



All Theses and Dissertations

2012-04-12

Method for the Simulated Layup of Composite Fabrics

David M. Christensen

Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>

 Part of the [Mechanical Engineering Commons](#)

BYU ScholarsArchive Citation

Christensen, David M., "Method for the Simulated Layup of Composite Fabrics" (2012). *All Theses and Dissertations*. 3559.
<https://scholarsarchive.byu.edu/etd/3559>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Method for the Simulated Layup of Composite Fabrics

David Christensen

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

C. Greg Jensen, Chair
Christopher A. Mattson
David T. Fullwood

Department of Mechanical Engineering

Brigham Young University

June 2012

Copyright © 2012 David Christensen

All Rights Reserved

ABSTRACT

Method for the Simulated Layup of Composite Fabrics

David Christensen
Department of Mechanical Engineering, BYU
Master of Science

Due to the complexity of designing advanced composite parts, many software tools have been developed to aid the designer and reduce design cycle time. Draping is one of those tools and is used to predict the fiber angles throughout the part. This application of draping is to simulate the actual hand layup process that a technician would go through while creating a multi-layered laminate composite part. This method is the first to use plies as an underlying surface for draping instead of just using an offset mold surface. This method can visualize full 3D ply geometry taking into account the thicknesses of the underlying plies and their drop-off regions which a designer could use to avoid superimposing ply drop-offs. Physical testing showed that this method predicts ply drop-off regions reasonably well and allows the designer to visualize the final shape of the laminate. The method also provides for re-ordering of the plies while keeping their cut-out shape the same because of its reverse-process draping technique.

Three methods of draping were explored in order to find the best method. Multiple test parts were created with specific features that are difficult to drape. The method to drop fabric to the surface was the most versatile while a method published by Wang was the best for convex surfaces and superior to spread-type draping. No one method worked well for all surfaces.

Keywords: draping, composites, simulated layup

ACKNOWLEDGEMENTS

I would like to thank my committee, Dr. Greg Jensen, Dr. Fullwood, and Dr. Mattson. They have been a great help in getting this thesis published. Also, thank you to Ryan Cox, Jim Roach, and Ammon Hepworth at Pratt & Whitney for the technical input they have given and Pratt & Whitney for funding this research. A big thank you to my wife, Amanda, for being patient while I finish school and this thesis.

TABLE OF CONTENTS

LIST OF TABLES ix

LIST OF FIGURES xi

1 Introduction 1

1.1 Problem Overview 1

1.2 Thesis Objective 2

1.3 Problem Delimitation..... 3

1.4 Thesis Organization 3

2 Background 5

2.1 Composites Primer..... 5

2.1.1 Relevant Material Types 5

2.1.2 Relevant Manufacturing Processes 5

2.1.3 Laminates 7

2.2 Current Design Tools 8

2.3 Tessellated Surfaces 9

2.4 Draping 10

2.4.1 Kinematic Approach 10

2.4.2 FEA Approach 11

2.4.3 Mixed Approach 12

2.4.4 Ply Deformation Methods 12

2.4.5 Material Property Determination 13

3 Method 17

3.1 Creation of the Flat Pattern 17

3.1.1 Input Parameters 17

3.1.2	Ply Structure.....	18
3.2	Draping Method.....	21
3.2.1	Warp and Weft Elements.....	21
3.2.2	Shear Elements.....	23
3.2.3	Bend Elements.....	24
3.2.4	Modeling Fabric Thickness.....	25
3.2.5	Optimization Method.....	26
3.2.6	Drop Draping.....	28
3.2.7	Spread Draping.....	28
3.2.8	Tendon and Diagonal Node Mapping.....	28
3.3	Method for Stacking Plies.....	29
4	Implementation.....	31
4.1	Ply Creation.....	31
4.1.1	User Input.....	32
4.1.2	Flat Pattern Curves Bounding Box and Data Structure Allocation.....	33
4.1.3	Node Grid Bounding Box.....	34
4.1.4	Determining Inner Nodes.....	34
4.1.5	Extra Outer Layer.....	38
4.1.6	Element Creation.....	39
4.1.7	Warp and Weft Element Strings.....	41
4.2	Draping Method.....	41
4.2.1	Force Calculation.....	42
4.2.2	Differences Between Optimization Methods.....	44
4.3	Ply Cutting Method.....	45
4.3.1	Triangulating the Flat Pattern.....	45

4.3.2	Removing Extra Fabric	46
4.3.3	Translation to Three Dimensions	46
5	Results	49
5.1	Physical Testing	49
5.1.1	Fabrication Process	49
5.1.2	Comparison of Results	52
5.2	Software Testing	56
5.2.1	Test Parts	56
5.2.2	Results Explanation	57
6	Conclusions	61
6.1	Recommendations for Future Work	62
6.1.1	Fabrication and Measuring Process	63
6.1.2	Friction	63
6.1.3	User Defined Tow Placement	63
6.1.4	Fiber Slippage	64
	REFERENCES.....	65

LIST OF TABLES

Table 1: Hemisphere Arc Length Error	53
Table 2: Hood Error	53
Table 3: Software Testing Results	57

LIST OF FIGURES

Figure 2-1:Resin Transfer Molding	6
Figure 2-2: Vacuum Bagging.....	7
Figure 2-3: Tessellated Surface	10
Figure 2-4: Fabric Shear	13
Figure 2-5: Picture Frame Test [20].....	14
Figure 2-6: Bias Extension Test [23]	15
Figure 2-7: FEA Unit Cell [20].....	15
Figure 3-1: Nodes Covering Flat Pattern	20
Figure 3-2: Warp/Weft Element Force	22
Figure 3-3: Locked Unit Cell Showing Lengths of Shear Elements	23
Figure 3-4: Bend Element Geometry	25
Figure 3-5: Surface Force	26
Figure 3-6: Step Size Reduction	27
Figure 3-7: After Tendon (a) and Diagonal (b) Node Mapping[17].....	29
Figure 4-1: Ply Parameters Input Dialog	32
Figure 4-2: Drape Method Selection Dialog.....	33
Figure 4-3: Flat Pattern Bounding Box Creation.....	34
Figure 4-4: Flat Pattern Showing Intervals.....	37
Figure 4-5: Flat Pattern Filled with Nodes	38
Figure 4-6: Nodes Overlapping Flat Pattern.....	39
Figure 5-1: 6 Inch Hemisphere and Scale Dodge Viper Hood Molds	50
Figure 5-2: Finished Parts.....	51
Figure 5-3: Ply Drop-off	51

Figure 5-4: Measurement Techniques	52
Figure 5-5: Ply Almost Parallel to Cross-Section.....	54
Figure 5-6: Test Parts.....	56
Figure 5-7: Shifted Drop Drape	58

1 INTRODUCTION

“Composite materials are used in a wide range of applications in structural design due to their superior mechanical properties over conventional materials. For example, in the aerospace industry, structures are usually built from materials with high strength and stiffness to weight ratios, and composite materials often serve these characteristics adequately. However, due to the complex behavior of composite materials, the design of a structural component using them is more complicated than one using conventional material.”[1] This design complexity has created a need for engineering tools to speed the design process.

1.1 Problem Overview

Due to the complexity of composite parts, there are many problems with accurately representing them in a virtual environment. Two of these problems are finding the correct fiber angles when a ply is draped on a mold and determining the location and geometry of ply drop-offs.

The first problem is commonly approached using a method called draping. Draping tries to predict and model the final shape and fiber angles of the fabric. In composite part manufacturing two relevant processes are vacuum bagging and resin transfer molding. In vacuum bagging and resin transfer molding, composite fabric plies are placed on the mold surface in layers. Each layer is made of some kind of fiber surrounded by a matrix material.

Fibers are usually made of fiberglass or carbon fiber and, in rarer cases, aramid. The matrix material is usually some type of polymer like polyester or epoxy. Vacuum bagging and resin transfer molding can take almost any shape. When each layer is placed in the mold, the fabric deforms to conform to the curvature of the mold. If the mold has curvature in two directions, it causes the fibers in the fabric to rotate relative to each other in the plane of the fabric which is called fabric shear. This deformation can lead to bunching and wrinkling of the fabric, bridging of the fabric, and due to the change in fiber directions, changes in the strength properties of the final laminate can occur which is one of the reasons draping exists. Due to the complexity of designing composites, many computer applications have been created to predict how a fabric will drape [2].

Ply drop-offs occur when a larger layer of fabric is placed over a smaller layer. Drop-offs are areas of stress concentration in the final part due to the free edge of the fabric. All of the algorithms to date have represented the fabric as a surface, neglecting the thickness of the fabric. The draping method in this thesis models the thickness of the fabric which allows for predicting the correct shear after a ply drop-off.

1.2 Thesis Objective

The objective of this thesis is to explain a method that has been created to simulate the draping of a fabric on a mold surface while taking into account the thickness of the fabric. The method specifically will:

- Create a representation of a fabric that incorporates both thickness and cut-out shape
- Predict the fiber angles throughout the part
- Develop true ply geometry including ply drop-offs

This method is meant to be a quick check for designers to make sure their part is able to be draped without wrinkles and to avoid superimposing ply drop-offs.

1.3 Problem Delimitation

This method is not meant as a precise analysis tool but as a quick approximation for use during the design phase of a product. This method does not include an external knowledge-system incorporating design rules. It also does not include a method to create an FEA model from the drape or a way to interactively change the ply boundaries once they are created. These are suggested as topics of future research.

1.4 Thesis Organization

Chapter 2 will give a brief background on composite materials, current design tools and a literature review on some draping methods and other topics relevant to this method. Chapter 3 discusses the new draping method generally while Chapter 4 goes through the implementation of this method. Chapter 5 explains the results from this method and results of testing. The sixth chapter discusses the conclusions and future research of this project.

2 BACKGROUND

2.1 Composites Primer

Composite design and analysis is much more complex than design using homogeneous materials. Despite the complexity, composites offer certain advantages over homogeneous materials. The main advantage of using composites over homogeneous materials is a higher strength to weight ratio according to Strong [3]. This has led to heavy use of advanced composites in the aerospace industry.

2.1.1 Relevant Material Types

Composites come in many forms but this thesis refers to continuous fiber reinforced composites. The fibers in these composites are usually a fabric or unidirectional material and can be either pre-impregnated called “pre-preg” or with the matrix material or dry without any matrix. Both pre-preg and dry fabrics can be modeled by the method found in this thesis. Unidirectional fabric does not drape well and is not considered.

2.1.2 Relevant Manufacturing Processes

Resin transfer molding and vacuum bagging are two processes that are relevant to this thesis. In resin transfer molding, a mold cavity is filled with dry woven and/or unidirectional

material in layers. The mold is closed, resin and a catalyst are mixed and injected through ports in the mold (See Figure 2-1:Resin Transfer Molding).

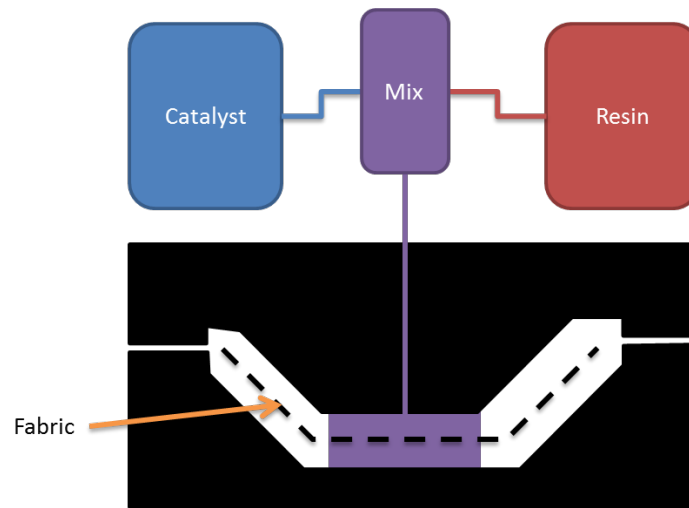


Figure 2-1:Resin Transfer Molding

In vacuum bagging only one side of the mold is needed. Either dry fabric is impregnated and then placed on the mold or pre-impregnated material is laid down and a bag is placed around the part. A vacuum is attached which removes the air and excess resin from the part which applies atmospheric pressure to the bag side of the mold (see Figure 2-2: Vacuum Bagging). Resin transfer molding has excellent wet-out which is a measure of how well the fibers are impregnated with resin. Good wet-out decreases the amount of voids found in the part which increases its strength [3]. Vacuum bagging generally has more voids than resin transfer molding, but the amount of voids can be decreased using pre-impregnated material.

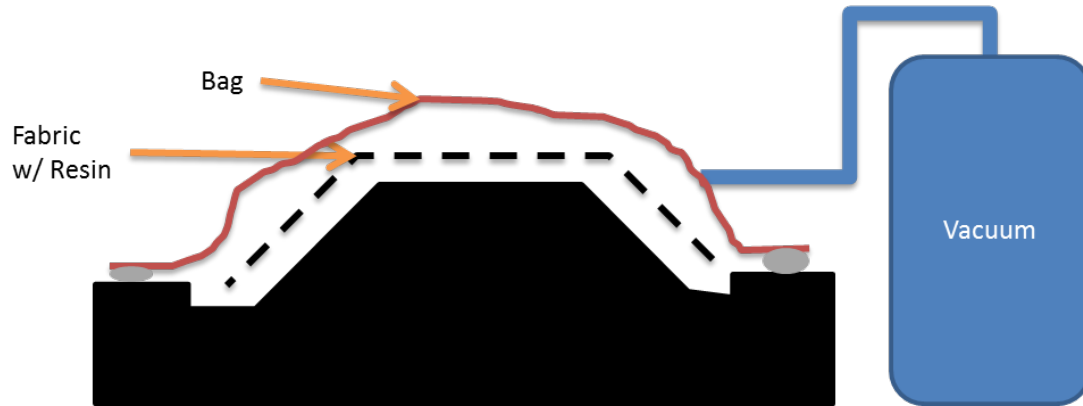


Figure 2-2: Vacuum Bagging

2.1.3 Laminates

One layer of fabric would have insufficient strength in most cases so most advanced composites are created using multiple layers laid on top of each other in a laminate. The designer's main goal is to define a ply. A ply is a layer of fabric or unidirectional material that has a specified cut-out shape and angle relative to the laminate coordinate system. A designer can choose an orientation for each ply thus optimizing the strength properties of the laminate by putting more of the fibers in the load direction and less in the other directions. This is one reason composites are lighter than isotropic materials.

