

Interner Bericht

DLR-IB-BT-ST-2025-161

Development and Expansion of Thermoplastic AFP Process Simulation

Master Thesis

Federica Bruno

Deutsches Zentrum für Luft- und Raumfahrt

Institut für Bauweisen und Strukturtechnologie
Pfaffenwaldring 38-40
70569 Stuttgart



DLR

Deutsches Zentrum
für Luft- und Raumfahrt

DLR-IB-BT-ST-2025-161

**Development and Expansion of Thermoplastic AFP
Process Simulation**

Masterarbeit

Federica Bruno

Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Bauweisen und Strukturtechnologie
Pfaffenwaldring 38-40, 70569 Stuttgart

Stuttgart, den 11.11.2025

Seiten: 138

Abbildungen: 105

Tabellen: 17

Geschäftsführung:

(Nicole Lützenburger)

Abteilungsleiter BT-BGF:

(Dipl.-Ing. S. Nowotny)

Betreuer:

(Daniel Fricke)

Summary

Fiber Reinforced Plastics (FRPs) are high-performance materials that combine an excellent strength-to-weight ratio, stiffness-to-weight ratio, and exceptional corrosion resistance. In the aerospace sector, FRPs are used in critical structural components such as fuselages, wings, and support elements. A key advantage of FRPs is the ability to align the load-bearing fibers along the load paths, leading to significant improvements in structural efficiency compared to traditional metallic constructions in lightweight designs. Automated Fiber Placement (AFP) technology is used as an alternative to manual tape laying. The proper setup of such processes is crucial for optimal performance and configuration. This research focuses on the thermal simulation of the Thermoplastic Automated Fiber Placement (T-AFP) process, simulated in ANSYS, particularly the non-Autoclave methodology for in-situ consolidation of thermoplastic tapes to create a multi-layer composite structure. A transient thermal analysis is performed, simulating the actual placement of tapes, which are simultaneously positioned and heated by a laser heat source. In the first phase of the research, a method was developed to define the paths on the surface of a tool flat panel, based on its geometric characteristics, the orientation angle, the lay-up strategy, and the dimensions of the tapes. The approach aims at extracting the coordinates of the initial and final points of the trajectories. The 2D geometry of the tool is created in CATIA, with a tetrahedral mesh generated using the open-source GHSM software. A Python script is implemented to create CSV output files containing the coordinates of the extreme points of the paths, modeling the tapes as "boxes" on the tool surface. This method does not require complex geometric formulations, making it applicable to flat, square, rectangular, and arbitrarily shaped flat panels. The formulation is based on mathematical and trigonometric principles, such as the parametrization of the segment and vector calculations for defining the normals to the plane. A study on curved panels is also proposed. The second phase focuses on configuring the transient thermal analysis. Temporal steps are defined where the deposited tape elements are selected, and necessary thermal conditions are applied to simulate the process, including

heating, cooldown, and optical control waiting times. To support the simulation, a Mechanical APDL script is written, as ANSYS does provide the necessary functionality for selecting the tapes during the analysis stages not in the GUI, only via script. Subsequently, the analysis is finalized by creating the 3D model of the laminate in contact with the tool, using ANSYS drawing tools and inserting all necessary geometric and thermal process parameters. The results obtained confirm the feasibility of the proposed method, providing consistent and comparable outcomes to the real process.

Contents

Summary	i
Contents	iii
Nomenclature	vi
List of Figures	vii
List of Tables	xi
1 Introduction	1
1.0.1 Outline	2
1.1 In-Situ AFP Process: State of Art	3
1.1.1 AFP Machine Hardware	3
1.1.2 Material and Design	5
1.1.3 Path Planning	6
1.1.3.1 Path Creation Strategies	8
1.1.4 Coverage Strategies	9
1.1.5 Tool	10
1.2 Objectives of the thesis	11
2 Developed Method: Path Planning	13
2.1 Libraries Used	15
2.2 First Section: <code>mesh_intersection.py</code>	17
2.2.1 Input Data	18
2.2.2 Process of Functions	21
2.2.2.1 Definition of the Intersection Planes	21
2.2.2.2 Intersection points	25
2.2.2.3 Extension of the lengths of the paths	31
2.2.3 Data Output	40
2.2.4 Core function	42
2.2.5 Flowchart : Full First Section	43

2.3	Second Section: <code>boxes_steps_mode.py</code>	44
2.3.1	Thermal Strategy Modes	45
2.3.2	Input Data	50
2.3.3	Process of Functions	52
	2.3.3.1 Division of the paths into discrete segments	52
	2.3.3.2 Projection Points	55
2.3.4	Core Process	58
	2.3.4.1 Step Times	60
2.3.5	Output Data	64
2.3.6	Flowchart : Full Second Section	67
2.4	Curved Panel	68
3	Thermal Transient Analysis	73
3.1	Ansys Parametric Design Language	75
3.2	Analysis Setting	77
	3.2.1 Model Geometry and Mesh	77
	3.2.2 Material and Boundary Conditions	80
	3.2.3 Workbench Project	82
3.3	Commands (APDL)	83
	3.3.1 Data Acquisition	84
	3.3.2 Variables Set-Up	85
	3.3.3 Analysis Initialization	85
3.4	Flowchart: Commands (APDL)	86
3.5	Commands (APDL) 2	87
	3.5.1 Differentiation of the Steps	88
	3.5.2 Selection of Elements	89
	3.5.3 Process of the Script	91
	3.5.3.1 Step setting	91
	3.5.3.2 Angle identification	91
	3.5.4 Activation of elements and application of BCs	93
	3.5.5 Flowchart : Commands (APDL) 2	97
4	Results	98
4.1	Test Conducted	98
4.2	Flat Square Panel	101
	4.2.1 4-Layer Laminates Results	103
	4.2.2 8-Layer Laminate Results	108
4.3	Flat Round Panel	111
5	Conclusions	115
5.1	Future Work	116

Contents	v
----------	---

6 Data Appendix	i
------------------------	----------

Bibliography	iii
---------------------	------------

Nomenclature

AFP Automated Fiber Placement

T-AFP Thermoplastic Automated Fiber Placement

L-AFP Laser-Assisted Fiber Placement

FRP Fiber Reinforced Plastic

UD Unidirectional Carbon Fibers

BC Boundary Condition

GUI Graphical User Interface

APDL ANSYS Parametric Design Language

MAPDL ANSYS Mechanical APDL

API Application Programming Interface

ACP ANSYS Composite PrePost

FEA Finite Element Analysis

CAD Computer-Aided Design

CAM Computer-Aided Manufacturing

CAE Computer-Aided Engineering

RS Reference System

DOF Degrees of Freedom

CSV Comma-Separated Values format

PLY Polygon File Format

AI Artificial Intelligence

List of Figures

1.1	Characteristics of the Heat Sources [15]	4
1.2	L-AFP Head	4
1.3	DLR Thermoplastic AFP facility at the Institute of Structures and Design [39]	5
1.4	DLR Thermoplastic AFP Head at the Institute of Structures and Design [39]	5
1.5	Characteristics of material types [15]	6
1.6	Definitions of the main common defects in the AFP Process [41]	7
1.7	Benchmark of surface differentiation	8
2.1	Standard Strategy	18
2.2	Out Strategy	19
2.3	Center Strategy	19
2.4	Standard: New mode showing single (left) and double (right) alternating modes	19
2.5	Out: New mode showing single (left) and double (right) alternating modes	20
2.6	Center: New mode showing single (left) and double (right) alternating modes	20
2.7	Plane with normal vector $[\tan(30^\circ), -1, 0]$	22
2.8	Test intersection planes 60° and 120°	22
2.9	Test intersection planes 30° and 45°	23
2.10	Comparison of Rectangular panels with different tape widths and 45° angle (200 mm - left and 100 mm - right)	24
2.11	Flowchart: Definition of the Intersection Planes	25
2.12	Flowchart: Boundary Edges Function	26
2.13	Localization intersection point	27
2.14	Flowchart: Boundary Intersection	28
2.15	Comparison of S strategy at 90° - left and 0° - right	28
2.16	Comparison of C strategy at 90° - left and O strategy at 90° - right	29

