

Design for Additive Manufacturing (DfAM) framework for integrating structural optimization, path planning and manufacturing signatures in automated composites manufacturing.

Egor Savin

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington
2026

Committee:

Avinkrishnan Ambika Vijayachandran

Marco Salviato

Program Authorized to Offer Degree:

Aeronautics & Astronautics

©Copyright 2026
Egor Savin

University of Washington

Abstract

Design for Additive Manufacturing (DfAM) framework for integrating structural optimization, path planning and manufacturing signatures in automated composites manufacturing.

Egor Savin

Chair of the Supervisory Committee:
Avinkrishnan Ambika Vijayachandran
Department of Aeronautics & Astronautics

As composite structures become more prevalent in aerospace applications, Automated Fiber Placement (AFP) has replaced traditional hand layup as a method of choice for manufacturing large, contoured structures. However, straight fiber paths still dominate current designs. Curvilinear fiber paths offer improved stiffness distribution but are limited by manufacturing constraints and increased defect formation. This work presents a framework to design curvilinear fiber paths that are amenable to AFP-based manufacturing. A Python-based tool is developed to generate AFP models by offsetting an initial fiber path defined as a Bezier spline on a given 3D surface. Further, nonlinear stability analysis is conducted using the commercial finite element software, Abaqus to ascertain the performance under axial compressive loads. As a proof of concept, an open rectangular cutout from a curved, conical-cylindrical shell as part of a rocket fairing is considered. First, an optimal straight-fiber configuration is selected and used as the benchmark. Curvilinear fiber paths are then explored, showing an increase in the critical buckling load. The influence of manufacturing parameters on the optimal solutions is also considered.

TABLE OF CONTENTS

List of Figures	iii
List of Tables	v
Glossary	vi
Acknowledgments	vii
Dedication	viii
1 Introduction	1
1.1 Thesis Motivation	1
1.2 VAT Structures Literature Review	3
1.3 AFP Manufacturing	9
1.4 Common AFP defects	11
2 Fiber-Path Generation Methodology	13
2.1 Chapter Scope and Objectives	13
2.2 Reference Workflow in VCP AFP Software	13
2.3 Overview of the Custom AFP Model Generation Framework	17
2.4 Geometric Representation of the Tooling Surface	18
2.5 Definition of the Guide Curve	22
2.6 Definition of the Part Boundary	24
2.7 Generation of Fiber Courses	25
2.8 Manufacturing Check	29
2.8.1 Minimum Steering Radius	29
2.8.2 Gaps and Overlaps	34
2.9 Generation of the Abaqus Model	37
2.9.1 Mapping Data to FE Mesh	38
3 Computational Methods	42
3.1 Chapter Scope and Objectives	42
3.2 Finite Elements Analysis	42
3.3 Nonlinear Analysis	44

3.3.1	Newton–Raphson Method for Nonlinear Equilibrium Problems	46
3.3.2	Riks Method	48
3.4	Differential Evolution Algorithm	53
3.5	Optimization Framework	56
4	Application of the Framework to a Representative Structural Problem .	58
4.1	Chapter Scope and Objectives	58
4.2	Problem Statement	59
4.2.1	Boundary and Loading Conditions	59
4.2.2	Layup and Material Properties	60
4.2.3	Manufacturing Signatures	60
4.3	Optimization	61
4.3.1	Finite Elements Model	61
4.3.2	Straight Fibers Optimization	62
4.3.3	Optimized Fiber Paths, 2-inch course width	64
4.3.4	Optimized Fiber Paths, 4-inch course width	66
4.4	Chapter conclusions	67
5	Conclusions and Future Work	68
5.1	Summary	68
5.2	Future Work	69
	Bibliography	70
A	Example of NURBS Surface Evaluation	76

LIST OF FIGURES

1.1	Boeing 787 Dreamliner composite fuselage section material scheme	1
1.2	Manufacturing Signatures	3
1.3	AFP development timeline	3
1.4	NURBS Surface Example	4
1.5	Manufactured panels using tow-drop and overlap methods	6
1.6	Typical buckling response of different types of structures	7
1.7	VAT cylinder	7
1.8	Workflow for optimization with DFM tool	9
1.9	Typical AFP system scheme	10
1.10	AFP system during operation	11
1.11	Common AFP defects representation	11
2.1	Example of the surface, part boundary and guide curve uploaded to VCP. . .	14
2.2	Fiber courses generated using VCP.	15
2.3	Angle deviation analysis using VCP	16
2.4	Python workflow for model generation	18
2.5	Geometrical representation of the NURBS surface.	19
2.6	Representation of the NURBS surface using Python.	19
2.7	Example of the case where some points cannot be mapped to the UV domain.	21
2.8	Example of the guide curve created using discussed method.	23
2.9	Part boundary visualization.	24
2.10	Tangent, normal, and binormal unit vectors along a curve on a surface. The surface is shown by black grid lines, and the curve is shown in red. The binormal vector is defined as the cross product of the tangent and normal vectors, $\hat{\mathbf{b}} = \hat{\mathbf{t}} \times \hat{\mathbf{n}}$	26
2.11	Fiber courses generated using Python.	29
2.12	Principal curvatures.	31
2.13	Geodesic curvature on the planar surface.	32
2.14	Geodesic curvature example.	33

2.15	Comparison of steering radius measurements.	34
2.16	Gaps/Overlaps visual representation.	35
2.17	Minimum rotated bounding rectangle.	36
2.18	Overlap visualization using the Python tool.	37
2.19	Example of property assignment to mesh elements	40
2.20	Example of dataset for several elements. Element 2 contains overlap in the first ply.	41
3.1	Geometry, loads and finite element meshes.	43
3.2	Nonlinear solution procedure.	47
3.3	Snap-through behavior.	47
3.4	Snap-back behavior.	48
3.5	Arc-length parameter interpretation.	49
3.6	Arc-length constraint interpretation.	50
3.7	Flow chart of optimization procedure executed using the Python-built tool integrated with Abaqus.	57
4.1	The selected geometry is the conical portion of a universal stage adapter for the SLS rocket where the highlighted section is the selected geometry for optimization and analysis.	59
4.2	Schematic of the problem setup detailing the plate geometry, boundary con- ditions, and loading conditions.	60
4.3	Finite elements mesh.	61
4.4	Finite elements mesh.	62
4.5	Convergency plot for Riks nonlinear analysis of for non-VAT panel	63
4.6	$(0/[80/-40/30/-20]_s)_2$ force-displacement plot of the analyzed structure for the given loading and BCs, using the Riks arc-length method.	63
4.7	Best found non-VAT paths	64
4.8	Equilibrium paths of Riks nonlinear analyses for different layups.	65
4.9	Optimized paths with 2-inch course width.	65
4.10	Equilibrium paths of Riks nonlinear analyses for different layups.	66
4.11	Optimized paths with 4-inch course width.	67

LIST OF TABLES

1.1	Hand layup vs AFP comparison	2
2.1	Comparison of minimum steering-radius results.	34
4.1	Material Properties	60
4.2	Critical buckling loads and initial stiffness for different layups, 2-inch course width.	64
4.3	Critical buckling loads and initial stiffness for different layups, 4-inch course width.	66

GLOSSARY

AFP: Automated Fiber Placement.

MS: Manufacturing Signatures.

DFM: Design for Manufacturing.

DFaM: Design for additive Manufacturing.

VAT: Variable Angle Tow.

VSP: Variable Stiffness Panel.

CoV: Coefficient of Variation.

VCS: VERICUT Composite Simulation.

VCP: VERICUT Composites Programming.

FEM: Finite Elements Method.

FEA: Finite Elements Analysis.

DE: Differential Evolution.

ACKNOWLEDGMENTS

There were many challenges throughout this two year path to the Master's degree. Here I wish to express my gratitude to the people who supported me on this journey.

Professor Avinkrishnan Vijayachandran, my advisor. He suggested me the topic for research project during his Mechanics of Solids class and helped me to stay confident since my first quarter at University of Washington.

I am deeply grateful to all teaching faculty of the University of Washington and particularly those in Aeronautics & Astronautics department for their guidance and encouragement. I would especially like to thank Dr. Marco Salviato, Dr. Ed Habtour, and Dr. Peter Mackenzie-Helnwein, each of whom contributed to my academic journey in their own way, but all inspired me in my decision to begin a Ph.D.

I also personally thank my family for supporting me during these years. My special gratitude goes to my parents, Sergey and Margarita, and my grandparents, Liubov, Anatoliy, and Tatiana. I want to acknowledge my aunt, Elena and my uncle, Oleksii. I am equally thankful to my wife, Iuliia. I would not be where I am today without all of you.

Finally, I would like to thank all my friends for their encouragement and companionship along the way.

DEDICATION

to my parents, Sergey and Margarita and to my wife, Iuliia

Chapter 1

INTRODUCTION

1.1 Thesis Motivation

Composite materials play a key role in modern aerospace industry. They provide high stiffness and strength to weight ratio, which allows to design modern structures with incredible characteristics. Most famous examples are Boeing 787 Dreamliner Airplane and Airbus A350 XWB. The Boeing 787 is comprised of 80% composite materials by volume and 50% by mass [1]. The largest share is used in fuselage and wings as shown in Figure 1.1. Composites allow the 787 to maintain pressurization at an altitude of about 6,000 feet, which helps reduce the health impacts of long flights [2].

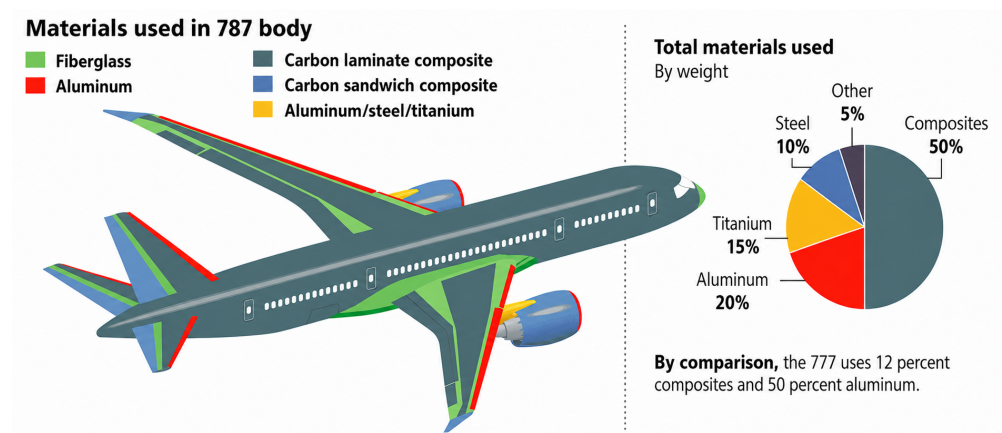


Figure 1.1: Boeing 787 Dreamliner composite fuselage section material scheme [1].

Since late 1990's automated fiber placement (AFP) has been actively replacing traditional manual layup methods. This is especially important in applications where high repeatability, production efficiency, and precise fiber placement are required. For example, repeatability coefficient of variation (CoV) for AFP parts is less than 5%. This and other key comparison metrics are presented in Table 1.1. Transition from hand layup to AFP is important both from a manufacturing point of view and a structural design perspective, because AFP makes it possible to create laminate configurations that are difficult or even impossible to manufacture using hand layup.

One of the key opportunities provided by AFP is the use of curvilinear fiber paths.

Table 1.1: Hand layup vs AFP comparison [3].

Metric	Hand Layup	Modern AFP	Improvement
Production Speed	2–3 kg/hour	10–150 kg/hour	40x faster
Material Waste	20–50%	less than 6%	88% reduction
Labor Requirement	5–10 skilled workers	1–2 operators	80% reduction
Placement Accuracy	± 2 –5 mm	± 0.05 –0.08 mm	50x more precise
Repeatability (CoV)	greater than 10%	less than 5%	2x better
Setup Time	Minutes	2–4 hours	Slower initial setup
Part Cost Reduction	Baseline	43% at 150+ parts/year	43% savings
Capital Investment	\$10K–100K	\$300K–500K, or \$3.5K/month	Higher but accessible

In traditional composite laminates, fibers are usually placed along constant directions. This approach is well known and used in most real-world applications. The disadvantage of this method is that it limits the designer’s ability to locally adapt the parameters of the structure. Many composite components experience non-uniform stress fields, especially near holes and cut-outs or just in distributed-loading cases. In such situations, straight fiber orientations may not be the most efficient solution, because they cannot fully follow the load paths inside the structure.

Structures with curvilinear fibers are also called variable stiffness or variable angle tow (VAT) structures. The concept of variable stiffness panels (VSP) was introduced by Gurdal and Olmedo in 1993 [4]. Following the earlier development of variable-stiffness laminates, Weaver et al. [5] adopted the term VAT to describe laminates with varying fiber orientations. In such parts the fiber distribution can be changed according to mechanical requirements. As a result, structural performance may be improved without an increase in mass or a change in geometry. This is very relevant in aerospace structures, where geometry of composite parts is usually depends on aerodynamics and mass is critically important for the fuel efficiency. The practical use of VAT structures introduces new challenges. The design of each ply becomes more complex, because the fiber orientation is no longer described by a constant angle, but by some distribution across the structure. This distribution should be chosen not just to fulfill mechanical requirements, but also adhering to manufacturing signatures (MS). MS can be divided by three big groups as shown in Figure 1.2.

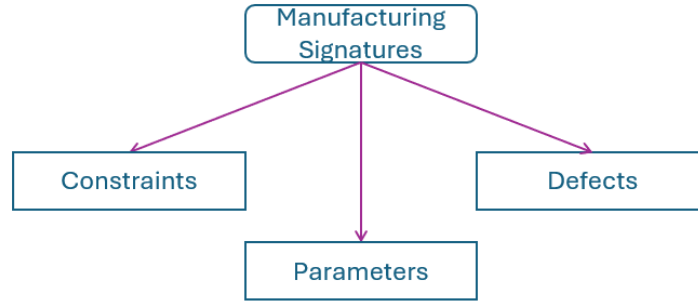


Figure 1.2: Manufacturing Signatures.

Constraints are conditions that cannot be violated, such as the minimum steering radius. Parameters are manufacturing characteristics that can be selected from a set of available values, for example, the fiber course width or curing temperature. Gaps and overlaps are treated as defects, although their size can also be constrained during optimization.

A better understanding of the impact of variable fiber orientations on structural behavior can help with the creation of more efficient composite parts. The work on the boundary between structural design and manufacturing is important for transferring mechanical concepts from theoretical studies to practical industry applications.

1.2 VAT Structures Literature Review

The timeline of AFP development is presented in Figure 1.3.

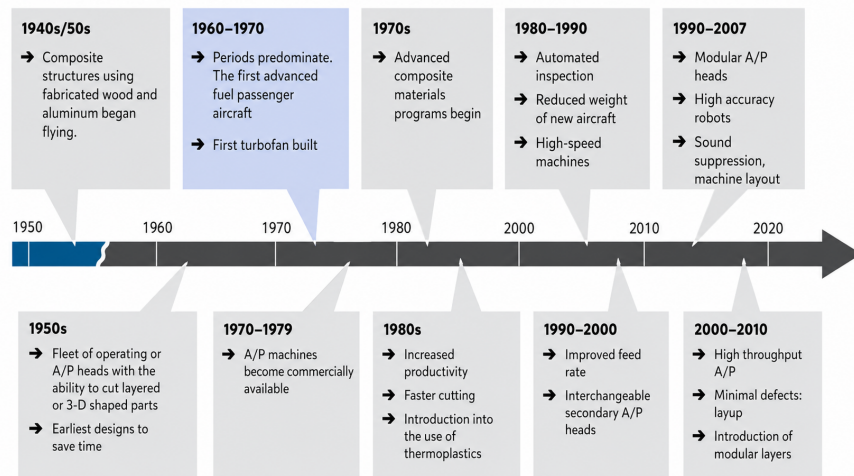


Figure 1.3: AFP development timeline [6].

One of the first and well-known papers about curvilinear fibers application appeared

in 1991 by Hyer and Lee. They reported the increase of buckling resistance for composite panel with a circular hole [7]. Their study established the structural potential of curvilinear fibers, although the conversion of the optimized orientation field into manufacturable finite-width tow courses was not the main focus. Later Gurdal and Olmedo introduced concept of variable stiffness [4]. Their work provided closed-form solutions of several mechanical problems describing the in-plane response of laminates whose stiffness properties vary across the structure. From the beginning of this area two different approaches of parameterization started to form. The first one is fiber paths optimization on finite element level [7]. The second is geometry-level optimization of paths with future discretizations for analysis. An example of this strategy was presented by Nagendra et al. [8]. Authors of this manuscript use NURBS (Non-Uniform Rational B-Splines) approach for optimization.

NURBS are mathematical representations used to model smooth curves and surfaces. Their geometry is controlled through control points, weights, and knot vectors, providing high flexibility and local control. Example of surface created by this method is presented in Figure 1.4. Compared with elementwise orientation optimization, geometry-based path descriptions are more closely connected to CAD and CAM practice. They also provide a more natural basis for introducing geometric and manufacturing constraints, such as path continuity, curvature limits, and tow steering requirements [9].

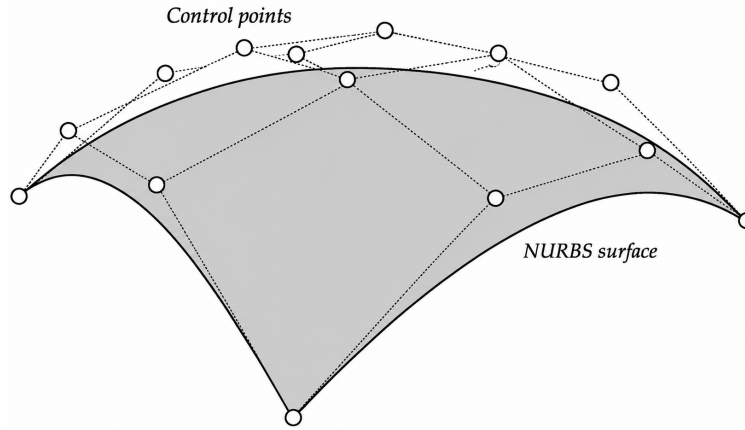


Figure 1.4: NURBS Surface Example [10].

Following the early numerical studies, Tatting and Gurdal suggested mathematical formulation for fiber paths with varying angle [11]. Their work connected path definition, finite-element analysis, and automated tow placement. They also developed the tool for design and analysis of VAT flat panels and manufactured several of these panels, which became an important milestone.

The reference path is defined by six scalar parameters: x_0 , y_0 , T_0 , ϕ , d , and T_1 , where

- $(x_0, y_0) \rightarrow$ coordinates of the origin of the reference path,
- $T_0 \rightarrow$ fibre orientation angle of the reference path at the origin,
- $\phi \rightarrow$ principal direction of the fibre-angle variation,
- $d \rightarrow$ characteristic length measured in the ϕ -direction,
- $T_1 \rightarrow$ fibre orientation angle of the reference path at d .

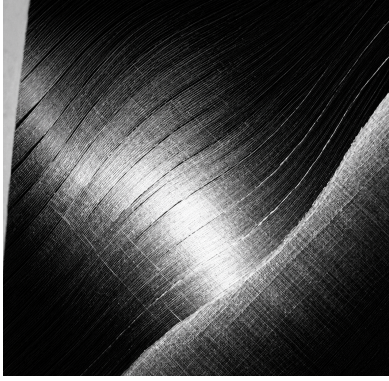
These parameters define a reference path with a linear variation of the fiber orientation angle:

$$\theta(\gamma) = \begin{cases} \phi + \frac{(T_1 - T_0)\gamma}{d} + T_0, & 0 < \gamma < d, \\ \phi + \frac{(T_0 - T_1)(\gamma - d)}{d} + T_1, & d < \gamma < 2d. \end{cases} \quad (1.1)$$

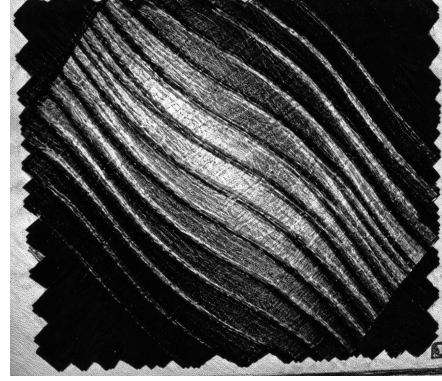
To generate a complete ply from the reference path, additional courses are created by shifting the curve in a direction perpendicular to the ϕ -direction. The shift distance is selected so that no gaps occur between adjacent courses. However, due to the curvature of the reference path and the offset-based construction of neighboring courses, overlaps may develop between adjacent tow paths [11].

Two strategies were suggested to manage these overlaps. In the *tow-drop method*, individual tows are cut or terminated during placement to prevent local thickness build-up in overlapping regions. In the *overlap method*, the complete tow courses are retained, allowing adjacent paths to overlap and thereby producing localized increases in laminate thickness [11]. Panels manufactured using both of these strategies presented in the Figures 1.5a and 1.5b.

Meanwhile, researchers tried to develop alternative approaches for VAT structures optimization. Parnas et. al presented methodology where layer thicknesses and fiber angles are represented by Bézier surfaces and cubic Bézier curves, respectively [12]. The coordinates of the Bézier control points were used as design variables. This strategy both ensures continuity of fiber paths and allows to work with complex, curvilinear parts. The authors applied this approach to several planar problems and successfully minimized weight. During the 2000's, much of the VAT literature remained focused mostly on numerical optimization and structural analysis rather than on detailed manufacturing effects [13], [14]. Lopes et al. [15] performed buckling, progressive damage and failure analyses with testing. For the flat panel with a circular hole, a 56% strength increase was reported for curvilinear-fiber laminates



(a) Tow-drop method.



(b) Overlap method.

Figure 1.5: Manufactured panels using (a) the tow-drop method and (b) the overlap method [11].

compared to straight-fiber laminates.

Gürdal et al. [16] further clarified the mechanics of variable-stiffness panels. In straight-fiber angle-ply laminates, changing the fiber orientation to modify the panel's in-plane stiffness also alters its buckling load, whereas in variable-stiffness panels, one of these characteristics can be changed while the other is kept constant.

Most papers about VAT structures are concentrated on flat structures. This is suitable for demonstrating the basic mechanics and validating numerical methods, but most real aerospace applications are curved structures, including cylindrical and conical shells, fuselage sections, pressure vessels, and fuel tanks. The response of curved shells cannot be inferred directly from flat-plate behaviour because shell curvature strongly influences the prebuckling state, geometric imperfections, instability modes, and postbuckling response. Under axial compression, the buckling response of a geometrically perfect structure can differ substantially from that of an imperfect structure. Thin-walled cylindrical shells are very sensitive to initial geometric imperfections, which can lead to buckling loads significantly below the classical predictions for perfect shells [17]. This pronounced imperfection sensitivity is one of the main reasons why large buckling-load knockdown factors are commonly used in shell design. Comparison of the behavior of different structures under an axial load is given in Figure 1.6.

