# 6. Analysis of automatically generated T-joint Models for Optimisation

## 6.1 Introduction

In Chapter 5, a description was provided of how meso-scale 3D woven T-Joint geometry models were automatically generated using TexGen’s Python scripting interface. The next requirements for using this to find optimum weaving patterns for the woven reinforcements are automatic mesh generation and a method of scoring each weave or evaluating an objective function value.

In this chapter, the meshing of the T-joints using both voxel and octree refined voxel mesh methods for use in optimisation is described. The tensile pull-off test that was performed in Chapter 3 was simulated using finite element analysis for use in the optimisation routine in the next chapter. The finite element meshes are compared before one is chosen and the modelling scheme to be used to score each weave in the optimisation is set out.

## 6.2 Meshing of T-Joint Weave Models

Once the meso-scale geometry models are produced, a domain is created using the *CPrismDomain* TexGen class, which will be included in an upcoming release of TexGen [10], which allows the domain of the model to be specified to conform to the outline of the T-shape. This domain can then be meshed. Before this work, only voxel meshes could be generated on prism shaped domains. The *COctreeVoxelMesh* class was originally designed to mesh rectangular block domains and so it was necessary to adapt it to be able to mesh the outline of the T-profile. This was achieved by checking whether each element’s nodes were in the bounds of the specified prism outline, which in this case was the T-profile.

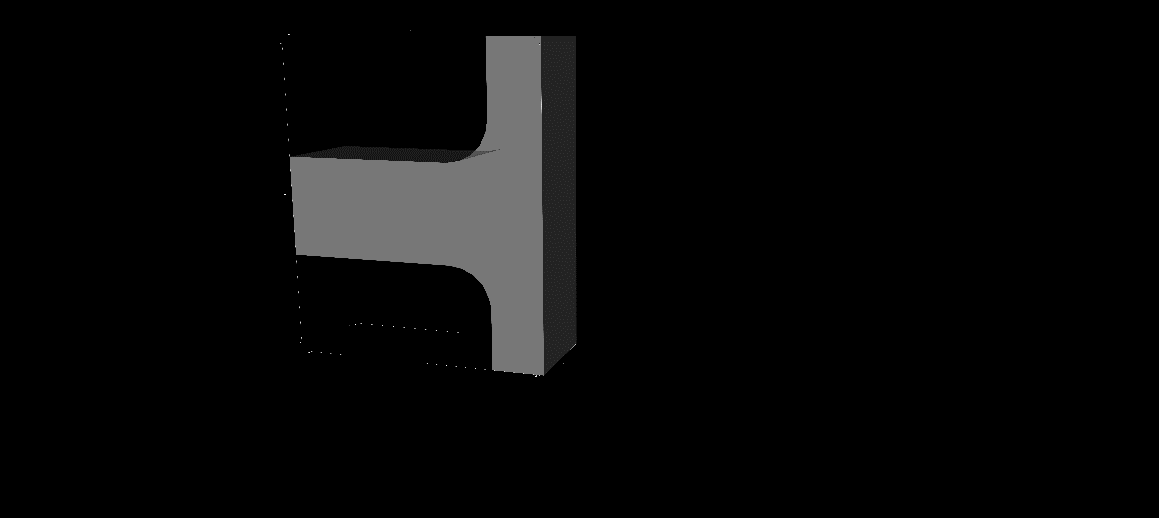


Figure 6-1 TexGen prism Domain with T-joint outline. The T shape is defined as a set of points and then provided with a length to form the full 3D shape.

This allows any profiled shape domain to be meshed with the octree voxel mesh in TexGen, allowing the possibility for users to use it to mesh complex domains such as I and Pi shaped domains among others. In this work, both voxel and octree refined voxel meshes were compared to be able to identify the most suitable for use to mesh the geometries in an optimisation routine. Factors considered included the number of elements, the geometry conformity and the solution accuracy.

### 6.2.1 Voxel Mesh

Models of composite textile structures have often been meshed with voxel meshes which provide a robust solution to the problems associated with the automatic generation of conformal meshes, namely their inability to capture the yarn geometry with elements of sufficient quality. Additionally, matching nodes on opposite surfaces allows for easy application of periodic boundary conditions evaluating unit cell textile models. Conformal meshes have been shown to produce distorted elements upon meshing yarns in proximity to each other with the solution in [93] to be the reduction of the cross sections to provide contact clearance. This is because of their rigid constraints of conformity to, and displacement continuity across, the interfaces between the yarns and between the yarns and the matrix.

