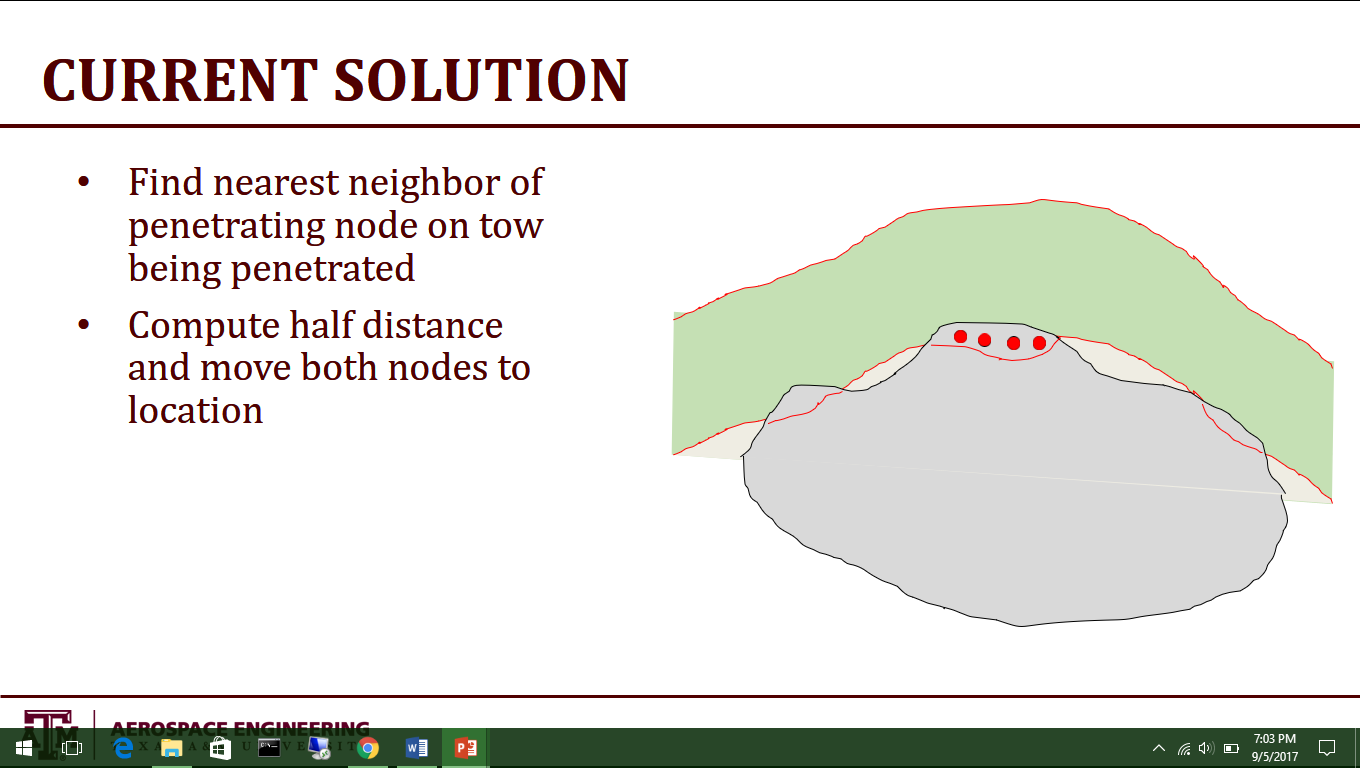
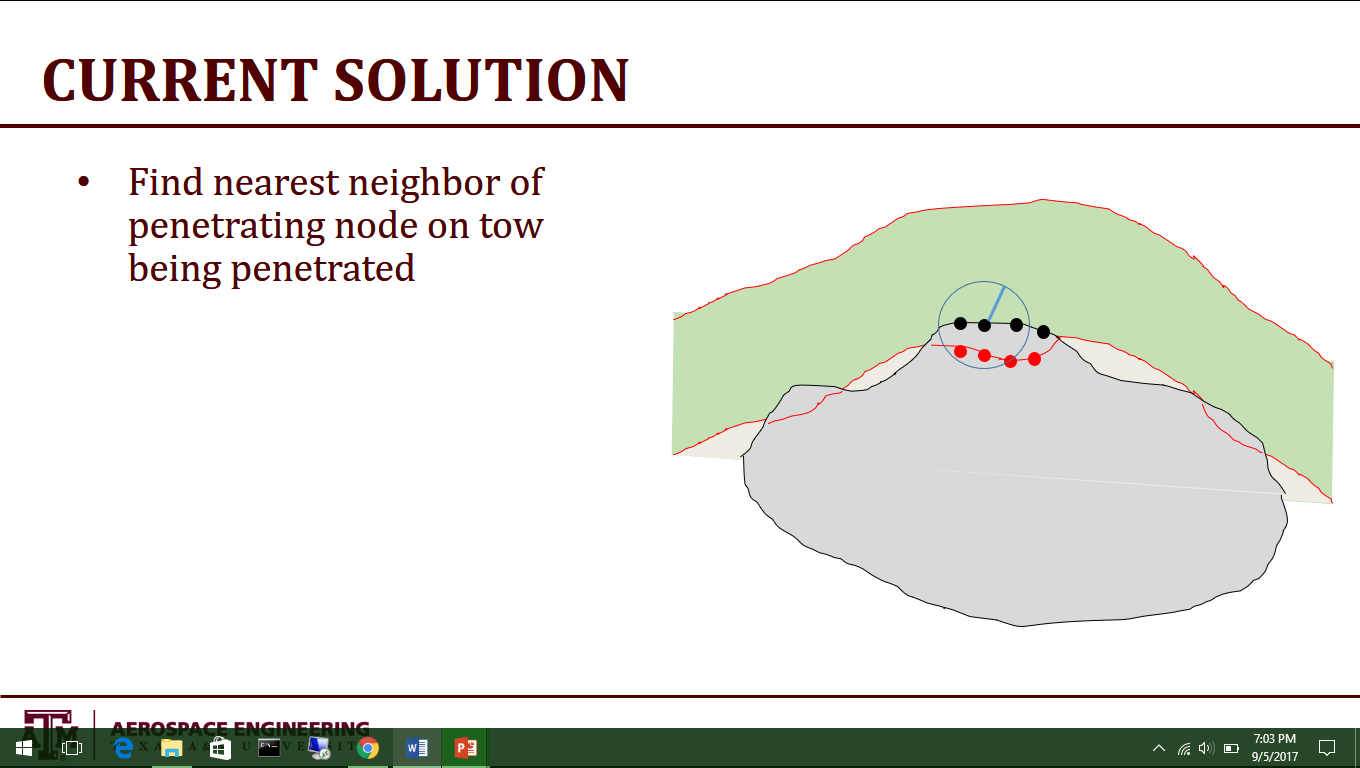