Blom et al. [19] extended VAT research from flat panels to curved structures by considering conical shells. The authors developed the method of tracing local fiber orientations and overlaps on a conical shell with a variable-stiffness ply constructed by fiber placement. This work was an important step toward applying variable stiffness to more representative aerospace geometries. Blom et al. [20] later investigated a manufactured variable-stiffness composite cylinder. Experimental testing showed 18 % buckling load increase of VAT cylin-

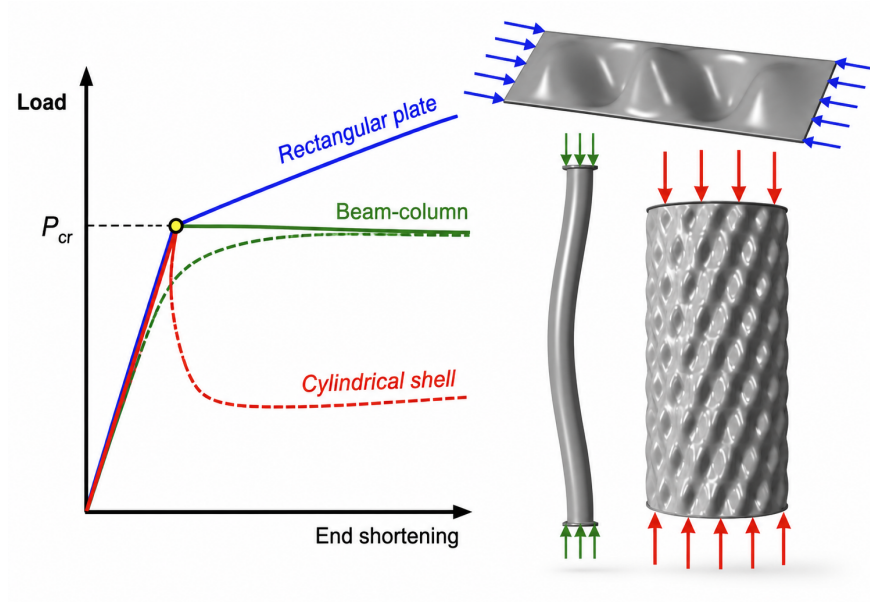


Figure 1.6: Typical buckling response of different types of structures [18].

der compared to the chosen straight fiber baseline. The manufactured cylinder is presented in Figure 1.7.



Figure 1.7: VAT cylinder [20].

Although VAT structures prototypes were manufactured through 1990'-2000's, manufacturing defects received limited attention. Early studies mentioned tow drops, gaps, overlaps, and local thickness changes, but detailed experiments, defect modeling, and defect-aware optimization became more common in the early 2010's.

Kim et al. described manufacturing limitations of VAT panels and possible defects [21]. The authors say that most process-induced defects are caused by in-plane bending of

the tow, which also limits the minimum achievable tow-path radius of curvature. In 2013 Fayzabakh et al. [22] captured effects of gaps and overlaps in variable stiffness laminates made by AFP. Their results showed that when gaps are included in the model, the predicted buckling improvement decreases. Lozano et al. [23] reviewed almost three decades of research on variable stiffness laminates. The authors concluded that the practical implementation of these findings remains challenging because the integration of manufacturing considerations into the design phase of variable-stiffness structures has not yet been successfully achieved. Nevertheless, they admit that the use of curvilinear fibers presents great potential to improve the performance of composite structures.

The absence of real-world applications highlighted the need for approaches and models that integrate design and manufacturing. Belnoue et. al discussed how to predict defects formed during AFP layup process [24]. Their work showed that not only fiber paths are important, but also curing process parameters, such as temperature and pressure.

Vijayachandran et al. [25] proposed a fiber-course modeling approach based on Bézier splines. Individual tow courses were represented explicitly, allowing manufacturing signatures such as minimum radius of curvature and gaps/overlaps to be considered during the optimization process. This represented a further shift toward optimizing manufacturable tow trajectories. During the 2020s, manufacturing constraints became increasingly important in VAT optimization. Vijayachandran and Waas [26, 27], and Zhao and Kapania [28] investigated steered-fiber designs for different structural problems and found optimized steered fiber paths. Manufacturing signatures such as minimum steering radius and gap/overlap size were considered during optimization process. These studies linked manufacturing parameters to mechanics definitions such as stress concentration, compressive response, and buckling resistance.

The concept of Design For Manufacturing (DFM) is not new. Its application to VAT structures design is still developing. Integrating AFP process planning into the optimization stage can exclude designs that are good from structural point on view, but difficult or impossible to manufacture. Tools such as VERICUT Composite Programming (VCP) can be used to evaluate tow paths, course boundaries, steering limits, and cut-and-restart operations for designs during optimization. Noeverre and Jamora [29] presented their framework (Figure 1.8) which unites structural optimization, process planning and path generation. This type of workflow represents the current trend of VAT research, where structural performance and manufacturability are evaluated together during the design process.

Digital-twin approaches provide a further extension of this idea. Francis et al. [30] also discuss digital twins concept in AFP to connect design, manufacturing, inspection, and structural analysis. Such approaches are beneficial for VAT structures because the true

manufactured fiber paths and defects can differ from the ideal design and may be subject to include in the final structural model.

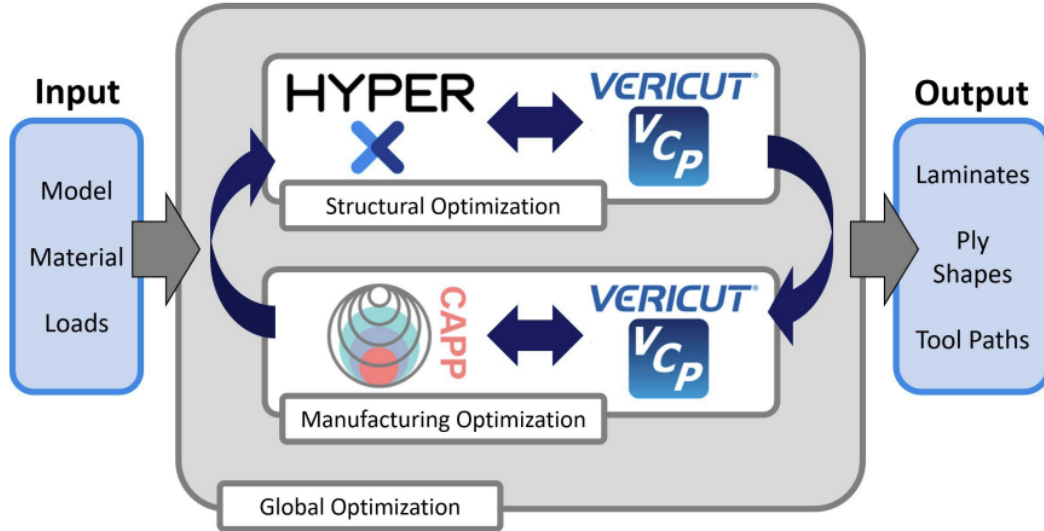


Figure 1.8: Workflow for optimization with DFM tool [29].

The literature shows a transition from the numerical optimization of idealized fiber-orientation fields toward integrated design frameworks that include manufacturing signatures (such as tow-path generation, manufacturing constraints, defect prediction, process planning) and structural analysis of the precise digital twin.

1.3 AFP Manufacturing

AFP is advanced technology to manufacture composite parts. The material is normally supplied as prepreg: continuous fibers that have already been impregnated with a partially cured polymer resin. AFP developed from automated tape laying (ATL), but instead of one wide tape, an AFP machine places a band of several narrow prepreg strips, commonly called tows. This allows the material to follow more highly curved and geometrically complex surfaces.

The main part of AFP machine is the placement head, which is mounted on an industrial robot. Individual tows are unwound from spools and guided through the head. Before deposition, a backing film may be removed and the material is heated to soften the resin. The tows then pass beneath a compaction roller, which presses them against the mold or against previously deposited layers. The combined effects of heat and pressure hold the new course in position and reduce trapped air between layers. The scheme of AFP system is presented in Figure 1.9.

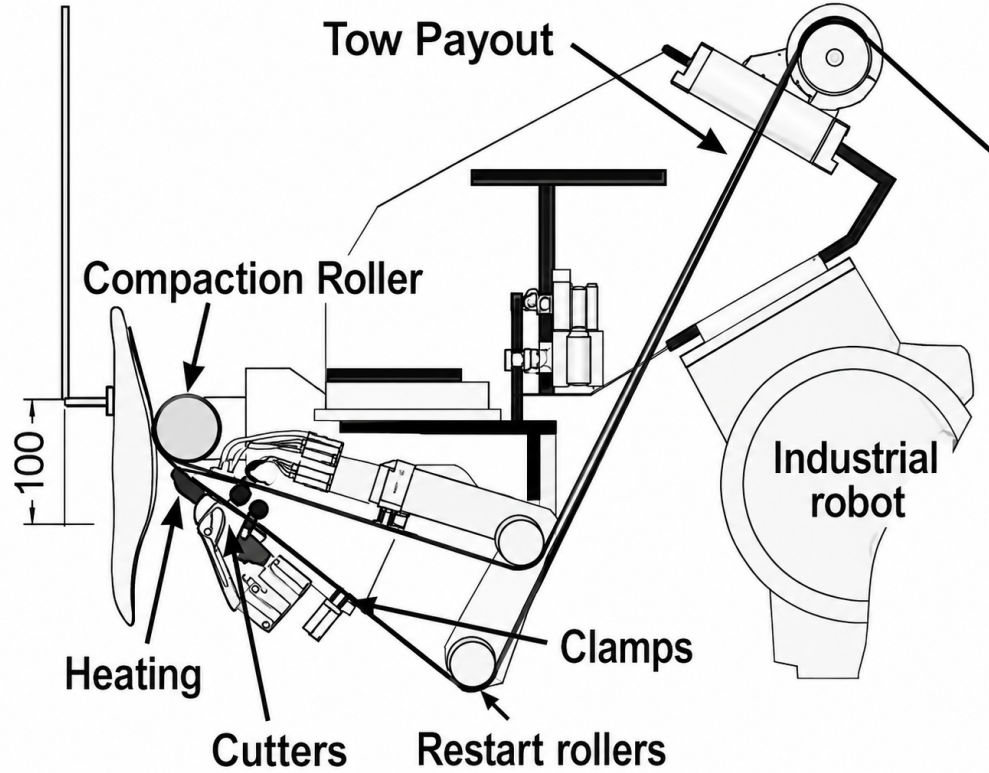


Figure 1.9: Typical AFP system scheme [31].

During manufacturing, the placement head follows a programmed path over the tool surface. Several tows are deposited side by side to form a course. Each tow can normally be started, stopped, and cut independently. Consequently, the machine can change the width of a course, place material around openings and follow prescribed ply boundaries.

A complete component is produced by placing many courses and layers. The fiber direction can be changed between layers according to the laminate design. Courses may also be steered along non-straight paths. However, excessive steering can cause the material to buckle, wrinkle, separate, or produce gaps and overlaps between adjacent tows. Accurate path planning and control of the processing parameters are therefore essential parts of AFP operation.

For thermoset preregs, the placed laminate is usually vacuum-bagged and subsequently cured in an oven or autoclave. In thermoplastic AFP, stronger heating may be applied during placement so that the incoming material bonds or consolidates directly with the layer below. In both cases, AFP performs the controlled positioning and compaction of the reinforcement, while the final curing or consolidation stage forms the finished composite structure.



Figure 1.10: AFP system during operation [32].

1.4 Common AFP defects

Defects across each individual ply may occur during AFP layup. Harik et al. [33] done the systematic classification of such defects. They defined 14 possible defects and described anticipation, existence, significance and progression of each of them. Schematic images of all of them are presented in Figure 1.11

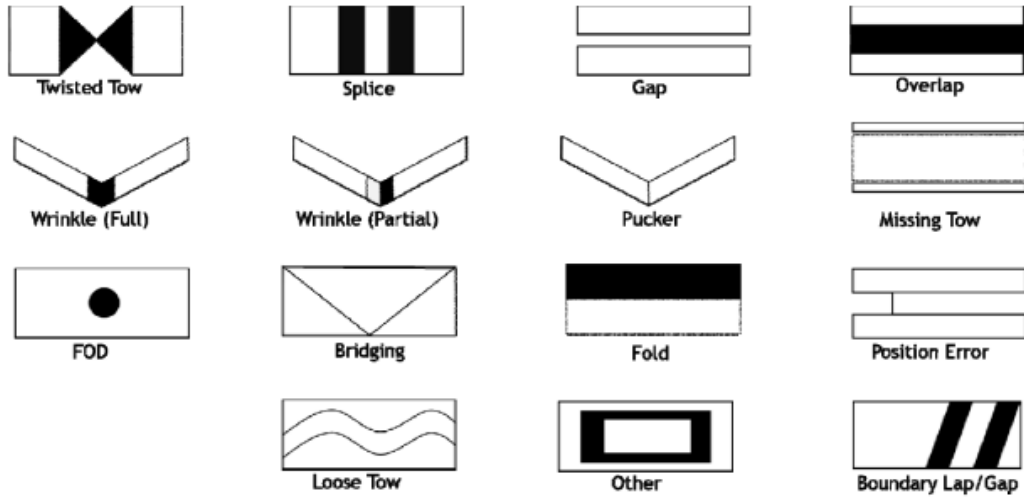


Figure 1.11: Common AFP defects representation [34].

For the purpose of this research, defects can be organized by three big groups, although this classification is conditional.

The primary focus is on defects that are mostly related to path planning and machine parameters, so they can be anticipated during the design phase. These include gaps, overlaps, puckers, wrinkles, and boundary-coverage defects. Their occurrence or likelihood can be evaluated using parameters such as tow width, path planning, steering radius, surface geometry, ply-boundary definition and some prescribed roller/head parameters. Of course they can occur even after the accurate simulations, but the main characteristic of this group is that they can be identified before material deposition. The second group contains defects that almost equally depend on the design, but external process factors also have a strong impact on them. This group includes loose tows, wandering tows, twists, folds, angle deviations, and bridging. Geometry and head trajectories may indicate regions of risk, but reliable prediction also requires material adhesion, deviations in tow tension, tow supply and machine calibration. The third group organizes defects that generally cannot be predicted from nominal path-planning parameters or roller/head parameters. Missing tows, foreign-object debris, splices, and position errors usually result from material-supply conditions, cutting failures, machine or calibration errors. These defects may be detected or prevented through process monitoring and quality control, but they are unlikely to predict during the design phase. The first defect group is the most important for this research, since it contains defects whose formation can be predicted primarily from known geometric and process-planning parameters.

Chapter 2

FIBER-PATH GENERATION METHODOLOGY

2.1 Chapter Scope and Objectives

This chapter presents the methodology developed to generate VAT composite laminate models. The main goal of the generation procedure is to create a model based on tooling geometry, physical part boundary and guide curve definition. This model can be later studied for potential defects, transferred into FEM input file or just serve as a visual representation of the laminate.

A custom tool was therefore developed in Python to reproduce the principal geometric operations required for the study. These operations include import of the tooling surface, layup boundary and initial guide curve, iterative generation of fiber courses, identification of defects, and transfer of the resulting fiber architecture to a Finite-Element Model in commercial in Abaqus, a commercial software. The developed tool should be interpreted as a research-oriented framework rather than as a complete replacement for commercial AFP process-planning software. In particular, it does not model the complete kinematics of the AFP machine, tow-cutting and restart operations, collision avoidance or detailed process parameters. Its purpose is to make conservative models which can be later used for structural analysis.

A commercial AFP modeling solution named Vericut Composites Programming (VCP) is also reviewed in this chapter and used as a reference. The common steps are standard and share the principal stages of the modeling workflow, including surface handling, boundary definition, guide curve creation, course generation, manufacturing check, and export to finite elements software.

2.2 Reference Workflow in VCP AFP Software

VERICUT Composite Programming (VCP) is a commercial software developed for the design of AFP parts. For the purposes of this research, the VCP workflow is considered through four principal stages: laminate definition, process setup, course generation, and manufacturability analysis.

The laminate-definition stage establishes the geometric basis of the model. The tool-

ing surface and the physical part boundary are imported from a CAD model. For constant-angle laminates, a separate guide curve is not required because the fiber direction can be prescribed according to the selected coordinate system. For VAT laminates the guide curve is a critical input because it controls the initial fiber trajectory and strongly impacts on the fiber paths across the surface. Example of surface with boundary and guide curve is presented in Figure 2.1.

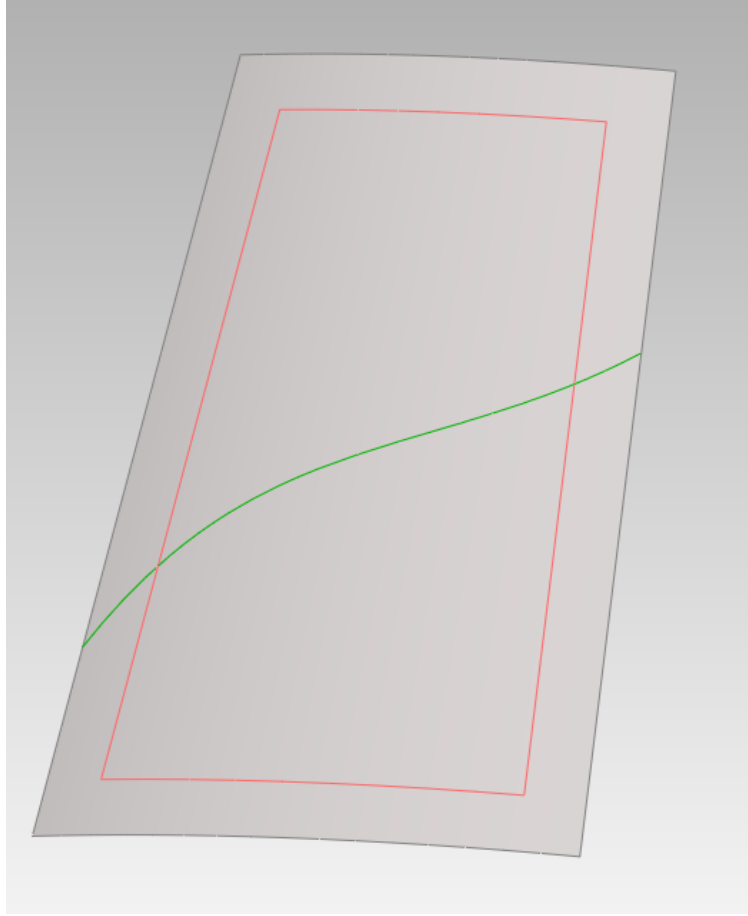


Figure 2.1: Example of the surface, part boundary and guide curve uploaded to VCP.

The setup stage defines the material, head, roller and layup parameters. Material properties important for course generation include the thickness, individual tow width, allowable steering radius, and permissible angle deviation from the prescribed direction. Roller settings include the allowable compression. Crucial head parameters include the number of tows within a course and the minimum allowable length. The layup settings define the permissible gap and overlap limits and also include the course-offset algorithm together with additional specific options. During the generation stage, the software constructs the AFP courses over the defined surface. Course generation is initiated from a prescribed starting

location and is performed relative to the selected coordinate system, guide curve and other settings. Example of generated fiber courses is presented in Figure 2.2.

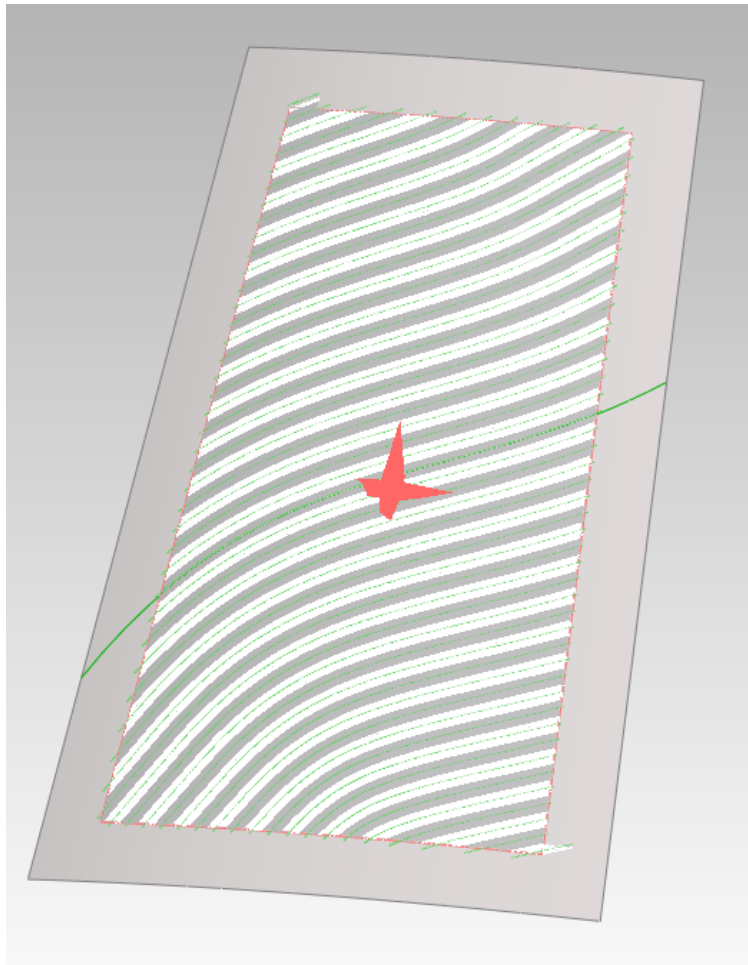


Figure 2.2: Fiber courses generated using VCP.

The analysis stage is used to evaluate the manufacturability of the generated layup. Typical checks include violations of the minimum steering radius, angular deviation, local gaps and overlaps size, tow-length restrictions. The results allow the user to identify regions in which the proposed layup may be difficult or impossible to manufacture. Example of angle deviation analysis using VCP is presented in Figure 2.3.

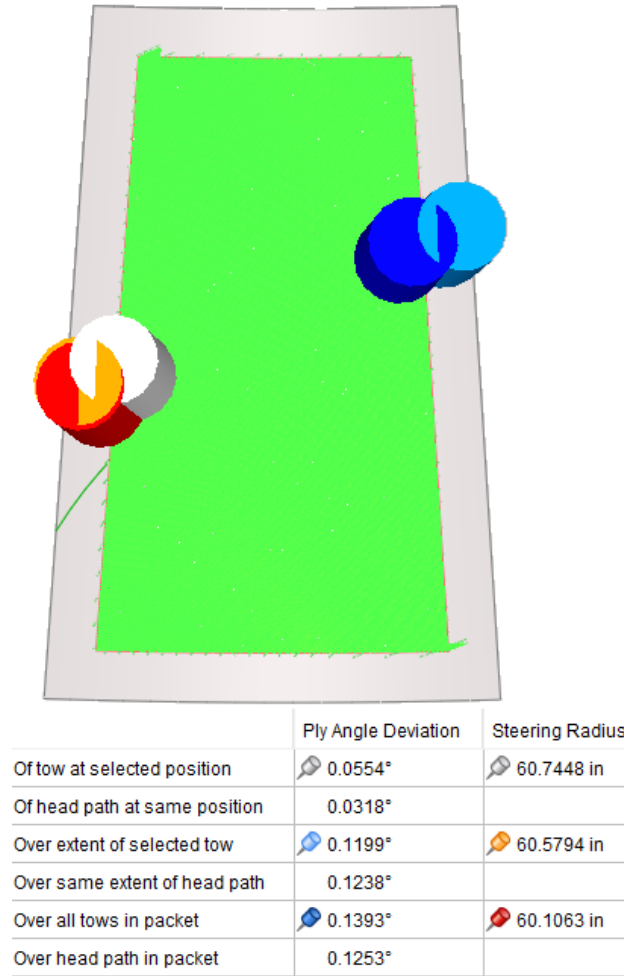


Figure 2.3: Angle deviation analysis using VCP.