One of the disadvantages to voxel meshes is the jagged, step-like interface between the yarns and the matrix elements formed when the interface is not aligned along any of the three global cartesian axes in which the model was meshed. This can cause stress concentrations and requires the user to be selective in the uses for voxel meshes. One such area where it is unclear the voxel mesh is suitable is when cohesive surfaces or elements are employed.

The effect of the interface depends on the stress state of the elements on either side. If tensile stress normal to the interface dominates then the effect is reduced. Conversely, shear locking of elements causes much higher stresses at the interface. A check should be made that the shear stress should be low compared to the direct stresses at the interfaces between the yarns and matrix elements.

Another disadvantage of rectangular voxel meshes is their relative inability to capture the yarn cross section geometry accurately. They require many more elements than those required in conformal meshes to approximate the cross-section geometry.

### 6.2.2 Octree Voxel Mesh

Matveev et al. [12] proposed and implemented in TexGen an improvement to the standard voxel mesh that addresses the problems caused by the step-like interface. It uses the dual methods of octree refinement, where the elements that straddle the material interfaces are divided and stored in an octree, and smoothing, where the element boundary nodes along the interfaces are moved so that the mesh elements better approximate the interface boundary and is closer to a conformal mesh. The combination of refinement and smoothing improve the performance of the voxel mesh. Hereafter, this method shall be referred to as the octree voxel mesh and it will be specified as to whether the smoothing algorithm has been applied.

Each mesh element straddling the boundary is divided and the process is repeated with straddling elements divided until the user-specified maximum level of refinement is reached. The memory structure in c++ that holds the node structure information is called an octree. The mesh is “balanced” so that there is a gradual change in the level of refinement at further distances from the material interfaces (see figure 6-2). Using this method can result in fewer elements being needed to discretise the entire model when compared to the voxel mesh, as not all the elements need to be as refined to fully resolve the model geometry.

A picture containing text, athletic game

Description automatically generated

Figure 6-2 Illustration of the octree voxel meshing of a yarn cross section with subdividing of elements that contain material boundaries [12]. Each successive level of refinement refines elements along the boundary allowing the yarn surface geometry to be better resolved.

The drawbacks to this method include the extra hanging nodes that are generated due to the different element sizes caused by the various levels of refinement. These require extra multi-point constraint equations to ensure the refined elements maintain continuity, which slows down the analysis.

One further advantage of using the octree voxel mesh is the ability to output element surfaces. This allows the use of cohesive surfaces as applied in Chapter 4, removing the requirement to add in cohesive surfaces by hand using third-party software such as HyperMesh [72].

## 6.3 Finite Element Modelling of T-Joint Weaves under Tensile Pull-off Load

The tensile pull-off test from Chapter 2 was modelled to evaluate the difference between each T-Joint weave and provide a basis for scoring each weave during the optimisation.

### 6.3.1 Boundary Conditions

In the tensile pull-off test, the T-Joint samples were clamped at the flanges and load applied to the web. To simulate this in the finite element model, the nodes on the outward facing surfaces of each flange were fixed in all directions using Abaqus’ encastre boundary conditions, with the tensile pull-off load applied to the top surface of the web using a displacement boundary condition to a driver node connected to the top surface nodes by linear constraints (see figure 6-3).

A blue rectangular object with red arrows

Description automatically generated

Figure 6-3 Finite element T-joint model boundary conditions for tensile pull off.

There is periodicity along the length of the 3D woven T-joint reinforcement so periodic boundaries can be applied along the length. However, single unit repeats of the weave pattern in the x-direction were used with no periodic boundary condition along the x-axis since it had been reported by Yan [45] that this had little impact on the elastic behaviour of the T-Joints.

In the experimental set up discussed in Chapter 3, the T-joint flanges were not fixed at the ends but clamped at the top and bottom at a point further in-board from the end. This has the effect of constraining the movement of the T-joint at the clamp locations. It is proposed that for the purpose of comparing the effect of the different weave patterns on the mechanical properties of the T-joint the difference will not have much of an effect because the boundary conditions are the same for each finite element analysis in the optimisation.