Both vacuum bagging and resin transfer molding build up a laminate by laying plies on top of plies over a mold surface. The method for simulated layup found in this thesis will show what each ply looks like when laid down along with its corresponding fiber angles. Only fabric plies are considered for this thesis; not unidirectional material.

2.2 Current Design Tools

Current laminate design systems such as PACKS (developed at Boeing, purchased by Unigraphics), SCADS (developed at the University of Kansas), AUTOLAY and others have already been developed because of the complexity of modern composites [1], [2], [4], [5]. They have shown significantly decreased design cycle time and cost savings of up to 60% while designing composite parts [2].

Typically, “a designer focuses his or her energy on manipulating lines in space, with the intention of eventually describing the concept of a ply. Unfortunately, the actual details of the ply have historically been neglected. Specifically, the true 3D geometry of the ply is not considered, merely an approximation of the geometry. Generally, that approximation consists of wireframe ply boundaries projected to the original tool surface, neglecting the thickness effect of adjacent plies or thickness transition details at a ramp between two constant thickness regions,” [2]. In the literature review for this research, not other method was found that modeled the thickness effects of the ply drop-offs. In the same paper Hale describes how composite parts are frequently divided into zones and each zone is defined independently which causes problems due to incompatible stacking sequences and ply drop-offs. PACKS tries to mitigate those problems by using a global database for the whole part and incorporates design rules and knowledge while still keeping the part divided into zones.

Hepworth et al. describe a method for automatically creating ply geometry [6]. This paper takes a step away from the zone based definition which eases the problems associated with designing in zones. In another paper, Hepworth et al. describe an optimization approach to designing a laminate using a global ply table and a mid-surface representation of the laminate [7]. Elements are defined using the global ply table adding plies by priority and the thickness of

the part at the center of each element. While this is a quick way of producing an approximation, it does not take into account the actual ply geometry. Neither paper by Hepworth takes into account the local fiber angles of the ply due to draping. The method in this thesis takes into account ply thicknesses unlike current design tools. The lack of a zone based definition allows for a more accurate representation of plies including the drop-off zone.

2.3 Tessellated Surfaces

Tessellated surfaces are surfaces made up of many individual triangle facets. This type of surface is used in computer graphics extensively and is the basis for displaying almost all interactive 3d models on a computer. The advantages of tessellated surfaces over other types of surfaces such as NURBS surfaces are:

- They can take any shape
- Accelerated graphics cards can display them
- Local operations can be performed on them such as removing a face or filling a hole
- Parts of the surface can be changed without affecting other parts of the surface
- CAD systems already create them to display the part on the screen as shown in

Figure 2-3: Tessellated Surface

There are some disadvantages to tessellated surfaces also:

- They are not continuous
- They require more data
- They are not parameterized like NURBS

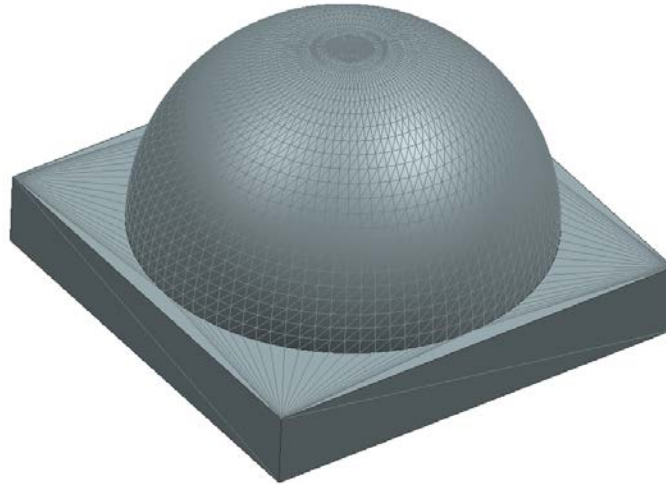


Figure 2-3: Tessellated Surface

The implementation of this method uses a tessellated surface representation because of the advantages listed above – in particular the advantage of taking on any shape. With current computer speeds and memory capacities, the larger amount of data required is becoming less important.

2.4 Draping

Draping is a field of research that predicts the fiber directions of a fabric after it is placed on the mold surface. There are three main methods to calculate the draped ply: the kinematic approach, the FEA approach, and the mixed approach. These approaches are explained as well as the methods of fabric deformation and the way material properties are determined for a composite fabric which is necessary for the FEA and mixed approaches.

2.4.1 Kinematic Approach

The first way to calculate a draped ply is called the kinematic approach. The kinematic approach builds a pin-jointed net. First, the fabric is built outward by specifying an initial

application point and direction. Next, geodesic curves are created with a specified length (usually the tow spacing) in the initial direction and the perpendicular direction. Finally, the corner nodes to the unit cells are found by determining the geodesic curves whose endpoints are coincident and whose length is equal to the tow spacing[8]. This approach does not take into account any material properties of the fabric except the tow spacing. The kinematic approach assumes that the tows are inextensible and the shear deformation is solely a geometric phenomenon and has nothing to do with the fabric properties.

2.4.2 FEA Approach

The second method uses finite elements to model the fabric, the tooling, and the interaction between the two[9–16]. Multiple draping elements have been developed for different FEA systems in both triangular and quadrilateral form. Early work into the FEA approach was done using bar elements and membrane elements by Cherouat [10]. Dong [11] also used membrane elements but with an updated material law due to the coordinate system of the element changing during the simulation. Boisse [9] used a finite element approach to model both the shear and extension properties of the fabric. These properties were determined by modeling the unit cell of the fabric in FEA and comparing to physical tests. A second paper by Boisse [14] shows the difference between having shear properties in the finite element versus neglecting shear by doing a physical forming test on a hemisphere. It was shown that when including shear of the fabric, wrinkles could be predicted. Further work by Hamila and Boisse [15] created a triangular shell finite element to predict drape and wrinkles with only displacement degrees of freedom. They also determined that in-plane shear stiffness is mainly important for predicting wrinkle initiation and that bending stiffness determines the shape of those wrinkles.

Another triangular shell element was created by Thije [13] which takes into account inter-ply friction due to multiple layers of fabric being formed at the same time. Skordos [12] showed that shear rate has a significant effect on the final fabric shear for pre-preg materials. Skordos also applies friction between the fabric and the mold surface during forming. The FEA approach is the most accurate, but is also the slowest and requires FEA programs.

2.4.3 Mixed Approach

The third method mixes the kinematic and FEA methods and provides a compromise between the two [17], [18]. The surface is discretized, and the kinematic approach is used to provide initial node locations. The nodes are connected by spring elements along the fiber directions and on the diagonals to mimic the shear resistance of the fabric. The strain energy is calculated from the displacement of the spring elements and is minimized using an optimization routine. The boundary conditions caused by the tooling in this type of simulation are usually ignored.

2.4.4 Ply Deformation Methods

Most of the research on draping and ply deformation methods has been performed on fabrics due to the ease with which they can be formed on a surface. Unidirectional materials in general are harder to form on a doubly curved surface.

The main deformation methods of fabric are fabric shear, fiber extension, fiber buckling, and slippage of fiber bundles [19]. Of these methods of deformation, the most studied is fabric shear because it is the largest deformation method and has the most impact on draping. Fabric shear is shown in Figure 2-4.

Fabric shear is resisted at first by friction between the tows. This friction is low in most fabrics. Once a fabric reaches its shear locking angle, the shear is also resisted by interference between and compaction of the tows. The shear locking angle is the most important material property in a draping simulation. Once the shear locking angle is reached, the resistance to fabric shear increases rapidly and the fabric can exhibit out of plane deformation causing wrinkles. Wrinkles decrease strength by introducing voids in the final part.

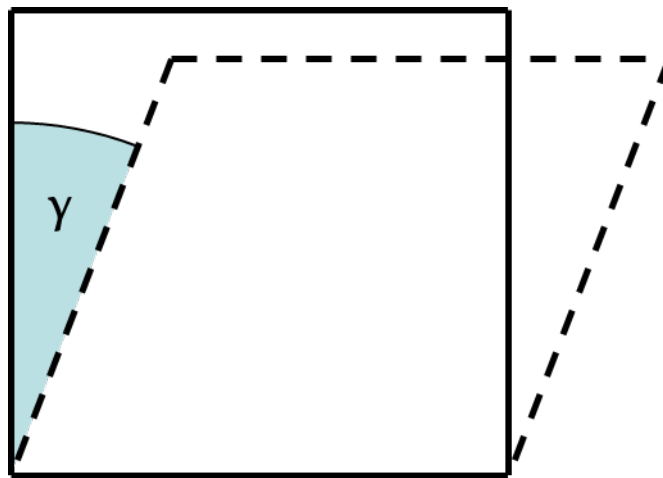


Figure 2-4: Fabric Shear

2.4.5 Material Property Determination

There are three accepted ways of characterizing the shear behavior of the fabric: the picture frame test, the bias extension test, and modeling with FEA. The picture frame test is performed using a rig that clamps the four sides of a small square of composite fabric (see Figure 2-5). There are pin joints on each corner to allow the rig to shear with the fabric. The rig is pulled from opposite corners in a tensile tester and the force/strain curve is recorded. The picture frame

test is very sensitive to boundary conditions of the fabric. If the fabric is not perfectly aligned, it can stretch or relax the fibers which decreases the accuracy of the test.

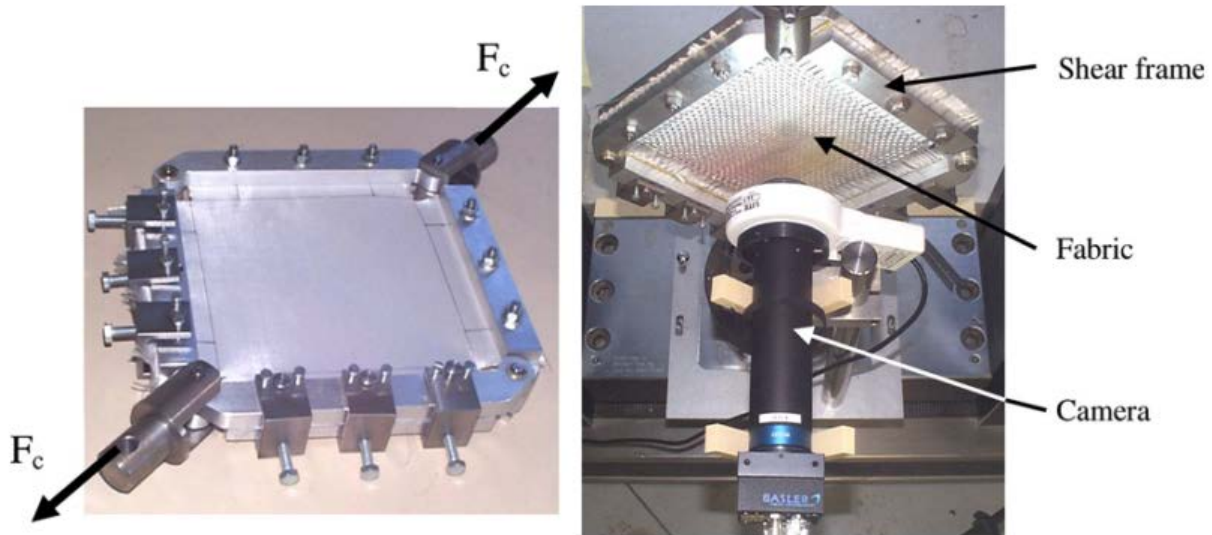


Figure 2-5: Picture Frame Test [20]

The bias extension test takes a sample of fabric that has been cut in the bias direction (45°) and is usually at least twice as long as it is wide. Clamps are placed on each short edge of the fabric and the sample is pulled in the tensile tester (see Figure 2-6). The force/strain is measured again and the data is normalized to determine the shear properties of the fabric. An approach has also been proposed by Harrison [21] that combines the picture frame and bias extension tests. A variation on the bias extension test, called the wide-strip bias extension test, has been published by Potluri [22]. Angles of the fibers were measured using both a camera and using an adjusted global strain method.

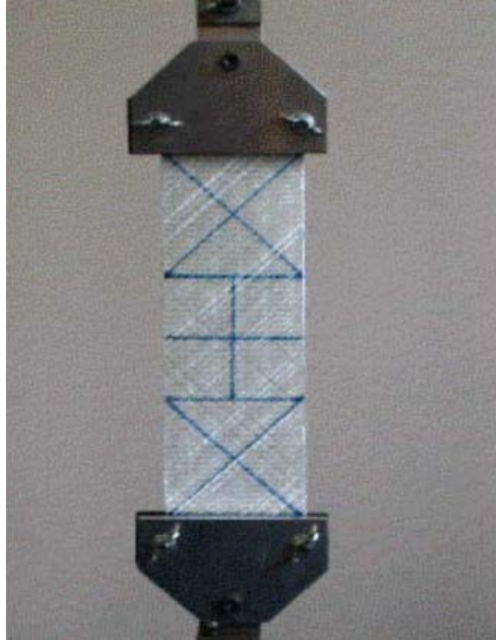


Figure 2-6: Bias Extension Test [23]

The wide-strip bias extension test was shown to agree closely between the camera angles and the adjusted global strain method up to about 50° shear angle. After 50° , the fabric began to slip and results were inaccurate. The wide strip bias extension test has the benefit of being insensitive to boundary conditions while using a square pure-shear region of the fabric like the picture frame test.