Pin color meanings

- White** – Tow characteristics at the selected position
- Light blue** – Angle deviation along the selected tow
- Dark blue** – Angle deviation across the tows in the course
- Orange** – Radius of curvature along the selected tow
- Red** – Radius of curvature across the tows in the course

The description provided here represents only a high-level overview of the VCP workflow. In practice, the software provides a wider range of tools for AFP. In the present work, VCP is used primarily as a reference for identifying the principal modeling stages that were reproduced in simplified form by the custom Python tool.

2.3 Overview of the Custom AFP Model Generation Framework

The custom AFP model generation framework was developed in Python to replicate principal steps to create VAT laminate models. The framework is based on three primary geometric inputs: the tooling surface, the physical part boundary, and an initial guide curve. The guide curve defines the reference trajectory from which the remaining courses are generated.

The developed procedure follows the same general modeling logic used in commercial AFP software, although its implementation is considerably simplified. First, the surface, boundaries, material data, course width, and guide curve definition are initialized. The guide curve is then treated as the centerline of the first fiber course. The two boundaries of this course are generated by offsetting the centerline by half of the prescribed course width. Additional course centerlines are subsequently produced by iteratively offsetting the previously generated centerline. The corresponding course boundaries are constructed in the same manner, and the procedure is repeated until the part area has been covered.

Since the fiber courses are generated on a curved surface, points along the centerline are offset in a direction that lies within the *local tangent plane* of the surface and is normal to the *local course tangent*. The displaced points are then projected back onto the tooling surface. A smooth spline is fitted through the resulting points to define the next course centerline or boundary. This procedure allows adjacent fiber courses to follow the surface geometry while preserving the prescribed nominal course width and spacing. The generated course boundaries are combined into closed polygons representing the regions occupied by individual courses. These polygons are used to determine whether newly generated courses are inside the part boundary and should therefore be retained. They are also used to identify gaps and overlaps between courses. The geometric operations are primarily performed in the parametric domain of the surface, where polygon containment, intersection, and Boolean operations can be evaluated more conveniently.

Finite-elements mesh is imported to the framework via the template input file. The element centroids and nodes are projected onto the surface parameter domain. After the full course layout has been generated, the laminate is transferred to the mesh. Each element has its own section which should represent ply stacking over the structure. The framework determines the courses that contain its centroid, evaluates the local fiber direction from the nearest course centerline, and assigns the corresponding material, orientation, and thickness data to the section. Regions without fiber coverage are represented as areas filled by purely matrix, whereas elements located within overlapping courses receive multiple local material properties. This process is iterative for all plies.

The final output of the framework is an Abaqus input file containing the composite

section definitions required for structural analysis. In addition to the finite-element model, the framework can provide a model of the generated laminate and information about gaps and overlaps. The overall procedure is presented in Figure 2.4.

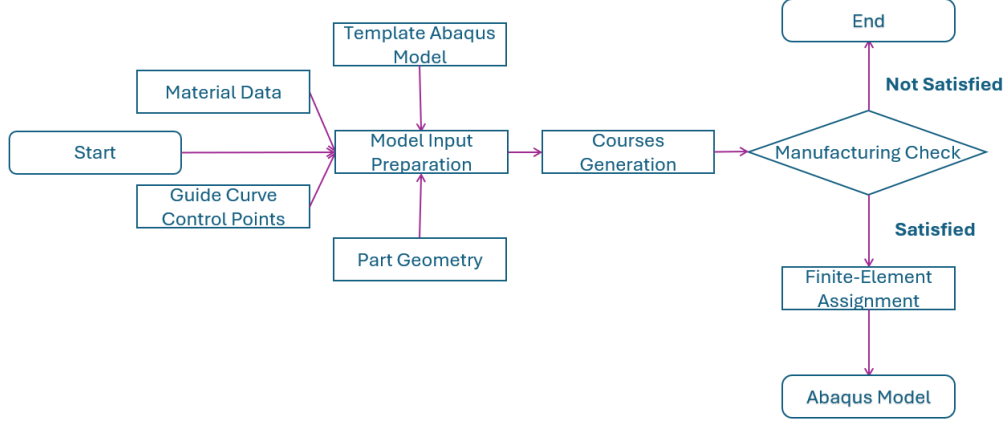


Figure 2.4: Python workflow for model generation.

2.4 Geometric Representation of the Tooling Surface

The entire geometry is represented using a non-uniform rational B-spline (NURBS) surface. NURBS are widely used in computer-aided design because they provide a compact mathematical representation of both free-form surfaces and common analytical geometries. A NURBS surface is defined by a set of control points, associated weights, polynomial degrees, and knot vectors in two independent parametric directions. In its general form, the surface is expressed as

$$\mathbf{S}(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) M_{j,q}(v) w_{ij} \mathbf{P}_{ij}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) M_{j,q}(v) w_{ij}}, \quad (2.1)$$

where $\mathbf{P}_{ij}$ are the surface control points, w_{ij} are the corresponding weights, and $N_{i,p}$ and $M_{j,q}$ are B-spline basis functions of degrees p and q , respectively. The variables u and v are surface parameters rather than physical coordinates. The main advantage of this method is that any surface can be presented in two dimensional parametric domain UV:

$$\Omega_{uv} = \{(u, v) \in \mathbb{R}^2 \mid u \in [0, 1], v \in [0, 1]\}, \quad (2.2)$$

and subsequently mapped to its physical three-dimensional geometry through $\mathbf{S}(u, v)$.

A complete numerical example of the mapping between the parametric and physical domains is provided in Appendix A. The geometrical representation is given in Figure 2.5.

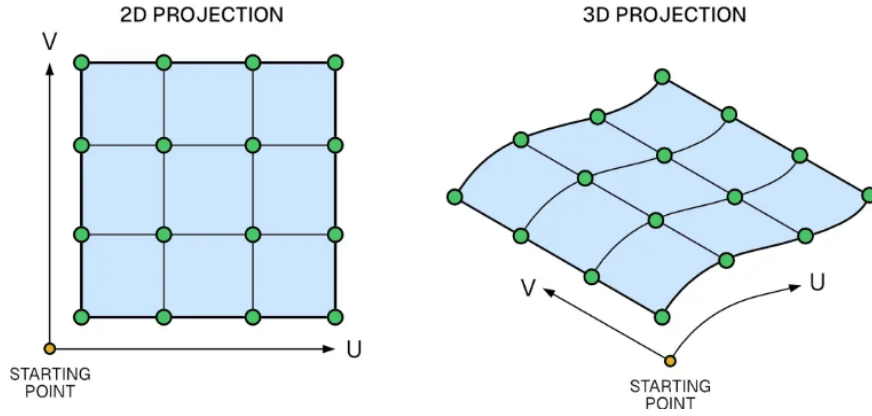


Figure 2.5: Geometrical representation of the NURBS surface [35].

To work with geometry in Python, the `geomdl` and `OCP` libraries were used. Original tooling surface is imported from the `.step` file and provides basis for the laminate model. All guide curves, fiber-course centerlines, course edges, layup boundaries, strictly and other geometric entities must be defined relative to it. Example of the NURBS surface with control points imported from CAD `.step` file is presented in Figure 2.6.

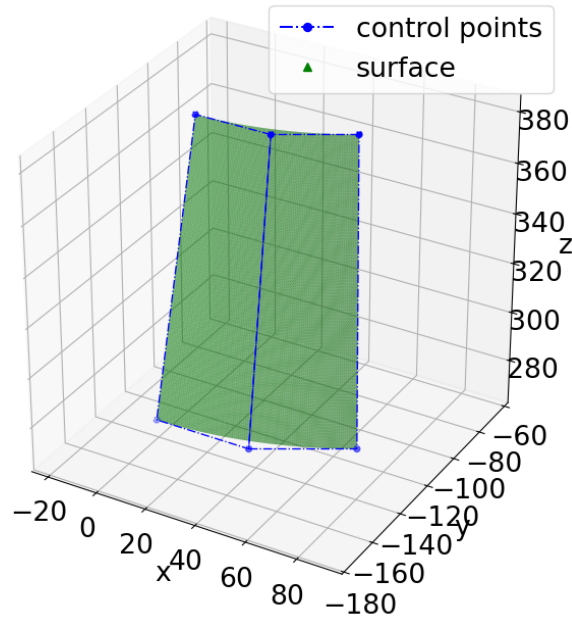


Figure 2.6: Representation of the NURBS surface using Python.

Most geometric operations are performed using UV domain. As can be seen from 2.1 and the example in Appendix A, it is not so difficult to map from UV domain to XYZ

domain using formulas, but the reverse operation remains challenging. For this reason, numerical methods are required to do this. Gauss–Newton algorithm was chosen for this research to make mapping from XYZ domain to UV domain possible. The problem is therefore formulated as a nonlinear least-squares problem and solved iteratively using the given method [36].

The NURBS surface is a vector-valued function of the two surface parameters, u and v . Its value represents the Cartesian coordinates of the corresponding point in the physical domain:

$$\mathbf{S}(u, v) = \begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{bmatrix}, \quad (2.3)$$

where $x(u, v)$, $y(u, v)$, and $z(u, v)$ are the Cartesian coordinate functions of the surface.

For a prescribed physical point $\mathbf{P}$ which should be mapped on this surface, the residual vector is defined as

$$\mathbf{r}(u, v) = \mathbf{S}(u, v) - \mathbf{P}, \quad (2.4)$$

where $\mathbf{P}$ is constant with respect to the surface parameters. Therefore,

$$\frac{\partial \mathbf{P}}{\partial u} = \frac{\partial \mathbf{P}}{\partial v} = 0 \quad (2.5)$$

And

$$\frac{\partial \mathbf{r}}{\partial u} = \frac{\partial \mathbf{S}}{\partial u} = \mathbf{S}_u, \quad \frac{\partial \mathbf{r}}{\partial v} = \frac{\partial \mathbf{S}}{\partial v} = \mathbf{S}_v. \quad (2.6)$$

The residual Jacobian is consequently

$$\mathbf{J}_r = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_u & \mathbf{S}_v \end{bmatrix}. \quad (2.7)$$

Starting from an initial estimate

$$\boldsymbol{\xi}^{(0)} = \begin{bmatrix} u \\ v \end{bmatrix}^{(0)}, \quad (2.8)$$

the Gauss–Newton method updates the surface parameters according to

$$\boldsymbol{\xi}^{(i+1)} = \boldsymbol{\xi}^{(i)} - (\mathbf{J}_r^T \mathbf{J}_r)^{-1} \mathbf{J}_r^T \mathbf{r}(\boldsymbol{\xi}^{(i)}). \quad (2.9)$$

The parameter increment is defined as

$$\Delta \boldsymbol{\xi}^{(i)} = \boldsymbol{\xi}^{(i+1)} - \boldsymbol{\xi}^{(i)}. \quad (2.10)$$

Therefore,

$$(\mathbf{J}_r^T \mathbf{J}_r) \Delta \boldsymbol{\xi}^{(i)} = -\mathbf{J}_r^T \mathbf{r}(\boldsymbol{\xi}^{(i)}). \quad (2.11)$$

This linear system is solved for

$$\Delta \boldsymbol{\xi}^{(i)} = \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}^{(i)}, \quad (2.12)$$

after which the parameter estimate is updated as

$$\boldsymbol{\xi}^{(i+1)} = \boldsymbol{\xi}^{(i)} + \Delta \boldsymbol{\xi}^{(i)}. \quad (2.13)$$

The iteration is terminated when

$$\|\Delta \boldsymbol{\xi}^{(i)}\| \leq \varepsilon. \quad (2.14)$$

There exist cases where this method cannot converge by definition. For example, during offsetting some parts of the curve might be outside the domain, so they should be trimmed. An illustrative example is shown in Figure 2.7.

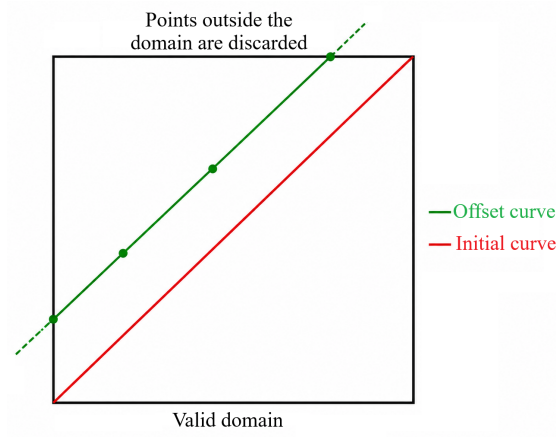


Figure 2.7: Example of the case where some points cannot be mapped to the UV domain.

2.5 Definition of the Guide Curve

The guide curve defines the reference trajectory from which the fiber courses of VAT laminate are generated. In the developed framework, the guide curve is represented by a cubic Bézier spline defined by four control points. The advantage of this approach is the granted continuity of modeling fibers geometry which is an important MS to be considered. A Bézier spline can be treated as a special case of a B-spline or NURBS curve with a clamped knot vector and no internal knots.

The guide-curve control points are specified in the parametric domain of the tooling surface. This strategy was used to ensure that the Bézier spline defined by these points is going to lie on the imported tooling surface. The expression of parametric guide curve is:

$$\mathbf{C}(t) = \sum_{i=0}^3 B_{i,3}(t) \mathbf{P}_i, \quad t \in [0, 1], \quad (2.15)$$

where $B_{i,3}(t)$ are the cubic Bernstein basis functions and P_i , are the guide-curve control points in the UV domain:

$$\begin{aligned} B_{0,3}(t) &= (1 - t)^3, \\ B_{1,3}(t) &= 3t(1 - t)^2, \\ B_{2,3}(t) &= 3t^2(1 - t), \\ B_{3,3}(t) &= t^3. \end{aligned} \quad (2.16)$$

Therefore, the curve can also be written explicitly as

$$\begin{aligned} \mathbf{C}(t) &= (1 - t)^3 \mathbf{P}_0 + 3t(1 - t)^2 \mathbf{P}_1 \\ &\quad + 3t^2(1 - t) \mathbf{P}_2 + t^3 \mathbf{P}_3. \end{aligned} \quad (2.17)$$

The same curve may be represented as a cubic B-spline:

$$\mathbf{C}(t) = \sum_{i=0}^3 N_{i,3}(t) \mathbf{P}_i, \quad (2.18)$$

with polynomial degree

$$p = 3 \quad (2.19)$$

and clamped knot vector

$$T = \{0, 0, 0, 0, 1, 1, 1, 1\}. \quad (2.20)$$

Because the knot vector contains no internal knots and the curve has $p+1 = 4$ control points, the cubic B-spline representation is equivalent to the cubic Bézier representation in Equation (2.15).

Evaluation of the guide curve in the parametric domain gives:

$$\mathbf{C}(t) = \begin{bmatrix} u(t) \\ v(t) \end{bmatrix}. \quad (2.21)$$

The corresponding three-dimensional curve is obtained by substituting $u(t)$ and $v(t)$ into the NURBS surface definition given in Equation (2.1):

$$\mathbf{S}(u(t), v(t)) = \begin{bmatrix} x(u(t), v(t)) \\ y(u(t), v(t)) \\ z(u(t), v(t)) \end{bmatrix}. \quad (2.22)$$

The first and last control points define the endpoints of the guide curve, while the two intermediate control points control its tangent directions and overall shape. The four control points provide a compact parameterization of the guide curve trajectory. The example of guide curve created using this method is presented in Figure 2.8.

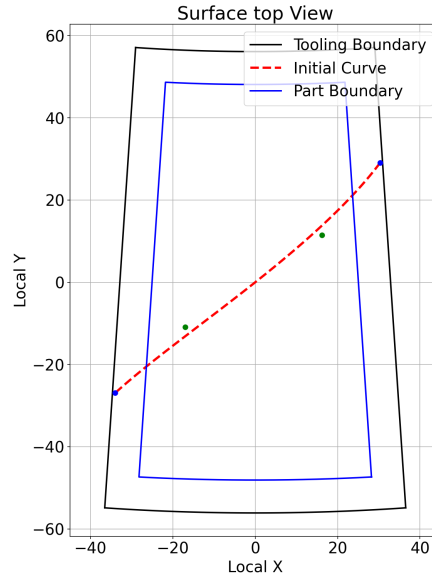


Figure 2.8: Example of the guide curve created using discussed method.

After mapping to the tooling surface, the guide curve is used as the centerline of the first fiber course. The remaining centerlines and course boundaries are generated by applying the offset procedure. Control-point coordinates in the UV domain are later treated as design variables for optimization.

2.6 Definition of the Part Boundary

The physical boundary of the manufactured part, which defines the final part contour, must lie on the tooling surface and be positioned strictly within its outer edges. During the AFP process, fiber courses may extend beyond this boundary. Typically, the extra material is removed by trimming along the contour.

The part boundary is initially created in a CAD and imported into the Python framework as a .step file. The imported geometry is interpreted as a set of points in the physical XYZ domain. These points are subsequently mapped to the parametric UV domain using the Gauss–Newton projection method described above. The main reason for transferring the boundary to the parametric domain is that geometric and Boolean operations can be performed more conveniently in two dimensions. A point classified as being inside the boundary polygon in the UV domain corresponds to a point inside the same physical boundary after mapping to the XYZ domain. In particular, the mapped boundary can be represented as a closed polygon in the UV domain and used for point-in-polygon tests and intersection operations. Representation of the part boundary on the tooling surface is shown in Figure 2.9.

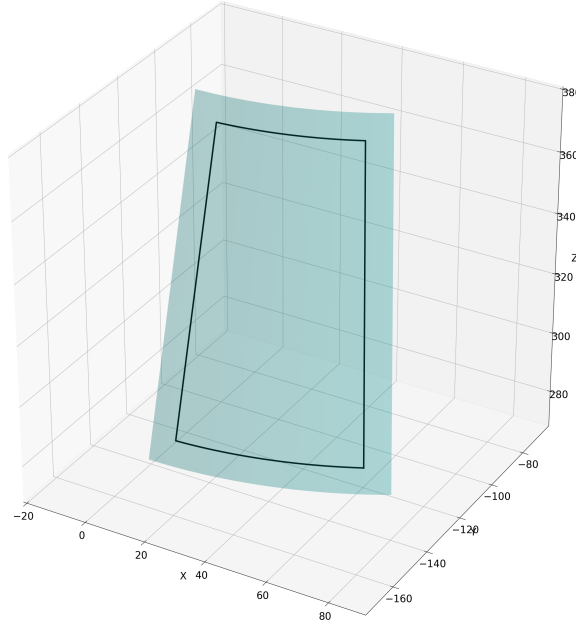


Figure 2.9: Part boundary visualization.

2.7 Generation of Fiber Courses

Fiber courses are generated using an offset procedure on surface. Each course is defined by a centerline and two boundaries. The initial guide curve is the centerline of the first course. Two boundaries are generated by offsetting the centerline by half of the prescribed course width. Every next course centerlines are generated by offsetting the previous centerline by the prescribed distance.

The centerline in the UV domain is:

$$\mathbf{C}(t) = \begin{bmatrix} u(t) \\ v(t) \end{bmatrix}, \quad t \in [0, 1]. \quad (2.23)$$

The offset procedure is applied to points, distributed along the centerline. Point in the physical domain is obtained from the tooling surface mapping:

$$\mathbf{P}(t) = \mathbf{S}(u(t), v(t)). \quad (2.24)$$

The tangent vector to the mapped centerline is calculated using the chain rule:

$$\mathbf{T}(t) = \frac{d\mathbf{S}(u(t), v(t))}{dt} = \frac{\partial \mathbf{S}}{\partial u} \frac{du}{dt} + \frac{\partial \mathbf{S}}{\partial v} \frac{dv}{dt}. \quad (2.25)$$

Here,

$$\frac{\partial \mathbf{S}}{\partial u} = \begin{bmatrix} \frac{\partial x}{\partial u} \\ \frac{\partial y}{\partial u} \\ \frac{\partial z}{\partial u} \end{bmatrix}, \quad \frac{\partial \mathbf{S}}{\partial v} = \begin{bmatrix} \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial v} \end{bmatrix}, \quad (2.26)$$

while du/dt and dv/dt are obtained from the parametric centerline $\mathbf{C}(t)$.

The unit tangent vector is defined as

$$\hat{\mathbf{T}} = \frac{\mathbf{T}}{\|\mathbf{T}\|}. \quad (2.27)$$

The unit normal $\hat{\mathbf{N}}$ to the tooling surface is evaluated at the same point using the build-in function of the `geomdl` Python library.

Then the binormal direction lying in the local tangent plane of the surface is:

$$\hat{\mathbf{B}} = \frac{\hat{\mathbf{N}} \times \hat{\mathbf{T}}}{\|\hat{\mathbf{N}} \times \hat{\mathbf{T}}\|}. \quad (2.28)$$

The sign of $\hat{\mathbf{B}}$ determines the offset side. Reversing it changes the offset direction from one side of the centerline to another. The schematic image of all three vectors is presented in Figure 2.10

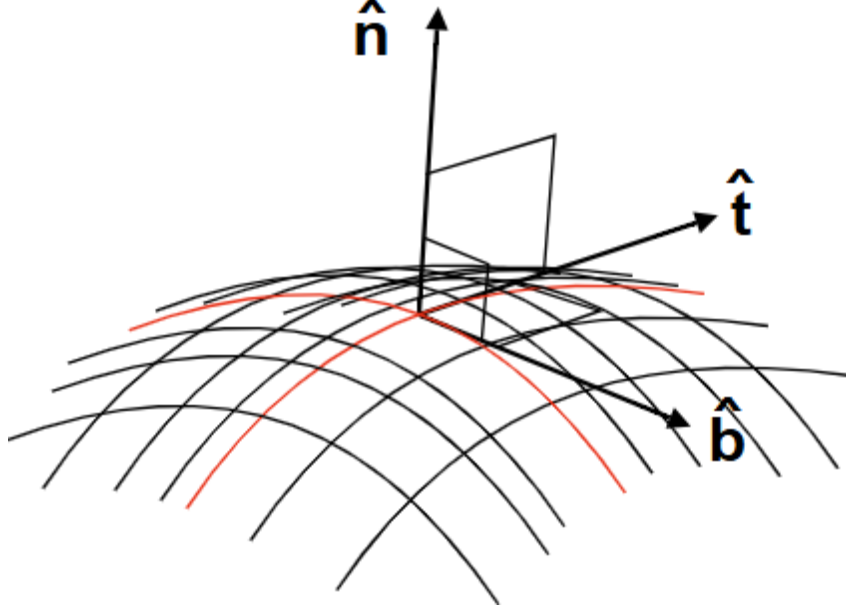


Figure 2.10: Tangent, normal, and binormal unit vectors along a curve on a surface. The surface is shown by black grid lines, and the curve is shown in red. The binormal vector is defined as the cross product of the tangent and normal vectors, $\hat{\mathbf{b}} = \hat{\mathbf{t}} \times \hat{\mathbf{n}}$ [37].

For each sampled point $\mathbf{P}_0$ on the centerline the corresponding offset point is calculated in the XYZ domain:

$$\mathbf{P}_{\text{off}} = \mathbf{P}_0 + d\hat{\mathbf{B}}, \quad (2.29)$$

or, in component form,

$$\begin{bmatrix} x_{\text{off}} \\ y_{\text{off}} \\ z_{\text{off}} \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + d \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix}. \quad (2.30)$$

The signed offset distance d depends on the curve being generated. For the two course boundaries this is the half of course width w , which is also an important MS:

$$d = \pm \frac{w}{2}, \quad (2.31)$$

For the centerline of an adjacent course this distance d is prescribed by the user.

The point obtained from Equation (2.29) generally lies in the local tangent plane

rather than exactly on the tooling surface. It should be projected back onto the surface using the Gauss–Newton method described previously.