### 6.3.2 Time Control

The quasi-static analysis was chosen to be solved using the Abaqus explicit solver as it was likely to reduce the analysis time. To model a quasi-static process such as the T-Joint pull-off test, a reduced time step can be used. This allows a much faster time step that is more suited to Abaqus explicit’s usage. Trial models using the implicit solver took too long to be used within an optimisation. Abaqus documentation [67] recommends finding a lower bound on the explicit time step by examining the model’s modes using the eigenvalue analysis, linear perturbation time step.

For the T-Joint models, this produced the first mode frequency of 270Hz, with a time period of 3.5ms. The general recommended lowest time period is at least 100 times this value to avoid inertial effects. This should result in a velocity graph for the machine head that increases, reaches a constant velocity and then slows down. The resulting kinetic energy and internal energy graphs ideally would then look like those presented in figure 6-3.

Chart

Description automatically generatedChart, line chart, histogram

Description automatically generated

Figure 6-3 Graphs showing ideal kinetic energy and internal energy graphs for a quasi-static process. Taken from the Abaqus documentation [67].

A time period of 300s was chosen to displace the top surface of the web 5mm, resulting in a loading rate of 1mm/min. This resulted in kinetic energy and internal energy graphs as seen in figure 6-5.

Graphical user interface, application, Word

Description automatically generated

Figure 6-4 Internal and kinetic energy graphs against time during the tensile pull-off process.

The process can be said to be quasi-static if the kinetic energy of the model remains small in comparison to the internal energy. From the graphs in figure 6-4 it can be seen that this is the case with the maximum internal energy approximately 80 times larger than the maximum kinetic energy. The deviation after 3mm of the kinetic energy graph may suggest that after this point the pull-off model is no longer quasi-static. This is due to oscillations in the solution caused by the reflection of the vibrational wave. Therefore, the models were cut-off after 3mm of displacement. This provided enough displacement to cause the models to reach the failure threshold discussed in the later sections.

### 6.3.3 Creating Boundary Node Sets for the Octree Refinement Mesh

Boundary conditions in Abaqus can be applied to one of Abaqus’ geometry references or to nodes and surfaces in the input definition. In this case, to model the T-Joint pull-off test, boundary conditions are required on the top surface of the web and the outward facing side faces of the flanges. For a voxel mesh, the consistent nature of the elements in size and shape renders it easy to find the node numbers labelling the nodes on the surface boundary. They will be consistent from one geometry to the next and can be calculated if the element size and prism domain shape are known. In the octree voxel mesh, element sizes and shapes vary across the model depending on the proximity to a yarn/matrix interface surface and the number of smoothing iterations applied. These nodes vary depending on the weave model geometry and parameters used to generate an octree voxel mesh. Therefore, there is no set method to determine these node numbers when using this meshing method.

It was possible to test whether these nodes were on the domain surface. Using this method, nodes were added to a new class *CTJointBoundaries* which is used to create boundary conditions for the T-Joint models for creating the T-Joint finite element models.

These boundary nodes are then tied by linear constraint to driver nodes to which the boundary conditions are applied. The reason for this is so that the force applied to the node is distributed equally among the nodes on a surface.

### 6.3.4 Material Properties

Each yarn is considered as a unidirectional composite with transverse isotropy. Material properties for such yarns can be found using Chamis’ rule of mixtures [94] alongside using measured properties or those supplied by the manufacturer. As the purpose of this work is to find optimum reinforcement weave patterns, the material properties between each design produced by the optimisation algorithm will be the same. In this case, the material properties stated by the manufacturers of the materials used in Chapter 2 are used.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | E1 (GPa) | E2/E3  (GPa) | ν12/ν13 | ν23 | G12/G13  (GPa) | G23  (GPa) |
| Infused Hexcel Im7 Yarn | 174.4 | 8.9 | 0.3 | 0.3 | 4.2 | 4.2 |
| Gurit Prime 20LV Epoxy | 3.5 |  | 0.5 |  |  |  |

Table 6-1 Elastic properties of material constituents.

The strengths of the infused yarns can be found using empirical formulae [95].