2.17	Comparison of C strategy at 60° - left and 120° - right	29
2.18	Comparison of Round panels with different tape widths and 45° angle (200 mm - left and 100 mm - right)	30
2.19	Comparison S strategy at 60° - left and 0° - right with tape width 150mm	30
2.20	Comparison O strategy at 60° - left and 0° - right with tape width 150mm	31
2.21	Paths on the mesh	31
2.22	Schematic representation of the tape model	32
2.23	Tapes with endpoints coincident with the boundary intersections	33
2.24	Trigonometric Formulation for extensions added to the path	33
2.25	Extension points details	35
2.26	Extreme point of the path details	35
2.27	Tapes with endpoints coincident with the boundary intersections plus extensions	36
2.28	Extended tapes on a round shape	37
2.29	Round shape with tapes width of 12 mm	37
2.30	Round shape with tapes width of 75 mm	37
2.31	Flowchart: Extension Points Calculation	38
2.32	Flowchart: Max Distance Couple Points	39
2.33	Flowchart: First Section <code>mesh_intersection.py</code>	43
2.34	Thermal Strategy Modes Scheme	45
2.35	Modelling the Movement of the Laser Head	48
2.36	Laminate Example Model	49
2.37	Sequence of Time Steps for the Example	49
2.38	Flowchart: Division paths in segments	52
2.39	Subdivision of Tapes into Segments	53
2.40	Examples of Division Points for Square Panel: 0° (top left), 30° (top right), 45° (bottom left), 90° (bottom left)	54
2.41	Projection mathematical formulation	55
2.42	Examples of Projection Points for Square Panel: 0° (top left), 30° (top right), 45° (bottom left), 90° (bottom right)	56
2.43	Flowchart: Calculation of the Projection Points	57
2.44	Flowchart: Step Time Modes	62
2.45	Example Mode 1 : Flash Heating N, Waiting Type T	63
2.46	Example Mode 5 : Flash Heating Y, Waiting Type T	63
2.47	<code>info_process.csv</code>	64
2.48	<code>info_layer.csv</code>	64
2.49	<code>info_tape.csv</code>	64
2.50	Example: Plot Output Square Panel	65
2.51	Example: Plot Output Round Panel	66

2.52	Flowchart: <code>boxes_steps_mode.py</code>	67
2.53	Curved Panel Model : section of airliner fuselage	68
2.54	Representation of the intersection of the two planes with the tool surface mesh	69
2.55	Intersections at the boundary between offset planes and the mesh	69
2.56	Node position verification	70
2.57	Paths on the tool surface with a tape width of 150 mm	71
2.58	Paths on the tool surface with a tape width of 12.7 mm	71
2.59	Test with the geometry of one part of the tool surface	72
3.1	APDL Applications and Scripting Functions	76
3.2	Quadratic and Linear Elements	78
3.3	Entire Model: Tool (Yellow) and Laminate (White)	79
3.4	Tape Reference System	79
3.5	Depiction of the geometric constraints of the mesh	79
3.6	BCs Applied	82
3.7	Workbench Project and Commands (APDL) Details	82
3.8	Flowchart: Commands (APDL)	86
3.9	Detailed Functioning of <code>step_counter</code>	88
3.10	Element Selection Boxes	90
3.11	Central and Local Reference Systems	90
3.12	Flowchart: Angle Identification	92
3.13	Example: Selection of elements and application of BCs	94
3.14	Example: Coordinates List List of coordinates of the points of the segments and tapes	94
3.15	Flowchart: Elements Activation and Application BCs	96
3.16	Flowchart: Commands (APDL) 2	97
4.1	Lamination of an 8-layer flat round panel.	99
4.2	Model and mesh of the 8-layer flat round panel and tool	100
4.3	Lamination of an 8-layer flat square panel	100
4.4	Model and mesh of the 8-layer flat square panel and tool	100
4.5	S Mode. Angles: 0° at the top right, 90° on the left, -45° at the bottom left, and 45° at the bottom right	101
4.6	C Mode. Angles: 0° at the top right, 90° on the left, -45° at the bottom left, and 45° at the bottom right	102
4.7	O Mode. Angles: 0° at the top right, 90° on the left, -45° at the bottom left, and 45° at the bottom right	102
4.8	Detail of the placement of the second tape on the first layer, segment by segment	104

4.9	Comparison of the second tapes of the first layer, deposited with the flash heating strategy (right) and without it (left) . . .	104
4.10	A comparison of tape temperatures, with flash heating deactivated on the left and activated on the right, during the robot's waiting time	105
4.11	Deposition, with/without flash heating (right/left), of the central tape of the second layer, oriented at -45°	105
4.12	Heat diffusion along the various layers	106
4.13	Average temperature over time of the 4-layer model placed with Flash heating, shown in black, with a detailed view in red	107
4.14	Average temperature over time of the 4-layer model placed without Flash heating, shown in black, with a detailed view in red	108
4.15	Heat transfer along the thickness of the laminate	108
4.16	Average temperature over time of the 8-layer model placed with Flash heating, shown in black, with a detailed view in red	109
4.17	Average temperature over time of the 8-layer model placed without Flash heating, shown in black, with a detailed view in red	109
4.18	Average temperature over time of the 8-layer model placed without Flash heating, shown in black, with a detailed view in red, Center lay-up Strategy	110
4.19	S Mode. Angles: 0° at the top right, 90° on the left, -45° at the bottom left, and 45° at the bottom right	111
4.20	C Mode. Angles: 0° at the top right, 90° on the left, -45° at the bottom left, and 45° at the bottom right	112
4.21	Round panel: Average temperature over time of the 8-layer model placed with Flash heating, shown in black, with a detailed view in red, Standard lay-up Strategy	113
4.22	Round panel: Average temperature over time of the 8-layer model placed with Flash heating, shown in black, with a detailed view in red, Center lay-up Strategy	113
4.23	Heating step for a central tape of a round flat panel	114

List of Tables

2.1	Different Time Steps in Seconds	47
2.2	Different Time Steps in Seconds	48
2.3	Laminate Properties: Square Panel	50
2.4	Tapes Parameters	50
2.5	Different Time Steps in Seconds	50
2.6	Process Parameters	51
2.7	Different Time Steps in Seconds	61
3.1	Material parameters for CF/LM-PAEK	80
3.2	Start and End Time Definitions	95
3.3	Example: Start and End Time Definitions	95
6.1	Material parameters for CF/LM-PAEK	i
6.2	Different Time Steps in Seconds	i
6.3	Laminate Properties: Square Panel	i
6.4	Laminate Properties: Round Panel	ii
6.5	Tapes Parameters	ii
6.6	Different Time Steps in Seconds	ii
6.7	Process Parameters	ii

Chapter 1

Introduction

Over the past 50 years, the production of composite material components for the aerospace industry has increased exponentially, thanks to the benefits offered by these materials, such as high strength and stiffness, reduced weight, and exceptional resistance to fatigue and corrosion, which result in lower maintenance costs. Notable examples of the growing use of composite materials, replacing aluminium alloys, can be found in the latest models of the Boeing series [53], highlighting how these materials have become essential and difficult to replace in the aerospace sector. Additionally, the structures of next-generation aircraft, such as the Boeing 787 and the Airbus A350 XWB, contain approximately 50% of their weight in composite materials [30].

Alongside the increasing adoption of these materials, innovative advancements have been made in production and automation technologies. These technologies have been designed to make processes faster, more efficient, precise, and repeatable compared to traditional manual lay-up solutions, particularly in applications that require high levels of structural integrity and performance [17], while also ensuring a significant reduction in production costs. Automation reduces both labour costs and those associated with waste and material dispersion due to human errors. The Automated Fiber Placement (AFP) technology for thermoplastic materials reinforced with fibre meets both of these needs in this sector.

On one hand, the matrix of thermoplastic materials, as structured, can be repeatedly melted, formed, and solidified, enabling pressure forming operations and advanced joining methods for automatic assembly, allowing the combination of multiple simple parts into complex structures. On the other hand, fibre placement (especially in-situ placement) can be considered almost fully automated.

In fact, in-situ consolidation represents one of the strengths of the process as it eliminates the need for post-consolidation in an autoclave [46]. It involves

heating the thermoplastic material to its melting temperature, followed by cooling without crosslinking, thus combining the resin and fibre [20]. Heat is applied simultaneously, using a laser source at the nip point, and pressure is applied through the use of a compaction roller, facilitating the fusion of the thermoplastic matrix and the formation of bonds between reinforcing fibres (UD).