The projected parameter values are retained only when they remain inside the valid parametric domain:

$$0 \leq u \leq 1, \quad 0 \leq v \leq 1. \quad (2.32)$$

Points for which a valid projection cannot be obtained are discarded. After the two boundaries of a course have been constructed, a closed course polygon in the UV domain is being formed. This polygon represents the fiber course and is used for geometric operations such as intersection tests, gap and overlap evaluations and finite element properties assignment.

Course generation is continued while the newly generated course polygon intersects the part-boundary polygon. This is evaluated by checking whether at least one sampled point of the course polygon lies inside the part boundary. If no sampled point of the newly generated course polygon is located inside the part-boundary polygon, the course no longer contributes to the manufactured region and the generation procedure is terminated in that direction. This last generated course is not included to the model.

After an offset curve has been created, one or both of its endpoints may remain inside the part physical boundary. In this case, the curve does not cross the entire region to be covered and should be extended.

Let an endpoint of the candidate centerline be defined by

$$\boldsymbol{\xi}_0 = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}, \quad (2.33)$$

with the corresponding physical point

$$\mathbf{P}_0 = \mathbf{S}(u_0, v_0). \quad (2.34)$$

The endpoint tangent is first evaluated and normalized using Equations (2.25) and (2.27).

At the starting point of the curve, the extension direction is taken opposite to the local tangent. At the endpoint of the curve it follows the local tangent:

$$\hat{\mathbf{d}}_{\text{start}} = -\hat{\mathbf{t}}_{xyz}, \quad \hat{\mathbf{d}}_{\text{end}} = \hat{\mathbf{t}}_{xyz}. \quad (2.35)$$

A trial point is generated by advancing a prescribed physical step Δs along the current extension direction:

$$\mathbf{P}_{\text{trial}} = \mathbf{P}_k + \Delta s \hat{\mathbf{d}}_k. \quad (2.36)$$

Since the trial point does not generally lie exactly on the tooling surface, it is projected back onto the surface using the Gauss–Newton algorithm.

The projected point is added to the centerline definition, and the local extension direction is updated. To control the change in direction, the preceding and newly evaluated tangents are combined as

$$\mathbf{t}_{\text{blend}} = (1 - \alpha)\hat{\mathbf{t}}_k + \alpha\hat{\mathbf{t}}_{k+1}, \quad (2.37)$$

where $\alpha \in [0, 1]$ is a prescribed tangent-blending parameter. The resulting vector is normalized before the next extension step.

The extension procedure continues until the projected points leave the closed polygon representing the part boundary in the UV domain. A prescribed number of additional points is generated after the first boundary crossing to ensure that the curve extends far enough beyond the part boundary. The original and extended point sets are then combined. Duplicate points are removed before the extended centerline is reconstructed. The extension procedure permits a course to extend outside the part boundary, but all generated extension points must still be representable on the tooling surface.

Starting from the initial guide curve, courses are generated iteratively on both sides of the first centerline. For each new candidate centerline, the framework attempts to construct it using prescribed distance $w/2$ and the corresponding left and right boundaries using half of the course width $w/2$.

Near the edge of the tooling surface, a course may not always be fully representable. For example, current course might partially lie inside the part boundary, so algorithm thinks that the next course should be generated. But due to the geometry near the edge of the part, a new centerline might be completely outside of the tooling surface, so following common sense it is impossible to create this next centerline. There also exist cases where this next centerline is still on the tooling surface, but one of the boundaries not. For this reason, iterative filling procedure is terminated in the current direction when:

1. The next centerline cannot be constructed;
2. The first offset boundary cannot be constructed; or
3. The second offset boundary cannot be constructed.

Fully generated courses using the custom Python are presented in the Figure 2.11.

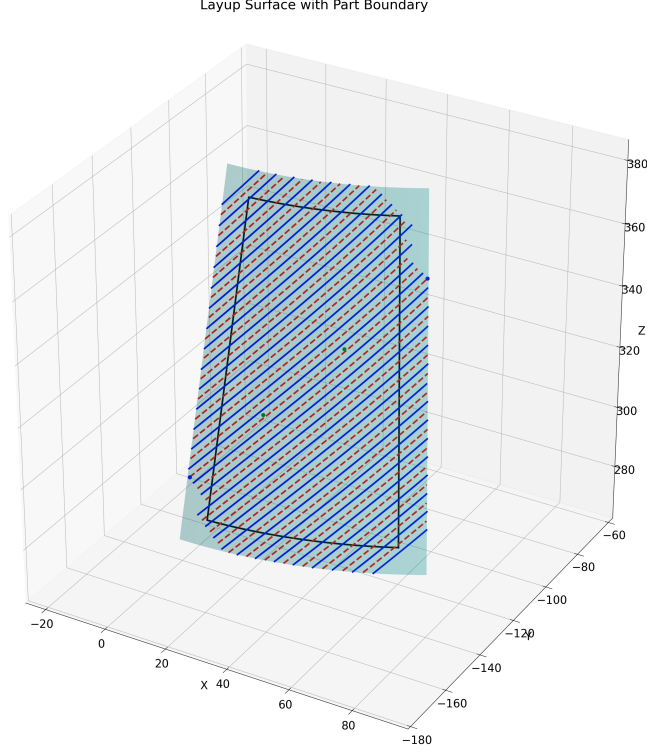


Figure 2.11: Fiber courses generated using Python.

2.8 Manufacturing Check

Several manufacturing constraints are considered in this work. The first is the head path, which is required to be continuous, with the start and end points of each centerline located outside the engineering boundary. This is generally ensured by the algorithm described in the previous section.

2.8.1 Minimum Steering Radius

The second is the minimum steering radius. It is measured as geodesic curvature of the course centerline. The geodesic curvature describes the turning of a curve within the local tangent plane of a surface [38].

$$R_g = \frac{1}{|\kappa_g|}, \quad (2.38)$$

where κ_g is the geodesic curvature. This curvature describes the turning of the centerline within the local tangent plane of the tooling surface.

The total curvature of a curve on surface consists of two components: the normal curvature κ_n , caused by the curvature of the surface itself, and the geodesic curvature κ_g

[38].

The centerline is already defined in Equation (2.23). Its tangent vector $\mathbf{T}(t)$ in the XYZ domain is defined in Equation (2.25).

The second derivative is:

$$\begin{aligned}\mathbf{Q}(t) = & \mathbf{S}_{uu} \left(\frac{du}{dt} \right)^2 + 2\mathbf{S}_{uv} \frac{du}{dt} \frac{dv}{dt} + \mathbf{S}_{vv} \left(\frac{dv}{dt} \right)^2 \\ & + \mathbf{S}_u \frac{d^2u}{dt^2} + \mathbf{S}_v \frac{d^2v}{dt^2},\end{aligned}\tag{2.39}$$

where

$$\mathbf{S}_{uu} = \frac{\partial^2 \mathbf{S}}{\partial u^2}, \quad \mathbf{S}_{uv} = \frac{\partial^2 \mathbf{S}}{\partial u \partial v}, \quad \mathbf{S}_{vv} = \frac{\partial^2 \mathbf{S}}{\partial v^2}.\tag{2.40}$$

The magnitude of the total curvature of the centerline can be defined using velocity (first derivative) and acceleration (second derivative) components [38]:

$$\kappa(t) = \frac{\|\mathbf{T}(t) \times \mathbf{Q}(t)\|}{\|\mathbf{T}(t)\|^3}.\tag{2.41}$$

At every point of the tooling surface, two mutually orthogonal principal directions can be defined in the local tangent plane. The associated principal curvatures, denoted by κ_1 and κ_2 , are respectively the maximum and minimum normal curvatures at that point [38].

The principal curvatures are obtained from the first and second fundamental forms of the surface. The first fundamental form is as follows [39]:

$$\mathbf{I} = \begin{bmatrix} E & F \\ F & G \end{bmatrix},\tag{2.42}$$

where

$$E = \mathbf{S}_u \cdot \mathbf{S}_u, \quad F = \mathbf{S}_u \cdot \mathbf{S}_v, \quad G = \mathbf{S}_v \cdot \mathbf{S}_v.\tag{2.43}$$

The second fundamental form is [39]:

$$\mathbf{II} = \begin{bmatrix} L & M \\ M & N \end{bmatrix},\tag{2.44}$$

with

$$L = \mathbf{S}_{uu} \cdot \hat{\mathbf{N}}, \quad M = \mathbf{S}_{uv} \cdot \hat{\mathbf{N}}, \quad N = \mathbf{S}_{vv} \cdot \hat{\mathbf{N}},\tag{2.45}$$

where $\hat{\mathbf{N}}$ is the unit surface normal.

The principal curvatures and their corresponding directions satisfy the generalized eigenvalue problem [39]:

$$\mathbf{II}\mathbf{q} = \kappa \mathbf{I}\mathbf{q}. \quad (2.46)$$

And the principal curvatures are the eigenvalues of

$$\mathbf{A} = \mathbf{I}^{-1}\mathbf{II}. \quad (2.47)$$

The eigenvectors define the principal directions in the surface parameter space and are mapped to the corresponding tangent directions in the physical domain.

Let $\hat{\boldsymbol{\tau}}$ denote the unit tangent to the centerline, and let $\hat{\mathbf{v}}_1$ denote the unit principal direction associated with κ_1 . Figure 2.12 is given for the reference.

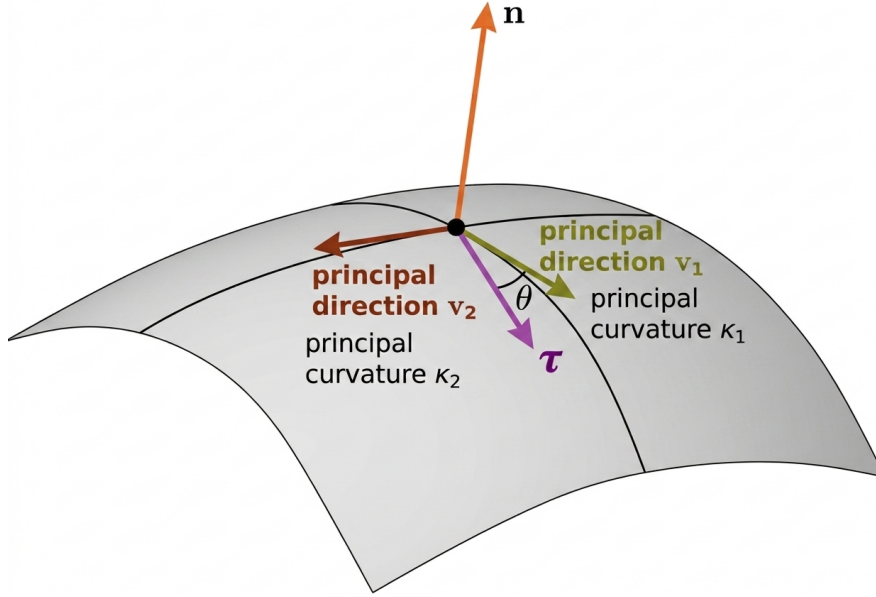


Figure 2.12: Principal curvatures. Adapted from [40].

The angle between these directions is:

$$\theta = \arccos(\hat{\boldsymbol{\tau}} \cdot \hat{\mathbf{v}}_1). \quad (2.48)$$

This angle is introduced only for the geometric curvature calculation. It is not the local fiber-orientation angle used to define the material properties of the laminate. According to Euler's curvature formula, the normal curvature in the centerline direction is:

$$\kappa_n(\theta) = \kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta. \quad (2.49)$$

The quantity κ_n describes the part of the centerline curvature that is defined by the local curvature of the tooling surface. Curve that is locally straight with respect to the surface are called geodesic. Such curves have zero geodesic curvature, although it may still be curved when viewed in three-dimensional space. On a planar surface, the normal curvature is zero. Therefore, the geodesic curvature is equal to the total curvature of the spline on this planar surface. Example is given in the Figure 2.13.

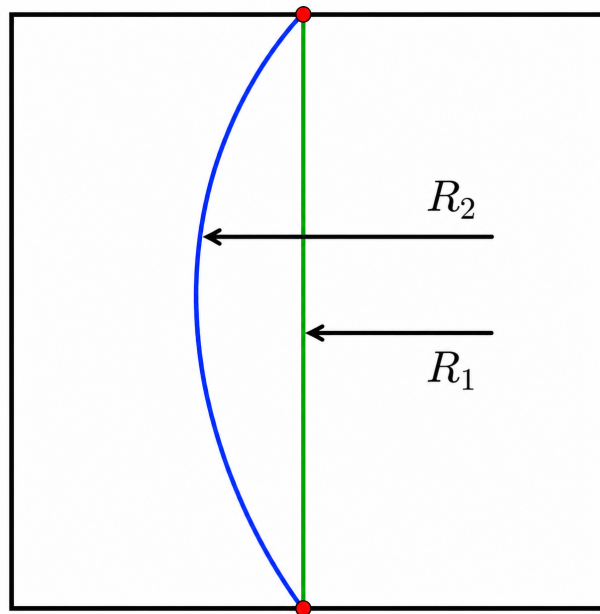


Figure 2.13: Geodesic curvature on the planar surface.

Straight line on the planar surface has infinite radius R_1 and therefore zero geodesic curvature, whereas a circular path of radius R_2 has

$$\kappa_g = \frac{1}{R_2}. \quad (2.50)$$

This property is more interesting on a curved surface. For example, when the Earth is idealized as a sphere, a flight path between two cities on the same latitude is curved in three-dimensional space and usually not lie on this latitude. Nevertheless, it is usually almost geodesic and therefore has small geodesic curvature. The flight route between Paris (48.86° N) and Seattle (47.61° N) can serve as an example. Figure 2.14 shows this in a two-dimensional map projection.

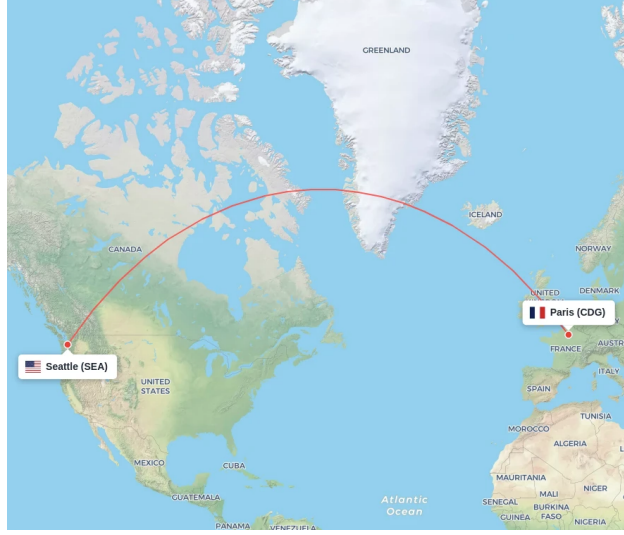


Figure 2.14: Geodesic curvature example [41].

The total, normal, and geodesic curvatures are related by [38]:

$$\kappa^2 = \kappa_n^2 + \kappa_g^2. \quad (2.51)$$

Therefore, the magnitude of the geodesic curvature is calculated as

$$\kappa_g = \sqrt{\kappa^2 - \kappa_n^2}. \quad (2.52)$$

The local steering radius is then obtained from Equation (2.38). For each course, the curvature is evaluated at a finite number of centerline points, and the minimum calculated steering radius is defined as

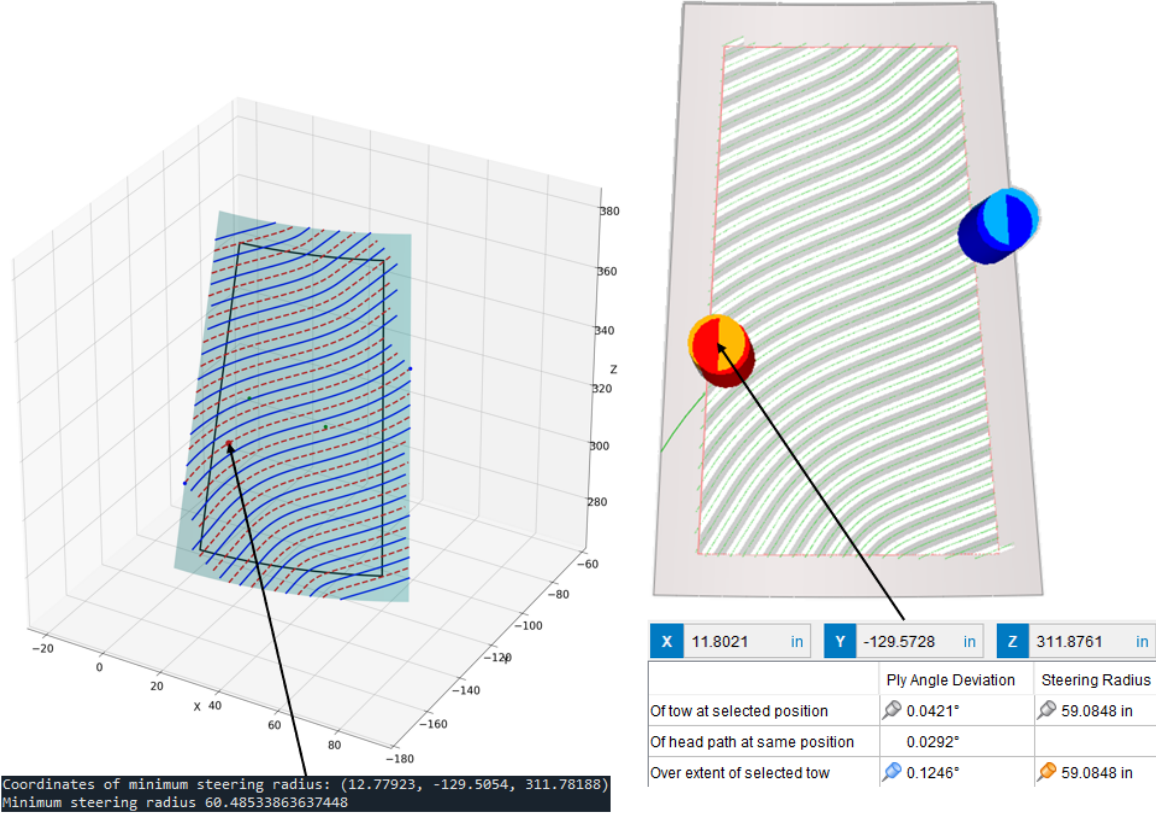
$$R_{g,\min} = \min_{k=1,\dots,n_s} R_g(t_k). \quad (2.53)$$

The course satisfies the adopted steering constraint when

$$R_{g,\min} \geq R_{\text{allow}}, \quad (2.54)$$

where R_{allow} is the prescribed minimum allowable steering radius.

For the verification, comparison with commercial VCP tool was performed. Minimum steering radius of initial centerline was measured using the custom Python tool. Both value and position were defined. Results are presented in Figure 2.15a. Minimum radius of curvature for closest tow to initial centerline was measured using VCP. Corresponding coordinates were also obtained. Results are presented in the Figure 2.15b.



(a) Steering radius measurement using the Python tool.

(b) Steering radius measurement using the commercial VCP software.

Figure 2.15: Comparison of steering radius measurements.

The Python framework predicted a minimum radius of 60.4853 in, while VCP reported 59.0848 in. The relative difference is approximately 2.37%. The locations of the minimum-radius points differed by approximately 0.98 in in the XYZ domain. The agreement is considered satisfactory for the present geometric implementation.

Table 2.1: Comparison of minimum steering-radius results.

Quantity	Python framework	VCP
Minimum steering radius, in	60.4853	59.0848
x -coordinate, in	12.7792	11.8021
y -coordinate, in	-129.5054	-129.5728
z -coordinate, in	311.7819	311.8761

2.8.2 Gaps and Overlaps

Gaps and overlaps are common AFP manufacturing induced defects. They can either be prescribed or unprescribed. They also can be used to improve structural performance, but

on the other hand can reduce overall quality in certain cases. Graphical representation of these manufacturing signatures is presented in Figure 2.16.

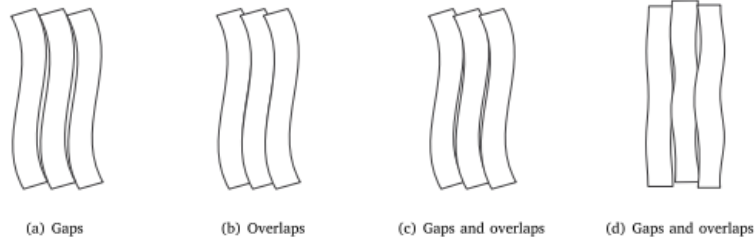


Figure 2.16: Gaps/Overlaps visual representation [42]

Even since these manufacturing signatures are very intuitive, some rules should have been established to define and measure them. Let Ω_{eng} denote the engineering-boundary polygon and let Ω_i denote the polygon occupied by the i -th fiber course, all represented in the uv -domain.

The overlap between a newly generated course and the preceding course is defined as

$$\Omega_{\text{ov},i} = \Omega_i \cap \Omega_{i-1} \cap \Omega_{\text{eng}}. \quad (2.55)$$

The intersection with the engineering boundary ensures that only overlap located within the manufactured part is retained.

After all courses have been generated, their combined coverage is

$$\Omega_{\text{cov}} = \bigcup_{i=1}^{n_c} \Omega_i. \quad (2.56)$$

The uncovered part of the engineering region is then defined as the gap region:

$$\Omega_{\text{gap}} = \Omega_{\text{eng}} \setminus \Omega_{\text{cov}}. \quad (2.57)$$

The Boolean operations are performed in the uv -domain, where the course and engineering boundaries are represented as planar polygons.

Gap and overlap regions may have irregular shapes. In the developed framework, the orientation of each region is first estimated using its minimum rotated bounding rectangle, defined as the rectangle of minimum area that fully contains the polygon. Shapely Python library is actively in the process of gaps/overlaps detection and measurement. Example of the minimum bounding rectangle is presented in Figure 2.17.

Although the minimum bounding rectangle is constructed in the UV domain, its orientation is used as an approximation of the corresponding longitudinal and transverse

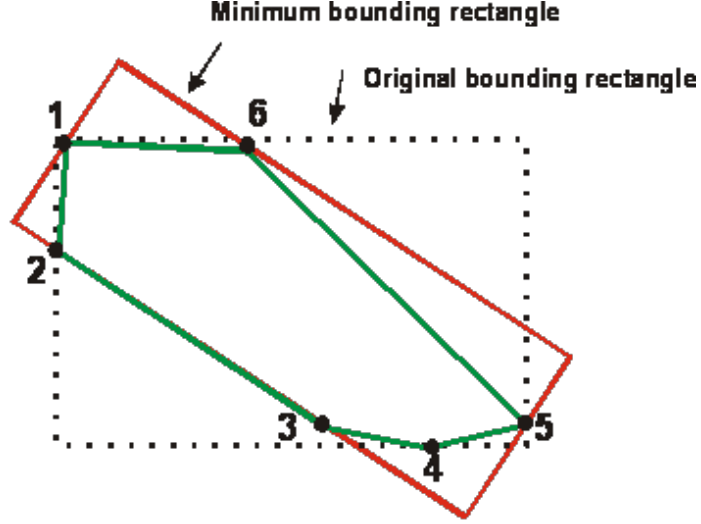


Figure 2.17: Minimum rotated bounding rectangle [43].

directions in the physical domain. This assumes that the surface parameterization is regular and does not introduce any distortions. For the smooth tooling surfaces considered in this work, the mapping from is expected to follow the local orientation of the defect well enough.

