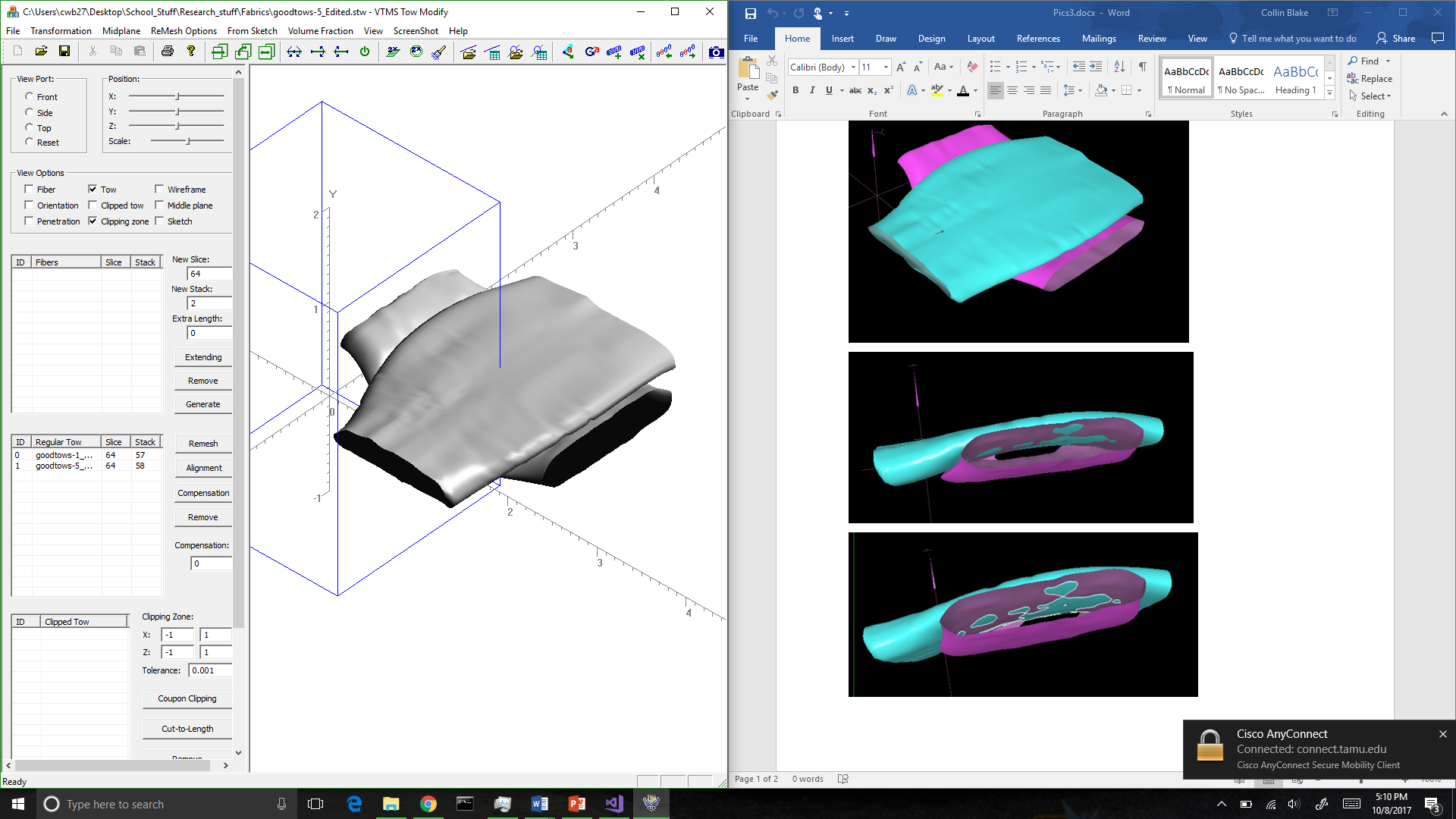
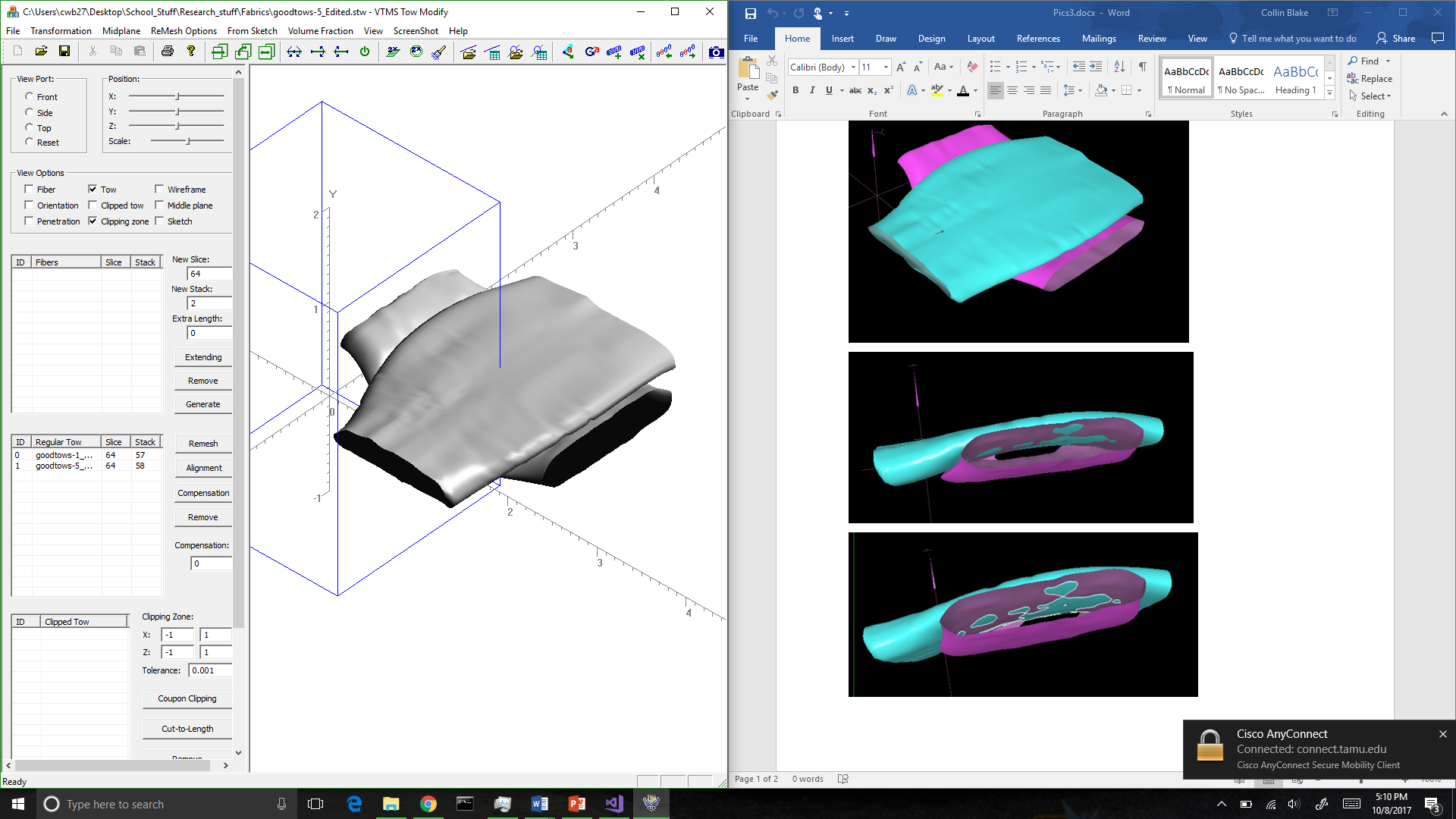
Figure 2 shows a representation of a series of inter-penetrations between two geometries, one running in the plane of the page, the other running out of the page. Illustrated are multiple regions that need to be resolved.