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | F1t  (MPa) | F1c  (MPa) | F2t/F3t  (MPa) | F2c/F3c  (MPa) | F12/F13  (MPa) | F23  (MPa) |
| Infused Hexcel Im7 Yarn | 3546 | 2754 | 116.7 | 233.4 | 116.7 | 116.7 |
| Gurit Prime 20LV Epoxy | 73 | 146 |  |  |  |  |

Table 6-2 Strengths of material constituents.

## 6.4 Mesh Sensitivity Study

Optimisation requires a compromise between finite element solution accuracy and analysis time. If the time for each weave to be analysed is too long then the overall optimisation time, which will use many individual analyses, will be too long to be useful to those wanting to perform optimisations. A mesh sensitivity study using models with no weft crossover was conducted to find the most appropriate for optimisation. The criteria was primarily the stiffness response then the time taken for the model to run was a secondary consideration. The results from the mesh sensitivity have been balanced against the time taken for the analysis.

A picture containing diagram

Description automatically generated

Figure 6-5 TexGen model with no crossover used for the tensile pull-off mesh sensitivity.

For the rectangular voxel mesh, each element was kept to the same aspect ratio so the only parameter is element size which can be reformulated as the number of voxels in the model. For the octree voxel mesh, there are three parameters: the number of voxels, the number of smoothing iterations to the surface element nodes and the level of refinement. This was used to find the correct balance between the three different parameters. Each mesh parameter has a different effect on the length of time it takes to generate the mesh and also the length of the analysis time, so these were both considered within the study.

For the mesh sensitivity study, each mesh was run using the boundary conditions set out earlier. Both the time taken and the force-displacement behaviour were plotted. Each weave was labelled according to the number of iterations of smoothing, the number of elements and the level of maximum refinement.



A blue line graph with numbers

Description automatically generated

Figure 6-6 Force-displacement graphs with meshes with refinement level 2 and 20 elements in each direction in plane of T-Joint profile. Graph 2 has the additional data set of the 20\_2\_r2 mesh to allow visibility of other data sets.

The smoothed meshes all had the same number of elements. It can be seen in Figure 6-6 that as the number of smoothing iterations increases the oscillations in the load-displacement response increases. This is because of the effect from the increasingly poorly shaped elements produced by successive smoothing iterations. The effect can be seen starting from the models with smoothing levels over 10, so the number of smoothing iterations was set at 5.

For models with varying levels of refinement meshes with maximum refinement levels of 1, 2 and 3 were run without any smoothing iterations. Further refinements led to poor mesh construction due to the number of elements. This produced the following graphs in figure 6-6.

A graph with a red line

Description automatically generated

Figure 6-7 Tensile pull-off force-displacement graph for maximum refinments 1, 2 and 3.

A graph showing a line graph

Description automatically generated with medium confidenceRefinement causes the number of elements to increase because of the subdivision of elements close to material boundaries. As can be seen the increase in refinement level between level 1 and 2 produced very similar graphs. As the refinement causes the cross-section shape to be better captured, refinement level 2 was judged to be the most appropriate.

Figure 6-8 T-Joint pull-off force-displacement graphs for models with 5 smoothing iterations, refinement level 2 and varying numbers of elements in the x-y plane.

As can be seen from figure 6-8, increasing the number of elements beyond 20 did not yield any difference in the force-displacement response. However, as the number of elements increases there is an increase in the oscillation in the result. This is caused by noise from the explicit procedure. Since further increasing the number of elements only produced extra noise without altering the stiffness response, the number of elements was set to 20.

The final parameters chosen for the octree voxel mesh were 20 elements in the x-y plane, smoothed with 5 iterations of the smoothing algorithm and refined to a maximum level of 2. This mesh took 20 minutes to run on a standard desktop PC with 4 cores. This is well within a reasonable time frame for optimisation.

## 6.5 Selecting an Objective Function Value

One of the main purposes of this work is to use the ability to automatically generate T-Joint designs to optimise the weaving pattern. This requires the selection of an objective function to minimise. One of the main design considerations is the failure resistance of the composite joints under tensile pull-off load. The usual finite element methods to characterise the failure behaviour of these joints is a full-scale failure analysis where the failure initiation and stiffness degradation of the joints is modelled using user subroutine [23]and sometimes cohesive modelling of interface damage[34]. While this can reproduce high model fidelity, the use of the subroutines add significant computational cost as the subroutine is called after every increment to re-evaluate the stress state. This introduces a barrier to the use of this sort of failure modelling in an optimisation routine where hundreds of designs need to be analysed.