The two long edges of the minimum rotated rectangle are assumed to follow the longitudinal direction of gap/overlap. A prescribed number of corresponding points is distributed along these two edges, and line segments joining the corresponding points are generated. These segments are approximately parallel to the short edges of the bounding rectangle and therefore represent width of the defect.

Each candidate segment is clipped by the original gap or overlap polygon. It ensures that portions lying outside the actual defect or the part boundary are removed. The retained segments are then mapped onto the tooling surface using the surface mapping $\mathbf{S}(u, v)$. The real length of each mapped segment is approximated by

$$l_j \approx \sum_{k=1}^{n_p-1} \|\mathbf{S}(u_{k+1}, v_{k+1}) - \mathbf{S}(u_k, v_k)\|, \quad (2.58)$$

where n_p is the number of interpolation points along the segment.

The reported defect width is the maximum length among the retained transverse sections:

$$w_{\text{def}} = \max_j l_j. \quad (2.59)$$

Figure 2.18 illustrates the implemented procedure. The red filled regions indicate the detected overlap polygons. The magenta segments indicate segments used to estimate the

local widths of the detected overlap.

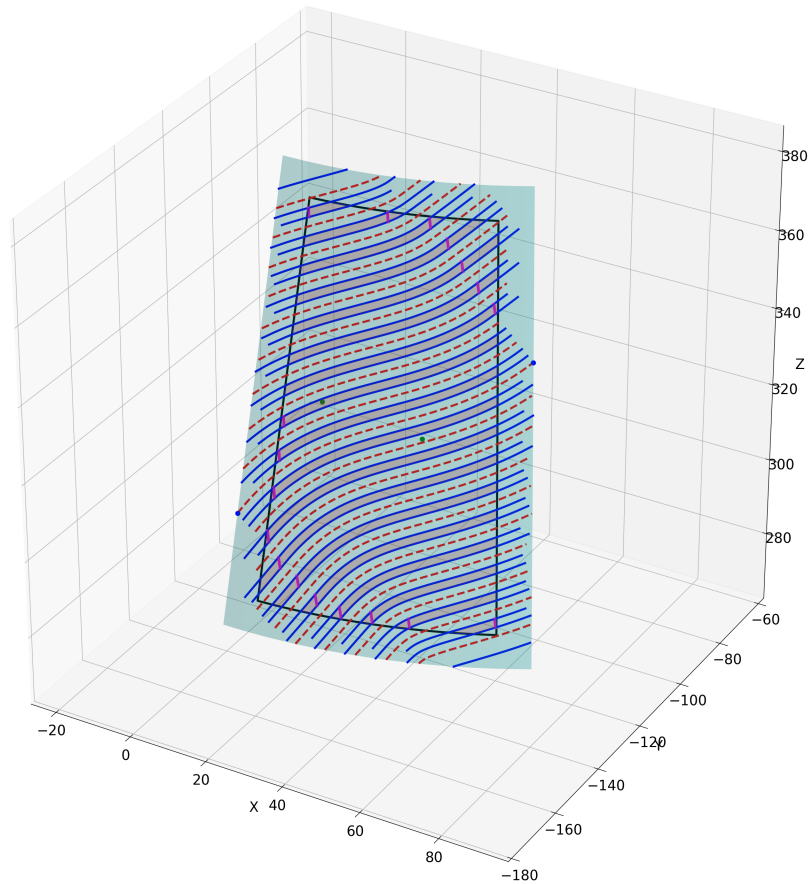


Figure 2.18: Overlap visualization using the Python tool.

2.9 Generation of the Abaqus Model

Input files are used to link Python laminate model to Abaqus. Theoretically, this approach should work not only for Abaqus, but for any type of commercial FEA software, like Simcenter Nastran or Altair Hyperworks. User should create a template input file. This file should contain almost entire information about the model: mesh definition, analysis settings, loading and boundary conditions, coordinate systems information and global zero direction of the layup. The material-property and composite-section blocks are generated by the Python framework and inserted into the template later. The user should prescribe material properties (thickness and mechanical properties) using the Python tool.

2.9.1 Mapping Data to FE Mesh

Python framework reads this data, project FE mesh and project element centroids coordinates to the tooling surface. Each element has its own section with unique ply stacking inside it to adequately represent angle distribution across the VAT structure.

For each laminate ply(layer), the projected centroid of every finite element is tested against all course polygons generated for that layer. Let $\mathbf{C}_e$ denote the centroid of element e , and let (u_e, v_e) be its projected coordinates in the UV domain. The set of courses covering the centroid is defined as

$$\mathbf{N}_e = \{i : (u_e, v_e) \in \Omega_i\}, \quad (2.60)$$

where Ω_i is the polygon occupied by course i .

An element is assigned to a course when its centroid lies inside the corresponding course polygon.

The material and thickness assignment depends on the number of courses covering the element centroid:

1. If $\mathbf{N}_e = 0$, the element contains no fiber course in the considered ply (gap region). A matrix-only properties are assigned with the prescribed orientation angle of 0° . This angle was chosen since the matrix material is isotropic.
2. If $\mathbf{N}_e = 1$, the element is covered by exactly one course, and the default composite properties are assigned.
3. If $\mathbf{N}_e > 1$, the centroid lies in an overlap region. Default mechanical properties are assigned, but the fiber thickness is modified as:

$$h_{ov} = f_{ov} h_f, \quad (2.61)$$

where h_f is the nominal fiber-layer thickness and $f_{ov} = 0.9$ is the adopted overlap-thickness factor.

For every course assigned to an element, the closest sampled point on the corresponding centerline is first identified in the XYZ domain. Let $\mathbf{P}_{i,k}$ denote the k -th sampled point of centerline i . The selected point index is

$$k^* = \arg \min_k \|\mathbf{P}_{i,k} - \mathbf{C}_e\|. \quad (2.62)$$

The centerline tangent at an interior sampled point is approximated using a central finite difference:

$$\mathbf{t}_{i,k^*} = \mathbf{P}_{i,k^*+1} - \mathbf{P}_{i,k^*-1}. \quad (2.63)$$

At the first and last centerline points, forward and backward differences are used, respectively. The tangent is subsequently normalized:

$$\hat{\mathbf{t}} = \frac{\mathbf{t}}{\|\mathbf{t}\|}. \quad (2.64)$$

The surface normal is evaluated at the corresponding centerline coordinates (u_{k^*}, v_{k^*}) . Let $\hat{\mathbf{n}}$ denote the unit surface normal and let $\mathbf{r}_0$ denote the global zero-degree reference direction supplied in the input template. A vector $\mathbf{a}$ is projected onto the local surface tangent plane using

$$\mathbf{a}_{\text{proj}} = \mathbf{a} - (\mathbf{a} \cdot \hat{\mathbf{n}}) \hat{\mathbf{n}}. \quad (2.65)$$

This operation is applied to both the global reference direction and the centerline tangent:

$$\mathbf{r}_{\text{proj}} = \mathbf{r}_0 - (\mathbf{r}_0 \cdot \hat{\mathbf{n}}) \hat{\mathbf{n}}, \quad (2.66)$$

$$\mathbf{t}_{\text{proj}} = \hat{\mathbf{t}} - (\hat{\mathbf{t}} \cdot \hat{\mathbf{n}}) \hat{\mathbf{n}}. \quad (2.67)$$

The projected vectors are then normalized to obtain $\hat{\mathbf{r}}_{\text{proj}}$ and $\hat{\mathbf{t}}_{\text{proj}}$.

The signed local fiber-orientation angle is calculated as:

$$\theta_e = \text{atan2} \left(\hat{\mathbf{n}} \cdot \left[\hat{\mathbf{r}}_{\text{proj}} \times \hat{\mathbf{t}}_{\text{proj}} \right], \hat{\mathbf{r}}_{\text{proj}} \cdot \hat{\mathbf{t}}_{\text{proj}} \right). \quad (2.68)$$

The second argument of the atan2 function is the dot product of the projected unit vectors and is therefore equal to $\cos \theta_e$. The first argument is the scalar triple product between the projected reference direction, projected fiber tangent and surface normal. Because both projected vectors lie in the local tangent plane, this triple product provides the signed value of $\sin \theta_e$. Its sign distinguishes clockwise and counterclockwise rotations relative to the orientation of the surface normal. Consequently, the atan2 formulation returns the signed fiber angle over the interval $(-\pi, \pi]$. The resulting angle, together with the course identifier, layer identifier, assigned thickness, and material, is stored for subsequent generation of the Abaqus composite-section definitions. Visual representation of the entire process is given in

Figure 2.19.

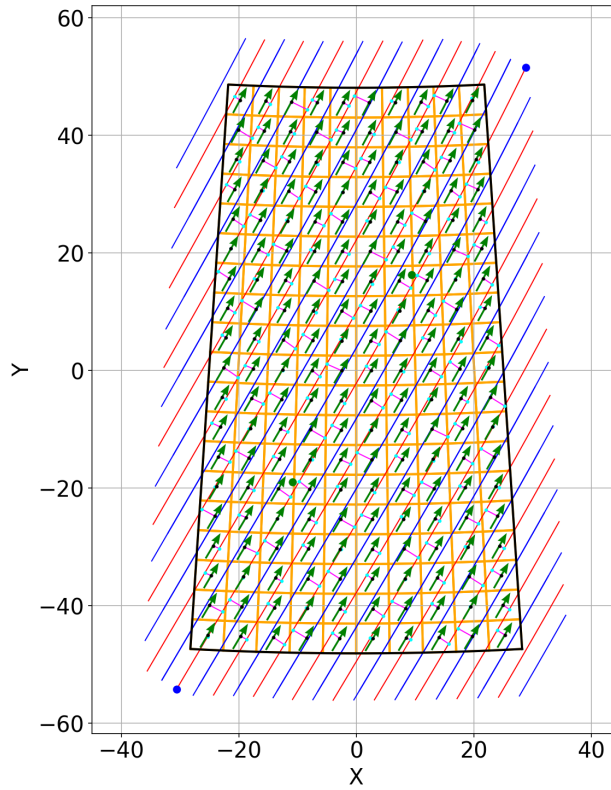


Figure 2.19: Example of property assignment to mesh elements based on the nearest course-centerline point.

Plot color meanings

- Black line** – Part boundary
- Orange lines** – Mesh
- Black points** – Element centroids
- Red and blue lines** – Course centerlines and boundaries
- Cyan points** – Nearest centerline points associated with the element centroids
- Magenta lines** – Connections between element centers and their corresponding centerline points
- Green arrows** – Assigned local fiber directions

Figure 2.20 shows an example of the element-level dataset generated by the framework. Multiple rows with the same element and layer identifiers indicate that the element centroid is covered by more than one course, so it lies in an overlap region. This data is later converted to Abaqus sections for each element.

element_id	x	y	z	layer_id	course_id	angle	thickness	material
1	11	11	0	1	1	45	0.005	Composite
1	11	11	0	2	6	0	0.005	Composite
2	13	11	0	1	1	45	0.0045	Composite
2	13	11	0	1	6	45	0.0045	Composite
2	13	11	0	2	6	0	0.005	Composite
3	15	11	0	1	6	45	0.005	Composite
3	15	11	0	2	6	0	0.005	Composite

Figure 2.20: Example of dataset for several elements. Element 2 contains overlap in the first ply.

As was said, most of the model setup is taken from the template input file which is pre-created by the user, so Python framework just adds material properties and section data.

Chapter 3

COMPUTATIONAL METHODS

3.1 Chapter Scope and Objectives

The previous chapter described custom Python framework for VAT structures modeling. This chapter describes the structural-analysis and optimization methods used around that framework. The principal objective of the optimization procedure is to improve structural performance of the laminate by changing control points of the initial guide curve. For this work, the targeted structural performance to improve is the critical buckling load. For each candidate guide curve, the VAT modeling framework generates the complete set of fiber courses, checks manufacturing constraints, maps the laminate to the finite-element mesh and creates an Abaqus input file. Candidate designs that do not pass manufacturing check are penalized before the FEA is performed.

This algorithm is general and different analysis methods can be used for problem solution. Since the goal of this work is to demonstrate that gains in critical buckling load can be achieved using VAT laminates, FEM in combination with arc-length nonlinear analysis are used. Riks nonlinear analysis method is available directly in Abaqus. The custom Python tool and the commercial finite element software Abaqus are integrated using an optimization loop, which is also described in this chapter. Differential Evolution (DE) optimization algorithm implemented in SciPy was used to perform this step.

The methods introduced in this chapter do not alter the geometric definitions established previously. The same course-generation, steering-radius, defect-detection, and finite-element mapping procedures are applied to every candidate design considered during optimization.

3.2 Finite Elements Analysis

Finite Element Analysis (FEA) is a powerful method that is widely used in engineering and research. Structural, aerodynamical, thermomechanical and other problems can be solved using FEA, but mechanical problems are the main focus in this work, specifically, buckling response. For this reason, only the aspects directly connected to such type of problems are described in this chapter.

The basic idea of finite-element method (FEM) is to divide the body into finite elements, connected by nodes, and obtain an approximate solution as shown in Figure 3.1. This is called the finite element mesh [44].

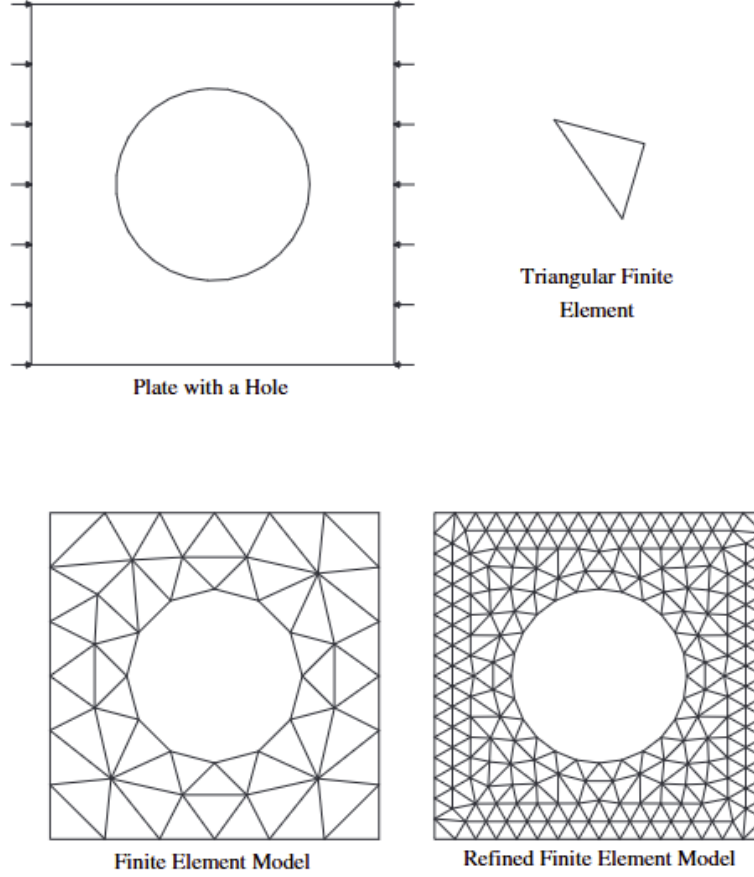


Figure 3.1: Geometry, loads and finite element meshes [44].

In general, a mechanical system composed of linear elastic elements is governed by a linear system of equations [45]:

$$\mathbf{K}\mathbf{U} = \mathbf{F}_{\text{ext}}. \quad (3.1)$$

where $\mathbf{K}$ is a model stiffness matrix, $\mathbf{U}$ is a vector of nodal displacements, $\mathbf{F}_{\text{ext}}$ is a vector of applied forces.

For the buckling problems two main approaches can be used: linear buckling analysis and nonlinear buckling analysis. Linear analysis is faster and is used for simpler models without imperfections. It can be formulated as [46]:

$$(K_0^{NM} + \lambda_i K_{\Delta}^{NM}) v_i^M = 0. \quad (3.2)$$

Here,

K_0^{NM} is the stiffness matrix corresponding to the base state, including the effects of any preloads, $\mathbf{P}^N$;

K_{Δ}^{NM} is the differential initial stress and load stiffness matrix associated with the incremental loading pattern, $\mathbf{Q}^N$;

λ_i are the eigenvalues;

v_i^M are the buckling mode shapes (eigenvectors);

M and N refer to degrees of freedom; and

i denotes the i th buckling mode.

This type of analysis works fast, but may not be very precise in case of imperfections and nonlinearities which are common to most real-world problems. In such situations more accurate approaches are needed. These approaches are described in the next section.

3.3 Nonlinear Analysis

There are three main types of nonlinearities in FEA. The first group are material nonlinearities. This is everything that violates the Hook's Law:

$$\sigma = E\varepsilon, \quad (3.3)$$

Examples are plasticity in metals, hyperelasticity in rubbers and other material laws other than (3.3). Such nonlinearities are not the subject of this work.

The second group are contact and constraint nonlinearities. They occur when boundary conditions or contacts in the model are subject of changes during the analysis. The example of such type of nonlinearity can be the brake pad engagement. A brake caliper presses pads against a rotor, transitioning instantly from zero stiffness (gap open) to high stiffness (gap closed) with severe friction.

The third group are geometric nonlinearities. They are mostly happen due to the large displacement. In the linear static analysis infinitesimal strain tensor is mostly used:

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}). \quad (3.4)$$

The indices i and j identify the row and column, respectively, of the 3×3 infinitesimal strain tensor. In three dimensions, $i, j \in \{1, 2, 3\}$. The notation $u_{i,j}$ denotes the partial derivative of the i th displacement component with respect to the j th reference coordinate:

$$u_{i,j} = \frac{\partial u_i}{\partial x_j}, \quad (3.5)$$

and

$$u_{j,i} = \frac{\partial u_j}{\partial x_i}. \quad (3.6)$$

Infinitesimal strain tensor can be used in case of small displacements, while for the large ones other strain definitions should be use. One of the most famous one is the Green-Lagrange strain tensor:

$$E_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i} + u_{k,i}u_{k,j}), \quad (3.7)$$

With respect to infinitesimal strain tensor, it has an additional term $u_{k,i}u_{k,j}$. This term can be written as:

$$\frac{\partial u_k}{\partial x_i} \frac{\partial u_k}{\partial x_j}. \quad (3.8)$$

For example, third component of the third row of Green-Lagrange strain tensor can be written as:

$$\frac{\partial u_3}{\partial x_3} + \frac{1}{2} \left[\left(\frac{\partial u_1}{\partial x_3} \right)^2 + \left(\frac{\partial u_2}{\partial x_3} \right)^2 + \left(\frac{\partial u_3}{\partial x_3} \right)^2 \right]. \quad (3.9)$$

It can be seen that term $u_{k,i}u_{k,j}$ considers the second-order terms which are negligible when displacements are small. In such a case the Green-Lagrange strain tensor comes down to the infinitesimal strain tensor from the Equation (3.4). For small-displacement analysis the infinitesimal strain measure is used [47]. The other strain measure available in Abaqus and used when the small-strain approximation is no longer valid is logarithmic strain [48]. It can be formulated as:

$$\mathbf{E}_H = \sum_{a=1}^3 \ln(\lambda_a) \mathbf{n}_a \otimes \mathbf{n}_a. \quad (3.10)$$

where λ_a is the principal stretch in the a th principal direction, defined as

$$\lambda_a = \frac{l_{\text{deformed}}}{l_{\text{original}}}, \quad (3.11)$$

and $\mathbf{n}_a$ is the corresponding unit vector defining that principal direction. The principal directions here are three mutually orthogonal directions in which the strain tensor is diagonal:

$$\mathbf{E}_H^{(p)} = \begin{bmatrix} \ln(\lambda_1) & 0 & 0 \\ 0 & \ln(\lambda_2) & 0 \\ 0 & 0 & \ln(\lambda_3) \end{bmatrix}. \quad (3.12)$$

So, when the strain tensor is expressed in the coordinate system other than principal, all shear strain components appear.

Due to the geometric nonlinearity of large-displacement problems, the equilibrium path is also nonlinear now.

3.3.1 Newton–Raphson Method for Nonlinear Equilibrium Problems

The Newton–Raphson method is one of the most common approaches for solving nonlinear equilibrium problems. Here is the example from the Altair OptiStruct manual [49]:

Consider the nonlinear equilibrium problem

$$\mathbf{F}_{\text{int}}(\mathbf{u}) = \mathbf{P}, \quad (3.13)$$

where $\mathbf{u}$ is the global displacement vector, $\mathbf{P}$ is the global external load vector, and $\mathbf{F}_{\text{int}}(\mathbf{u})$ is the nonlinear internal-force vector of the system. For a linear problem, this relation reduces to

$$\mathbf{F}_{\text{int}}(\mathbf{u}) = \mathbf{K}\mathbf{u}, \quad (3.14)$$

Where $\mathbf{K}$ is the global stiffness matrix.

Applying Newton’s method to the nonlinear equilibrium equation gives the iterative procedure

$$\mathbf{K}_t^k \delta \mathbf{u}^k = \mathbf{R}^k, \quad \mathbf{u}^{k+1} = \mathbf{u}^k + \delta \mathbf{u}^k. \quad (3.15)$$

Where

$$\mathbf{K}_t^k = \left. \frac{\partial \mathbf{F}_{\text{int}}(\mathbf{u})}{\partial \mathbf{u}} \right|_{\mathbf{u}=\mathbf{u}^k} \quad (3.16)$$

is the tangent stiffness matrix evaluated at the current displacement state $\mathbf{u}^k$, and

$$\mathbf{R}^k = \mathbf{P} - \mathbf{F}_{\text{int}}(\mathbf{u}^k) \quad (3.17)$$

is the residual-force vector.

The tangent stiffness matrix represents the local slope of the nonlinear equilibrium

path. The iteration is repeated until the residual norm is sufficiently small:

$$\|\mathbf{R}^k\| < \varepsilon_R, \quad (3.18)$$

where ε_R is a prescribed convergence tolerance.

This process is presented in Figure 3.2.

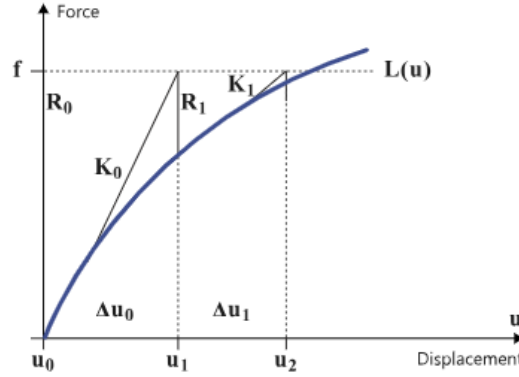


Figure 3.2: Nonlinear solution procedure. [49].

This equilibrium paths might be non-trivial, and two common difficulties may occur. The first one is snap-through, when the load response in the next increment is smaller than in the previous one. Illustration is given in the Figure 3.3. The second one is the snap-back, when the displacement in the next increment decreases with respect to the previous one. This is shown in the Figure 3.4.

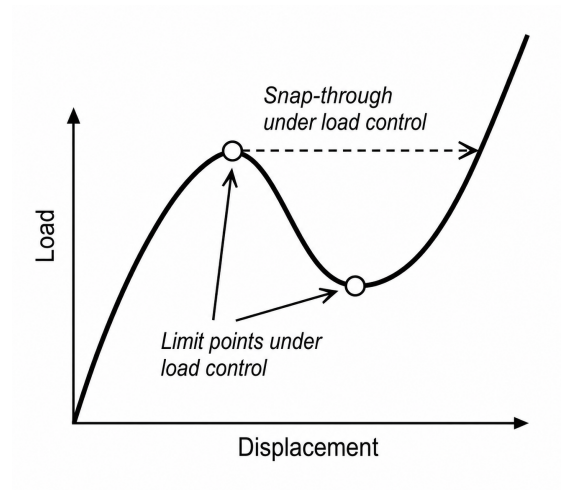


Figure 3.3: Snap-through behavior. [50].