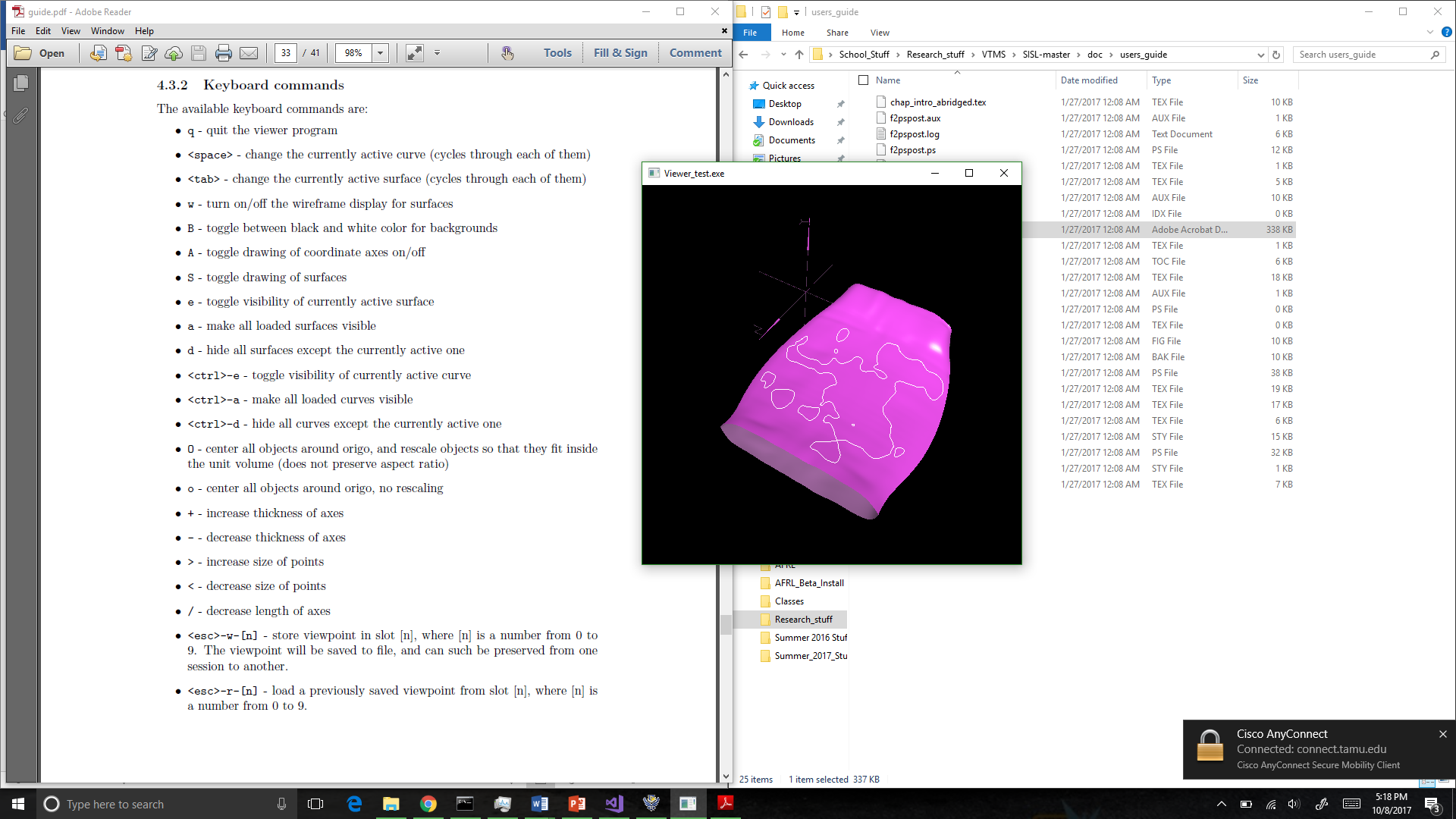
**Figure 2: Illustration of Penetration Phenomena**