To compare and score the different joints, the finite element models were run using perfectly bonded elements (ie. No cohesive surfaces) and with no user supplied material models in the form of subroutines. Once the models are run, for each frame a Python script was developed that iterated through each element and evaluated its stress state which was then compared to Hashin’s failure criteria [96] for the yarn elements and the pressurised Von Mises criteria for the elements in the matrix. This is to be able to determine the initial failure load which will form the basis of comparison between the different T-Joints in an optimisation. It has the advantage of being able to determine initial failure without the cumbersome cost from running user subroutines at every iteration in the finite element analysis.

### 6.5.1 Determining Initial Failure

To determine the point of initial failure, several methods were considered including checking surrounding elements of those failed to build up a chain of localised element failure that would stop once the chain reached a certain size to simulate crack growth. This was attempted but ultimately took too long for an optimisation process. This would be an interesting avenue for further research. The decided upon method determined that initial failure occurs when a failure threshold was met. This was set so that the global number of failed elements should not exceed 2% of the total elements in the mesh. This has the advantage of not only being significantly faster to compute but also being mesh size independent.

Without stiffness degradation upon failure, each of the elements will continue to be able to support local stress to the same level, causing an increasing error in the following solution. It was chosen therefore to make an early cut-off. Therefore, elements that have failed according to the failure criteria will cause the load-displacement response to deviate immediately from the true behaviour. Experimental validation of the finite element analysis was again not possible due to the Covid pandemic and the difficulty in obtaining woven samples which is discussed in more detail in Chapter 3.

### 6.5.2 Failure Criteria

The yarns are treated as homogeneous unidirectional plies with transverse isotropy. This allows them to be treated like any other unidirectional composite in terms of their failure behaviour. From the numerous available failure criteria that have been postulated, none has been shown able to predict mechanical failure onset for any 3D loading condition [97], [98] with the top ranked criteria only capable of predicting failure under two dimensional stress-strain. This question of the best failure criteria to use represents an entire field to itself, where there is no clear answer. Therefore, until a more complete theory is produced and verified, it is left to the user to make their own selection. In this case, Hashin’s failure criteria was chosen because it uses four criteria to predict failure: both transverse and longitudinal (in the fibre direction) failure in the yarns by compressive or tensile stress. These are given in equation below.

I1 ≥ 0

I1 ≤ 0

I2 ≥ 0 (3)

I2 ≤ 0 (4)

(5)

Where the Ii are the four stress invariants for which I ≥ 0 is for tensile stress and I ≤ 0 is for compression. The Fi are the tensile and compressive strengths longitudinal and transverse to the fibre direction. The di are the damage parameters for longitudinal tension and compression and transverse tension and compression respectively. Generally for the infused yarns, longitudinal failure indicates fibre failure while transverse failure indicates failure of the intra-fibre matrix within the yarn.

The matrix elements are homogeneous and isotropic, the pressurised Von Mises criteria was used [99].

(6)

Where dm is the matrix damage parameter and the Fm are the tensile and compression damage parameters.

## 6.6 Conclusions

A modelling schema for evaluating the fitness or the ability of the T-joint weaves under tensile pull-off load was set out to find the initial failure load for use as the objective function value in the optimisation in the next chapter. Abaqus explicit was used to achieve a faster analysis time. The time control was selected after setting a lower bound in accordance with the Abaqus user manual recommendation.

Mesh sensitivity studies for both the rectangular voxel mesh and octree voxel mesh were conducted to find a mesh that balances the accuracy required against the time taken for each analysis. This brings a subjectivity to the choice of which mesh to use as the priority between analysis time and accuracy is up to the modeller. For rectangular voxel meshes the main parameter is the number of elements as each element has the same aspect ratio. For the octree voxel mesh, there are additional parameters to be considered when conducting mesh sensitivity studies, namely the number of refinements at the material interfaces and the number of smoothing algorithm iterations. An octree voxel mesh was chosen with 5 smoothing iterations and a maximum refinement level of 2. This mesh conformed well to the yarn cross-section shape without causing over distortions to the mesh elements.

The location of failure indicated by the presence of failed elements corresponded to literature experimental data indicating that elements primarily failed along yarn/matrix interfaces within the noodle region of the T-joint.