The extremely precise control of the input energy, by accurately defining the power and size of the laser heating area, ensures a highly controlled and optimised process. By selecting optimal heating, compaction/pressure, and tensioning conditions, prepreg strips can be continuously laid and consolidated in-situ to form a single layer, with multiple layers creating a durable laminate ready for use in applications. To date, with the expanding adoption of AFP technology in the Western industry [30], considerable research has been conducted to support the improvement of this process, particularly in two areas: finite element analysis (FEA) to study the thermal effects that occur during and after deposition, as well as the effects of post-treatment on the material, examining its structural characteristics and composition, and the development of advanced algorithms and models for fibre deposition strategies, with the goal of improving process efficiency and minimising the formation of voids between adjacent tape layers. For example, in the field of simulations and modelling, numerous studies have been conducted by engineers such as Mantell and Springer [35], Lamontia [31], Tierney [47], Gennes [23], and Sonmez [45], aiming to optimise process parameters and achieve high-quality results.

This thesis presents a transient thermal simulation designed to replicate the actual AFP process in the production of multilayered thermoplastic composite flat panels, in order to capture the dynamic nature of the process.

1.0.1 Outline

The thesis report is structured as follows. Chapter 1 introduces the fundamental aspects of the AFP process, with Section 2 providing a comprehensive overview and a state-of-the-art analysis. Section 3 presents a comparative assessment of the most commonly used materials and the various heat sources, while Section 4 explores different path planning strategies. Chapter 2 details the methodology developed in this work for the simulation of multilayered thermoplastic flat panel models. Chapter 3 focuses on the implementation and development of the thermal simulation, followed by Chapter 4, which discusses the validation process and the analysis of the conducted tests. Finally, in the Chapter 5, the conclusions evaluate the reliability of the proposed approach and its potential contributions to the optimisation of AFP processes.

1.1 In-Situ AFP Process: State of Art

The AFP technique offers significant advantages, particularly in the production of large-scale thermoplastic composite components. In traditional manufacturing processes, such as those involving presses or autoclaves, laminates are first stacked and then heated above the melting temperature. Pressure is subsequently applied to facilitate complete consolidation of the component. The disadvantages of these conventional methods include low efficiency, slow cooling rates, and extended production times.

Another limitation of traditional techniques is the inability to produce large components, constrained by the size and capabilities of the pressing machines. In contrast, with the AFP production technique, especially when employing in-situ consolidation, this issue is considerably alleviated. The flexibility and precision of the AFP process enable the production of complex and large components, thereby reducing production time and enhancing overall process efficiency.

1.1.1 AFP Machine Hardware

The AFP in-situ consolidation process involves the automation in deposition of fibre-reinforced thermoplastic tapes onto the surface of a mould and subsequently onto the already deposited laminate. During the process, the tapes are deposited using a machine head, which allows for the simultaneous application of heat and pressure, Figure 1.2.

This machine head consists of several key components: the compaction roller, which can be solid or segmented, as described in [15], and the heat source, which in this case is a laser (L-AFP) [25], [36]. Additionally, the head is equipped with a feed unit that operates with rollers to deliver the tapes with the correct tension. In some cases, for complex geometries, different feed paths may be used to minimise defects [41]. The machine is also equipped with a cutting unit, which enables the trimming of continuous fibres once the path is completed. A study aimed at optimising cutting speed is presented in [40]. The general dimensions of the tapes are up to 35 mm, but in this study context, the standard dimension of 12.7 mm, or 1/2 inch, is used.

The motion system is essential for the precise placement of tapes on various geometries. While both vertical and horizontal gantries are used, the most common system is the robotic arm, as it offers six degrees of freedom (DOF), along with an additional linear motion axis. Furthermore, it can be coupled with a rotator, resulting in a total of eight DOF, providing enhanced flexibility in tape placement [15]. The use of the laser, compared to other heat sources such as hot gas torches, IR, or pulsed light, offers several advantages.

Firstly, it enables high energy density and a short response time, allowing for higher deposition speeds. In fact, the consolidation time is significantly reduced. According to [24], for a deposition speed of 200 mm/s and a selection area of 5 mm, the time is less than 25 ms. Furthermore, in Figure 1.1, the table summarising the various differences between heat sources, as analysed by [15], is presented.

Figures 1.3 and 1.4 illustrates the Robotic Arm and Head of the AFP machine available at DLR, which has been used as the reference model for the development of this thesis work.

Heating System	Material	Controllability	Temperature
HGT	Thermoset/Thermoplastic	Low	Medium
IR	Thermoset	Medium	Low
Laser	Thermoplastic	High	High
Pulsed Light	Thermoset/Thermoplastic	High	Medium/High

Figure 1.1: Characteristics of the Heat Sources [15]

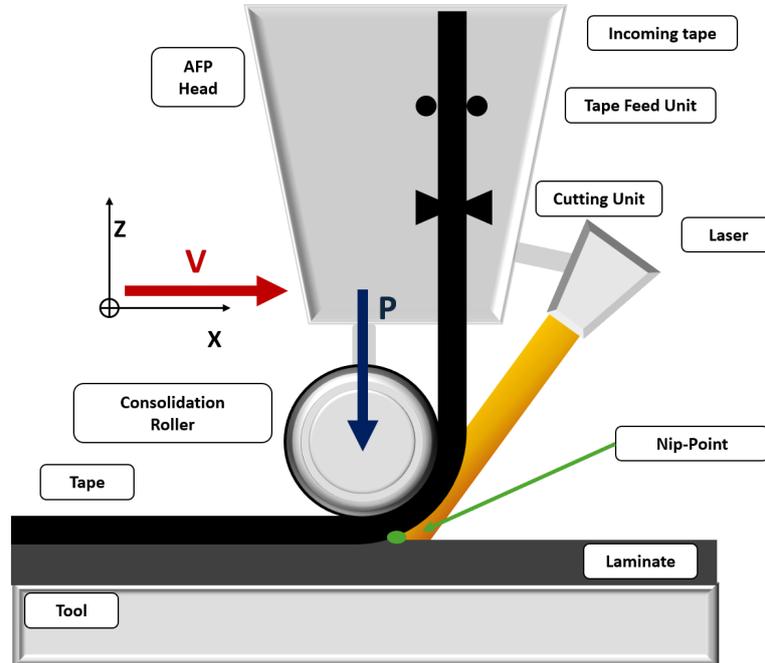


Figure 1.2: L-AFP Head



Figure 1.3: DLR Thermoplastic AFP facility at the Institute of Structures and Design [39]



Figure 1.4: DLR Thermoplastic AFP Head at the Institute of Structures and Design [39]

1.1.2 Material and Design

In the case under consideration, the in-situ consolidation process is simulated, which requires the use of thermoplastic materials. For a broader perspective on AFP processes, it is important to highlight that components can also be produced using thermosetting materials and dry fibre. These fibres are not "prepreg", meaning they are not pre-impregnated with resin. Thermosetting materials are widely used in this field due to their matrix, which reaches its melting point at a temperature very close to ambient conditions, requiring minimal energy for heating. These materials solidify into a three-dimensional structure [34], and in some cases, cooling is necessary to prevent the resin from polymerising too quickly, as it has a low viscosity. Dry fibre materials, on the other hand, are optimal for applications involving deformation or bending. Additionally, since they do not contain resin, they avoid clogging the machine head due to material accumulation [12]. However, they come

with disadvantages, such as the need for fusion at the end of the process or the application of an additional layer of another material to enable the adhesion of dry fibres [29]. Thermoplastic materials can be repeatedly heated and solidified, offering advantages such as recyclability, reworkability, and high-temperature performance. Furthermore, they do not require autoclave processing but do require very high temperatures for deposition, which can reach up to 400 °C [13]. In the case of the in-situ process, the heating phase, during which the material is warmed, must follow specific guidelines when selecting process parameters. It is particularly crucial to avoid overheating, which could lead to bond degradation and a reduction in performance. The heat is absorbed by the surface of the material and progressively conducted into the fibres and underlying layers. The heat source must be capable of maintaining a constant intensity. At the end of this phase, the matrix of the tape should be melted and, after passing through the melting point, pressed by the compaction roller to establish contact and form a bond with the underlying layer that was previously placed. Table 1.5, taken from [15], compares the key differences and characteristics of the various types of materials.

Material	Processing Temperature	Material Storage	Curing
Thermoset	Low	Frozen	Autoclave/Oven
Thermoplastic	High	Room Temp	<i>In-situ</i> /Autoclave
Dry Fiber	High	Room Temp	Infusion

Figure 1.5: Characteristics of material types [15]

1.1.3 Path Planning

In the overall workflow related to AFP layup, the layup strategy, which includes the starting point, path planning, and surface coverage, represents the initial phase of process planning [15]. It is a fundamental part of production with this technology and depends on various factors, including process and geometric parameters, such as the complexity of the shape of the production tool (2D or 3D), the materials used, and certification standards. There are several methods and mathematical implementations, but the selection of the layup strategy is mainly based on improving the mechanical and thermal properties of the final products and reducing defects [41]. The following table 1.6 lists the most common defects .