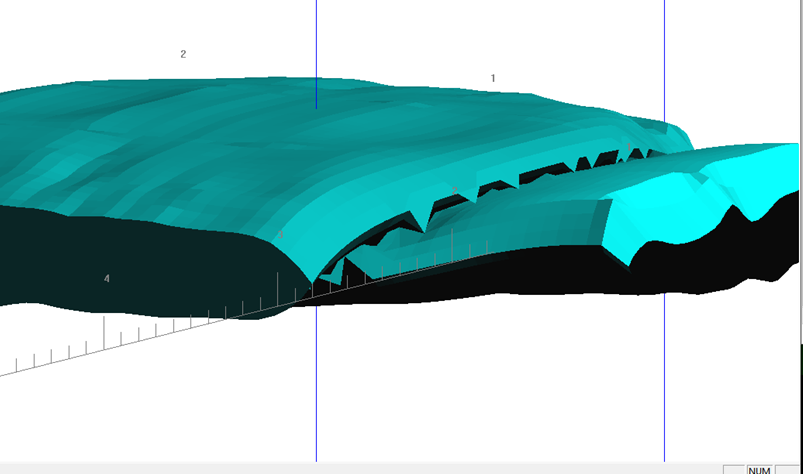
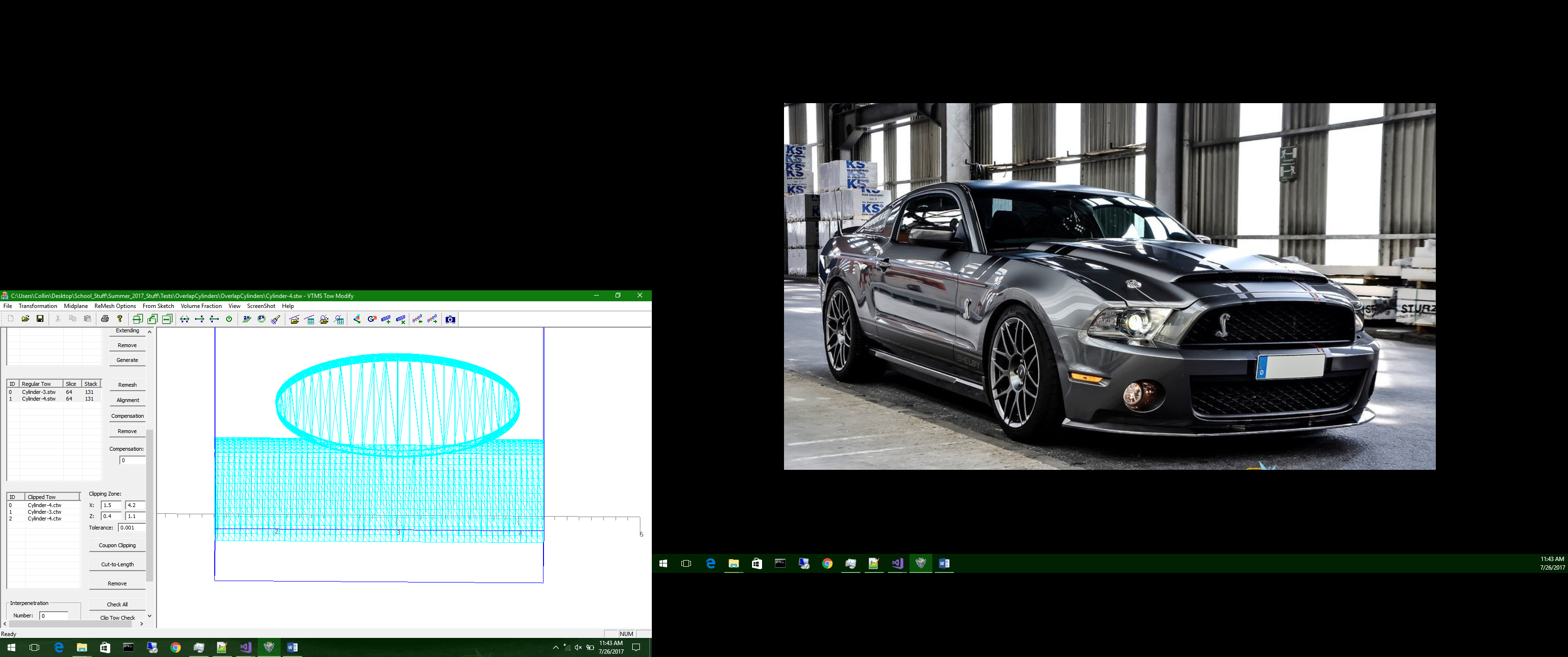
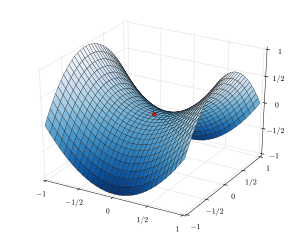
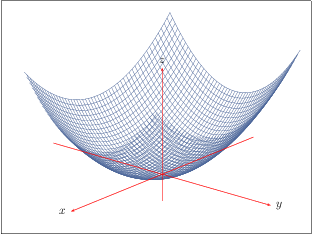
**Accomplished:**

