**Thesis Outline**

**ABSTRACT**

The abstract will briefly introduce the problem and a top-level summary of the methods. The abstract will also address the results of the methods and the implication of these results.

**INTRODUCTION**

This section will contain an introduction into both the problem at hand and the methods used.

**Composite Analysis**

* Give introduction to composite analysis
  + Serves to give background into why this problem is important
  + Also discuss reason for idealized vs. non-idealized geometries to introduce need for something like VTMS

**VTMS**

* Discuss digital chain process
  + Likely top-level discussion with references
* Discuss process of producing surface representations
  + Rolling pin method
* Possibly discuss clipped tow vs standard tow format if necessary

**PROBLEM**

The surface representations created after the physical simulation process used by VTMS contain interpenetrations between the surfaces that do not allow the use of traditional finite elements.

* Illustrate in detail interpenetrations between surfaces
* Discuss origin of interpenetrations
  + Error comes from linear segmented chains and sphere size
  + Also comes from approximate of individual filaments as one volume

**LITERATURE REVIEW**

**Interpenetration detection algorithms**

* Discuss detection in continuous vs. discrete domains
  + This will setup the discussion of using a meshed surface vs. a continuous surface description like NURBS
  + Continuous
    - Point-Marching formulations
  + Discrete

**Interpenetration resolution**

* Discussion of general resolution (fixes) for interpenetration between objects
  + Illustrate the multiple methods that could be potentially used to fix the interpenetrations
    - Unions
    - Subtractions
    - Common boundaries

**Other VTMS interpenetration fixes**

* Discuss other research that uses VTMS and how they overcome their issues
  + Discuss pros/cons of these methods and why another solution is needed

**METHODS**

**Creation of NURBS surface from VTMS**

* Discuss both format of VTMS standard tow and why NURBS are chosen (site reasonings from literature review)
* Discuss using VTMS data to create parametric space for NURBS surface
* Describe transform of parametric to real space for each tow surface

**Intersection curves**

* Use of NURBS surface intersection algorithm to create NURBS curves for intersections between surfaces
* Linearization of curves so that they can be visualize and also used for future algorithms

**Using Discrete surfaces (Method previously written up)**

Methodology is to include intersection curve in each surface to create a common boundary. Then, the region inside the curve is the interpenetration region. Can remove or repair as is appropriate.

* **Method**
  + Conversion of NURBS into meshed surface
    - Division of parametric space uniformly
    - Use of NURBS library to find 3D location of surface using uniformly divided parametric space
  + Use of Separating Axis Theorem to determine which intersection curve segments intersect with surface elements
  + Subdividing intersection curve elements where they intersect surface element boundaries
    - Done for both surfaces
  + Remeshing of surface element to include intersection segments within its boundary

**Using continuous surfaces**

* **Method**
  + Keep surface as NURBS
  + Use intersection curves (linearized) points to query nearest parametric coordinate set on surface
  + Record returned parametric coordinates of surface for each curve point
  + Use recorded parametric coordinates to create representation of intersection curves in the parametric space of the surface
  + “Mesh” the parametric space (including intersection curves) of the surface, resulting in 2D mesh
  + Use “mesh” of parametric coordinates to return 3D location of surface at those parametric coordinates
  + The result is a 3D mesh of the surface with the intersection curves included

**Collecting Interpenetrating elements**

* **Method**
  + For each surface element, a ray-intersection algorithm is conducted against each intersection curve segment
    - This method shoots a ray in a random direction (for us we choose the direction based off of one edge of the element)
    - If the ray intersects an odd number of segments, the element is inside of the curve
    - If the number of intersections is even, it is outside of the curve
  + The elements are then grouped by which curve they are inside of
  + Record is also kept if the element is in multiple curves
  + From here, the elements can be removed and replaced with one set, chosen from either surface

**Meshing Matrix (Keith)**

A separate app creates a matrix around the tows and volume meshes for the tows. This app also retains traditional finite element compatibility rules so that the resulting matrix and tow meshes can be used in traditional finite element software.

**RESULTS/DISCUSSION**This section will include a discussion on the pros and cons of each method. This will also include a discussion on why one method was abandoned.

**CONCLUSIONS**