Defects	Definitions
Gaps/Overlaps	Gaps or overlaps can occur between two tows or two courses. A gap will weaken the structure while an overlap will increase its weight.
Wrinkle	A wrinkle is a short section of fiber that has raised off the tool due to excess heat or excessive steering. In these cases, the fiber has failed to fully adhere to the previous layers and create an uneven surface.
Bridging	Bridging appears very similar to wrinkling, and describes the failure of fiber paths to adhere to the tool surface. The tensions of the tows overcome the tows' adhesion to the surface, inducing the bridging. This is typically found with ply drops.
Twisted Tow	The tow has rotated on itself before being placed on the surface by the compaction roller.
Splice	A splice is when a tow ends and the next tow is superposed to the previous one during many inches.
Missing Tow	When a tow is not laid on the surface or does not adhere on it. This defect can be linked to an issue during the feeding function.
Foreign Objects or Debris	An object or a debris are laid under or on a tow. This will have an impact on all the following plies.
Fold/Roping	When the feeding system does not deliver the tow with the right orientation or the right tension, the laid tow will have a cylindrical shape more than a planar one on the surface.
Position Error	A tow placed on the wrong location.
Loose Tow	A loose tow is a tow which does not fully adheres to the previous ply. This is typically link to heat or compaction issues.

Figure 1.6: Definitions of the main common defects in the AFP Process [41]

Almost all methods for generating the paths of the tapes on the surface of the tool to be covered are based on a reference path. These can be defined on parametric models, on mesh (preferably triangular), and in some cases using numerical approaches on the 3D point-cloud of the geometry of the shape [48].

In the research article [18], automated path planning on CAD is compared, with particular attention to the standard formats STL and IGES.

In the case of mesh, there is a complete representation of the geometric properties of the shape, such as the normal, position, area, and vertices for each individual triangle in the mesh. However, some information, such as the material of the part, remains unknown, and errors may occur due to missing elements or overlaps. Although data analysis is advantageous for its simplicity, it can result in high computational costs and significant calculation times, especially if the mesh is refined in detail.

On the other hand, the use of parametric models provides complete informations, including the physical properties of the material. Additionally, precise geometric entities, such as points, NURBS lines, and splines, are available, allowing for a more accurate and detailed understanding of the surface.

In any case, the best model is chosen optimally in relation to the specific requirements of the application to be carried out.

To define the trajectory on the tool surface, the available methods are classified based on the geometric characteristics of the surface [41]. A geometry benchmark is drawn in Figure 1.7 .

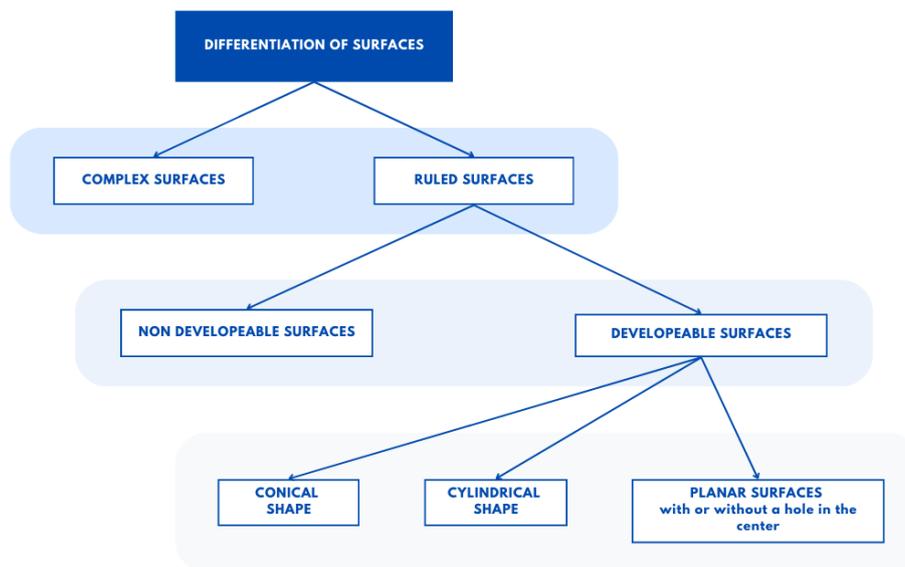


Figure 1.7: Benchmark of surface differentiation

1.1.3.1 Path Creation Strategies

Many methods presented in the literature for setting the paths of Automated Fiber Placement machines are based on the definition of an initial reference curve, which significantly impacts the overall layup outcome. The most common strategies are the fixed-angle strategy, the use of geodesics, and the variable-angle strategy. The fixed-angle strategy involves creating the curve starting from a point that has a constant angle across the entire surface, relative to a given axis or direction. Several solutions are found, using both mesh-based approaches [32] and parametric approaches [51], [44], [50].

In the first case, the described method utilises mesh information (such as normal vectors and triangles) to reconstruct a topology and generate reference curves at a fixed angle through intersection with tangent planes and appropriate algorithms. Successive curves are obtained by rotating the plane by a constant angle .

The parametric approach, which has more references and solutions in the literature, involves the intersection of a parametrised surface in a reference parametric plane and a major axis plane. An intersection equation of the planes is obtained, along which discretised points are identified according to a defined distance (also used for offset curves). Furthermore, by employing the Newton-Raphson method (NRM) [33], the starting point from which the curve propagates is determined, corresponding to the intersection of the

boundary and the projection of the major axis. To find the discrete points, the procedure moves along a defined direction with a specified step size, after which the NRM is used for convergence to the plane intersection line. Other variants have been developed based on this method, such as the insertion of a tangent point on the tool surface to simulate the presence of the compaction roller [51], the use of iterative algorithms employing a defined reference direction on surfaces [28], or even the application of this method to conical shapes, hence closed forms [14].

A geodesic curve is defined as the shortest path between two points on a three-dimensional surface in the Cartesian plane [37]. Naturally, in the case of a flat panel, the geodesic coincides with a straight line. To obtain the reference curve, it is necessary to know the starting point and the direction of the trajectory. If a parametric surface is given, a system of differential equations must be solved, governed by the curvature and geometry of the surface, with the initial conditions specified for the different cases [41].

Finally, variable-angle strategies involve the variation of the angle along the reference curve to create laminates with variable stiffness. Additionally, being more flexible, they overcome unidirectional constraints. However, this introduces greater complexity in calculations and optimisation. A study in this field of research is cited in [10]. The strategies for variable angles can be divided into three types: constant curvature [21], linear variation [38], and nonlinear variation[9].

1.1.4 Coverage Strategies

Coverage strategies are based on independent curves, offset curves, and shifted curves. The first method computes curves independently and is widely applied for solutions involving highly complex tool surfaces; each path is staggered with constant length and different directions. Furthermore, these strategies are commonly used for covering gaps and defects [15].

In the case of offset curves, the reference curve is used as the starting point, and a constant offset is applied to identify parallel curves until the total coverage of the laminate is achieved. In this case, both parametric-based approaches are used, which solve a system of numerical equations to find the subsequent curves, and mesh-based approaches, applying the Fast Marching Method, which is based on Eikonal equations for the propagation of the curve [52],[16]. Naturally, if the curvature of the reference curve changes, the parallel curves will follow the same variation.

For shifted curves, a perpendicular translation is applied; however, in the case of complex curves, this can lead to excessive formation of gaps or overlaps [26].

1.1.5 Tool

The geometry of the tool has a significant impact on the quality of the final product, particularly when the geometry is highly complex. In the case of closed-edge geometries, a loss of roller adhesion may occur, leading to a variation in compaction pressure [19]. In other cases, the formation of gaps and defects can be associated with geometries characterised by pronounced curvatures or non-homogeneous surface distributions. Harik et al. [27] contributed to the understanding of these phenomena by developing a methodology to anticipate defects such as gaps, overlaps, bridging, and twisting that arise during the deposition process on complex geometries. Wehbe et al. [49] propose a method for predicting the formation of wrinkles based on the curvature of the tool and the geometry of the deposition path.

Therefore, to achieve a good result, it is essential to optimise the manufacturing processes and appropriately combine the tool shape with the machine action.