Our current efforts are now focused on the implementation and use of a Non-Uniform Rational B-Spline (NURBS) surface algorithm (SISL) written by the Geometry Group at SINTEF ICT, Department of Applied Mathematics. Currently, we are able to fit a surface for a geometry (Figure 4) and detect full regions of penetrations (Figure 5).

This method shows promise allowing a more robust method for detecting and resolving penetrations in these models.



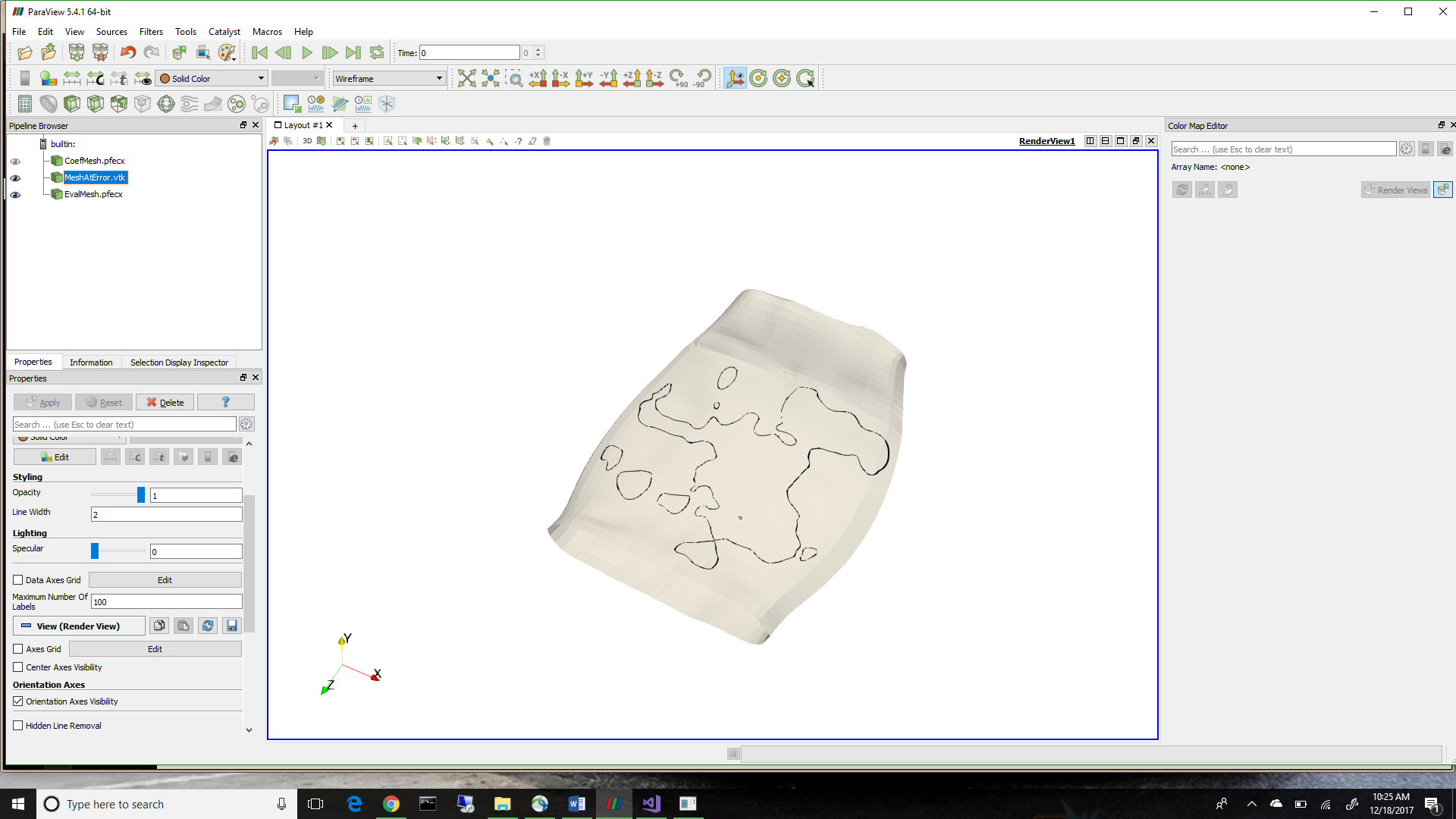
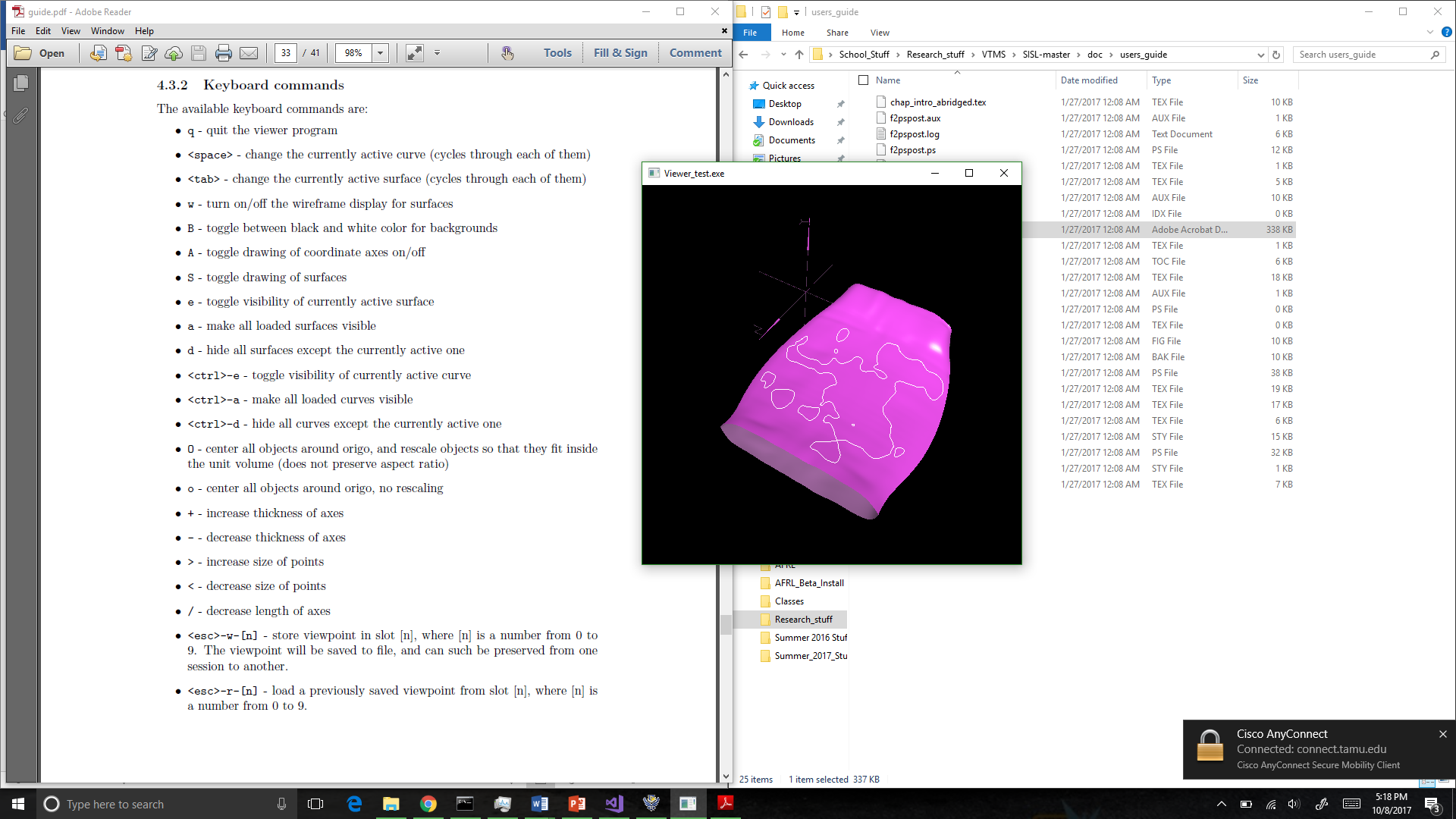
**Figure 3: VTMS Surface (Left) and a NURBS Approximation (Left) of the Same Surfaces**