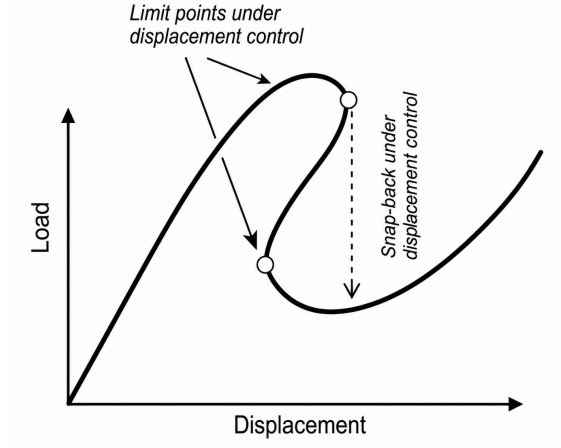


Figure 3.4: Snap-back behavior. [50].

3.3.2 Riks Method

Riks method is one of the most successful for such types of the problems solution. It was generally described by Eduard Riks [51]. One of its versions has been implemented in Abaqus. Here simplified version of the method is described.

There also exist two simpler methods in nonlinear structural analysis: load control and displacement control. They are more straightforward because, in load control, the load multiplier λ_i is prescribed at each step i , and the corresponding displacement vector $\mathbf{u}_i$ is determined.

The problem of this method is that it fails for the snap-through case, because $|\lambda_i|$ should always be greater than $|\lambda_{i-1}|$. In the displacement-control method, one displacement u_i is prescribed, and the load multiplier λ_i should be found for each step i . Then, the entire displacement vector $\mathbf{u}_i$ can be computed. This method fails for the snap-back case due to a similar issue: $|u_i|$ should always be greater than $|u_{i-1}|$.

The Riks method introduces the arc-length parameter Δs to follow the equilibrium path. A geometrical representation of this parameter is given in the Figure 3.5.

At increment i , the applied load is

$$\mathbf{P}_i = \lambda_i \mathbf{P}, \quad (3.19)$$

where λ_i is a dimensionless load multiplier. A scaling parameter α , with units of L^2 , is introduced, so the arc-length metric is

$$ds^2 = \alpha d\lambda^2 + d\mathbf{u} \cdot d\mathbf{u}. \quad (3.20)$$

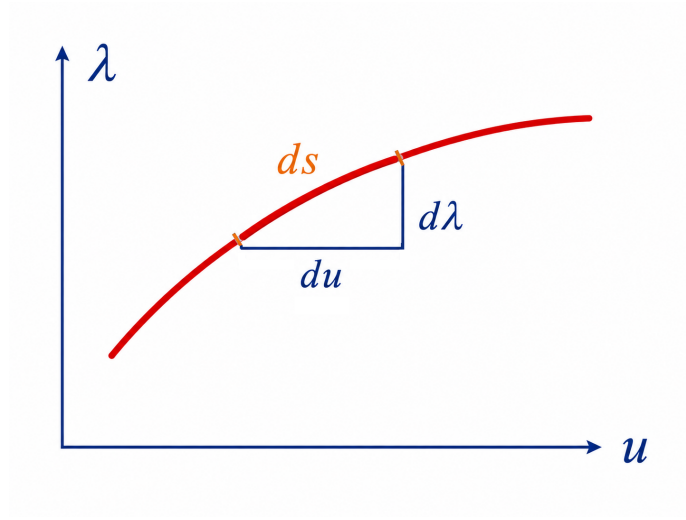


Figure 3.5: Arc-length parameter interpretation.

For one displacement degree of freedom, Equation (3.20) reduces to

$$ds^2 = \alpha d\lambda^2 + du^2. \quad (3.21)$$

The arc-length constraint is defined as

$$\begin{aligned} g(\lambda_{i+1}, \mathbf{u}_{i+1}) &= \alpha d\lambda_{i+1}^2 + d\mathbf{u}_{i+1} \cdot d\mathbf{u}_{i+1} - ds^2 \\ &= 0. \end{aligned} \quad (3.22)$$

A geometrical interpretation is given in the Figure 3.6.

For small arc-length increments,

$$ds \rightarrow \Delta s, \quad d\lambda_{i+1}^2 \rightarrow \Delta\lambda_{i+1}^2, \quad d\mathbf{u}_{i+1} \cdot d\mathbf{u}_{i+1} \rightarrow \Delta\mathbf{u}_{i+1} \cdot \Delta\mathbf{u}_{i+1}. \quad (3.23)$$

The finite increments are

$$\Delta\lambda_{i+1} = \lambda_{i+1} - \lambda_i, \quad (3.24)$$

and

$$\Delta\mathbf{u}_{i+1} = \mathbf{u}_{i+1} - \mathbf{u}_i. \quad (3.25)$$

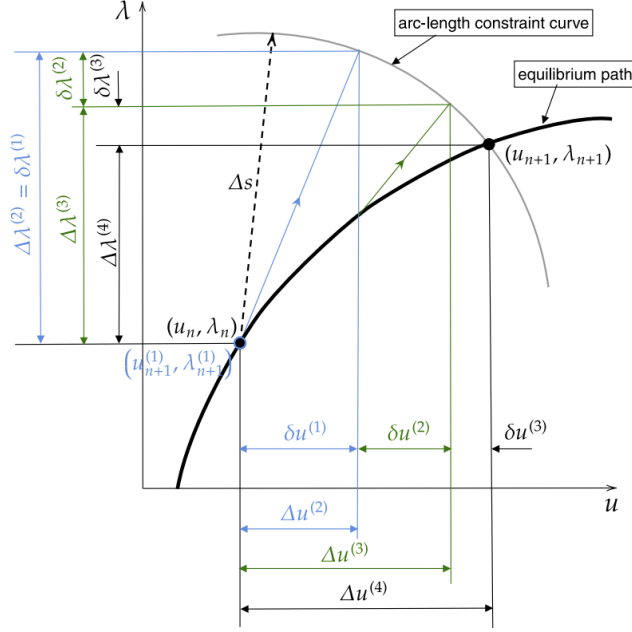


Figure 3.6: Arc-length constraint interpretation [52].

Therefore, the finite arc-length constraint becomes

$$\begin{aligned}
 g(\lambda_{i+1}, \mathbf{u}_{i+1}) &= \alpha (\lambda_{i+1} - \lambda_i)^2 \\
 &\quad + (\mathbf{u}_{i+1} - \mathbf{u}_i) \cdot (\mathbf{u}_{i+1} - \mathbf{u}_i) - (\Delta s)^2 \\
 &= 0.
 \end{aligned} \tag{3.26}$$

The known quantities are $(\lambda_i, \mathbf{u}_i)$ and $(\lambda_{i-1}, \mathbf{u}_{i-1})$, which are the last and previous converged points on the equilibrium path, together with the scaling parameter α and the prescribed arc-length increment Δs . The goal is to find $(\lambda_{i+1}, \mathbf{u}_{i+1})$.

At each Newton iteration, the tangent stiffness matrix $\mathbf{K}_t$ is evaluated at the current trial displacement, while the reference load vector $\mathbf{P}$ is prescribed and fixed. When converged, the residuals should be zero. Therefore, the system of equations can be formulated as

$$\begin{bmatrix} \mathbf{R}(\lambda_{i+1}, \mathbf{u}_{i+1}) \\ g(\lambda_{i+1}, \mathbf{u}_{i+1}) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}. \tag{3.27}$$

At Newton iteration k , Equation (3.27) is linearized as

$$\mathbf{R}^k + \delta \mathbf{R}^k = \mathbf{0}, \tag{3.28}$$

and

$$g^k + \delta g^k = 0. \quad (3.29)$$

The variation of the equilibrium residual is

$$\begin{aligned} \delta \mathbf{R}^k &= \frac{\partial \mathbf{R}^k}{\partial \lambda_{i+1}} \delta \lambda^k + \frac{\partial \mathbf{R}^k}{\partial \mathbf{u}_{i+1}} \delta \mathbf{u}^k \\ &= \mathbf{P} \delta \lambda^k - \mathbf{K}_t^k (\mathbf{u}_{i+1}^k) \delta \mathbf{u}^k. \end{aligned} \quad (3.30)$$

The variation of the arc-length constraint is

$$\begin{aligned} \delta g^k &= \frac{\partial g^k}{\partial \lambda_{i+1}} \delta \lambda^k + \frac{\partial g^k}{\partial \mathbf{u}_{i+1}} \cdot \delta \mathbf{u}^k \\ &= 2\alpha (\lambda_{i+1}^k - \lambda_i) \delta \lambda^k \\ &\quad + 2 (\mathbf{u}_{i+1}^k - \mathbf{u}_i) \cdot \delta \mathbf{u}^k. \end{aligned} \quad (3.31)$$

Equations (3.28)–(3.31) can be written in matrix form as

$$\begin{bmatrix} \mathbf{R}^k \\ -g^k \end{bmatrix} = \begin{bmatrix} \mathbf{K}_t^k & -\mathbf{P} \\ 2 (\mathbf{u}_{i+1}^k - \mathbf{u}_i)^T & 2\alpha (\lambda_{i+1}^k - \lambda_i) \end{bmatrix} \begin{bmatrix} \delta \mathbf{u}^k \\ \delta \lambda^k \end{bmatrix}. \quad (3.32)$$

The internal force vector $\mathbf{F}_{\text{int}} (\mathbf{u}_{i+1}^k)$ for the current trial state can always be computed. The Newton–Raphson iteration residuals $\mathbf{R}^k$ and g^k can therefore be directly computed as

$$\mathbf{R}^k = \lambda_{i+1}^k \mathbf{P} - \mathbf{F}_{\text{int}} (\mathbf{u}_{i+1}^k), \quad (3.33)$$

and

$$\begin{aligned} g^k &= \alpha (\lambda_{i+1}^k - \lambda_i)^2 \\ &\quad + (\mathbf{u}_{i+1}^k - \mathbf{u}_i) \cdot (\mathbf{u}_{i+1}^k - \mathbf{u}_i) - (\Delta s)^2. \end{aligned} \quad (3.34)$$

The selected Δs is also known and prescribed.

Let $(\lambda_i, \mathbf{u}_i)$ be the last converged point and $(\lambda_{i-1}, \mathbf{u}_{i-1})$ be the previous converged point on the equilibrium path. For equal consecutive arc-length increments, a secant predictor for $k = 0$ is

$$\begin{aligned} (\lambda_{i+1}^0, \mathbf{u}_{i+1}^0) &= (\lambda_i, \mathbf{u}_i) + [(\lambda_i, \mathbf{u}_i) - (\lambda_{i-1}, \mathbf{u}_{i-1})] \\ &= 2 (\lambda_i, \mathbf{u}_i) - (\lambda_{i-1}, \mathbf{u}_{i-1}). \end{aligned} \quad (3.35)$$

The displacement change between Newton–Raphson iterations, rather than between

arc-length steps, can be described as

$$\mathbf{K}_t^k \delta \mathbf{u}^k = \mathbf{R}^k + \mathbf{P} \delta \lambda^k. \quad (3.36)$$

Define

$$\mathbf{K}_t^k \mathbf{u}_R^k = \mathbf{R}^k, \quad \mathbf{K}_t^k \mathbf{u}_P^k = \mathbf{P}. \quad (3.37)$$

Equivalently,

$$\mathbf{u}_R^k = (\mathbf{K}_t^k)^{-1} \mathbf{R}^k, \quad \mathbf{u}_P^k = (\mathbf{K}_t^k)^{-1} \mathbf{P}. \quad (3.38)$$

These two quantities can be computed because the current residual, external force vector, and tangent stiffness matrix are known. Therefore,

$$\delta \mathbf{u}^k = \mathbf{u}_R^k + \mathbf{u}_P^k \delta \lambda^k. \quad (3.39)$$

Substituting Equation (3.39) into the linearized arc-length equation gives

$$\begin{aligned} 2 (\mathbf{u}_{i+1}^k - \mathbf{u}_i) \cdot \delta \mathbf{u}^k + 2\alpha (\lambda_{i+1}^k - \lambda_i) \delta \lambda^k \\ = -g^k. \end{aligned} \quad (3.40)$$

Using Equation (3.39),

$$\begin{aligned} 2 (\mathbf{u}_{i+1}^k - \mathbf{u}_i) \cdot \mathbf{u}_R^k + 2 (\mathbf{u}_{i+1}^k - \mathbf{u}_i) \cdot \mathbf{u}_P^k \delta \lambda^k \\ + 2\alpha (\lambda_{i+1}^k - \lambda_i) \delta \lambda^k = -g^k. \end{aligned} \quad (3.41)$$

Collecting the terms containing $\delta \lambda^k$ gives

$$\begin{aligned} 2 (\mathbf{u}_{i+1}^k - \mathbf{u}_i) \cdot \mathbf{u}_R^k + \delta \lambda^k \left[2 (\mathbf{u}_{i+1}^k - \mathbf{u}_i) \cdot \mathbf{u}_P^k \right. \\ \left. + 2\alpha (\lambda_{i+1}^k - \lambda_i) \right] = -g^k. \end{aligned} \quad (3.42)$$

Therefore,

$$\delta \lambda^k = \frac{-g^k - 2 (\mathbf{u}_{i+1}^k - \mathbf{u}_i) \cdot \mathbf{u}_R^k}{2 (\mathbf{u}_{i+1}^k - \mathbf{u}_i) \cdot \mathbf{u}_P^k + 2\alpha (\lambda_{i+1}^k - \lambda_i)}. \quad (3.43)$$

In Equation (3.43), the current value of g^k , together with $\mathbf{u}_R^k$ and $\mathbf{u}_P^k$, is known. Then, $\delta \mathbf{u}^k$ is obtained from Equation (3.39).

The trial displacement and load multiplier are updated as

$$\mathbf{u}_{i+1}^{k+1} = \mathbf{u}_{i+1}^k + \delta \mathbf{u}^k, \quad (3.44)$$

and

$$\lambda_{i+1}^{k+1} = \lambda_{i+1}^k + \delta\lambda^k. \quad (3.45)$$

Therefore, at each Newton–Raphson step:

1. Compute the tangent stiffness matrix $\mathbf{K}_t^k$ and the internal force vector $\mathbf{F}_{\text{int}}^k$.
2. Compute the residuals $\mathbf{R}^k$ and g^k .
3. Solve for $\mathbf{u}_R^k$ and $\mathbf{u}_P^k$.
4. Compute $\delta\lambda^k$.
5. Compute $\delta\mathbf{u}^k$.
6. Update $\mathbf{u}_{i+1}^{k+1}$ and λ_{i+1}^{k+1} .

The procedure is repeated until

$$\|\mathbf{R}^k\| < \varepsilon_R, \quad |g^k| < \varepsilon_g. \quad (3.46)$$

Abaqus commercial software uses modified version of this method, but the general pattern remains the same.

3.4 *Differential Evolution Algorithm*

Differential Evolution (DE) was selected because the optimization problem is treated as a simulation-based black-box problem. The design variables are the Bézier-spline control points that define the guide curve of each composite ply. For every candidate set of control-point values, the framework constructs the corresponding ply guide curves and generates the laminate model. This process described in details in the previous chapter. The goal is to maximize the critical buckling load P_{cr} . Explicit analytical relations between the buckling load and control points coordinates are not directly available. Differential Evolution is suitable for this task because it requires only evaluations of the objective function and does not rely on analytical derivatives. In general, any other optimization method which specific researcher consider appropriate can be implemented in the framework.

DE method was described by Storn and Price in their 1997 paper [53]. The general description of the algorithm is given based on this manuscript using the example directly related to this research.

The system has some real-valued properties:

$$y_m, \quad m = 0, 1, 2, \dots, M - 1. \quad (3.47)$$

There might be some constraints as well:

$$y_m, \quad m = M, M + 1, \dots, M + K - 1. \quad (3.48)$$

In this research, there is only one property of the system which is being evaluated, P_{cr} . There are also no constraints introduced.

This property depends on design variables:

$$x_n, \quad n = 0, 1, 2, \dots, N - 1. \quad (3.49)$$

This series is a multiple of eight, because a single ply of the composite laminate is defined by one guide curve, and four control points with two coordinates each are needed to initialize it. Since all control points are given in the UV domain (Chapter 2), all values of the design variables should be between 0 and 1:

$$0 \leq x_n \leq 1, \quad n = 0, 1, 2, \dots, N - 1. \quad (3.50)$$

Therefore, the problem can be described as a variation of the N -dimensional parametrization vector:

$$\mathbf{x} = (x_0, x_1, \dots, x_{N-1})^T. \quad (3.51)$$

until the property y_m , P_{cr} , is optimized. In a more compact way, it can be formulated as a minimization problem:

$$f(\mathbf{x}) = -P_{\text{cr}}(\mathbf{x}). \quad (3.52)$$

Storn and Price introduce parallel direct-search parameter vectors:

$$\mathbf{x}_{i,G}, \quad i = 0, 1, 2, \dots, NP - 1. \quad (3.53)$$

This vector represents the complete parametrization of the i -th candidate design in generation G . Its individual components are denoted by $x_{i,n,G}$, where n identifies a particular design variable. For example, in the case of a single-ply laminate, there are 8 design variables. The user should choose the size of the population NP , how many vectors are going to be evaluated in parallel, and the number of generations. In this research, it is determined by the number of available software licenses to run multiple analyses simultaneously. Let us say that $NP = 10$. The number of generations is the number of optimization steps. If five population states G are considered, the resulting set of vectors can be described as a $10 \times 8 \times 5$ array. The initial population can be chosen randomly or based on some known

solutions. In this work, a known laminate design is included in the initial population, while the remaining population members are generated as admissible variations of that design. For $NP = 10$, we can define it as:

$$\mathbf{x}_{i,0}, \quad i = 0, 1, 2, \dots, 9. \quad (3.54)$$

Each vector contains 8 design variables:

$$\mathbf{x}_{i,0} = (x_{i,0,0}, x_{i,1,0}, \dots, x_{i,7,0})^T. \quad (3.55)$$

Then mutation of the initial population should be performed, so mutation vectors should be defined. It can be done using many strategies; Storn and Price introduce two of them in their paper. The **SciPy** library for Python, which is used in this research, has the **best1bin** strategy, so that specific method is described here. An example with $NP = 10$ and 8 design variables is followed here. First, the best current solution should be chosen from the initial population. The function in Equation (3.52) is evaluated for each candidate. The candidate with the lowest $-P_{\text{cr}}$ is now $\mathbf{x}_{\text{best},0}$. The mutation vector for $\mathbf{x}_{0,0}$ is defined as:

$$\mathbf{v}_{0,0} = \mathbf{x}_{\text{best},0} + F(\mathbf{x}_{r_1,0} - \mathbf{x}_{r_2,0}). \quad (3.56)$$

where F is the mutation coefficient defined by the user, and $\mathbf{x}_{r_1,0}$ and $\mathbf{x}_{r_2,0}$ are randomly chosen vectors from the current population. In the same way, but with different random vectors each time, $\mathbf{v}_{1,0}$, $\mathbf{v}_{2,0}$, etc., are defined up to $\mathbf{v}_{9,0}$. Thus, at this point, there are 10 candidate vectors in the population, and each of them has a corresponding mutation vector. After that, for each pair of $\mathbf{x}_{i,0}$ and $\mathbf{v}_{i,0}$, **SciPy** creates a trial vector $\mathbf{u}_{i,0}$. It has $N = 8$ components as well. For each component, **SciPy** generates a random value q_n from 0 to 1. It should be compared with the recombination parameter CR , which is chosen by the user. One component of the trial vector should be taken from the mutant vector; **SciPy** makes this choice randomly. Let us call this component n_{rand} .

$$u_{i,n,0} = \begin{cases} v_{i,n,0}, & q_n < CR \text{ or } n = n_{\text{rand}}, \\ x_{i,n,0}, & \text{otherwise.} \end{cases} \quad (3.57)$$

Here,

$$n = 0, 1, \dots, 7. \quad (3.58)$$

Now the trial vectors $\mathbf{u}_{i,0}$ are evaluated as:

$$f(\mathbf{u}_{i,0}) = -P_{\text{cr}}(\mathbf{u}_{i,0}). \quad (3.59)$$

Then the new population is formed using the rule:

$$\mathbf{x}_{i,1} = \begin{cases} \mathbf{u}_{i,0}, & -P_{\text{cr}}(\mathbf{u}_{i,0}) \leq -P_{\text{cr}}(\mathbf{x}_{i,0}), \\ \mathbf{x}_{i,0}, & \text{otherwise.} \end{cases} \quad (3.60)$$

After that, the new iteration G starts. The optimization runs until the prescribed G is reached or until the convergence criterion is achieved:

$$\text{std}(-P_{\text{cr},i}) \leq \text{atol} + \text{tol} |\text{mean}(-P_{\text{cr},i})|. \quad (3.61)$$

where $\text{std}(-P_{\text{cr},i})$ and $\text{mean}(-P_{\text{cr},i})$ are the standard deviation and mean value of the objective function inside the population, while tol and atol are tolerance coefficients. By default, they are equal to 0.01 and 0, respectively, in **SciPy**.

3.5 Optimization Framework

The general description of the framework is given below. The individual elements of this framework are already described, so this goal of this section is to show the big picture. The framework is visualized in Figure 3.7.

The workflow begins with the Python **SciPy** optimizer, which generates a set of design parameters for the current optimization iteration. These parameters are passed to the simulation run script, which manages the execution of the finite element simulations. Commercial finite element software Abaqus is being utilized for the nonlinear buckling analysis using Riks method. The script retrieves the required finite elements model definitions (mesh, loading, boundary conditions), which are stored as Abaqus input files from the file system. Geometry is taken as *.step file, also from the file system. Using these inputs and the parameters provided by the optimizer, the script calls the model file generation script described in the Chapter 2. This script modifies the template model and geometry definitions according to the specified parameters and generates a complete Abaqus input file for the current simulation. During this process, the model should pass the manufacturing check. If the check fails, the case is penalized, and a new set of control points is generated. Otherwise, after the input file has been generated, it is returned to the simulation run script, which initiates the Abaqus job. Abaqus performs the finite element analysis and produces an output database

*.odb file (ODB) containing the simulation results. Once the analysis is completed, the simulation run script accesses the ODB file and performs post-processing, extracting the critical load P_{cr} for the optimization procedure.

The optimization process is performed using the DE algorithm. Based on the results obtained, the algorithm generates a new population of design variables for the next iteration. The updated parameters are then passed back to the simulation workflow, and the process is repeated until the optimization procedure reaches its termination criteria.