1.2 Objectives of the thesis

In the context of the Automated Fiber Placement (AFP) process, particularly when dealing with scenarios characterised by high thermal gradients, non-linear material behaviour, and the presence or absence of crystallisation, predicting the stresses and deformations induced by the process becomes an inherently complex task. Although post-processing solutions exist to obtain these predictions, pre-processing solutions remain underdeveloped and more limited.

In this regard, it is essential to develop a simulation that can establish a robust foundation for the study and understanding of the AFP process dynamics, enabling the estimation of outcomes prior to the actual construction of laminated components. Such an approach would facilitate the optimisation of process parameters and production environment, as well as the prediction of the final product quality.

In the context of industrial-scale production, the adoption of such simulations would lead to a significant reduction in physical trials, thereby reducing both costs and production time. This would result in increased efficiency and acceleration of the production process. Moreover, in research, the implementation of advanced simulations would provide considerable advantages, improving the ability to make more accurate predictions and fostering innovation.

This work draws inspiration from the simulation produced by D. Fricke [22], who conducted a thermal-structural analysis for predicting the deformations of thermoplastic laminates produced via AFP. The objective of the present study is to optimise and enhance the existing simulation, making it more versatile and applicable to a broader range of scenarios. One of the primary goals is to enable the processing of flat panels with arbitrary, curved, and square shapes, using information derived solely from the mesh of these surfaces, without the need for any additional data other than the spatial domain. Another key objective is to ensure greater clarity in the development of supporting codes, as well as to facilitate future implementations. Furthermore, the work aims to increase the automation of the process, enabling debugging at intermediate steps and enhancing the speed of computation. The work is divided into two fundamental parts: the first involves the development of a novel path planning method on the surface of the tool, and the second concerns the setup of the simulation analysis within the ANSYS environment. The proposed method is based on the intersection of planes placed at a defined distance by a predetermined offset, which intersect the mesh plane of the tool's surface. This approach is inspired by the fixed angle theory for the intersection between the mesh plane and the intersection plane, the cover-

age theory with parallel curves at a defined offset, and integrates additional mathematical and geometric techniques implemented within the workflow of the method, which will be explained in detail in the following chapters.

For the modelling of the tapes, the process is carried out as "boxes", i.e. rectangles for geometry definition, and as rectangular cuboids in the three-dimensional space. This approach enables the implementation of element selection in ANSYS and the application of boundary conditions. Chapter 3 will provide a detailed explanation of the implementation steps in three-dimensional space.

Chapter 2

Developed Method: Path Planning

The algorithm for defining paths on the surface is developed using the Python programming language (Python 3.12.0), with support from several specific libraries.

The Python programming language was chosen due to its versatility, simplicity, and readability, as well as its widespread use across various domains, making it well-known and accessible. Its clear syntax and dynamic typing facilitate ease of learning and implementation. Furthermore, Python offers a vast range of libraries and frameworks, making it suitable for numerous applications and particularly well-suited for the work outlined below. Additionally, it is cross-platform, supports object-oriented programming, and benefits from an active community that continuously contributes to its development.

The implementation is divided into two parts, each corresponding to a separate script. This choice is motivated by two main reasons:

1. Modularity – Separation of core functions for enhanced code clarity and organization.
2. Maintainability – Facilitation of debugging, modifications, and extensions of the algorithm.

The first section of the algorithm, implemented in the script `mesh_intersection.py`, focuses on identifying the coordinates of the tape start and end. The second section, `boxes_data.py`, is dedicated to modeling the tapes across different layers, where they are classified as "boxes". Furthermore, this module generates the necessary data for the second part of the simulation in the ANSYS environment, including the coordinates and

geometric characteristics of the tapes. These data are also crucial for the thermal simulation process, as they include input parameters such as laser temperature, tool temperature, and various thermal convection values. To achieve these solutions, several functions have been developed and organized into specific subfunctions, some of which rely on mathematical and logical principles that will be explained later in the discussion.

2.1 Libraries Used

By utilizing Python's standard libraries, the primary advantage lies in the speed of obtaining results with low computational cost. Additionally, the algorithm is highly replicable and easily deployable on different devices without requiring significant computational resources. The following is an overview of the libraries employed:

- **Os:** The `os` library provides an interface for interacting with the underlying operating system. It enables file and directory management operations such as creation, renaming, deletion, and retrieval of file and folder information. Thanks to its ability to interact with the operating environment, it is essential for saving and reading files, as well as recalling directories. It also manages data export for use in the Ansys environmen [1].
- **NumPy:** Essential for scientific computing in Python, NumPy allows performing various operations and numerical calculations, such as computing the Euclidean distance, which is useful in a crucial step of tape modeling. It is also essential for managing multidimensional arrays and matrices [2].
- **Pandas:** Pandas is a fundamental Python library for data manipulation and analysis, particularly useful for working with tabular data similar to spreadsheets or SQL tables. In Pandas, data can be structured in `DataFrame` and `Series` objects. In this study, `DataFrame`, a two-dimensional data structure resembling a table with rows and columns (where each column can contain different data types such as numbers, strings, and dates), was utilized [3].
- **Trimesh:** Trimesh is an open-source library used for managing and processing meshes throughout the entire path definition process, especially for mesh intersections, surface normal analysis, and exporting 3D models in common formats such as OBJ, STL and PLY [4].
- **Matplotlib.pyplot:** Chosen as the tool for data visualization in Python, it is part of the Matplotlib library, a powerful package for generating 2D and 3D plots. It also supports animations and interactive visualizations. The `pyplot` module is designed to provide an API similar to MATLAB, facilitating the adoption of Python for plotting purposes [5].
- **mpl_toolkits.mplot3d:** The `Poly3DCollection` class, included in the `art3d` module of `mpl_toolkits.mplot3d` within Matplotlib, is specifically designed for three-dimensional visualization. This class, used in

this study, allows the creation and management of collections of three-dimensional polygons. In particular, Poly3DCollection is employed to visualize surfaces and meshes, including complex structures, offering aesthetic customization options such as surface color and transparency [6].

- **combinations** : is a function from the **itertools module**, not a standard Python library. It generates all possible combinations of a given length from a set of elements, without repetition and regardless of order. This function is used to define specific point pairs within groups of four [7].

2.2 First Section: `mesh_intersection.py`

The `mesh_intersection.py` script has three main purposes:

- Intersect planes with the mesh.
- Identify boundary intersections at the mesh's edge and order them according to the layer production strategy and define the paths.
- Generate output files in TXT format to be used in the second Python script.

In addition, additional lines of code are included to visualize the intersections in the mesh, allowing a visual inspection of the process.

2.2.1 Input Data

To formulate the solution, the following input data are required:

- **Mesh directory:** The directory containing the mesh file to be loaded into the simulation. In this study, Dassault Systèmes' CATIA software (V5-6R 2018) was employed for CAD, CAM, and CAE operations to design the reference surface, while the open-source software GMSH was used for mesh generation, which was subsequently exported in STL format. Furthermore, the implementation of two additional support scripts is provided: one for visualising the STL file, useful for model verification, and another for converting the STL files to PLY format with a triangular mesh.
- **Deposition angle:** An array containing the placement angles of the tapes.
- **Tape Width:** Specifies the width of the tapes, a fundamental parameter for defining the intersection planes.
- **Lay-up Strategy:** A configuration variable defining the strategy for layer creation. Standard, Out and Center modes are currently available, but in the future, additional modes may be implemented.
 - Standard S: The fiber placement strategy progresses from the left to the right side of the shape.

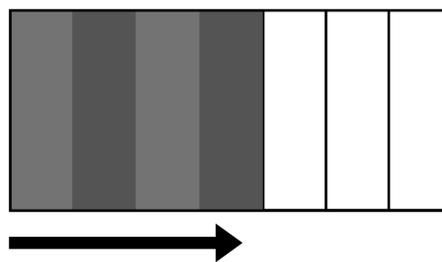


Figure 2.1: Standard Strategy

- Out O: The opposite strategy, where placement moves from the right to the left side.

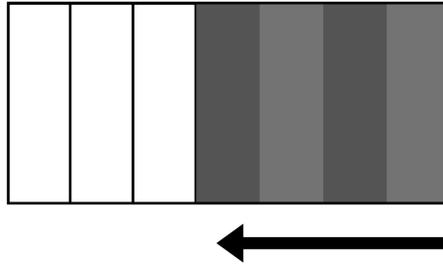


Figure 2.2: Out Strategy

- Center C: The first tape is positioned at the Center of the surface, and subsequent tapes are added symmetrically until the layer is completely filled.