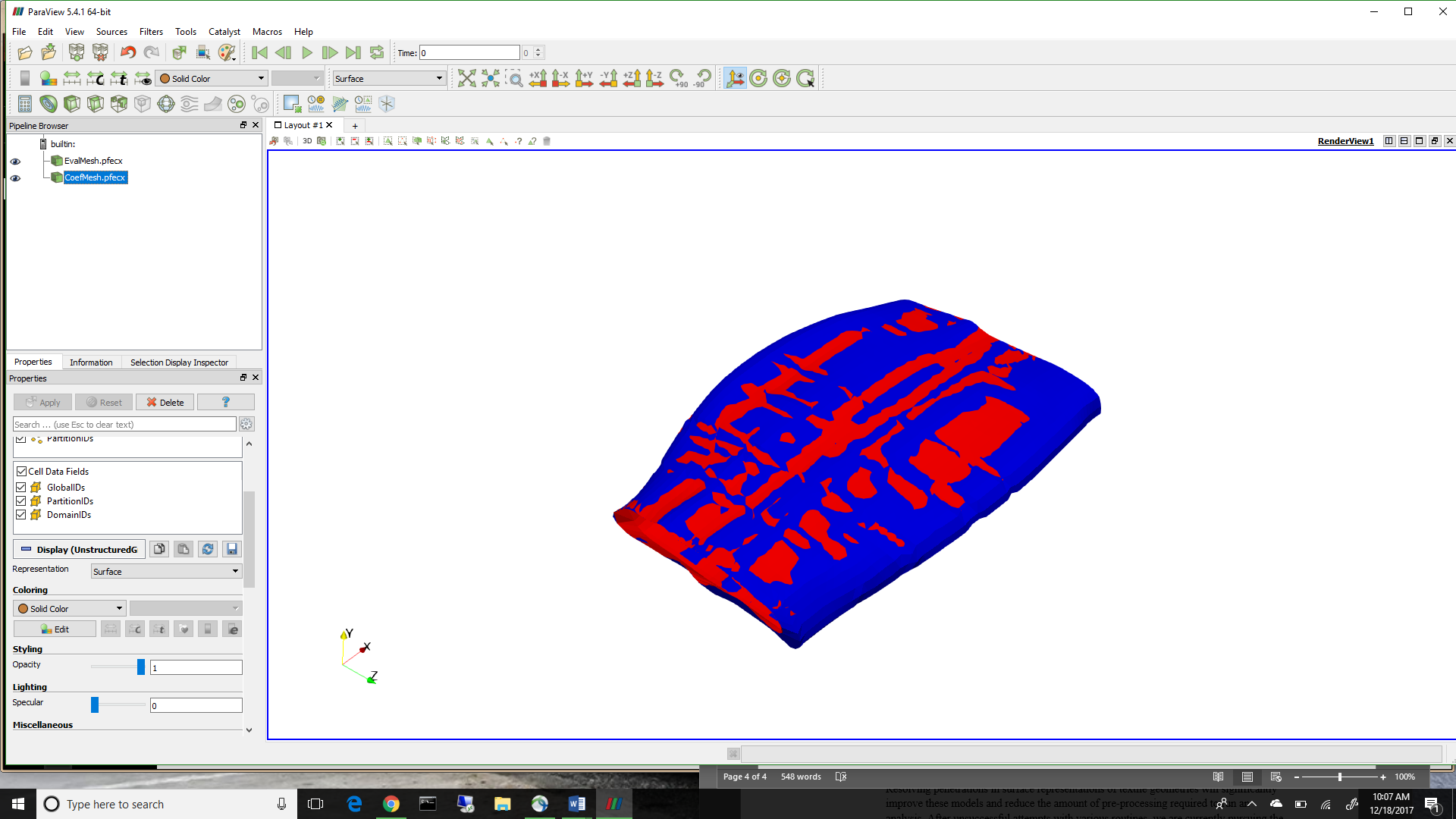
To this point, we have been depending on the viewer supplied by the SISL library to view the NURBS representation and intersection curves (shown in Figure 4). It became apparent that a more powerful viewing program was needed. Therefore, we developed a way translate the data so that the viewer ParaView could be used. This allows us to show node, control point, and element numbers for tracking specifics of the problem.