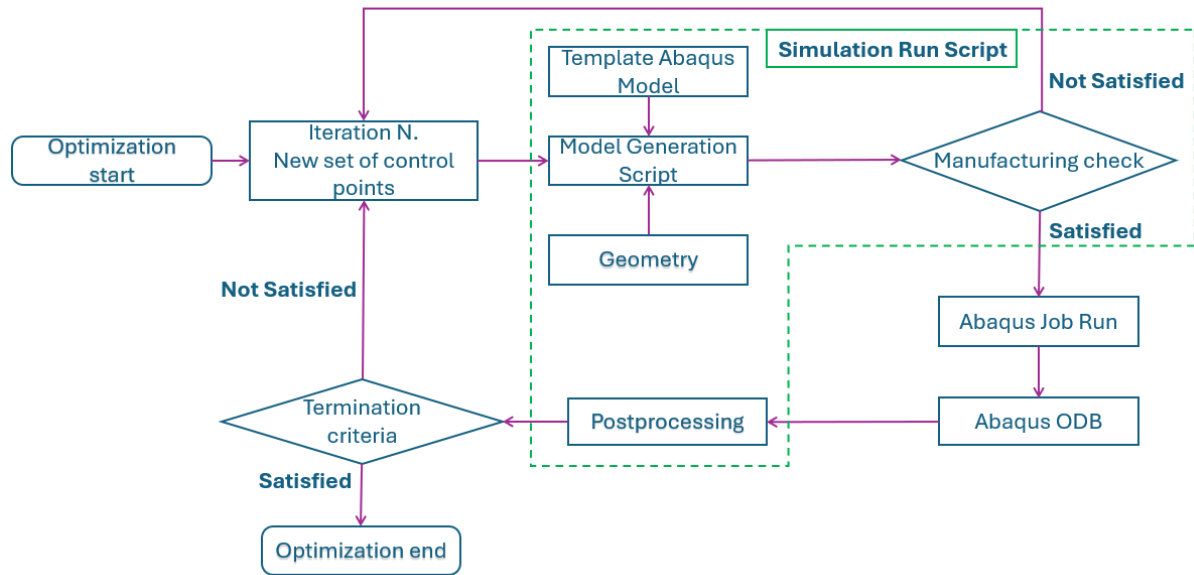


Figure 3.7: Flow chart of optimization procedure executed using the Python-built tool integrated with Abaqus.

Chapter 4

APPLICATION OF THE FRAMEWORK TO A REPRESENTATIVE STRUCTURAL PROBLEM

4.1 Chapter Scope and Objectives

The assumption that VAT laminates may improve structural performance of composite parts was described in the Chapter 1. The only reliable approach to manufacture such parts is AFP. Due to the manufacturing signatures of this technology, optimization of VAT structures seems much more complex than just to find the best mathematical solution of the given problem. To consider these manufacturing signatures during optimization, automated framework is needed. This framework is described in previous chapters. The main goal of this chapter is to test it on suitable problem and show that VAT laminates may improve structural response.

The suitable structural problem should have the complex geometry and be curved. It also should be broadly connected to structures where application of VAT structures can be useful in the future. The selected geometry was pulled from a notional design representative of NASA's Universal Stage Adapter for the Block 1B of the Space Launch System (SLS) rocket. This fairing-like structure serves to transition the exploration upper stage (EUS) to the Orion spacecraft[54]. As a key structural component of the rocket body, a primary loading condition for design optimization is the buckling load and mode. Because the adapter is carried through the EUS trans-lunar injection burn to send Orion to the Moon, weight is a critical design consideration. Further, the conical shape provides a 3D curved surface to assess the impact manufacturability of steered fiber paths optimized for buckling response. For these reasons, it was chosen as the artifact of the study. The notional surface was created using Dassault's CATIA V5 software as presented in Figure 4.1 where the highlighted portion is the selected geometry for analysis. The upper radius is 118.83 in with the radius of the lower edge being 139.70 in. Following suit, the length of the diagonal edges is 100.21 in.

The framework and methods used by the author have some limitations, so several important assumptions were made. First, the conical plate is analyzed in isolation to determine optimal fiber paths. In reality, a conical structure, due to its radial reduction, requires tow dropping. This process itself creates meaningful overlaps and gaps in the layup. These manufacturing signatures can have a significant impact on optimized steered fiber paths. Second, a distributed rather than a uniform load is used in this research due to the author's

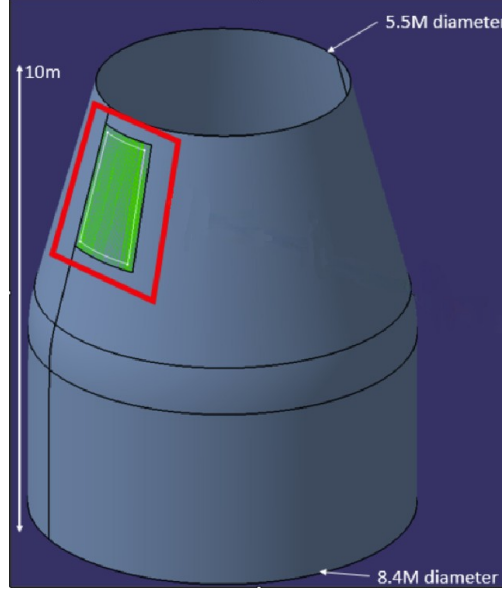


Figure 4.1: The selected geometry is the conical portion of a universal stage adapter for the SLS rocket where the highlighted section is the selected geometry for optimization and analysis.

view that VAT structures might be more effective in cases with a non-uniform load distribution. Ignoring this conditions means that the obtained results are not valid for a complete conical payload shell, but rather only for the conical plate itself.

Therefore, this study seeks to apply the ideas described in the previous chapters to optimize the fiber layup of a curved, rectangular-plan, thin plate in order to maximize the critical buckling load under the stated assumptions. The study serves as a proof-of-concept demonstration of the capabilities of the developed optimization framework.

4.2 Problem Statement

4.2.1 Boundary and Loading Conditions

The boundary and loading were conditions set as presented in Figure 4.2. The bottom edge was fixed for all translations and out-of-plane rotations, the lateral edges were simply supported to prevent out-of-plane displacement, and the top edge was supported in a way to not allow out-of-plane displacement and rotation. The loading via distributed load was placed on the top edge in a way so that it was purely an in-plane compressive load which is required for a non-linear buckling analysis.

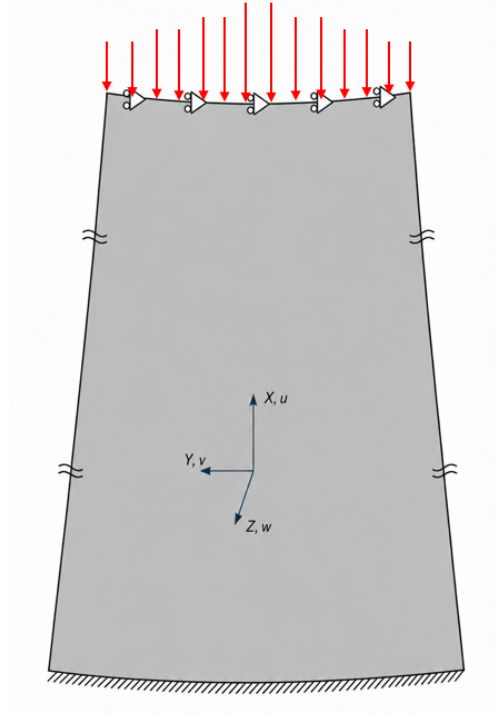


Figure 4.2: Schematic of the problem setup detailing the plate geometry, boundary conditions, and loading conditions.

4.2.2 Layup and Material Properties

The chosen layup for optimization is $(0/[\theta_1/\theta_2/\theta_3/\theta_4]_2)_s$. First 0° is needed to check if fiber stuck well to the tooling surface. The material properties of the selected composite are presented in Table 4.1.

Table 4.1: Material Properties

Property	Value	Property	Value
E_{11} (ksi)	18274.764	G_{12} (ksi)	725.190
E_{22} (ksi)	725.190	G_{13} (ksi)	725.190
E_{33} (ksi)	725.190	G_{23} (ksi)	725.190
ν_{12}	0.3	ν_{13}	0.05
ν_{23}	0.05	Ply thickness (in)	0.0075

4.2.3 Manufacturing Signatures

There exist many AFP manufacturing signatures and implementation of all of them is difficult at this point. In general, they can be divided by three big groups: constraints, defects and parameters. By constraints things which cannot be violated are meant, for instance,

the minimum steering radius. The example of parameter is fiber course width. It can be chosen among several available widths, but not directly constrained. Gaps and overlaps can be attributed to defects. Although defects are treated as a separate category, their size, such as gap or overlap size, can still be constrained in the optimization.

Several manufacturing signatures are considered in this research. The first is the head path, which is required to be continuous, with the start and end points of each centerline located outside the engineering boundary. The second is the minimum steering radius, which should not be smaller than the fiber course width. The sizes of gaps and overlaps are constrained to be no greater than 0.5 inches. Fiber course width MS is also included, 2-inch and 4-inch course widths are considered.

4.3 Optimization

4.3.1 Finite Elements Model

The finite element analysis was conducted using the commercial software Abaqus. The finite element model comprised 2176 fully integrated four-node quadrilateral shell elements (S4) and 2275 nodes. The mesh was kept fixed during the optimization process. Curvilinear fiber courses were modeled by assigning discrete fiber angles at the centroid of each finite element. Finite elements mesh is presented in Figure 4.3. Mesh convergency analysis was performed, results shown in Figure 4.4.

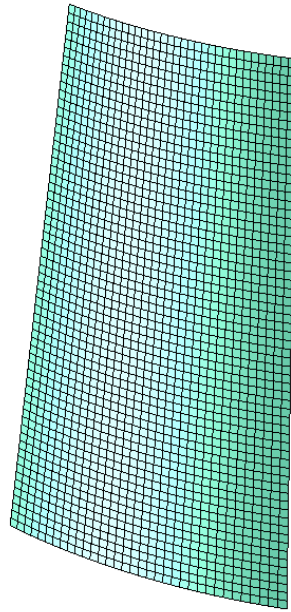


Figure 4.3: Finite elements mesh.

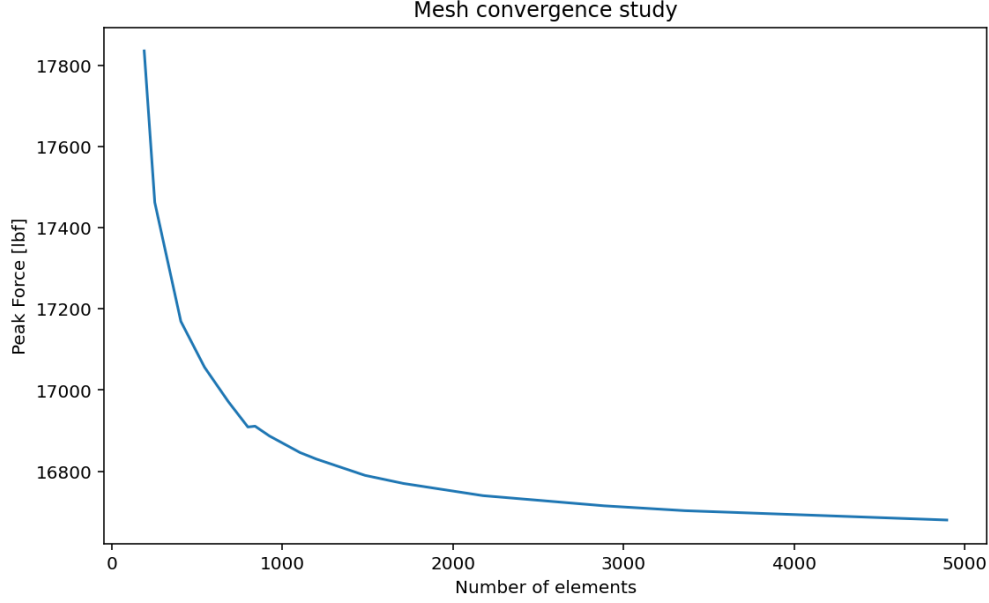


Figure 4.4: Finite elements mesh.

An initial sinusoidal imperfection was applied to enable the nonlinear load-response problem using Riks arc length method. Following analytical expressions for expected buckling shape, a double Fourier series of the form:

$$\sum_{m=1}^k \sum_{n=1}^l A_{ij} \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right) \quad (4.1)$$

is seeded into the mesh by offsetting the coordinates of the nodes. First few buckling modes are considered with the weightage term A_{mn} . A mesh convergence study was performed to ensure numerical accuracy.

4.3.2 Straight Fibers Optimization

To find best solution with straight fibers (non-VAT), Riks nonlinear buckling was used in conjunction with differential evolution. Convergency plot is presented in Figure 4.5. The best found solution is $(0/[80/-40/30/-20]_s)_2$. Force-displacement plot and typical mode evolution is presented in Figure 4.6. The visualization of the ply layups is presented in Figure 4.7.

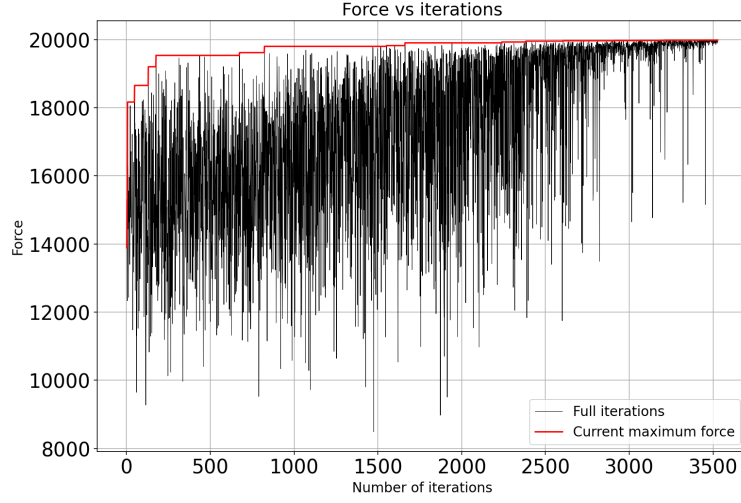


Figure 4.5: Convergency plot for Riks nonlinear analysis of for non-VAT panel

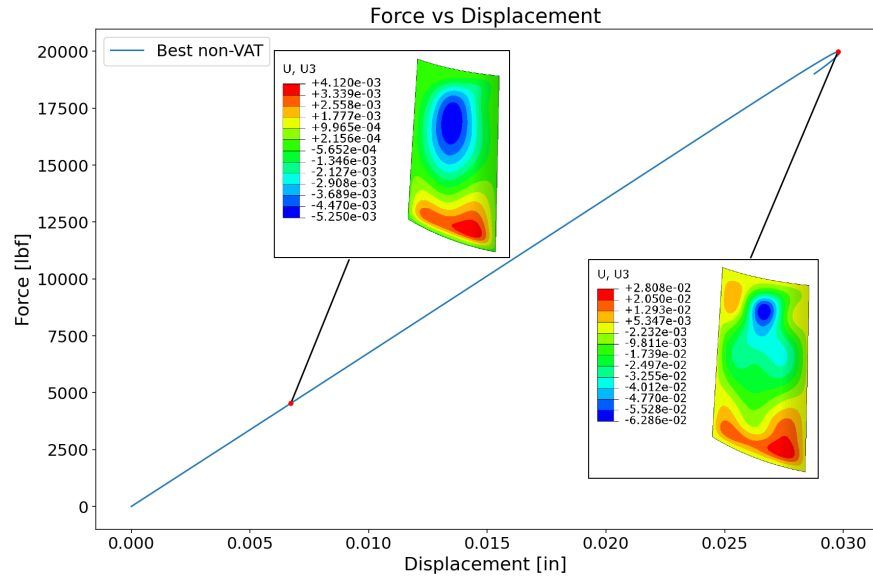


Figure 4.6: $(0/[80/-40/30/-20]_s)_2$ force-displacement plot of the analyzed structure for the given loading and BCs, using the Riks arc-length method.

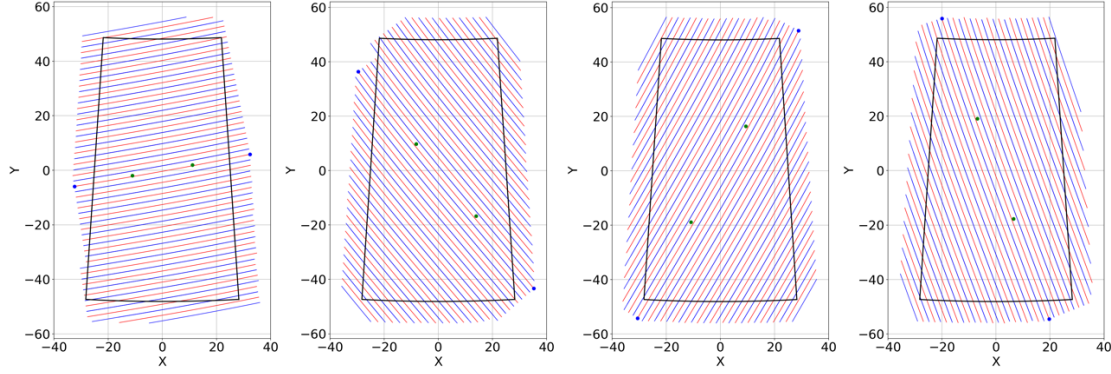


Figure 4.7: Best found non-VAT paths

4.3.3 Optimized Fiber Paths, 2-inch course width

$(0/[65/-50/-20/10]_s)_2$ laminate was chosen for optimization initiation point, since it demonstrated one of the best results among non-VAT laminates. Results were compared with $(0/[90/-45/45/0]_2)_s$ (equivalent almost-quasi-isotropic) and $(0/[80/-40/30/-20]_s)_2$ (best found non-VAT) solutions. The results and optimized VAT fiber paths for this width are presented in Table 4.2 and Figure 4.9. The Force-Displacement plots are presented in Figure 4.8.

Table 4.2: Critical buckling loads and initial stiffness for different layups, 2-inch course width.

Layup	P_{cr} , lbf	Initial stiffness, lbf/in	ΔP_{cr}	Δ stiffness
$(0/[90/-45/45/0]_2)_s$	19267	628377	—	—
$(0/[80/-40/30/-20]_s)_2$	19986	673298	+3.73%	+7.15%
Optimized VAT	21252	627171	+8.4%	-1.26%

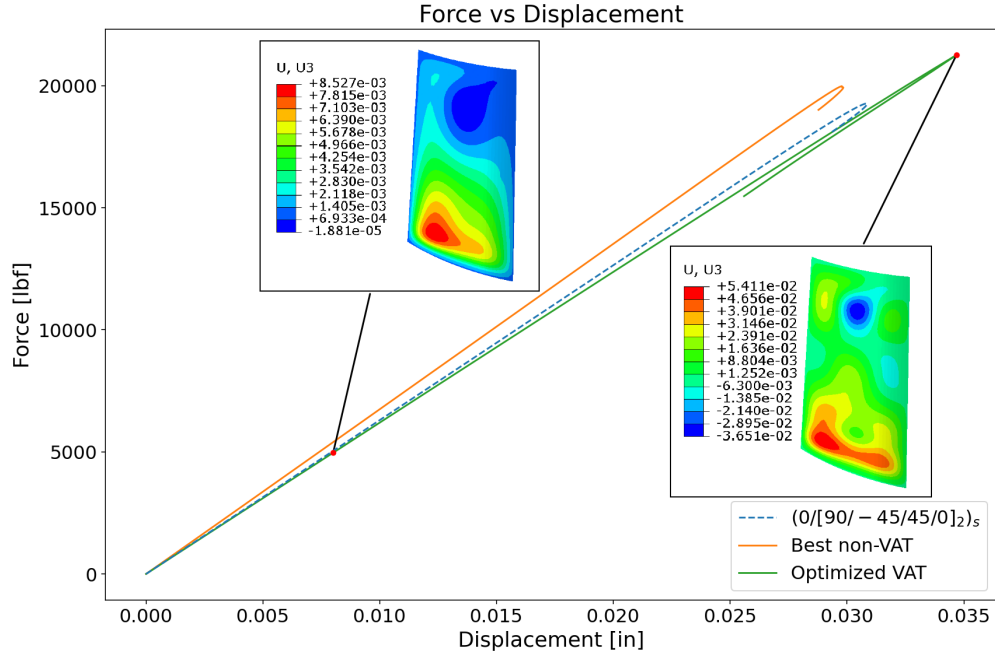


Figure 4.8: Equilibrium paths of Riks nonlinear analyses for different layups.

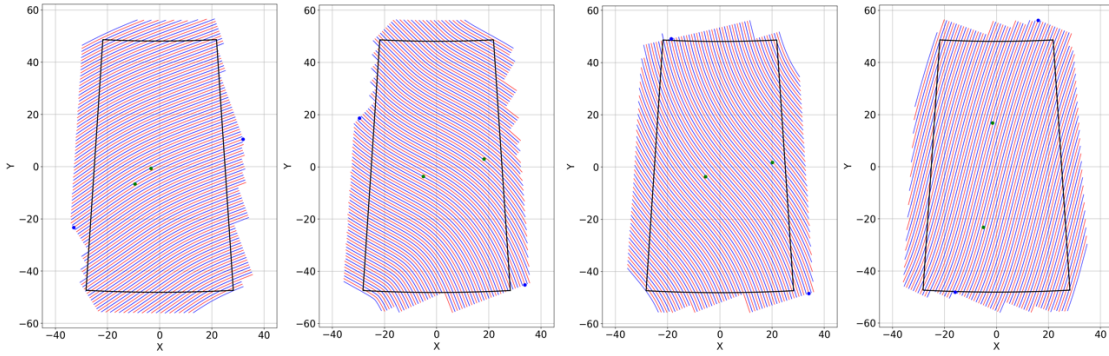


Figure 4.9: Optimized paths with 2-inch course width.

4.3.4 Optimized Fiber Paths, 4-inch course width

$(0/[80/-40/30/-20]_s)_2$ laminate was chosen for optimization initiation point, since it demonstrated one of the best results among non-VAT laminates. Results were compared with $(0/[90/-45/45/0]_2)_s$ (equivalent almost-quasi-isotropic) and $(0/[80/-40/30/-20]_s)_2$ (best found non-VAT) solutions. The results and optimized VAT fiber paths for this width are presented in Table 4.3 and Figure 4.11. The Force-Displacement plots are presented in Figure 4.10.

Table 4.3: Critical buckling loads and initial stiffness for different layups, 4-inch course width.

Layup	P_{cr} , lbf	Initial stiffness, lbf/in	ΔP_{cr}	Δ stiffness
$(0/[90/-45/45/0]_2)_s$	19267	628377	—	—
$(0/[80/-40/30/-20]_s)_2$	19986	673298	+3.73%	+7.15%
Optimized VAT	21658	627171	+12.41%	-0.19%

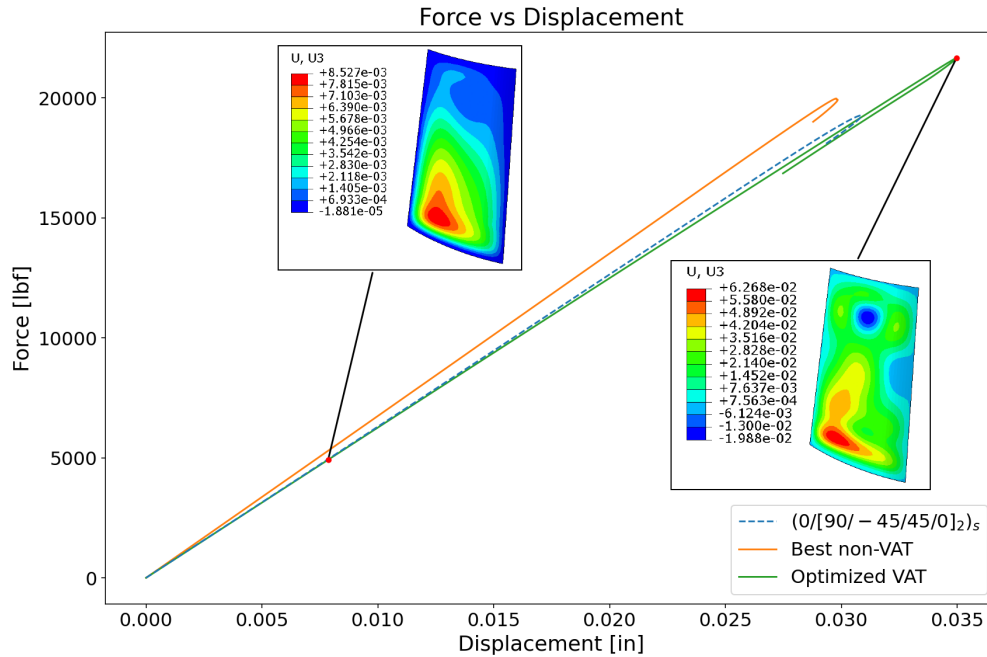


Figure 4.10: Equilibrium paths of Riks nonlinear analyses for different layups.