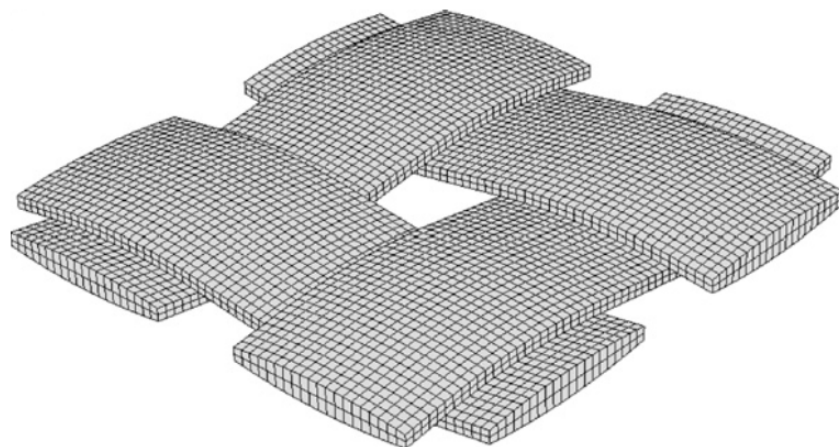


Figure 2-7: FEA Unit Cell [20]

The FEA method models one unit cell of the fabric using solid elements (see Figure 2-7). The model is subjected to shear loads and the force/strain curve is determined. FEA models avoid the need for expensive equipment and have good agreement with experimental testing according to Badel [20].

A parametric model to characterize the shear behavior of glass woven reinforcements was created by Lomov [24]. This allows for material property determination based solely on measurements of the fabric. While this is a good method for the type of fabric used by Lomov, more research would be needed before all fabric could be modeled using that method. The main way to determine shear properties of fabric is still bias extension testing and picture frame testing. Bias extension testing was used in this implementation because a tensile tester was available and special hardware is not required.

Draping is the main component used in this method. It is the part of the method that predicts the fabric boundaries and fiber angles. While current methods of draping ignore the thickness of the fabric, this method does not and allows for prediction of ply drop-off effects.

3 METHOD

This chapter discusses the simulated layup method. The simulated layup method includes creation of the flat pattern, the draping method with thickness modeling, and the method for stacking multiple plies.

3.1 Creation of the Flat Pattern

This section explains the inputs needed for the creation of a ply, the structure of a ply, and the method of creating the structure.

3.1.1 Input Parameters

Each ply in a layup has a specific shape which is called the flat pattern. To fully define a ply, the following data is needed:

- A flat pattern outline
- An application point on the ply
- An orientation angle for the warp direction of the fabric
- A shear locking angle
- A cell size

The flat pattern is made up of lines, arcs, splines or any mixture of the three. The curves all must lie in a plane and must form closed loops. Cut-out holes can be made in the flat pattern by forming an inner loop as long as the inner loop is surrounded by the outer loop of the ply.

An application point represents the point on the flat pattern that will first contact the surface of the mold when it is draped. It can be represented as a point in the plane of the flat pattern. This becomes the origin of the ply.

The orientation angle represents the strong direction of the fabric or the direction of the fibers. Many fabrics have the same strength in the warp and weft directions and in that case it does not matter which fiber direction is used. The shear locking angle as explained in 2.4.4 is the point at which the resistance to fabric shear increases rapidly. It is a property specific to each type of cloth. Cell size is the distance from the middle of one tow to the middle of the neighboring tow. In balanced fabrics the distance between the tows in the warp direction and weft direction are the same. Two cell sizes would be needed for an unbalanced fabric. This method only used one cell size and can only be used in its current state for balanced fabrics but it could be easily expanded to use two cell sizes.

3.1.2 Ply Structure

Each ply is made of two main types of data: nodes and elements. Nodes are locations in space and are represented by points. They are located at the intersections between the tows of fabric. Nodes serve to define the ply shape in 3d space when draping. Each node is connected to surrounding nodes by elements.

Elements are represented as springs. In this method there are many types of springs. The types are listed next.

- Warp elements
- Weft elements
- Shear elements
- Bend elements

Warp elements connect two nodes in the warp fiber direction and represent the stiffness of the fabric in the warp direction. In general, the warp direction of a fabric is very stiff compared to the shear direction.

Weft elements connect two nodes in the weft fiber direction and represent the stiffness of the fabric in the weft direction. For a balanced fabric, warp elements and weft elements have the same spring constant.

Shear elements connect diagonal nodes in both directions and they represent the resistance of the fabric to shear. These elements have a much lower spring constant than the warp and weft elements and are responsible for most of the deformation of the fabric. When

Bend elements connect three nodes in a line along the warp, weft and both shear directions. They resist out of plane bending which makes the fabric resist folding over on itself.

To build the ply structure, a grid of nodes must first be created that completely surrounds the flat pattern. The bounding box of the flat pattern curves is determined. The number of nodes in the node grid x direction is

$$N_x = \text{trunc} \left(\frac{X_{\max}^{bbox} - X_{\min}^{bbox}}{l_{cell}} \right) + 6 \quad (3-1)$$

where N_x is the number of nodes in the node grid x direction, X_{\min}^{bbox} is the minimum x value of the bounding box, X_{\max}^{bbox} is the maximum x value of the bounding box, and l_{cell} is the cell length.

This makes sure the node grid completely overlaps the flat pattern with at least one extra column of nodes on each side. The number of nodes in the y direction can be determined similarly. The center node index is

$$i_x^{center} = trunc\left(\frac{X^{origin} - X_{min}^{bbox}}{X_{max}^{bbox} - X_{min}^{bbox}} * (N_x - 6)\right) + 3 \quad (3-2)$$

where i_x^{center} is the index of the center node when the node indices start at 0 and X^{origin} is the x-coordinate of the ply application point. The y index of the center node can be found in a similar manner. The x-coordinate of any node can be found given its index i :

$$X_i = X^{origin} - l_{cell} * i_x^{center} + i * l_{cell} \quad (3-3)$$

Again, the y value of any node can be found in a similar manner.

Nodes are created on the grid, but only on the inside the flat pattern. Another layer of nodes is added so the flat pattern curves are completely covered so they can be used later to trim the fabric as shown in Figure 3-1. The elements are then created between nodes.

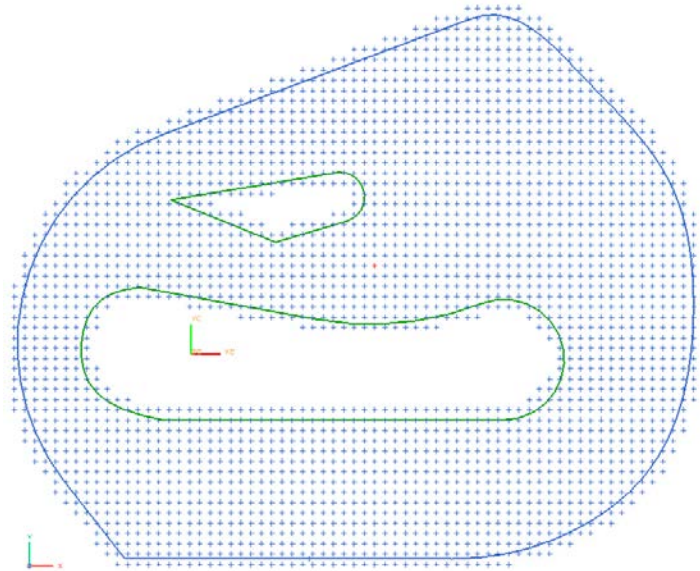


Figure 3-1: Nodes Covering Flat Pattern

3.2 Draping Method

The draping process is:

1. Create flat pattern and ply structure
2. Set initial location of nodes (two different methods are explored)
3. Calculate forces on nodes
4. Move nodes in direction of resultant force
5. Repeat 3 and 4 until optimum is reached

The fabric is modeled as a pin-jointed net. Some pin-jointed net models do not take into account any fabric properties. This method uses springs in the warp and weft directions to simulate the extension properties in their respective directions and springs on the diagonals to mimic the shear properties of the fabric. Bend elements prevent the fabric from folding over on itself and resists fabric bending. This thesis explores three draping methods: drop draping, spread draping, and a method found in Wang[17] called tendon node mapping and diagonal node mapping. All three have similar optimization methods, but different ways of applying it. This section includes a summary of each element type's behavior, the method for modeling fabric thickness, the optimization method, and drop, spread, tendon and diagonal node mapping types of draping.

3.2.1 Warp and Weft Elements

Warp and weft elements are connected in strings along fiber directions. The elements are modeled as a simple linear spring for simplicity, but non-linear springs could also be implemented. Forces on nodes from warp or weft elements are calculated from a string of warp or weft elements instead of just one at a time. The forces from each element in the string are

added together to find the total force. The force from a warp element to the right of a node is calculated by

$$\vec{F}_{warp}^{right} = \sum_{right} [k(l-l_0)] * \frac{\vec{p}_2 - \vec{p}_1}{\|\vec{p}_2 - \vec{p}_1\|} \quad (3-4)$$

where \vec{p}_1 is the three dimensional point of the node, \vec{p}_2 is the three dimensional point of the other node of the warp element, l is the length of the warp element, l_0 is the un-stretched length and k is the spring constant. The geometry is shown in Figure 3-2.

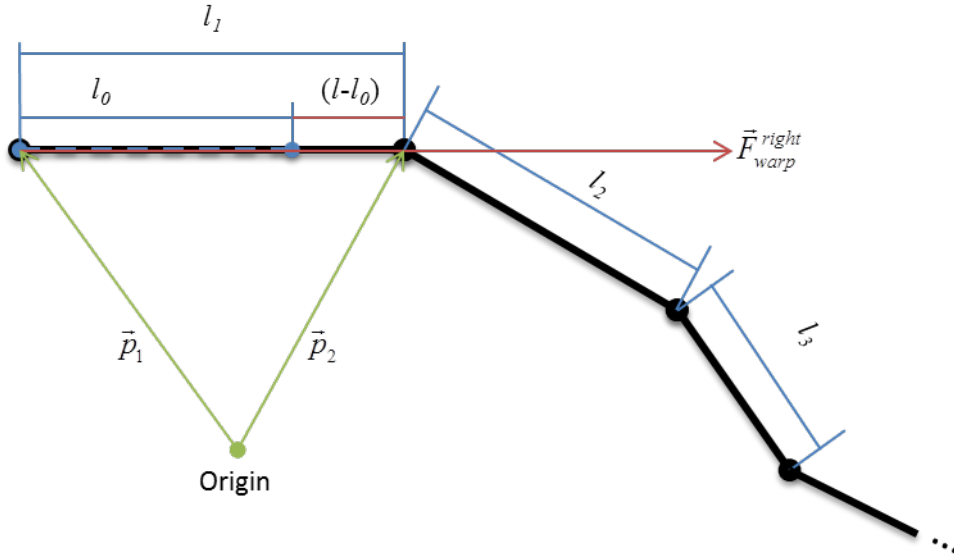


Figure 3-2: Warp/Weft Element Force

Basically, all tensile and compressive forces from warp elements on the right are summed as the magnitude of the force, and the direction is defined as $\frac{\vec{p}_2 - \vec{p}_1}{\|\vec{p}_2 - \vec{p}_1\|}$. Both the magnitude and the direction are applied to the node.

3.2.2 Shear Elements

There are two options for calculating shear force from a shear element. One option is to use a spring-based model and the other is to use a curve fit from the bias extension test.

For the first option, shear elements are modeled as linear springs for the first part of travel. The un-stretched length of the shear elements can be found using the Pythagorean Theorem

$$l_0 = \sqrt{l_{cell}^2 + l_{cell}^2} \quad (3-5)$$

The length at which the shear element becomes stiffer when compressed is

$$l_{short} = l_0 \sqrt{1 - \sin(\theta_{lock})} \quad (3-6)$$

and when stretched is

$$l_{long} = l_0 \sqrt{1 + \sin(\theta_{lock})} \quad (3-7)$$

where θ_{lock} is the shear locking angle. See Figure 3-3 for geometry.

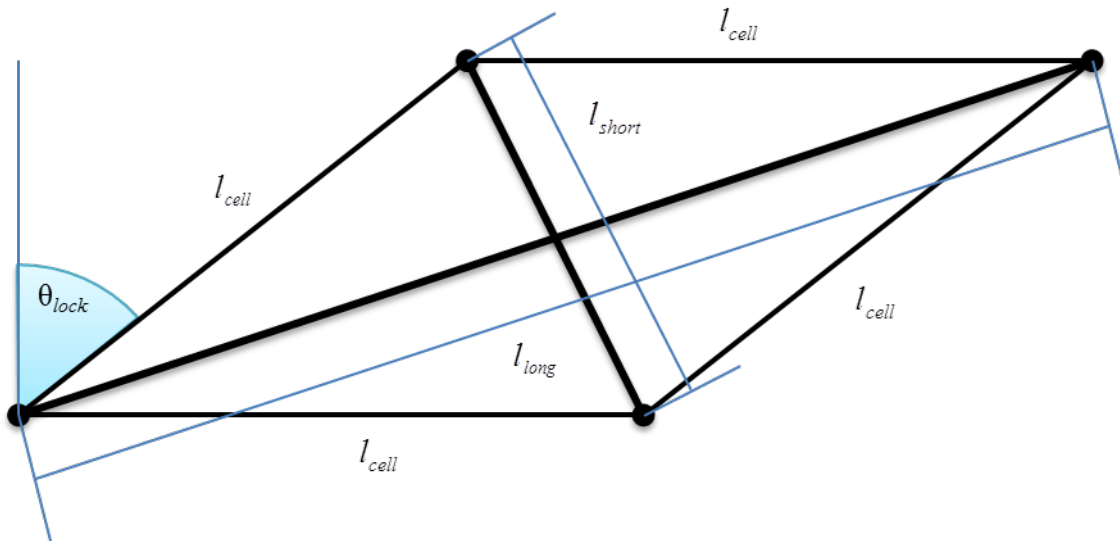


Figure 3-3: Locked Unit Cell Showing Lengths of Shear Elements

Once the shear locking angle is reached, another spring force is added to the node in parallel with the first spring along the same direction to model the increasing resistance to shear. The force magnitude is then

$$F_{shear} = \begin{cases} -k_1(l-l_0) - k_2(l-l_{short}) & \text{for } l < l_{short} \\ -k_1(l-l_0) & \text{for } l_{short} < l < l_{long} \\ -k_1(l-l_0) - k_2(l-l_{long}) & \text{for } l > l_{long} \end{cases} \quad (3-8)$$

where k_1 and k_2 are the spring constants. The direction of the force is towards the node along the shear element if in compression and away from the node if in tension.