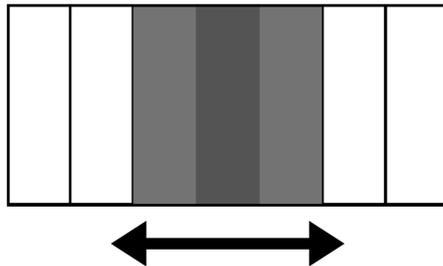


Figure 2.3: Center Strategy

- Other strategies :

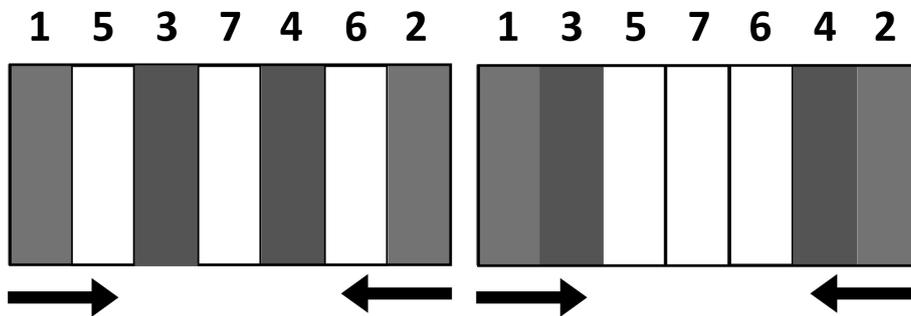


Figure 2.4: Standard: New mode showing single (left) and double (right) alternating modes

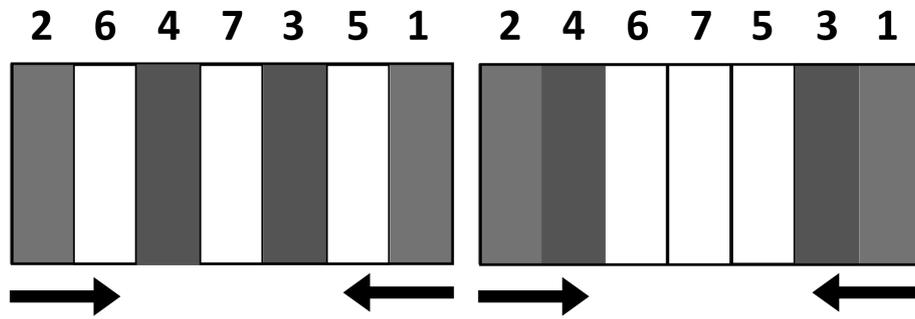


Figure 2.5: Out: New mode showing single (left) and double (right) alternating modes

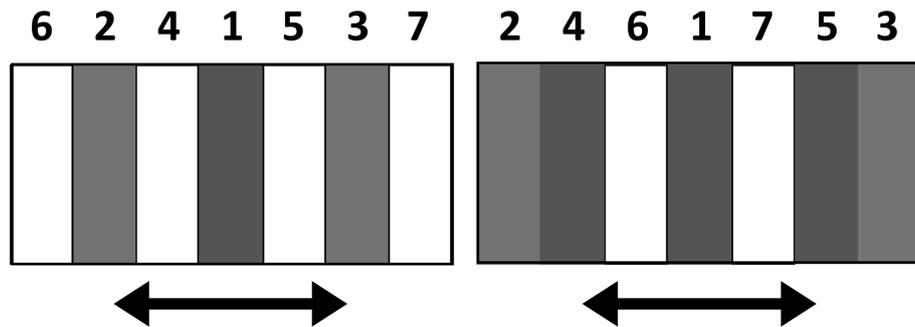


Figure 2.6: Center: New mode showing single (left) and double (right) alternating modes

2.2.2 Process of Functions

Once the data have been entered, the script proceeds with the definition of several subfunctions and their invocation within the core functions. These can be divided into two categories: one related to computation, data processing and export, and the other for visualisation. To understand their operation, it is essential to first express the mathematical concepts underlying the process.

2.2.2.1 Definition of the Intersection Planes

To define the correct orientation of the intersection plane in three-dimensional space and satisfy the requirement of the angle defining the paths, the normal to the plane is calculated. In the context of analytic geometry and computational physics, calculating the normal to a plane is crucial not only for determining the surface position, but also for applications involving light reflection and interactions between solid objects. Such applications require knowledge of the various angles relative to the reference system in use.

In this specific case of study, the reference system XYZ is placed at the centroid of the mesh. The Z -axis is aligned with the thickness of the model, and the XY -plane is parallel to the surface. The intersection plane must also be centered at the Centroid (2.2), parallel to the Z -axis, and inclined along the X -axis within the XY -plane, with the inclination defined by the angle chosen for positioning the paths.

From these considerations, it follows that the plane is defined by the normal vector 2.1, which has a relationship between X and Y with slope $m = \tan(\theta)$, where θ is the angle of inclination with respect to the X -axis. The normal has a negative component along the Y -axis and no component along the Z -axis. This implies that for every unit increment in x , the variation in Y is inversely proportional to the slope m . In other words, moving the point along the X -axis, the Y -component will adjust in such a way as to maintain the slope defined by m , with Y changing in the opposite direction to X :

Thus, the normal vector to the plane has the following form:

$$\vec{n} = \begin{pmatrix} m \\ -1 \\ 0 \end{pmatrix} = \begin{pmatrix} \tan(\theta) \\ -1 \\ 0 \end{pmatrix} \quad (2.1)$$

By substituting the centroid point, the point through which the plane passes 2.2, and the normal vector into the Cartesian equation (2.3) of the plane, the final form is obtained Eq. 2.4 .

$$p_0 = (0, 0, 0) \quad (2.2)$$

$$Ax + By + Cz + D = 0 \quad (2.3)$$

$$\tan(\theta)x - y + 0 = 0 \quad (2.4)$$

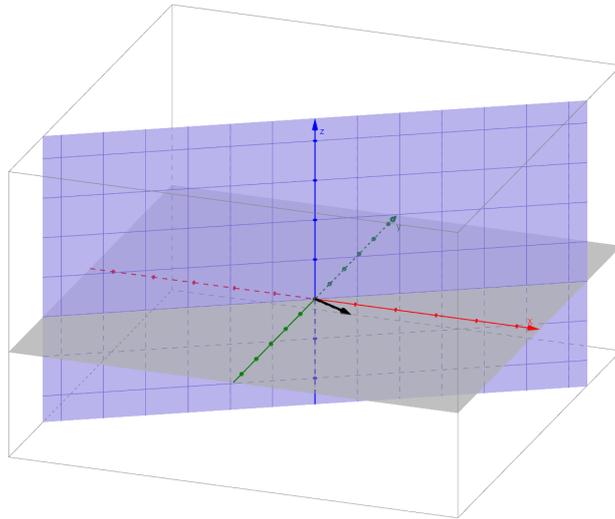


Figure 2.7: Plane with normal vector $[\tan(30^\circ), -1, 0]$

Intersection tests are carried out to analyze the variations of the intersection planes at different angles, while performing a verification and check of the implemented formula before incorporating this formulation into the final script, Figures 2.7, 2.8, 2.9 .