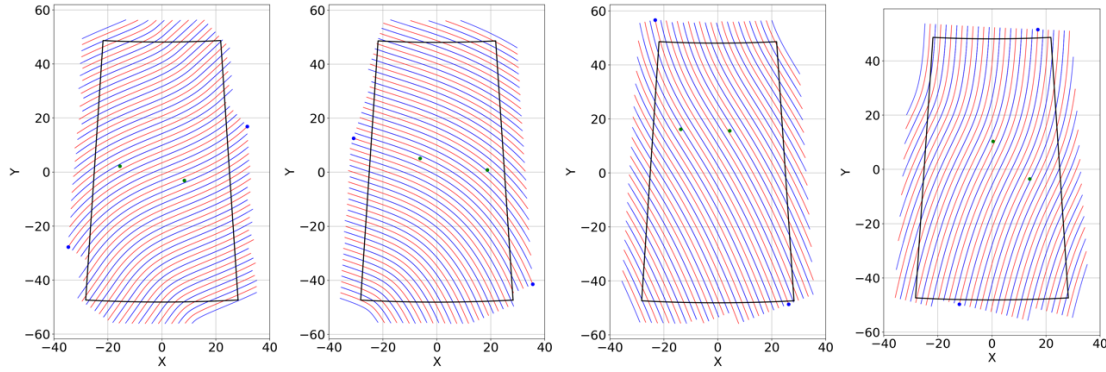


Figure 4.11: Optimized paths with 4-inch course width.

4.4 Chapter conclusions

It is demonstrated that curvilinear fibers can be used to improve the critical buckling load of thin-walled plate sections of large, curved structures. Optimization was performed using Python-based framework and commercial FEA software Abaqus, to optimize steered fiber paths that improve the buckling response of the plate. For the 2-inch course width, the corresponding improvements were 10.3% relative to the quasi-isotropic solution and 6.3% relative to the best non-VAT solution. For the 4-inch course width, the steered fiber design increased the critical buckling load by 12.4% and 8.4% with respect to the same baselines. These results demonstrate that the VAT design, enabled through AFP manufacturing, provides a measurable improvement in buckling performance over both quasi-isotropic and optimized non-VAT configurations.

CONCLUSIONS AND FUTURE WORK

5.1 *Summary*

This work presents the development and implementation of a general Design for Additive Manufacturing (DfAM) framework for AFP manufactured composite structures. The framework integrates optimization, FEA, automated fiber placement (AFP) path planning, and the evaluation of manufacturing signatures within a unified computational workflow. Its primary purpose is to enable automated structural optimization while accounting for the requirements, constraints, and characteristics associated with the manufacturing process.

Such an integrated framework is needed in composite engineering because structural design, manufacturing planning, and process-related effects are often considered separately. Combining these elements within a single workflow makes it possible to evaluate the design from a broader perspective and helps ensure that an optimized structure is both mechanically efficient and compatible with the intended manufacturing process.

In an idealized form, the framework consists of the following main components:

1. An AFP path-planning and analysis tool.
2. FEA software.
3. An optimization algorithm.
4. A manufacturing-process simulator.
5. A head program that combines and controls all elements.

All components except the manufacturing-process simulator were implemented in the created framework. The framework was successfully applied to the representative structural problem, confirming that the individual tools can be integrated into a single automated workflow for structural optimization considering manufacturing signatures.

Manufacturing signatures are defined broadly as the constraints and characteristics associated with the AFP process. Manufacturing constraints represent requirements that must be satisfied for a design to remain manufacturable, whereas manufacturing characteristics are selectable process parameters that have an impact on the generated fiber paths

and the resulting structure. The manufacturing signatures considered in this work include continuous AFP head paths, the placement of course start and end points outside the engineering boundary, the minimum allowable steering radius, limitations on gap/overlap sizes and fiber course width, which can be chosen from several options.

A key feature of the developed workflow is its ability to solve general engineering problems. The geometry of the analyzed part was not prescribed analytically, but imported directly from CAD. This shows the compatibility of the framework with existing design tools.

The results obtained for the representative structural problem demonstrate that the optimized VAT laminate achieved a 8.4% higher critical buckling load than the the best non-VAT laminate optimized for the same objective. This improvement shows that variation of the fiber orientation across the plies provides additional design freedom and enables the stiffness distribution to be tailored more effectively for enhanced buckling resistance.

The main conclusions of this work are therefore as follows:

1. VAT laminates can improve the buckling performance of composite structures compared with optimized constant-stiffness laminates.
2. A general DfAM framework that integrates structural optimization, AFP path planning, finite element analysis, and manufacturing-signature evaluation was successfully developed and implemented.

5.2 Future Work

Because the current framework has significant limitations, including its ability to process only open surfaces, major improvements are needed. Future work will focus mostly on integrating commercial AFP software into the developed framework. VERICUT Composite Programming (VCP), developed by CGTech, is planned for use in AFP path planning and path-quality analysis. VERICUT Composite Simulation (VCS), developed by the same company, will be incorporated to provide manufacturing-process simulation, which is not included in the framework presented in this research.

The integration of these tools will extend the current workflow by introducing more detailed evaluation of AFP manufacturing signatures and by improving the connection between structural path planning, process simulation, FEA and optimization. This development will support a more complete DfAM framework in which structural performance and manufacturing feasibility can be considered within a single automated process.

BIBLIOGRAPHY

- [1] Parveez, B. and Kittur, M. I. and Badruddin, I. A. and Kamangar, S. and Hussien, M. and Umarfarooq, M. A. Scientific advancements in composite materials for aircraft applications: A review. *Polymers*, 14(22):5007, 2022.
- [2] Boeing. 787 Dreamliner by Design. <https://www.boeing.com/commercial/787/by-design>, 2026. Accessed: 2026-08-12.
- [3] Addcomposites. Afp vs hand layup: The manufacturing revolution reshaping composite production. <https://www.addcomposites.com/post/afp-vs-hand-layup-the-manufacturing-revolution-reshaping-composite-production-ojc6p>, 2025. Accessed: 2026-07-10.
- [4] Gürdal, Zafer and Olmedo, Reynaldo. In-plane response of laminates with spatially varying fiber orientations—variable stiffness concept. *AIAA Journal*, 31(4):751–758, 1993.
- [5] Paul M. Weaver, Kevin D. Potter, K. Hazra, M. A. R. Saverymuthapulle, and M. T. Hawthorne. Buckling of variable angle tow plates: From concept to experiment. In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Palm Springs, California, 2009. American Institute of Aeronautics and Astronautics.
- [6] Harik, R. and Brasington, A. *Automated Fiber Placement: Status, Challenges, and Evolution*. SAE International, 2025.
- [7] M. W. Hyer and H. H. Lee. The use of curvilinear fiber format to improve buckling resistance of composite plates with central circular holes. *Composite Structures*, 18(3):239–261, 1991.
- [8] Nagendra, S. and Kodiyalam, S. and Davis, J. E. and Parthasarathy, V. N. Optimization of tow fiber paths for composite design. In *36th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, New Orleans, Louisiana, 1995. American Institute of Aeronautics and Astronautics.
- [9] Dassault Systèmes. What is NURBS modeling and why is it crucial to CAD software? <https://www.3ds.com/store/cad/nurbs-modeling>. Accessed: 2026-07-10.

- [10] Peter Eigenraam and Andrew Borgart. Reverse engineering of free form shell structures: From point cloud to finite element model. *HERON*, 61(3):193–210, 2016.
- [11] Brian F. Tatting and Zafer Gürdal. Design and manufacture of elastically tailored tow placed plates. NASA Contractor Report NASA/CR-2002-211919, NASA Langley Research Center, Hampton, Virginia, August 2002.
- [12] Levend Parnas, Süha Oral, and Ümit Ceyhan. Optimum design of composite structures with curved fiber courses. *Composites Science and Technology*, 63(7):1071–1082, 2003.
- [13] Shahriar Setoodeh, Mostafa M. Abdalla, and Zafer Gürdal. Design of variable-stiffness laminates using lamination parameters. *Composites Part B: Engineering*, 37(4–5):301–309, 2006.
- [14] K. Chauncey Wu and Zafer Gürdal. Variable stiffness panel structural analyses with material nonlinearity and correlation with tests. In *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Newport, Rhode Island, 2006. American Institute of Aeronautics and Astronautics.
- [15] C. S. Lopes, P. P. Camanho, Z. Gürdal, and B. F. Tatting. Progressive failure analysis of tow-placed, variable-stiffness composite panels. *International Journal of Solids and Structures*, 44(25–26):8493–8516, 2007.
- [16] Zafer Gürdal, Brian F. Tatting, and K. Chauncey Wu. Variable stiffness composite panels: Effects of stiffness variation on the in-plane and buckling response. *Composites Part A: Applied Science and Manufacturing*, 39(5):911–922, 2008.
- [17] Hilburger, Mark W. Buckling of Thin-Walled Circular Cylinders. Technical Report NASA/SP-8007-2020/REV 2, National Aeronautics and Space Administration, 2020.
- [18] Vishwa Mohan Tiwari, Royan J. D’Mello, Avinkrishnan A. Vijayachandran, and Anthony M. Waas. The axial compressive response of thin, elastic, polygonal shells. *Journal of Applied Mechanics*, 91(6):061002, 2024.
- [19] A. W. Blom, B. F. Tatting, J. M. A. M. Hol, and Z. Gürdal. Fiber path definitions for elastically tailored conical shells. *Composites Part B: Engineering*, 40(1):77–84, 2009.
- [20] Adriana W. Blom, Mostafa Rassaian, Patrick B. Stickler, and Zafer Gürdal. Bending test of a variable-stiffness fiber-reinforced composite cylinder. In *51st*

- AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Orlando, Florida, 2010. American Institute of Aeronautics and Astronautics. AIAA Paper 2010-2600.
- [21] B. C. Kim, K. Hazra, P. Weaver, and K. Potter. Limitations of fibre placement techniques for variable angle tow composites and their process-induced defects. In *Proceedings of the 18th International Conference on Composite Materials*, Jeju, Republic of Korea, August 2011.
 - [22] K. Fayazbakhsh, M. Arian Nik, D. Pasini, and L. Lessard. Defect layer method to capture effect of gaps and overlaps in variable stiffness laminates made by automated fiber placement. *Composite Structures*, 97:245–251, 2013.
 - [23] Gustavo Gonzalez Lozano, Ashutosh Tiwari, Christopher Turner, and Simon Astwood. A review on design for manufacture of variable stiffness composite laminates. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 230(6):981–992, 2016.
 - [24] J. P.-H. Belnoue, T. S. Mesogitis, O. J. Nixon-Pearson, J. Kratz, D. S. Ivanov, I. K. Partridge, K. D. Potter, and S. R. Hallett. Understanding and predicting defect formation in automated fibre placement pre-preg laminates. *Composites Part A: Applied Science and Manufacturing*, 102:196–206, 2017.
 - [25] Avinkrishnan A. Vijayachandran, Paul Davidson, and Anthony M. Waas. Optimal fiber paths for robotically manufactured composite structural panels. *International Journal of Non-Linear Mechanics*, 126:103567, 2020.
 - [26] Avinkrishnan A. Vijayachandran and Anthony M. Waas. Steered fiber paths for improved in-plane compressive response of aerostructural panels: Experimental studies and numerical modeling. *Composite Structures*, 289:115426, 2022.
 - [27] Avinkrishnan A. Vijayachandran and Anthony M. Waas. Minimizing stress concentrations using steered fiberpaths and incorporating realistic manufacturing signatures. *International Journal of Non-Linear Mechanics*, 146:104160, 2022.
 - [28] Wei Zhao and Rakesh K. Kapania. Buckling analysis and optimization of stiffened variable angle tow laminates with a cutout considering manufacturing constraints. *Journal of Composites Science*, 6(3):80, 2022.

- [29] August Noevere, Von Jamora, Matt Godbold, and Brian Tatting. Framework for optimization of AFP composite structures. Presentation at the 24th International Conference on Composite Materials, August 2025. NASA Technical Reports Server, Document ID 20250007422.
- [30] Ben J. Francis, Max B. Kirkpatrick, Drew B. Sander, Alex R. Brasington, and Ramy Harik. Towards a virtual manufacturing architecture to capture the automated fiber placement lifecycle. In *SAMPE 2023 Conference and Exhibition*, Seattle, WA, 2023.
- [31] D. O. Evans. *Fiber Placement*. Cincinnati Machine, Cincinnati, Ohio, 1997.
- [32] Electroimpact. Composites manufacturing customer solutions. <https://www.electroimpact.com/Products/composites-manufacturing/customer-solutions.aspx>. Accessed: 2026-07-14.
- [33] Ramy Harik, Clint Saidy, Stephen J. Williams, Zafer Gurdal, and Brian Grimsley. Automated fiber placement defect identity cards: Cause, anticipation, existence, significance, and progression. In *Proceedings of the SAMPE 2018 Conference and Exhibition*, Long Beach, California, USA, May 2018. Society for the Advancement of Material and Process Engineering.
- [34] Christopher Sacco, Anis Baz Radwan, Ramy Harik, and Michael Van Tooren. Automated fiber placement defects: Automated inspection and characterization. In *Proceedings of the SAMPE 2018 Conference and Exhibition*, Long Beach, California, USA, May 2018. NASA report no. NF1676L-29116, NASA NTRS document ID 20190027133.
- [35] Plasticity Manual. NURBS overview. <https://doc.plasticity.xyz/cad-essentials/nurbs-overview>. Accessed: 2026-08-03.
- [36] Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. Methods for non-linear least squares problems. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, 2004.
- [37] Maschen. Tangent normal binormal unit vectors. Wikimedia Commons, 2011. SVG image, public domain. Accessed: 2026-07-16.
- [38] Kristopher Tapp. *Differential Geometry of Curves and Surfaces*. Undergraduate Texts in Mathematics. Springer, Cham, 2016.
- [39] Mark Powell. Introduction to differentiable geometry. Course lecture notes: <https://www.maths.gla.ac.uk/~mpowell/DiffGeomCurvesSurfaces.html>. Accessed: 2026-07-16.

- [40] Lia Vas. Surfaces: Normal and gaussian curvatures. Differential Geometry course notes: <https://liavas.net/courses/math430/files/Surfaces.pdf>. Accessed: 2026-07-16.
- [41] Directflights. Direct flights from Seattle to Paris. <https://www.directflights.com/SEA-CDG>. Accessed: 2026-08-04.
- [42] Siddharth A. Pantoji, Christos Kassapoglou, and Daniël M. J. Peeters. Predicting gaps and overlaps in automated fiber placement composites by measuring sources of manufacturing process variations. *Composites Part B: Engineering*, 291:111891, 2024.
- [43] Scientific Lib. Minimum bounding box. <https://www.scientificlib.com/en/ComputationalGeometry/MinimumBoundingBox.html>. Accessed: 2026-07-17.
- [44] Jacob Fish and Ted Belytschko. *A First Course in Finite Elements*. John Wiley & Sons, Ltd, 2007.
- [45] Dennis M. Kochmann. Introduction to finite element analysis. Course lecture notes: https://ethz.ch/content/dam/ethz/special-interest/mavt/mechanical-systems/mm-dam/documents/Notes/IntroToFEA_red.pdf. Accessed: 2026-07-20.
- [46] Dassault Systèmes. Eigenvalue buckling prediction. <https://docs.software.vt.edu/abaqusv2024/English/?show=SIMACAEANLRefMap/simaanl-c-eigenbuckling.htm>. Accessed 2026-07-20.
- [47] Dassault Systèmes. General and perturbation procedures. <https://docs.software.vt.edu/abaqusv2024/English/?show=SIMACAEANLRefMap/simaanl-c-linearnonlinear.htm>. Accessed 2026-07-20.
- [48] Dassault Systèmes. Stress measures. <https://docs.software.vt.edu/abaqusv2024/English/?show=SIMACAETHERefMap/simathe-c-stressmeas.htm>. Accessed 2026-07-20.
- [49] Altair Engineering, Inc. Nonlinear static analysis. https://help.altair.com/hwsolvers/os/topics/solvers/os/analysis_nonlinear_quasi_static_c.htm. Accessed 2026-07-20.
- [50] Ion Leahu-Aluas and Farid Abed-Meraim. A proposed set of popular limit-point buckling benchmark problems. *Structural Engineering and Mechanics*, 38(6):767–802, 2011.
- [51] E. Riks. An incremental approach to the solution of snapping and buckling problems. *International Journal of Solids and Structures*, 15:529–551, 1979.

- [52] Chennakesava Kadapa. A simple extrapolated predictor for overcoming the starting and tracking issues in the arc-length method for nonlinear structural mechanics. *Engineering Structures*, 234:111755, 2021.
- [53] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.
- [54] National Aeronautics and Space Administration. Space launch system/universal stage adapter. <https://www.nasa.gov/glenn/glenn-expertise-space-exploration/sls-usa/>, 2024. Accessed: Mar. 13, 2026.
- [55] Carl de Boor. Package for calculating with B-Splines. *SIAM Journal on Numerical Analysis*, 14(3):441–472, 1977.

Chapter A

EXAMPLE OF NURBS SURFACE EVALUATION

This appendix illustrates the mapping of a point from a normalized parametric domain to the physical domain of a NURBS surface. To keep the numerical example compact, all control-point weights are set equal to one. Under this assumption, the NURBS formulation reduces to a non-rational B-spline surface.

Consider a quadratic surface defined by a 3×3 control-point net, where

$$i = 0, 1, 2, \quad j = 0, 1, 2. \quad (\text{A.1})$$

The control points are

$$\begin{aligned} \mathbf{P}_{0,0} &= (0, 0, 0), & \mathbf{P}_{0,1} &= (0, 20, 0), & \mathbf{P}_{0,2} &= (0, 40, 0), \\ \mathbf{P}_{1,0} &= (20, 0, 0), & \mathbf{P}_{1,1} &= (20, 20, 0), & \mathbf{P}_{1,2} &= (20, 40, 0), \\ \mathbf{P}_{2,0} &= (40, 0, 0), & \mathbf{P}_{2,1} &= (40, 20, 0), & \mathbf{P}_{2,2} &= (40, 40, 0). \end{aligned} \quad (\text{A.2})$$

The polynomial degrees in both parametric directions are

$$p = q = 2, \quad (\text{A.3})$$

and the corresponding clamped knot vectors are

$$U = V = \{0, 0, 0, 1, 1, 1\}. \quad (\text{A.4})$$

Because all weights satisfy

$$w_{ij} = 1 \quad \forall i, j, \quad (\text{A.5})$$

the rational NURBS surface equation reduces to the B-spline form. Furthermore, since the knot vectors are clamped, contain no internal knots, and the number of control points in each parametric direction is equal to the polynomial degree plus one, the resulting surface is also a quadratic Bézier surface.

The surface equation is

$$\mathbf{S}(u, v) = \sum_{i=0}^2 \sum_{j=0}^2 N_{i,2}(u) M_{j,2}(v) \mathbf{P}_{i,j}. \quad (\text{A.6})$$

The B-spline basis functions are evaluated using the Cox–de Boor recursion [55]. The zeroth-degree functions are defined as

$$N_{i,0}(u) = \begin{cases} 1, & u_i \leq u < u_{i+1}, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.7})$$

$$\begin{aligned} N_{0,0}(0.5) &= 0 && \text{since } [0, 0) \text{ is an empty interval,} \\ N_{1,0}(0.5) &= 0 && \text{since } [0, 0) \text{ is an empty interval,} \\ N_{2,0}(0.5) &= 1 && \text{since } 0 \leq 0.5 < 1, \\ N_{3,0}(0.5) &= 0 && \text{since } [1, 1) \text{ is an empty interval.} \end{aligned} \quad (\text{A.8})$$

For degree $p \geq 1$,

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \quad (\text{A.9})$$

A term in Equation (A.9) is defined as zero when its denominator is zero.

For

$$u = 0.5, \quad (\text{A.10})$$

the nonzero zeroth-degree basis function is

$$N_{2,0}(0.5) = 1. \quad (\text{A.11})$$

The first-degree basis functions are therefore

$$\begin{aligned} N_{0,1}(0.5) &= 0, \\ N_{1,1}(0.5) &= 0.5, \\ N_{2,1}(0.5) &= 0.5, \\ N_{3,1}(0.5) &= 0. \end{aligned} \quad (\text{A.12})$$

The quadratic basis functions are

$$\begin{aligned}
N_{0,2}(0.5) &= \frac{u_3 - 0.5}{u_3 - u_1} N_{1,1}(0.5) = \frac{1 - 0.5}{1 - 0} (0.5) = 0.25, \\
N_{1,2}(0.5) &= \frac{0.5 - u_1}{u_3 - u_1} N_{1,1}(0.5) + \frac{u_4 - 0.5}{u_4 - u_2} N_{2,1}(0.5) \\
&= \frac{0.5 - 0}{1 - 0} (0.5) + \frac{1 - 0.5}{1 - 0} (0.5) = 0.50, \\
N_{2,2}(0.5) &= \frac{0.5 - u_2}{u_4 - u_2} N_{2,1}(0.5) = \frac{0.5 - 0}{1 - 0} (0.5) = 0.25.
\end{aligned} \tag{A.13}$$

Thus,

$$[N_{0,2}(0.5), N_{1,2}(0.5), N_{2,2}(0.5)] = [0.25, 0.50, 0.25]. \tag{A.14}$$

Since the surface has the same degree and knot vector in the v -direction,

$$[M_{0,2}(0.5), M_{1,2}(0.5), M_{2,2}(0.5)] = [0.25, 0.50, 0.25]. \tag{A.15}$$

Substitution into Equation (A.6) gives

$$\begin{aligned}
\mathbf{S}(0.5, 0.5) &= (0.25)(0.25)\mathbf{P}_{0,0} + (0.25)(0.50)\mathbf{P}_{0,1} + (0.25)(0.25)\mathbf{P}_{0,2} \\
&\quad + (0.50)(0.25)\mathbf{P}_{1,0} + (0.50)(0.50)\mathbf{P}_{1,1} + (0.50)(0.25)\mathbf{P}_{1,2} \\
&\quad + (0.25)(0.25)\mathbf{P}_{2,0} + (0.25)(0.50)\mathbf{P}_{2,1} + (0.25)(0.25)\mathbf{P}_{2,2}.
\end{aligned} \tag{A.16}$$

Using the control-point coordinates from Equation (A.2),

$$\mathbf{S}(0.5, 0.5) = \begin{bmatrix} 20 \\ 20 \\ 0 \end{bmatrix}. \tag{A.17}$$

Therefore, for this planar example, the center of the normalized parametric domain is mapped to the center of the physical surface.