For the second option, a polynomial can be fit to the normalized bias extension test force vs. deflection data. This gives a more accurate definition for the shear force. One thing to keep in mind while using a curve fit is to make sure the force continually increases outside the bounds of the bias extension test data. This is done by using an odd ordered polynomial.

3.2.3 Bend Elements

Bend elements represent the stiffness of the bundles of fibers and are modeled as torsional springs. Bend elements have three nodes that are collinear when the fabric is not deformed. When deformed, the fabric tows are no longer straight and the three nodes form an angle in 3d space. This is the deflection angle for the torsional spring. The moment that this bend element creates is

$$M_{bend} = k(\pi - \theta) \quad (3-9)$$

where θ is the angle between the nodes and k is the spring constant. Since the nodes have only position in space, the moment is converted into two couple moments with the forces at the nodes as shown in Figure 3-4. The forces are perpendicular to the line between the outer nodes and the middle node and in the plane formed by the three nodes. To prevent the fabric from folding over

on itself, extra stiffness can be added to the bend element after a certain angle such as $\frac{\pi}{2}$ in a similar manner as was done with the shear elements.

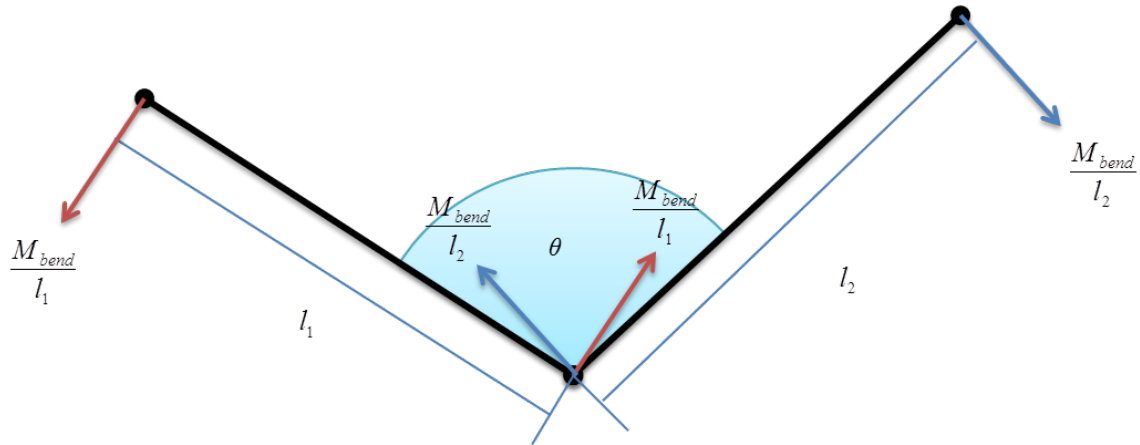


Figure 3-4: Bend Element Geometry

3.2.4 Modeling Fabric Thickness

The fabric thickness is modeled also using a spring type analogy. The difference is that the spring is connected from a node to the closest point on the surface of the layup. The direction of the force is along the line from the node to the closest point on the surface. If the node is closer than the thickness of the cloth to the surface, the sign of the force is changed and the node is pushed away from the surface.

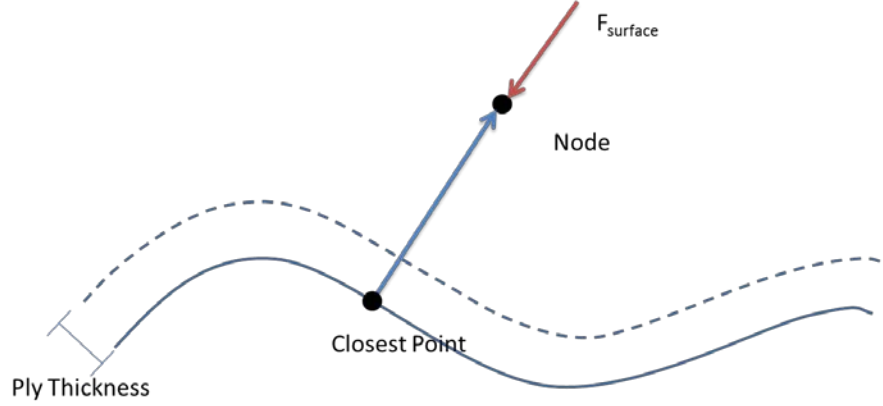


Figure 3-5: Surface Force

3.2.5 Optimization Method

A diffusion type optimization method was employed similar to the one found in Wang's paper [17]. The force vectors on each node are calculated from the elements connected to it and summed. The nodes are then moved a certain amount in the direction of the resultant force. The maximum distance allowed each step was limited to half the cell size to prevent the unit cells from crossing over each other. The step size is

$$d_{step} = d_{max} \left(1 - \frac{c}{c + \|\vec{F}\|} \right) \quad (3-10)$$

where d_{step} is the step size taken, d_{max} is the allowable step size, $\|\vec{F}\|$ is the magnitude of the force resultant, and c is an arbitrary constant on the order of $\|\vec{F}\|$. The step is taken in the direction of the resultant force. An addition to the optimization method employed by Wang was to decrease the maximum allowable step size as the optimization progressed which helped speed up convergence and made the value of c less critical. The down side is that to make sure there

were enough iterations in the optimization, extra iterations had to be run to make sure the optimization did not stop before the nodes were in the correct place. The fraction of the maximum step size at the n^{th} step in the optimization was

$$\alpha = \left(1 - \frac{n}{N}\right)^\beta \quad (3-11)$$

where α is the fraction, n is the step number, N is the maximum number of steps, and β is a constant greater than 0. A value of 1 for β will make the step size decrease linearly, higher than 1 will have more time for fine adjustment and less than 1 will have more time for coarse adjustment (see Figure 3-6). The optimization was run a set number of times instead of stopping when the strain energy was minimized because controlling the step size prevented the unit cells of elements from getting twisted.

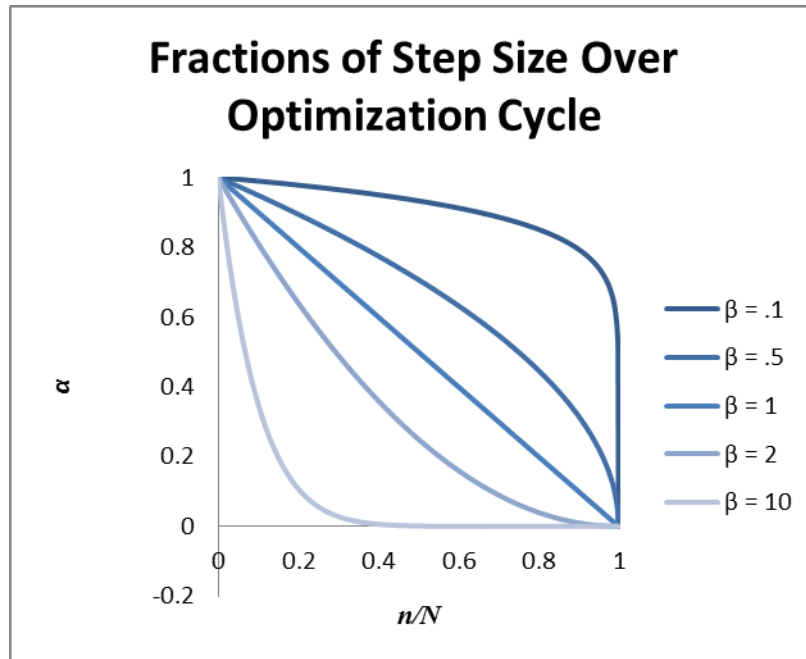


Figure 3-6: Step Size Reduction

3.2.6 Drop Draping

In drop draping, the fabric is created all at once in a plane above the mold surface. A simulated gravity force is added to each node. The optimization is run on all nodes at the same time. When a node is within a specified minimum distance of the surface of the layup, the gravity force is removed and the surface force takes over and pulls the fabric in further. The optimization runs indefinitely with a maximum step size until all nodes have switched over to using the surface force which allows the fabric to quickly drop to the surface. Once the fabric has reached the surface, the optimization runs a set number of times while decreasing the step size. Once the set number of runs is completed, the fabric has been draped.

3.2.7 Spread Draping

Spread draping does not model the entire fabric at once. The application point is the first node placed on the surface. The adjacent nodes are placed in a plane normal to the surface of the layup. These first nodes are optimized a set number of times while decreasing the step size. Once the optimization on those nodes is done, those nodes are deactivated and the nodes within one degree of separation are activated. The new set of active nodes is optimized in the same manner as the first set. The process is repeated thus “spreading” the fabric over the mold. Once all nodes have been optimized in sets, the entire fabric is activated and is optimized together while decreasing step size. Once the optimization is completed, the fabric is draped.

3.2.8 Tendon and Diagonal Node Mapping

Wang’s method is similar to spread draping because it starts in the middle and spreads out from there. The difference is in the order nodes are created. In spread draping they are created in layers while Wang’s method starts with what he calls tendon node mapping. Tendon

node mapping lays out the warp and weft fibers passing through the application point first by using the position of each previous warp or weft element to find an initial position and it is immediately optimized see Figure 3-7 (a). Once the tendon nodes are laid out, the diagonal node mapping is used to fill in the four quadrants created by the tendon nodes see Figure 3-7 (b). Since each diagonal node can have a great influence on the positions of subsequently created diagonal nodes, each diagonal node is optimized immediately after it is created. Finally, a boundary propagation algorithm is used to create the nodes that were unable to be created by the tendon and diagonal node mapping.

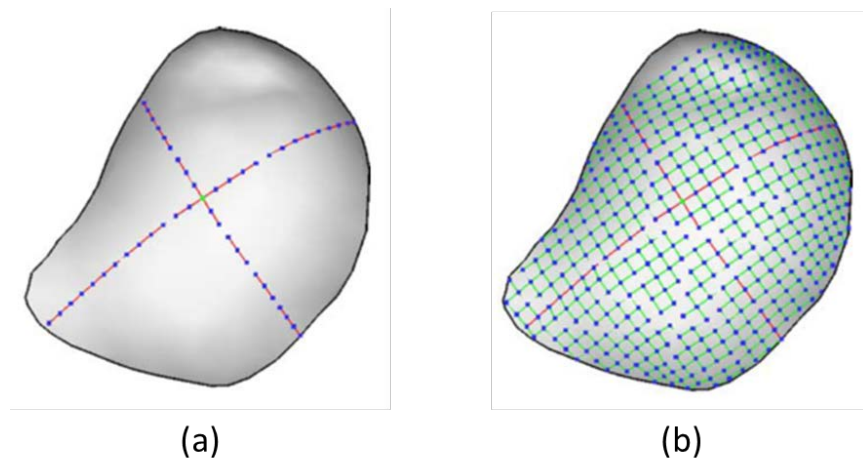


Figure 3-7: After Tendon (a) and Diagonal (b) Node Mapping[17]

3.3 Method for Stacking Plies

The key to stacking multiple plies is the tessellated surface representation. The mold surface is represented as a tessellated surface and the ply structure provides a convenient way to create a tessellated surface from the ply node grid. Since plies and the mold are represented using the same type of surface, the surface force can be calculated from a ply in the same way as the mold

thus plies can be draped on plies. Ply drop-offs naturally happen with this method since when one ply ends, the ones underneath can become the drape surface.

4 IMPLEMENTATION

The method in chapter 3 was implemented in a computer program which, given the definition of a ply, will drape plies onto a given surface. The surface can be a simple mold surface or a complex surface based on the plies already draped on a mold surface. The program was created in C++ using Siemens NX6 for visualization and geometry inputs. GSNLib created by Solid Modeling Solutions was used to compute arc/line/spline intersections and CGAL, an open source computational geometric algorithms library, was used to handle tessellated surfaces.

This chapter explains in detail the implementation of the method defined in chapter 3. Certain sections of the code are highlighted for clarity. The specific process of ply creation is explained, and then the application of the draping method is discussed. Finally the method of ply trimming is described.

4.1 Ply Creation

The steps of ply creation in this implementation are:

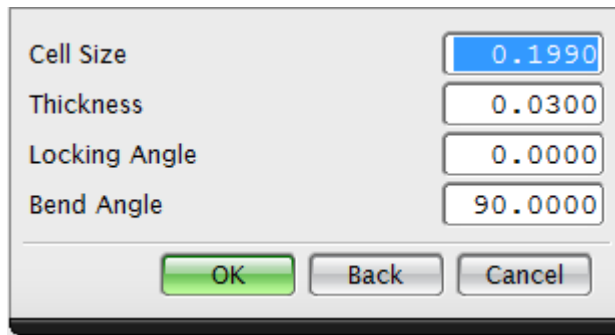
1. Get user input
2. Calculate flat pattern curves bounding box and allocate node grid data structure
3. Determine bounding box and center node
4. Determine which nodes are inside the ply
5. Add an extra layer of nodes

6. Create elements connecting nodes together
7. Create strings of warp and weft elements

Each of these will be explained in the following sections.