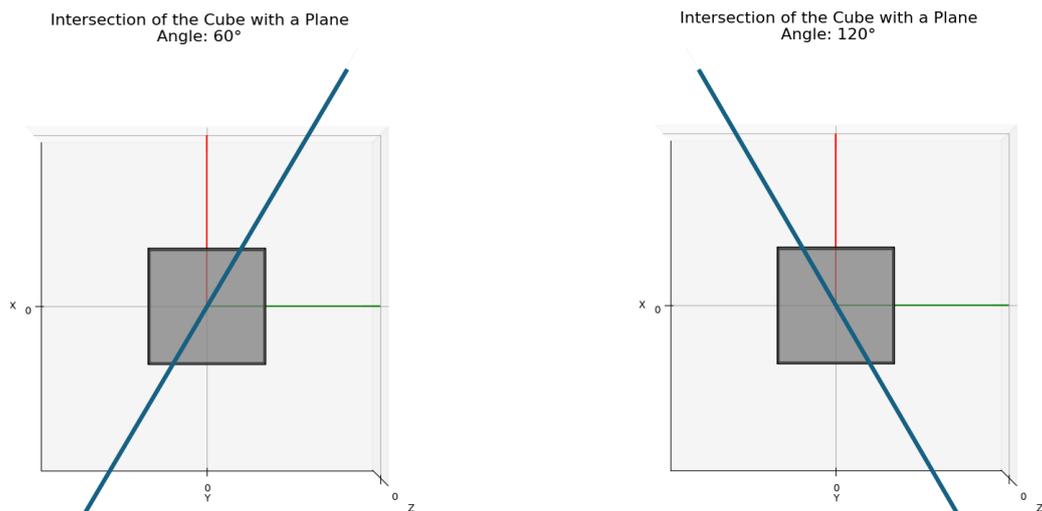


Figure 2.8: Test intersection planes 60° and 120°

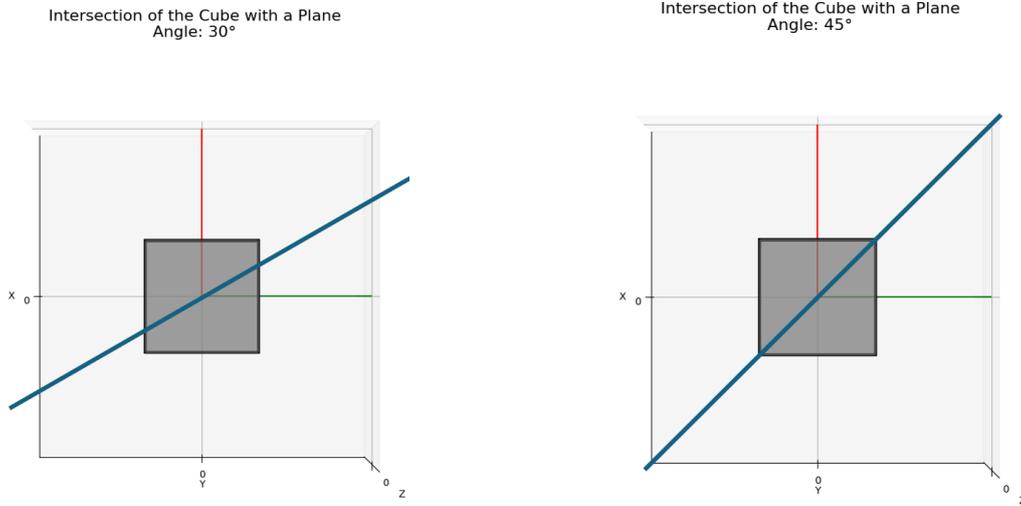


Figure 2.9: Test intersection planes 30° and 45°

Furthermore, for subsequent developments of the code, it is necessary to find the direction vector, which must be parallel to the XY plane and lie within the plane. This vector is obtained by computing the cross product (2.6) between the previously calculated normal vector of the plane 2.1 and the Z -axis 2.5. The result of this operation is a vector that is perpendicular to both the plane's normal vector and the Z -axis.

After calculating the direction vector of the reference plane at the origin of the coordinate system (i.e. at the centroid of the surface), to simulate the various tapes, new intersection planes are created parallel to the initial one. Respecting an offset equal to the width of each strip, each plane represents the center line of the tapes in their midpoints. Some output examples are shown in Figures 2.10.

$$\vec{n}_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (2.5)$$

$$\vec{v}_d = \vec{n} \times \vec{n}_1 = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \tan(\theta) & -1 & 0 \\ 0 & 0 & -1 \end{vmatrix} \quad (2.6)$$

To develop this concept in Python, in the absence of information regarding the spatial domain of the mesh more actions are carried out (Flowchart 2.11).

1. **Determine mesh bounds:** Identify the minimum and maximum bounds of the mesh along the axis most aligned with the normal of the reference plane.

2. **Interval Calculation:** Calculate the number of intervals (spaces) between the planes by dividing the distance between the bounds by the width of the strip. This determines how many times the plane can be shifted along the chosen axis from the minimum to the maximum bound of the mesh. An additional plane is included for the one located at the minimum bound. It is important to have an integer number of planes. Finally, a list of representing the positions of the offset planes is created.
3. **Compute Plane Offsets and Symmetry:** Begin by calculating the standard mode, followed by the offset distance in the direction of the reference plane normal. To ensure that the planes are generated symmetrically, the range is initially extended in the negative direction, and subsequently in the positive direction. This approach facilitates the creation of the lay-up from the left side towards the right side. For the Out and Center strategies, the positions of the points are reordered. In the case of the Out strategy, the list is reversed. For the Center strategy, the starting index, corresponding to the midpoint of the list length, is first determined. Then, the list is iteratively divided and alternated between points located in the upper and lower halves of the list.

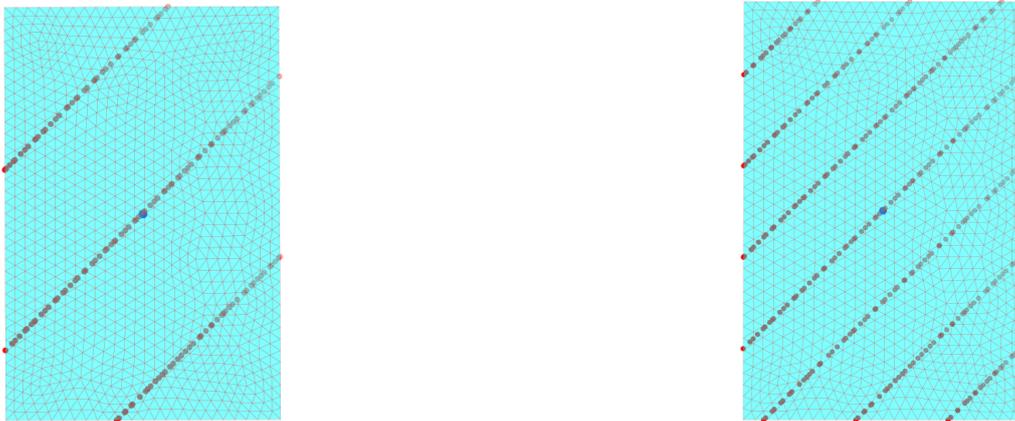


Figure 2.10: Comparison of Rectangular panels with different tape widths and 45° angle (200 mm - left and 100 mm - right)

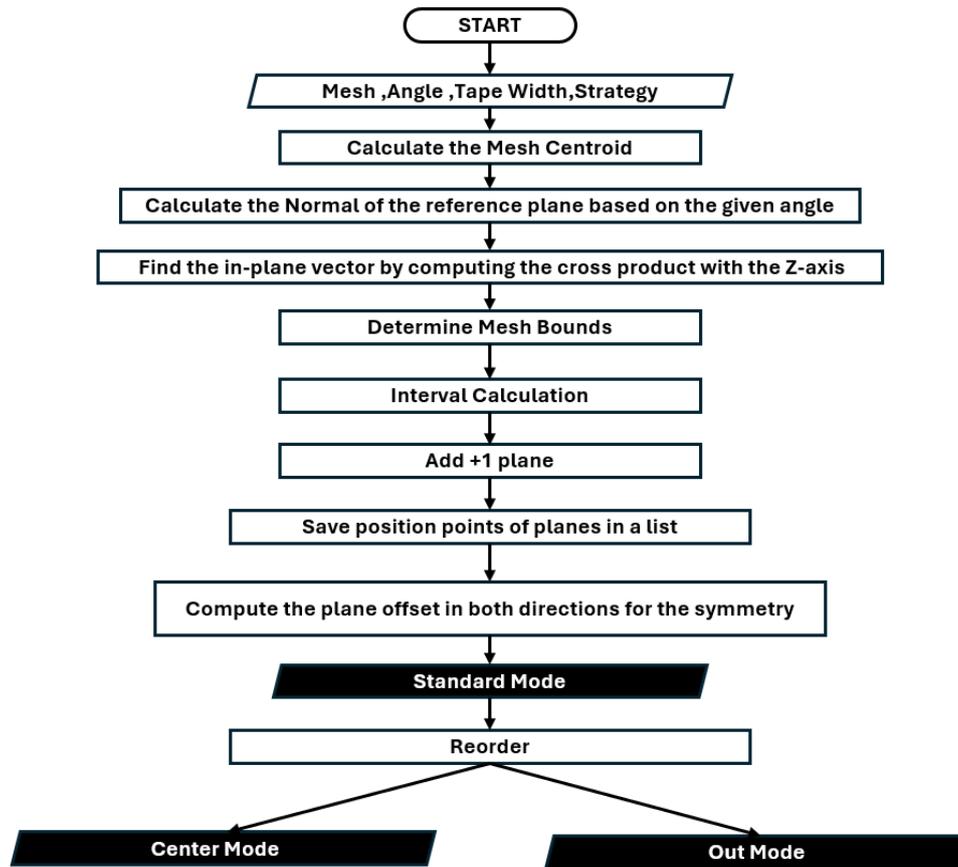


Figure 2.11: Flowchart: Definition of the Intersection Planes

2.2.2.2 Intersection points

According to the logic of this approach, for a complete specification of the paths, it is necessary to identify not only the orientation but also the endpoints. These endpoints correspond to the intersections between the mesh and the planes at the surface boundary. Through this process, information about the edges of the triangular elements at the boundary is obtained.