**Figure 4: Surface with Penetrating Regions Outlined in White**

The next problem to be solved is how to visualize the intersection curves within ParaView. The method for identifying the penetration regions returns a curve that lies on the intersection between the two surfaces. From this curve, we can identify the points that describe the curve and use them to describe the curve in ParaView. To do this, we connect each pair of points with a linear, two node element, and generate a “mesh” of the nodes. This allows us to identify each curve separately. We can also use all of our previously developed mesh manipulation algorithms on these curves to do different operations we may find necessary. The results of translating intersection curves and NURBS approximations can be see below in Figure 5.

**Figure 5: ParaVeiw vs. SISL viewer of Translated Data**



To display the surface representation, there are two methods that can be used. The first is to simply use the point data supplied by VTMS and connect them by triangular elements to create a surface mesh. The other is to take the NURBS surface and evaluate the surface on an evenly spaced grid in the two parametric directions. This method will give us a more accurate method in displaying the NURBS surface over simply using the control points that describe the surface. This is also what the SISL library uses to evaluate the intersection between the two surfaces. The result of the two representations can be seen below in Figure 6. The blue surface is the VTMS data/control points and the red is the evaluated NURBS surface. You can see that they are very similar but not exact, which is expected. The difference between the two is minimal, however we have no metric as of yet to determine how much of a difference exists. A good metric to measure the difference in the two would be the volume occupied by each geometry.

**Figure 6: A Comparison of NURBS surface and VTMS Data**

1. **CONCLUSIONS/ANALYSIS TO DATE**

The decision to use NURBS surfaces required the translation of NURBS data into something that could be visualized. Therefore, we developed the translation of this data so that we could view it in a more powerful viewer known as ParaView. Now the data can be viewed with much more information. We also know that there is a difference between the evaluated NURBS surface and the data passed in that is used to control the surface. Therefore, the NURBS approximation should be used if we wish to get the same data used for the intersections. Now we must deal with the curves given from the intersection algorithm

1. **WORK FORECAST AND PLANS**The curves returned from the algorithm are not guaranteed to be either unique or closed (even though every intersection should return a closed curve). Therefore, we will work on the best algorithm to combine open curves and delete any duplicates. Once we have identified all of the unique, closed curves we will determine the best way to add the curves to the mesh of each surface and begin determining the best way to handle these regions.