4.1.1 User Input

NX's Open C API was used to get information from the user. The API provides selection capability and some pre-defined number input boxes. First, the user selects the mold surface. Next, each ply is created sequentially. The first step for a ply is to define the cell size, thickness, and shear locking angle of the ply. All are input at once using the uc1608 function which creates the dialog in Figure 4-1 below.



Cell Size	0.1990
Thickness	0.0300
Locking Angle	0.0000
Bend Angle	90.0000

OK Back Cancel

Figure 4-1: Ply Parameters Input Dialog

Next, the user selects the application point coordinate system, the ply coordinate system, and the ply curves. Finally, to select the type of draping, a button box is used (see Figure 4-2). The lines arcs and splines are then converted to splines in the format GSNLib uses.

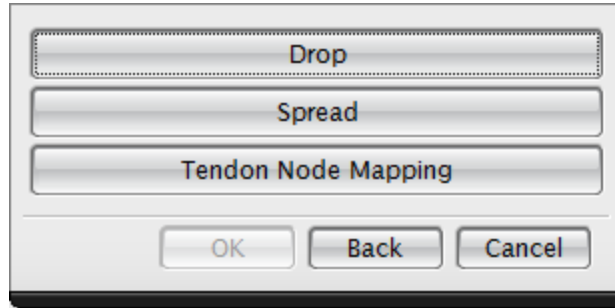


Figure 4-2: Drape Method Selection Dialog

4.1.2 Flat Pattern Curves Bounding Box and Data Structure Allocation

The flat pattern curves bounding box is found by computing each curve's bounding box and testing to see if the flat pattern fits within the composite bounding box of the trimming curves. If it does not completely fit inside the flat pattern bounding box, the flat pattern bounding box is expanded to fit the composite trimming curve bounding box (see Figure 4-3). Bounding boxes for splines are easy to calculate since splines follow the convex hull property. A curve follows the convex hull property if the entire curve fits inside the convex hull formed by the control points of the curve. If the convex hull contains the spline, a bounding box containing the control points (and therefore the convex hull) will always contain the spline itself. After creating the composite bounding box for the flat pattern, the structure for the node grid can be created. The node grid size is then adjusted to be able to fit the specified number of nodes in both the x direction and the y direction. The size of the node grid data structure is found by using equation (3-1). The node grid is a rectangular matrix of type Node.

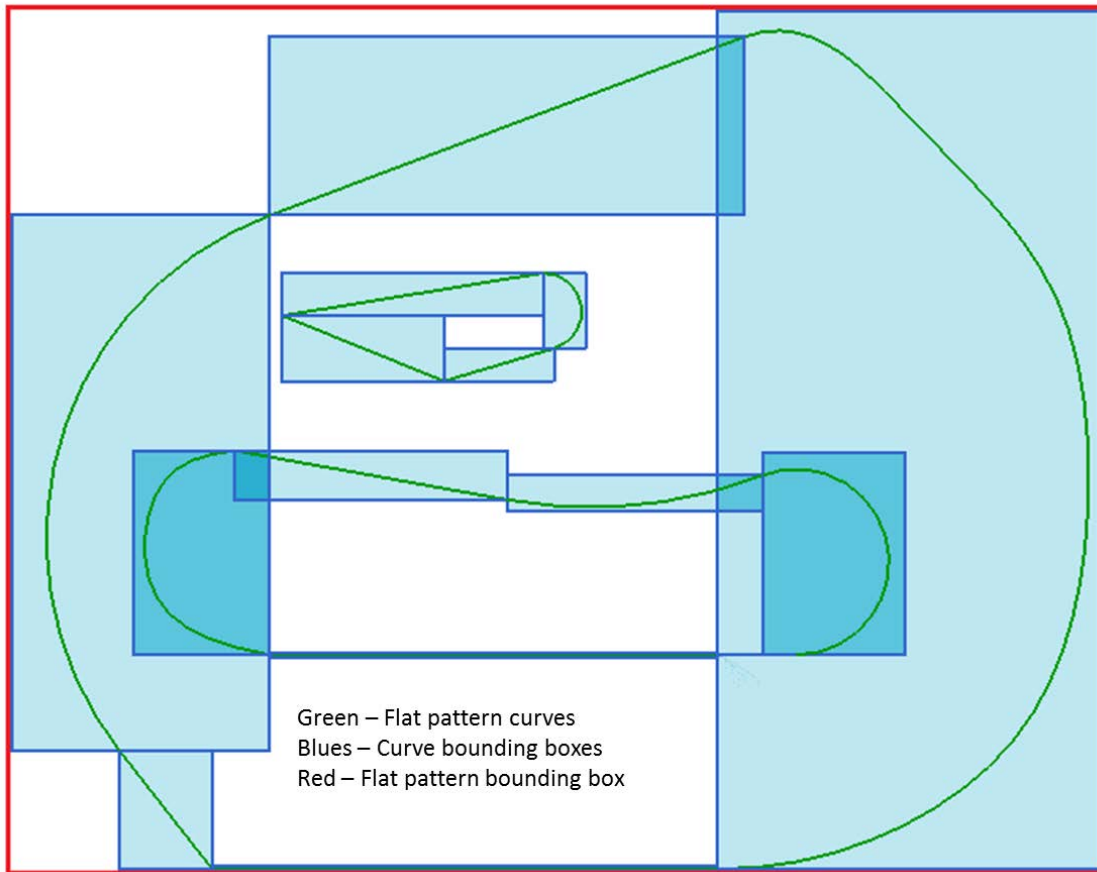


Figure 4-3: Flat Pattern Bounding Box Creation

4.1.3 Node Grid Bounding Box

The center node is found using equation (3-2). The node grid bounding box is then created by finding the bottom left node (indices 0,0) and the top right node (indices I,J). The node locations can be found from the indices using equation (3-3).

4.1.4 Determining Inner Nodes

To determine inner nodes, a scan line algorithm similar to one used in computer graphics was implemented. The algorithm follows these steps:

1. Create vertical line
2. Intersect line with all boundary curves
3. Save the intersections
4. Sort the intersections vertically
5. Create intervals between intersections and save
6. Loop 1-5 until all intervals are created

That code is shown next showing each step separately:

Step 1: Create vertical line

```
std::vector<std::vector<Interval> > yIntervals;
for(int i=0; i<sizeofDataStructureX; i++)
{
    //Create line along vertical node column
    std::vector<double> yVals;
    yVals.push_back(bottomLeftNode.y);
    yVals.push_back(topRightNode.y);
    std::vector<Interval> tempIntervals;
    IwPoint3d stPt(bottomLeftNode.x+this->cellSize*i,bottomLeftNode.y,0);
    IwPoint3d enPt(bottomLeftNode.x+this->cellSize*i,topRightNode.y,0);
    IwLine* newLine;
    IwLine::CreateLineSegment(localContext,3,stPt,enPt,newLine);
    IwObjDelete autoDestruct(newLine);
}
```

Step 2: Intersect with all boundary curves

```
for(int j=0; j<flatPatternCurves.size(); j++)
{
    IwSolutionArray solutions;
    flatPatternCurves.at(j)->GlobalCurveIntersect(
        flatPatternCurves.at(j)->GetNaturalInterval(),
        *newLine,newLine->GetNaturalInterval(),
        .001*this->cellSize,solutions);
    for(int k=0; k<solutions.GetSize(); k++)
    {
        IwVector3d newLineTangent, flatPatternTangent;

        //Evaluate the intersection point and derivative
        //from the intersection parameter of the line
        IwVector3d newLinePointAndDeriv[2];
        newLine->Evaluate(
            solutions.GetAt(k).m_vStart.m_adParameters[1],
            1,TRUE,newLinePointAndDeriv);
        newLineTangent = newLinePointAndDeriv[1];
        newLineTangent.Unitize();
    }
}
```

```

//Evaluate the intersection point and derivative
//from the intersection parameter of the curve
IWVector3d flatPatternPointAndDeriv[2];
flatPatternCurves.at(j)->Evaluate(
    solutions.GetAt(k).m_vStart.m_adParameters[0],
    1,TRUE,flatPatternPointAndDeriv);
flatPatternTangent = flatPatternPointAndDeriv[1];
flatPatternTangent.Unitize();

```

Step 3: Save the intersections

```

//Check for tangency. If not tangent, add to yVals
if(1.0-fabs(flatPatternTangent.Dot(newLineTangent))>.00001)
{
    IWPoint3d tempPoint = newLinePointAndDeriv[0];
    yVals.push_back(tempPoint.y);
} //End if
} //End solutions loop
} //End flat pattern curve loop

```

Step 4: Sort the intersections vertically

```

std::sort(yVals.begin(),yVals.end());

```

Step 5: Create intervals between intersections and save

```

for(int j=0; j<yVals.size()-1; j++)
{
    Interval tempInterval;
    tempInterval.SetMinMax(yVals.at(j),yVals.at(j+1));
    tempIntervals.push_back(tempInterval);
    IWPoint3d p1, p2;
    p1.Set(
        bottomLeftNode.x+static_cast<double>(i)*this->cellSize,
        yVals.at(j),0);
    p2.Set(
        bottomLeftNode.x+static_cast<double>(i)*this->cellSize,
        yVals.at(j+1),0);
}
yIntervals.push_back(tempIntervals);
} //End loop of steps 1-5

```

Step 6 is the loop surrounding steps 1-5. Once this code is run, the program has intervals in the y direction where nodes exist and where they do not exist see Figure 4-4. Those intervals alternate from non-existing nodes to existing-nodes.

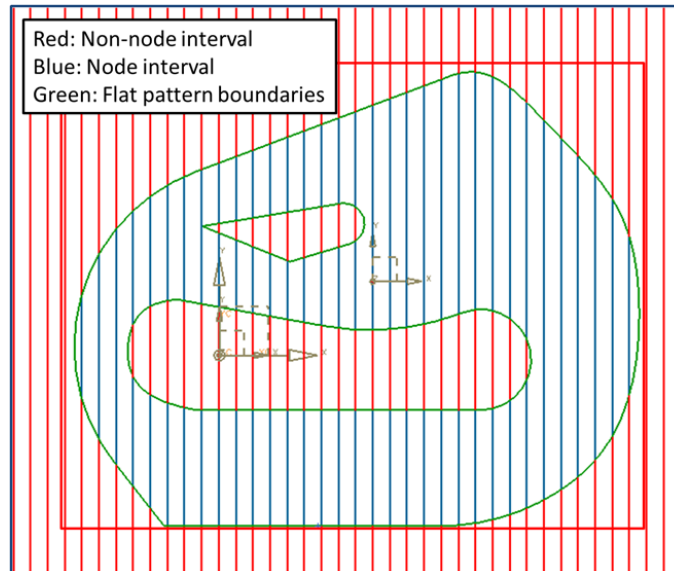


Figure 4-4: Flat Pattern Showing Intervals

The next code snippet creates the nodes along the intervals where they exist (or the blue intervals in Figure 4-4):

```

for(int i=0; i<sizeofDataStructureX; i++)
{
    for(int j=0; j<sizeofDataStructureY; j++)
    {
        for(int k=1; k<yIntervals.at(i).size(); k+=2)
        {
            if(
                yIntervals.at(i).at(k).ContainsValue(
                    bottomLeftNode.y+static_cast<double>(j)*this->cellSize)
                == TRUE)
            {
                this->nodeGrid.at(i).at(j) = new Node(this->layup);
                this->nodeGrid.at(i).at(j)->xIndex = i;
                this->nodeGrid.at(i).at(j)->yIndex = j;
            }//End if
        }//End interval loop
    }//End node y loop
}//End node x loop

```

When this block of code is finished, the flat pattern is filled with nodes as shown in Figure 4-5.

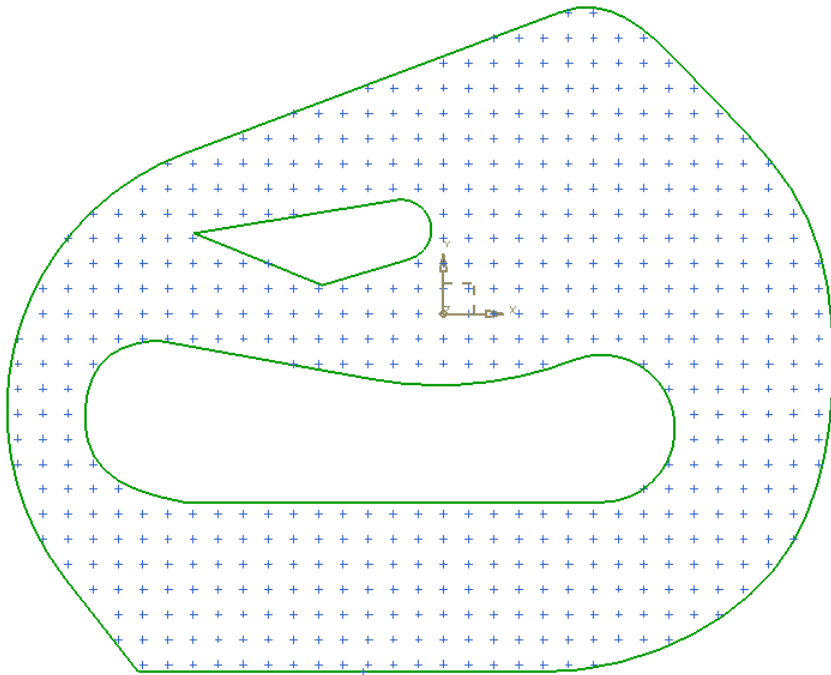


Figure 4-5: Flat Pattern Filled with Nodes

4.1.5 Extra Outer Layer

Another layer of nodes is created to ensure that the fabric completely overlaps the flat pattern. This enables the flat pattern curves to be used to trim the ply boundary. These nodes are added by taking each active node, finding the nodes around it, and adding them to a list if they previously did not exist. Once all inactive nodes are found, they are added to the node grid matrix. Once this is finished it looks like Figure 4-6.