1. Node to Node pairing using CTW data format (Typical Mesh data with nodes, connectivity, and elements)
   * Process takes a node that is known to be penetrating, uses a nearest neighbor algorithm to find closest node on opposing tow (that is penetrated), calculates the half distance and vector between the nodes, and assigns both nodes to the half distance location



* + **Attributes**
    - This method forces compatibility between nodes of opposing tows in penetrating regions
    - Simple implementation within coding structure of VTMS
    - Alleviates the need to define a contact surface between the two geometries
  + **Drawbacks**
    - Can create collapsed elements (two nodes of the same element occupying the same location)
      * Can be overcome by simply remeshing the nodes defining the surface, while discarding duplicate nodes
    - Nodes may have to significantly move location to reach the half-distance position, thereby skewing elements and creating high aspect ratio elements
    - Hard to take into account every scenario (i.e. more than two intersecting tows)
    - Can also create “horns” on boundary of penetrating region (see below) due to false penetration detections



1. Contact surface with STW data format (geometry separated into slices along tow path, each slice defined with a pre-determined number of nodes)
   * Current uses an algorithm that finds all penetrating nodes in a penetration region (from both tows), finds the centroid of all penetrating nodes, determines the average tow path in the region for each tow, and defines a plane orthogonal to both tows at the centroid. The flat plane variation works well but does not create physically realistic surfaces seen in tow contact regions.
   * **Attributes**
     + Straight forward to implement
     + Creates a perfect surface of contact between the two geometries that has a mathematical definition
   * **Drawbacks**
     + Finding right surface definition can be hard (if not using a flat plane)
       - The most ideal solution would be a best fit surface in three dimensions (think saddle shape of bi-variate quadratic surfaces)
     + Compatibility between surface nodes still needs to imposed. This is difficult with this particular data format due to orthogonal slices that
2. Modularization of written code in header and source format
   * This makes it easier to edit along with separating the particular functions need for these operations from the rest of VTMS
   * Allows us to edit algorithms offline without needing the entirety of VTMS
   * Could be implemented in other meshing software (i.e. BetaMesh)

**Need to do:**

* Analytical surface for tows – NRUBS “SISL” Library (Keith’s thoughts)
  + **Should be our first focus as it could solve a lot of our issues**
  + With analytical surface definitions of the tow geometries, boundary curves of intersecting regions can be found that can be used to define the contact surface between tows
  + A good first attempt would be subtraction of penetrating regions from the surface (due to the relative size of penetrating regions, this could solve the problem with minimal effects on the volume
* Create method to run VTMS without the GUI (Keith’s thoughts)
  + Will be accomplished rather quickly and will be finished during the implementation of NURBS
  + This will push us towards being able to script penetration removal and other various VTMS algorithms without having to navigate the bulky UI
  + Will also reduce the time needed to generate textile models using VTMS
* Git repository for VTMS (Email Lauren)
  + I will ask our contacts at AFRL if this is possible so that any changes made by us or Eric Zhou will be logged and stored in one central place
  + Depends on the distribution rights of the code itself
* Optimizing/Analysis of the relaxation module of VTMS (if time permits)
  + The relaxation module is a significant bottleneck of generating textile models as it can take multiple tries over multiple days to generate one textile geometry
  + IF the relaxation module can be made more efficient and faster, generating models will be significantly quicker.
* Surface/Compatibility resolution (with NURBS)
  + If NURBS surfaces become the best way to generate contact surfaces, node compatibility will still need to be imposed on the surface between the two (or three) tows

**Tried:**

* Surface Fitting to all penetrating points
  + Tough to define correct polynomials that describe the surface that best fits all of the points. This goal of this strategy is not to accurately fit a surface to the data but rather have a general shape of the average of the penetrating nodes to then use the surface as the contact surface
  + Still need to deal with compatibility of nodes on the surface
* Polyhedron detection
  + Used in video games but depends on time stepping and usually for avoidance of penetration. Does not resolve the case of actually penetrating geometries
  + Difficult for many sided Polyhedra as the algorithm has to iterate over all of the facets
  + Will still need distance data to solve penetrations. This would require either finding an additional library or writing one ourselves