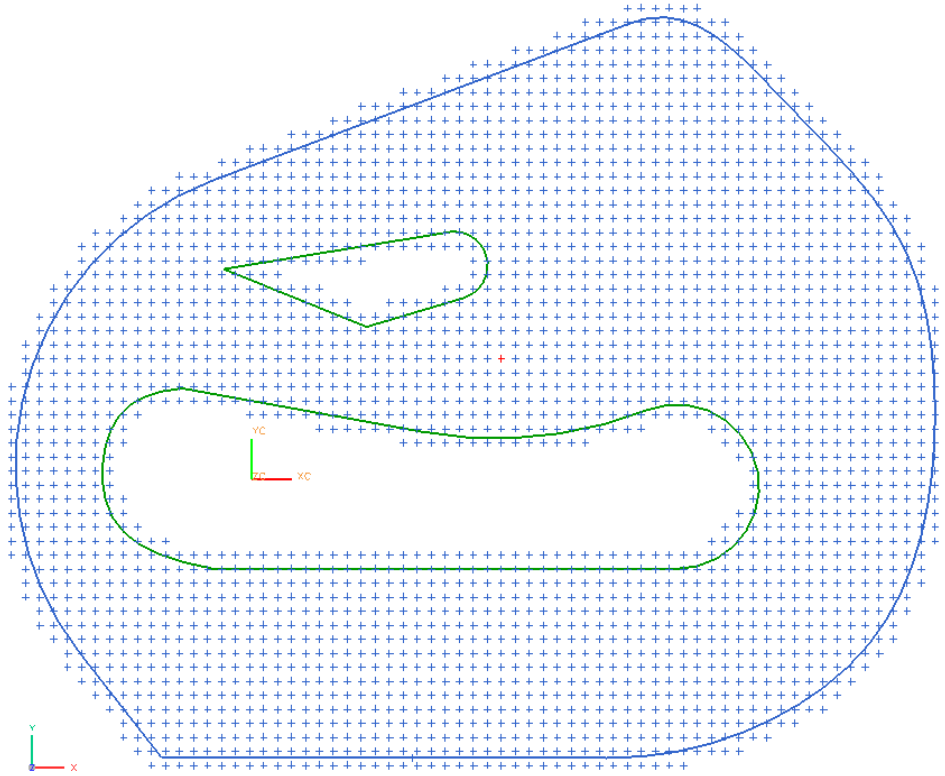


Figure 4-6: Nodes Overlapping Flat Pattern

4.1.6 Element Creation

After nodes are created, warp, weft, and shear elements can be connected between them. The node grid is traversed, connecting elements to the right and above the node if both end nodes for each element exist. The other constant data is also added to each element such as the unstretched length and spring constant.

```
//Create elements between active nodes
//Add node pointers to elements and element pointers to nodes
for(int i=0; i<this->nodeGrid.size()-1; i++)
{
  for(int j=0; j<this->nodeGrid.at(i).size()-1; j++)
  {
    //Warp elements
    if(nodeGrid.at(i).at(j) != NULL && nodeGrid.at(i+1).at(j) != NULL)
    {
      WarpElement *tempWarpElement = new WarpElement(this->layup);
    }
  }
}
```



```

tempWarpElement->n1 = nodeGrid.at(i).at(j);
tempWarpElement->n2 = nodeGrid.at(i+1).at(j);
tempWarpElement->springConstant = this->warpSpringConstant;
tempWarpElement->unstretchedLength = this->cellSize;
this->elementList.push_back(tempWarpElement);
nodeGrid.at(i).at(j)->AddElement(tempWarpElement);
nodeGrid.at(i+1).at(j)->AddElement(tempWarpElement);
} //End warp elements

//Weft elements
if(nodeGrid.at(i).at(j) != NULL && nodeGrid.at(i).at(j+1) != NULL)
{
    WeftElement *tempWeftElement = new WeftElement(this->layup);
    tempWeftElement->n1 = nodeGrid.at(i).at(j);
    tempWeftElement->n2 = nodeGrid.at(i).at(j+1);
    tempWeftElement->springConstant = this->weftSpringConstant;
    tempWeftElement->unstretchedLength = this->cellSize;
    this->elementList.push_back(tempWeftElement);
    nodeGrid.at(i).at(j)->AddElement(tempWeftElement);
    nodeGrid.at(i).at(j+1)->AddElement(tempWeftElement);
} //End Weft elements

//Shear elements
if(nodeGrid.at(i).at(j) != NULL && nodeGrid.at(i+1).at(j+1) != NULL)
{
    ShearElement *tempShearElement = new ShearElement(this->layup);
    tempShearElement->n1 = nodeGrid.at(i).at(j);
    tempShearElement->n2 = nodeGrid.at(i+1).at(j+1);
    tempShearElement->springConstant = this->shearSpringConstant;
    tempShearElement->unstretchedLength = this->cellSize*sqrt(2.0);
    tempShearElement->lockingAngle = this->lockingAngle;
    this->elementList.push_back(tempShearElement);
    nodeGrid.at(i).at(j)->AddElement(tempShearElement);
    nodeGrid.at(i+1).at(j+1)->AddElement(tempShearElement);
}
if(nodeGrid.at(i+1).at(j) != NULL && nodeGrid.at(i).at(j+1) != NULL)
{
    ShearElement *tempShearElement = new ShearElement(this->layup);
    tempShearElement->n1 = nodeGrid.at(i+1).at(j);
    tempShearElement->n2 = nodeGrid.at(i).at(j+1);
    tempShearElement->springConstant = this->shearSpringConstant;
    tempShearElement->unstretchedLength = this->cellSize*sqrt(2.0);
    tempShearElement->lockingAngle = this->lockingAngle;
    this->elementList.push_back(tempShearElement);
    nodeGrid.at(i+1).at(j)->AddElement(tempShearElement);
    nodeGrid.at(i).at(j+1)->AddElement(tempShearElement);
} //End shear elements
} //End vertical loop
} //End horizontal loop

```

Bend elements are different than warp, weft, and shear elements because they use three nodes instead of just two. Bend elements run in all directions and represent the stiffness of the fabric out of plane. The function to create a bend element is as follows:

```

bool Ply::CreateBendElement(Node* n1, Node* n2, Node* n3)
{
    if(n1!=NULL && n2!=NULL && n3!=NULL)
    {
        BendElement* be = new BendElement(this->layup);
        //Add nodes to element and initialize
        be->n1 = n1;
        be->n2 = n2;
        be->n3 = n3;
        be->myType = ET_BEND;
        be->springConstant = bendSpringConstant;
        be->lockingAngle = this->bendLockingAngle;
        this->elementList.push_back(be);
        //Add element to nodes
        n1->AddElement(be);
        n2->AddElement(be);
        n3->AddElement(be);
        return true;
    }
    return false;
}

```

4.1.7 Warp and Weft Element Strings

Since the warp and weft directions represent the tows which are essentially inextensible, the warp and weft element forces are calculated differently than shear or bend forces. The elements must be connected in strings to be able to calculate the force. The node grid is traversed and the warp elements to the left and right of node (i,j) are linked and the weft elements on top and bottom are linked. Cuts in tows are naturally modeled this way because if a node is missing in the string of elements, the warp or weft elements are not connected due to the blank node.

4.2 Draping Method

The implementation of the three draping methods, drop draping, spread draping, and Wang's method are explained in this section. Drop draping simulates a fabric dropping onto the surface due to a simulated gravity force. Spread draping and Wang's method are similar to putting one point of the fabric down, and pushing the fabric onto the surface in an ever-widening

ring. All three methods employ the same code to calculate forces and the optimization method is the same except for some minor differences. The force calculation is explained first since it belongs to all three methods of draping and then the differences between the implementations of the optimization methods are described.

4.2.1 Force Calculation

The force calculation is done from a list of active elements and active nodes. Each node keeps a vector sum of all the forces acting on it. The CalculateAndAddForces() function in the following block of code uses the equations in sections 3.2.2 to 3.2.3 (pages 23 to 24) to calculate the force of the element on each node connected to it and add it to the corresponding vector sum. The warp and weft elements are done slightly differently. Equation (3-4) includes the sum of all the tensile and compressive forces along the string of elements. Each element's tensile or compressive force is calculated first in the CalculateTensileForce function. See below:

```
for(std::list<Element*>::iterator it=activeElements.begin();
    it!=activeElements.end(); it++)
{
    switch((*it)->myType)
    {
        case ET_WARP:
        case ET_WEFT:
            (*it)->CalculateTensileForce();
            break;
        case ET_SHEAR:
            (*it)->CalculateAndAddForces();
            break;
        case ET_BEND:
            (*it)->CalculateAndAddForces();
            break;
        default:
            break;
    }
}
```

The active nodes are then traversed and warp and weft vector forces are calculated as seen in the next block of code:

```
for(std::list<Node*>::iterator it=activeNodes.begin();
    it!=activeNodes.end(); it++)
{
    (*it)->AddForce((*it)->CalculateWarpForce());
    (*it)->AddForce((*it)->CalculateWeftForce());
}
```

The CalculateWarpForce and CalculateWeftForce functions traverse the string of elements, do the scalar sum of the tensile forces, and multiply the scalar force by the unit vector in the direction of the warp or weft element connected to the node. A warp or weft force for each side of the node is calculated and added to the node.

After all the other forces have been calculated, the surface force is calculated. The surface force is based on the distance to the surface. CGAL was used to find the distance to the surface. An axis aligned bounding box tree (AABB tree) was constructed on the faces of the tessellated surface of the mold and plies. This allows for fast intersection and distance queries. When the closest point is queried, CGAL internally traverses the AABB tree using a ball search until it finds the bounding box that is closest at which point it switches over to using the triangle contained inside the bounding box to find the actual closest point on the triangle. Once the closest point is determined, the distance to that point and the direction to that point can be determined. The force from the surface is found using a spring equation with different spring constants for tension and compression. The un-stretched length of the spring is the thickness of the ply being draped. The direction of the force goes from the closest point on the surface to the node.

4.2.2 Differences Between Optimization Methods

Once all forces on all the nodes have been calculated, the nodes are moved in the direction of the resultant force as seen in the function below:

```
void Node::Move(double maxDist)
{
    double stepLength = maxDist*(1 - 50/(50+forceSum.Length()));
    IwVector3d step = forceSum;
    step.Unitize();
    step = step * stepLength;
    this->location = this->location + step;
    this->ResetForceSum();
}
```

where maxDist is set to less than or equal to half the cell size. This prevents most nodes from crossing over each other during optimization causing folds in the fabric. Once the node is moved, the force sum is set to zero for the next round of optimization.

For drop draping, the fabric is created all at once in a plane above the mold surface. All nodes are moved during the entire optimization. When nodes are first created a simulated gravity force is added to each one. Once the nodes are within a specified distance to the surface, the gravity force is switched off and the surface force is switched on. The optimization continues indefinitely with a constant maximum step size of half the cell size until all nodes have switched over to the surface force. Once all nodes have switched over, the step size is decreased gradually using equation (3-11) until the optimization ends.

For spread draping, only the application point and first layer of surrounding nodes are placed initially. Once the first nodes are placed, they are optimized a certain number of iterations. The initial nodes are de-activated and the next layer of surrounding nodes is then formed and optimized. This process repeats until the entire ply has been placed. It is now close to the optimum shape. Then, all nodes are activated at the same time for a final optimization.

For Wang's method, the first four tendon nodes are created and optimized, then the next tendon nodes are created and optimized. This happens until there are no more tendon nodes to map. Next the diagonal nodes are mapped by creating a node, then optimizing it before creating the next node. The boundaries are propagated until all nodes have their initial position. Once all nodes have been placed, the entire fabric is optimized at the same time.

4.3 Ply Cutting Method

Plies initially are created to overlap their flat pattern boundaries by at least one node. This is the shape that is used for the active fabric so there are no partial elements. Once the ply is draped, the extra nodes and elements around the boundary must be cut to the proper length to properly define the edges of the fabric. This is done in three steps. The first step is to triangulate the two dimensional flat pattern with the borders, the second is to remove the extra fabric around the borders, and the third is to translate the two dimensional triangulation to the three dimensional ply.

4.3.1 Triangulating the Flat Pattern

The flat pattern triangulation was done using CGAL. The type of triangulation used was a constrained Delaunay triangulation. To get the constraints, the warp and weft elements were intersected with the border curves of the flat pattern using GSNTLib and line segments were created between adjacent intersections. This creates a loop of non-intersecting constraints. The constraints are passed into the triangulation along with the node locations from the flat pattern. Once all nodes and constraints are passed in, the triangulation is finished. A face list is created from the triangulation and used in the next step.

4.3.2 Removing Extra Fabric

During creation of the flat pattern, the extra nodes were created. The pointers to those nodes were kept and any face in the triangulation face list connected to one of those nodes is removed. This removes all external faces.

4.3.3 Translation to Three Dimensions

The face list contains two dimensional points from the flat pattern. To translate to the third dimension, the node grid is parameterized along the columns and along the rows. The parameter values start at 0 at the first node in a row or column and increase by 1 each node. Each node's parameter is therefore its index in the node grid. Around the edges of the ply, the points from the constraints will likely lie between nodes. Their parameter values are interpolated between the surrounding nodes and will not be integers. For nodes, the three dimensional point is just the 3d point found by the draping algorithm. For points that do not lie on integer parameter values, the three dimensional point is interpolated from the surrounding nodes. The function for this operation is as follows:

```
IwPoint3d Ply::Get3dFromFPPoint(IwPoint3d fpPoint)
{
    IwPoint3d point;
    std::pair<double,double> params = GetParametersFromFPPoint(fpPoint);
    std::pair<int,int> indices = GetIndexFromParameter(params);
    std::pair<double,double> diffs = ParamsMinusIndices(params,indices);
    IwPoint3d origin = nodeGrid.at(indices.first).at(indices.second)->GetLoc();

    if(diffs.first > .000001 && diffs.second > .000001)
    {
        IwPoint3d point1 = (nodeGrid.at(indices.first+1).at(indices.second)
            ->GetLoc()-origin)*diffs.first+origin;
        IwPoint3d point2 = (nodeGrid.at(indices.first).at(indices.second+1)
            ->GetLoc()-origin)*diffs.second+origin;
        point = (point2-origin)+(point1-origin)+origin;
    }
    else if(diffs.first > .000001)
    {
        point = (nodeGrid.at(indices.first+1).at(indices.second)->GetLoc()-
            origin)*diffs.first+origin;
    }
}
```

```

}
else if(diffs.second > .000001)
{
    point = (nodeGrid.at(indices.first).at(indices.second+1)->GetLoc()-
        origin)*diffs.second+origin;
}
else
{
    point = origin;
}
return point;
}

```

The function compares the difference between the truncated parameter value and the full parameter value. If a non-integer parameter is found in both directions, the first *if statement* creates the three dimensional point by interpolating in both directions. The next two *if statements* only interpolate in either the column or the row direction and the *else statement* is for nodes that lie on exact parameter values.

Once the three dimensional points are determined, the face list and new three dimensional points can be used to create a Polyhedron_3 from CGAL. This is the *class* that forms the basis for each ply and the mold surface. This is also the *class* that can create the AABB tree used in finding the distance between the next ply's nodes and the surface of the layup.

5 RESULTS

To validate the model, physical testing was done as well as exploration of the three different draping methods to determine which was best for certain situations. In this chapter the physical testing is explained as well as observations and comparisons of the different draping methods.

5.1 Physical Testing

The physical testing was done by creating a mold, laying up a fiberglass/epoxy part, filling it with foam, making cross-sectional cuts, polishing the cut edges and finding the ply drop-offs under a microscope. The locations of the drop-offs were marked and measured and compared to the model created by the draping program. Two molds were used: a 6 inch hemisphere as a standard and a scaled down dodge viper hood. This section will briefly explain the process to create and measure the samples and discuss the results in detail.

5.1.1 Fabrication Process

First molds for the parts were acquired. The molds were a 6 inch hemisphere and a dodge viper hood shown in Figure 5-1.



Figure 5-1: 6 Inch Hemisphere and Scale Dodge Viper Hood Molds

The material used was a heavy plain-weave dry fiberglass with a tow width of .2 inches/tow and a thickness of .03 inches. The matrix material was an epoxy with a pot life of one hour and a cure time of 24 hours. Each ply shape was marked and cut out with heavy-duty scissors. The molds were prepped with a wax mold release and laid on the vacuum bag. Vacuum bagging was the only process available, it is relevant to this method and is common in industry. Each ply was wet-out with the epoxy on a plate and transferred from the plate and laid on the mold at the correct angle. Hand layup was used as the only method. After all plies were placed, the molds were covered with a perforated release film, a felt breather pad, and sealed in the vacuum bag using a sticky polymeric tape. A vacuum was applied immediately and for the duration of the cure.

Once cured, the parts were removed from the molds and filled with an expanding sealant foam commonly found at a hardware store. Figure 5-2 shows the finished parts.



Figure 5-2: Finished Parts

The parts were then marked with radial lines originating at the application point. The lines in Figure 5-2 are marked every 10° but only every 3rd line was used for cutting because a 10° slice would have been too thin to sand and polish. The parts were then cut using a Dremel tool with a cutting wheel along the lines.

Next the cut edges were sanded and polished until smooth. The polished samples were looked at under a microscope to find the ply drop-offs. An example of a ply drop-off is shown in Figure 5-3. Once the drop-offs were found, they were marked and measured.

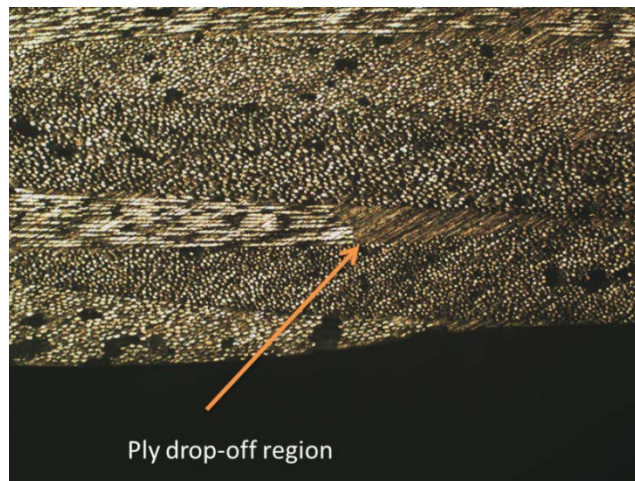


Figure 5-3: Ply Drop-off

The measurement method differed between the hemisphere and the viper hood. The hemisphere was measured using a 6 inch protractor with the 0° mark lined up with the bottom of the part. Each angle was measured to the closest degree. The hood measurements were taken from the outer edge of the hood surface since it was the only point of reference to measure with. Ply drop-offs on the hood were measured as a point to point distance.

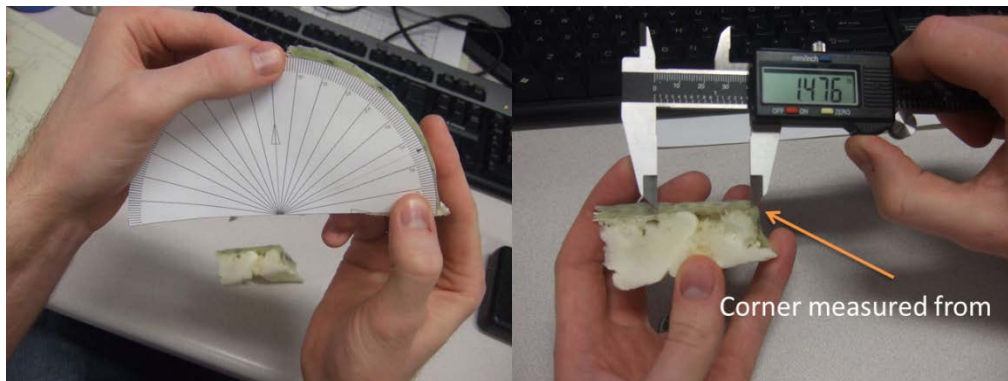


Figure 5-4: Measurement Techniques

5.1.2 Comparison of Results

The physical results in general agreed well with the results predicted by the implementation described in chapter 4. The results predicted by the implementation were measured by cross-sectioning the part at the same locations as the physical model. The differences in the results were compared against each other and are shown next for the hemisphere:

Table 1: Hemisphere Arc Length Error

Hemisphere	Arc length error (in)						
Cross-section	Ply 1	Ply 2	Ply 3	Ply 4	Ply 5	Ply 6	Ply 7
1	full ply	-0.20	0.06	0.07	0.09	-0.03	full ply
4	full ply	0.08	0.10	0.39	-0.06	-0.02	full ply
7	full ply	0.34	0.31	-0.17	0.12	0.24	full ply
10	full ply	-0.31	0.26	0.13	0.28	-0.08	full ply
13	full ply	0.08	0.36	-0.55	0.05	0.15	full ply
16	full ply	0.19	0.21	-0.07	0.12	0.10	full ply

Plys 1 and 7 covered the entire hemisphere and were not measured. The error in the cells of Table 1 are found using this formula:

$$arc_length_error = (\theta_{physical} - \theta_{CAD}) * radius \quad (5-1)$$

Where $\theta_{physical}$ and θ_{CAD} are the angles (in radians) measured from the top of the hemisphere to the ply drop-off of the physical part and the CAD representation. The error for the hood is

Table 2: Hood Error

Hood	Physical Measurement -CAD Measurement (in)							
Cross-section	Ply 1	Ply 2	Ply 3	Ply 4	Ply 5	Ply 6	Ply 7	Ply 8
1	full ply	-0.16	-0.08	-0.05	-0.05	-0.05	-0.23	full ply
4	full ply	-0.10	-0.01	0.03	-0.04	-0.06	-0.20	full ply
7	full ply	0.13	-0.05	-0.24	0.25	-0.11	-0.24	full ply
10	full ply	0.15	-0.05	-0.10	0.05	0.27	0.01	full ply
13	full ply	0.25	-0.15	-0.15	0.07	0.20	-0.13	full ply
16	full ply	0.16	-0.09	-0.05	0.04	0.06	0.17	full ply
19	full ply	0.07	0.02	-0.03	-0.02	0.21	0.09	full ply
22	full ply	0.09	-0.03	-0.18	-0.08	0.05	0.06	full ply
25	full ply	-0.11	-0.24	-0.05	-0.22	-0.14	-0.02	full ply
28	full ply	-0.23	0.29	0.14	0.14	0.19	0.05	full ply
31	full ply	-0.17	-0.27	0.13	-0.10	-0.17	-0.08	full ply
34	full ply	-0.14	-0.04	0.04	-0.09	-0.03	-0.20	full ply

where each cell is just the physical measurement minus the CAD measurement.

Much of the error in the results may have been introduced with the measurement techniques used. The samples were all polished by hand and therefore had varying amounts of material removed. If a ply edge was almost parallel to the cross-section as shown by the green ply in Figure 5-5, small changes in the position of the cross-section plane and small errors in the angle of the ply placement could introduce large discrepancies between the physical measurement and the CAD measurement. The cells in Table 1 and Table 2 that are marked red are the cells where this type of error is found. As the edges of the plies become more perpendicular to the cross-section like the red and yellow plies in Figure 5-5, the measurements are less affected by the amount of material removed during polishing.

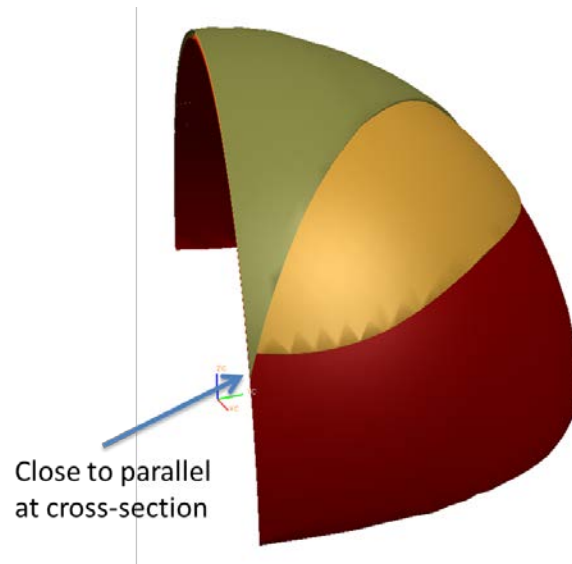


Figure 5-5: Ply Almost Parallel to Cross-Section

Also, since the process used was hand layup, some ply orientations may have been slightly off in the physical part which could compound the error caused by varying the amount of material

removed such as ply 4, cross-section 13 in Table 1. To get more accurate results a laser projection system could be used (if available) to make sure the ply was placed at the correct angle and cutting the cross-section large and grinding down to the correct cross-section would both help reduce the measurement error.

The maximum error in the hemisphere is larger than in the hood. One reason for this may be that the hemisphere has higher curvature and thus higher shear angles than the hood. One of the main deformation modes of fabric at high shear angles is fiber slippage. Since fiber slippage was not included in this method, it may have contributed to some of the error. Fiber slippage is suggested as a topic of future work in section 6.1.4. Another small source of error may be that the plies were traced from a pattern by hand and cut using hand shears. One final source of error is that because of the free edge of the fabric along the cuts, some fiber bundles fell out from the weave of the fabric, especially on the smaller plies. This made finding the true edge of the drape difficult under the microscope because one of the fiber directions was missing.

The average of the absolute value of the hemisphere error is .17 inches. After removing the red cells, the average is .15 inches. The maximum error drops from .55 inches to .36 inches. The average of the absolute value of the error for the hood is .12 inches and after removing the red cells, the average is .11 inches. The maximum error is .29 inches which could have been caused by tows falling out or misalignment of the ply.

These results are reasonable for a part of this size. If a larger part were manufactured, the distances between the ply drop-offs would likely be less of an issue. What really matters to a composite designer is whether the part will fail or not. This varies greatly with the fiber angles, locations of drop-offs, the stress state around the drop-offs and factor of safety. Since this

method is a way of visualization, the designer has one more tool to help figure out whether the design will fail.

5.2 Software Testing

This section explains what was done to test the robustness of the software. First the test parts are shown with explanations of why they were chosen. Next, an explanation of the results of using each of the drape methods on all four parts is offered.

5.2.1 Test Parts

There were four test parts as shown in Figure 5-6. Part 1 was included as a standard draping shape which should have no surprises. Part 2 was included because it was expected to create wrinkled fabric in the corners of the pocket. Part 3 was included because wrinkles can occur at the corners of the square protrusion. Part 4 was included as another easy draping shape that was not standard.

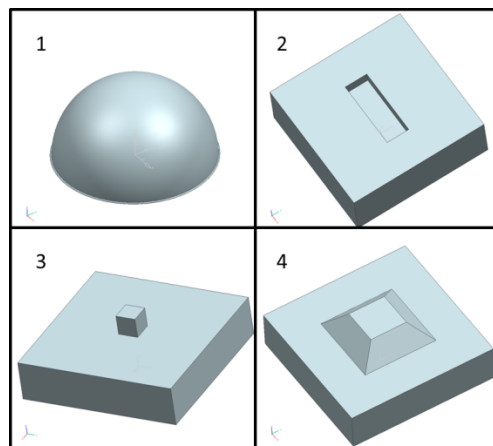


Figure 5-6: Test Parts

5.2.2 Results Explanation

The results for the testing are shown in Table 3.

Table 3: Software Testing Results

	Drop	Spread	Wang
Part 1	Application point shift	Uneven shear	Good
Part 2	Good	Self-intersections	Self-intersections
Part 3	Good	Hourglass unit cell	Hourglass unit cell
Part 4	Good	Uneven shear	Good

A drape was considered good if there were no artifacts in the simulation that would be impossible or unlikely in the real world. Some examples illustrated were self-intersecting fabric, hourglass shaped unit cells, large shifts in application point, and uneven shear. In the physical world, fabric cannot self-intersect. Hourglass shaped unit cells happen when the cell is in torsion, and would not happen in a normal draping situation. Uneven shear is possible in the physical world, but usually does not happen because the technician takes care to align the fabric properly.

Drop draping is the most versatile method and works well for parts 2 through 4 but the drape has a tendency to shift away from the application point if the part is asymmetrical or the fabric is draped asymmetrically on the part. Small plies work well on the hemisphere, but larger plies tend to get pulled off-center because as the fabric is dropping, some nodes are using the surface force and others are using the drop force (See Figure 5-7). If more of the nodes on one side of the fabric are using the drop force than the other side, the forces in the fabric are unbalanced which is why the drape gets pulled off-center. In the physical world, the technician creating the layup controls where the fabric lies by using a laser projection system or some other method of aligning the ply. Also, friction between the ply and the layup resists the shifting motion seen in the drop draping method. Adding a way for the user to define the placement of

the middle warp and weft fibers and/or adding friction to the software model are possible avenues for future work.

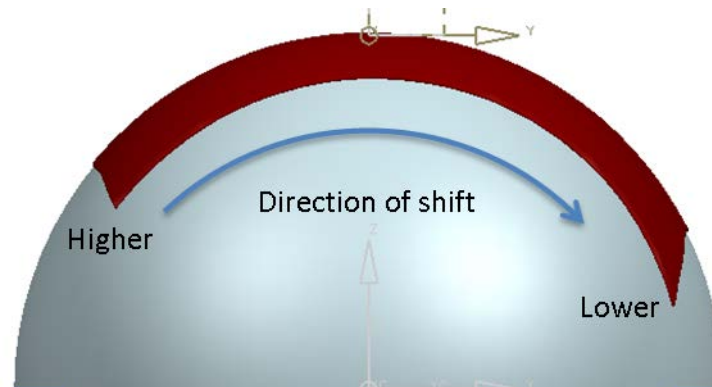


Figure 5-7: Shifted Drop Drape

Spread draping works well for parts 1 and 4. On part 2 as the drape expands into the corners of the pocket, the initial locations of the nodes cross over each other causing hourglass shaped unit cells and self-intersection of the fabric. Part 3 behaves similarly once the drape progresses past the inside corner at the base of the protrusion. In the physical world, the locations where the software drape folds over itself are regions of high shear and would likely cause wrinkles in the fabric. This method also is somewhat dependent on cell size of the fabric. Smaller cell sizes have more of a tendency to fold over themselves than larger cell sizes.

Wang's method gives the best and fastest drape of both parts 1 and 4. Since this method is very similar to spread draping, its behavior is almost identical. There is a slight difference though. The shear in the drapes created with Wang's method are usually more uniform than drapes created using spread draping. This happens because the tendon nodes do not move while the diagonal nodes are being created and optimized. This effectively separates the quadrants from each other and defines the overall shear for the quadrant. Defining the shear for the

quadrant affects the shear for each unit cell in the quadrant. Wang's method could be expanded to allow the user to define the tendon node paths, thus more completely defining the drape of the fabric.

Since no method worked for all parts, a designer should choose which method to use. The main factor that caused problems for Wang's method of draping was the presence of highly curved concavities. If a part was convex or gently concave, Wang's method was the best. If a part had sharp concave curvature, drop draping was the best method.

6 CONCLUSIONS

The objective of this thesis was to create a representation of fabric that could be used to create a simulated layup. The conclusions formed were:

1. No other method has modeled the local thickness of a layup
2. No one draping method works with all part shapes
3. This research predicted the physical layup ply boundaries to an average absolute error of .15 inches for the hemisphere and .11 inches for the hood
4. The methods developed for this research help designers by visualizing plies thus avoiding superimposing ply drop-offs
5. This method can re-order plies without changing cut-out shape

The literature review for this research found no other method that modeled the thickness of the laminate. To date, most draping methods have focused on draping a single ply on a surface. While this is helpful in many cases, a true representation of a layup must include fabric thickness. It was shown that this method takes into account the true layup thickness by forming a data structure where plies can be draped on top of plies. This becomes increasingly important with thick plies or thick laminates and areas where there are ply drop-offs.

It was also shown that so far, one draping method does not fit all draping scenarios. The drop draping method was shown to be the most versatile method but had some issues with centering the drape on the application point. Spread draping was shown to stay on the application

point, but was inferior to Wang's method because of uneven shear. Wang's method was shown to work well on convex parts and was faster than drop draping. More on these methods is explained in the next section on future work.

From the testing done on the hemisphere and hood parts, it was shown that this predicts the ply drop-off edges to an average of .15 inches and .11 inches for the two parts studied. Due to the hand measuring technique used, inaccuracies and anomalies in the hand layup process, and near-parallel ply boundaries and cross-sections, it was not possible to determine if the method predicted the boundaries better. It is expected that with a more tightly controlled physical layup technique the error would go down. The section on future work suggests ways to improve this measuring process.

The visualization of the plies in this method provides an easy way for the designer of a composite part to visualize the laminate. Cross-sections can be taken in the CAD part and superimposed ply drop-offs can be seen easily and thus avoided. Plies can also be hidden or shown which would allow the designer to see where the entire boundary of the ply falls.

Due to the nature of the data structure used, plies can be draped in any order while keeping their cut out shape. Most draping methods to date start with a boundary on a surface and drape to the boundary to find the flat pattern shape. This method reverses that process in order to be able to re-order plies if needed.

6.1 Recommendations for Future Work

There are many avenues for future work on this topic. These areas of future work explained in this section include:

1. Using a more accurate physical fabrication and measuring process
2. Adding friction to the drop drape method

3. Adding a user defined tow placement option to the drop draping method and Wang's method
4. Adding fiber slippage to the fabric model

6.1.1 Fabrication and Measuring Process

Since there were many sources of error in the physical layup and measuring technique, a more accurate process should be used. For the manufacturing process, a laser projection system would be helpful for placing plies correctly. For the cross-sectioning process, leaving an allowance for sanding and polishing would increase the accuracy because the part could be sanded and polished down to the measurement line to ensure the correct cross-section is measured. The layup of the part could be changed to include only plies whose boundaries are not close to parallel with any radial lines thus reducing the error caused by small differences in cross-section angle.

6.1.2 Friction

Other future work could include finding a way to keep the drop draping method from sliding off-center from the application point. Adding friction to the drop draping method is one option. Friction could be modeled by decreasing the step size of each node that is within a certain tolerance of the surface of the layup. This would cause the parts of the fabric that are on the surface to “stick” and would keep the drop drape on center.

6.1.3 User Defined Tow Placement

Another option to increase the accuracy of the drape might be to specify more deterministic boundary conditions for the fabric such as mixing the drop method with Wang's

method. This would be done by attaching tendon nodes to a user defined curve while dropping. This would have the benefits of Wang's method of draping such as more even shear while keeping the versatility of the drop draping method. In addition, a designer would have the option of tweaking the drape of the fabric to optimize the strength of the part.

6.1.4 Fiber Slippage

Fiber slippage was not included in this draping model. Fiber slippage becomes a significant deformation method at high shear angles. Currently, very few researchers have attempted to model fiber slippage. Slippage could possibly be included by adding a plastic deformation mode to the warp and weft springs.

REFERENCES

- [1] J. Kim, "Development of a user-friendly expert system for composite laminate design," *Composite Structures*, vol. 79, no. 1, pp. 76-83, Jun. 2007.
- [2] R. Hale, "Knowledge-Based Software Systems for Composite Design, Analysis and Manufacturing," in *SAE Technical Paper Series*, 2002, no. 724.
- [3] A. B. Strong, *Fundamentals of Composites Manufacturing*, 2nd ed. Society of Manufacturing Engineers, 2008, pp. 1-17,389-406,417-435.
- [4] B. Prakash, "AUTOLAY-a GUI-based design and development software for laminated composite components," *Computers & Graphics*, vol. 23, no. 1, pp. 95–110, 1999.
- [5] J.-S. Kim, C.-G. Kim, and C.-S. Hong, "Knowledge-Based Expert System for Optimal Stacking Sequence Design of Composite Structures," *Journal of Composite Materials*, vol. 33, no. 13, pp. 1244-1274, Jul. 1999.
- [6] A. I. Hepworth, C. G. Jensen, and J. T. Roach, "A CAD Independent Approach to Automate Laminate Composite Design and Analysis," *Computer-Aided Design and Applications*, vol. 6, no. 2, pp. 147-156, 2009.
- [7] A. I. Hepworth, C. G. Jensen, and J. T. Roach, "Methods to Streamline Laminate Composite Analysis and Optimization," *Computer-Aided Design and Applications*, vol. 7, no. 3, pp. 377-386, 2010.
- [8] F. Van Der Ween, "Algorithms for draping fabrics on doubly-curved surfaces," *International Journal for Numerical Methods in Engineering*, vol. 31, no. 7, pp. 1415-1424, 1991.
- [9] P. Boisse, "Meso-macro approach for composites forming simulation," *Journal of Materials Science*, vol. 41, no. 20, pp. 6591-6598, Sep. 2006.
- [10] A. Cherouat and J. L. Billo\et, "Mechanical and numerical modelling of composite manufacturing processes deep-drawing and laying-up of thin pre-impregnated woven fabrics," *Journal of materials processing technology*, vol. 118, no. 1-3, pp. 460–471, 2001.

- [11] L. Dong, C. Lekakou, and M. G. Bader, "Processing of Composites: Simulations of the Draping of Fabrics with Updated Material Behaviour Law," *Journal of Composite Materials*, vol. 35, no. 2, pp. 138-163, Jan. 2001.
- [12] a Skordos, C. Monroyaceves, and M. Sutcliffe, "A simplified rate dependent model of forming and wrinkling of pre-impregnated woven composites," *Composites Part A: Applied Science and Manufacturing*, vol. 38, no. 5, pp. 1318-1330, May 2007.
- [13] R. H. W. ten Thije and R. Akkerman, "A multi-layer triangular membrane finite element for the forming simulation of laminated composites," *Composites Part A: Applied Science and Manufacturing*, vol. 40, no. 6-7, pp. 739-753, Jul. 2009.
- [14] P. Boisse, B. Zouari, and J. Daniel, "Importance of in-plane shear rigidity in finite element analyses of woven fabric composite preforming," *Composites Part A: Applied Science and Manufacturing*, vol. 37, no. 12, pp. 2201-2212, Dec. 2006.
- [15] N. Hamila, P. Boisse, and F. Sabourin, "A semi-discrete shell finite element for textile composite reinforcement forming simulation," *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, vol. 79, no. May, pp. 1443-1466, 2009.
- [16] P. Boisse, B. Zouari, and a Gasser, "A mesoscopic approach for the simulation of woven fibre composite forming," *Composites Science and Technology*, vol. 65, no. 3-4, pp. 429-436, Mar. 2005.
- [17] C. Wang, K. Tang, and B. Yeung, "Freeform surface flattening based on fitting a woven mesh model," *Computer-Aided Design*, vol. 37, no. 8, pp. 799-814, Jul. 2005.
- [18] S. Sharma and M. Sutcliffe, "Draping of woven fabrics: Progressive drape model," *Plastics, rubber and composites*, vol. 32, no. 2, pp. 57-64, 2003.
- [19] J. Wang, J. R. Page, and R. Patod, "Experimental investigation of the draping properties of reinforcement fabrics," *Composites Science and Technology*, vol. 58, pp. 229-231, 1998.
- [20] P. Badel, E. Vidalsalle, and P. Boisse, "Computational determination of in-plane shear mechanical behaviour of textile composite reinforcements," *Computational Materials Science*, vol. 40, no. 4, pp. 439-448, Oct. 2007.
- [21] P. Harrison, J. Wiggers, and a. C. Long, "Normalization of Shear Test Data for Rate-independent Compressible Fabrics," *Journal of Composite Materials*, vol. 42, no. 22, pp. 2315-2344, Aug. 2008.
- [22] P. Potluri, D. Perezciurezu, and R. Ramgulam, "Measurement of meso-scale shear deformations for modelling textile composites," *Composites Part A: Applied Science and Manufacturing*, vol. 37, no. 2, pp. 303-314, Feb. 2006.

- [23] H. Kong, "Tensile extension properties and deformation mechanisms of multiaxial non-crimp fabrics," *Composite Structures*, vol. 66, no. 1-4, pp. 249-259, Dec. 2004.
- [24] S. Lomov and I. Verpoest, "Model of shear of woven fabric and parametric description of shear resistance of glass woven reinforcements," *Composites Science and Technology*, vol. 66, no. 7-8, pp. 919-933, Jun. 2006.