By examining the sorted edges of the mesh, all edges are retrieved through the list of vertex pairs that define each edge. Then, the unique or boundary edges are identified by setting their count to 1. These edges are not shared between multiple triangles, meaning they appear only once. This operation is performed by a Python function, which takes the mesh as input. In Figure 2.12 the flowchart is shown. Finally, the boundary edges are stored in the boundary edges array.

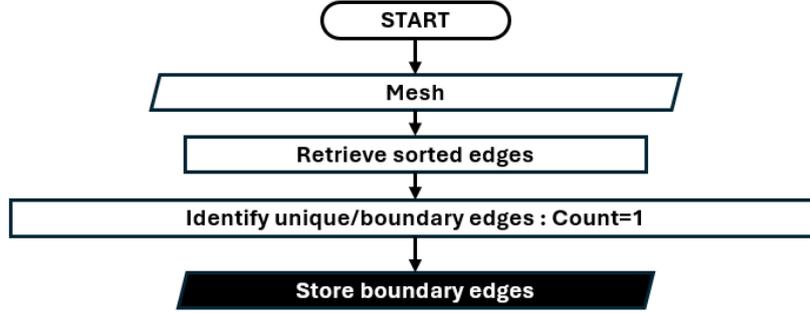


Figure 2.12: Flowchart: Boundary Edges Function

The next level is represented by an iterative process that takes a pair of points at the boundary of the mesh, calculates the segment between their coordinates in three-dimensional space along with its direction vector, and determines whether or not there is an intersection with the currently selected plane, using the plane point previously calculated.

This function is based on the parametrization of the segment, which is useful for determining the positions of points lying on the segment itself, depending on the value of the parameter t , known as the scaling factor.

Given the two endpoints of the segment, $p_1(x_1, y_1, z_1)$ and $p_2(x_2, y_2, z_2)$, the segment can be expressed as follows:

$$\gamma : p_1 \longrightarrow p_2(t) = \begin{cases} x(t) = x_1 + t \cdot (x_2 - x_1) \\ y(t) = y_1 + t \cdot (y_2 - y_1) \\ z(t) = z_1 + t \cdot (z_2 - z_1) \end{cases} \quad (2.7)$$

where $t \in [0, 1]$.

Specifically, if $t = 0$, the corresponding point is p_1 , while if $t = 1$, the point corresponds to p_2 .

For intermediate values of t , the position of points along the segment is determined by the linear relationship between p_1 and p_2 .

If the scaling factor falls outside this range, no intersection occurs.

Let \vec{p}_1 and \vec{p}_2 represent the points in space, and \vec{dir} their segment direction vector.

$$\vec{dir} = \vec{p}_2 - \vec{p}_1 \quad (2.8)$$

Then, it is possible to rewrite equation 2.7 :

$$p_{\text{segment}}(t) = \vec{p}_1 + t \cdot \vec{dir} \quad (2.9)$$

The figure 2.13 illustrates how the position of the point is defined relative to the total length of the segment.

It follows that it is necessary to calculate the factor in order to confirm the existence of the intersection and to determine its position relative to the segment. The following equation is used, which defines the segment between the two selected points p_{plane} and p_{segment} , and studies its alignment with respect to the currently selected plane, identified by the normal vector \vec{n} . p_{plane} is a generic point belonging to the plane, in this case replaced by the point of the plane previously calculated, and p_{segment} is a point belonging to the segment of the triangle mesh under consideration.

This is a dot product, so in order to satisfy the perpendicularity condition, it must be zero.

$$\vec{n} \cdot (p_{\text{segment}} - p_{\text{plane}}) = 0 \quad (2.10)$$

$$\vec{n} \cdot (p_1 + t \cdot \vec{dir} - p_{\text{plane}}) = 0 \quad (2.11)$$

From this, it can be derived that the scaling factor is equal to :

$$t = \frac{\vec{n} \cdot (p_1 - p_{\text{plane}})}{\vec{n} \cdot \vec{dir}} \quad (2.12)$$

Therefore, if the condition of perpendicularity and alignment between the plane and the segment is satisfied and the scaling factor falls within the defined interval, the position of the intersection point between the plane and the boundary of the mesh can be precisely localized.

Attention is drawn to the calculation of the denominator of the fraction of the parameter t . In intersection search iterations, the case where this value is equal to zero or relatively close to zero is excluded. Flowchart 2.14 shows the correct iterations.

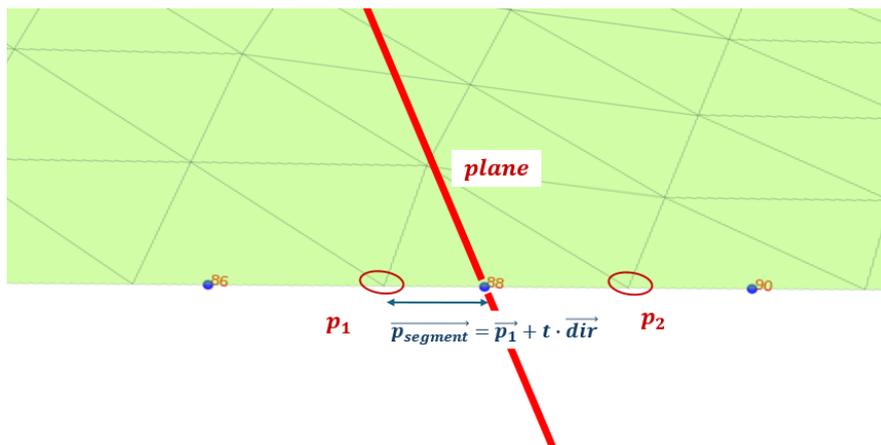


Figure 2.13: Localization intersection point

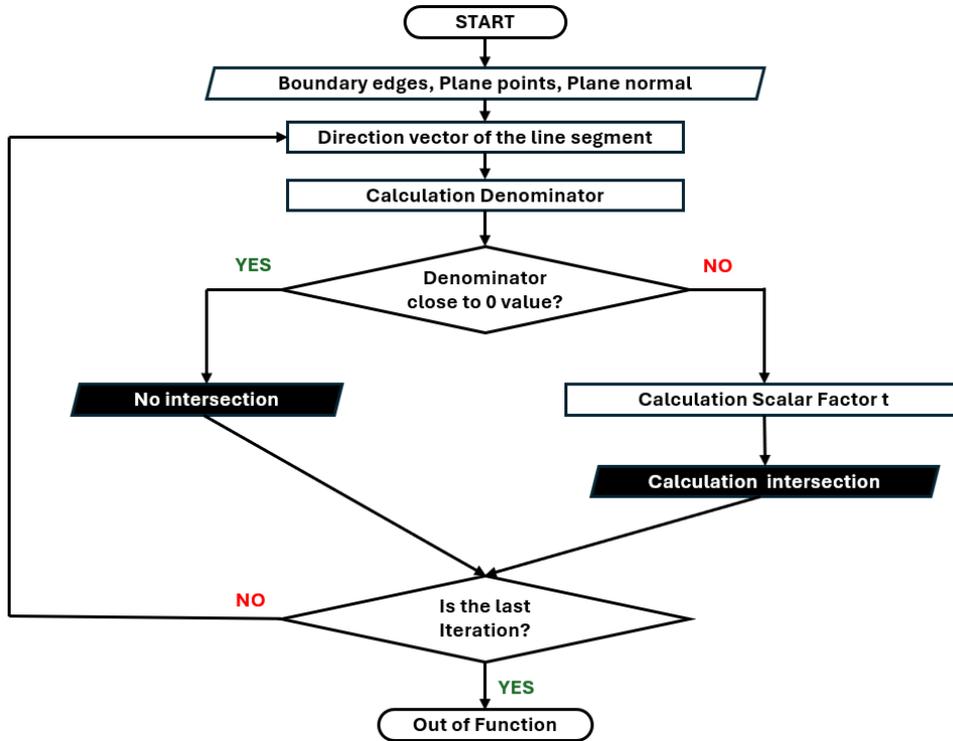
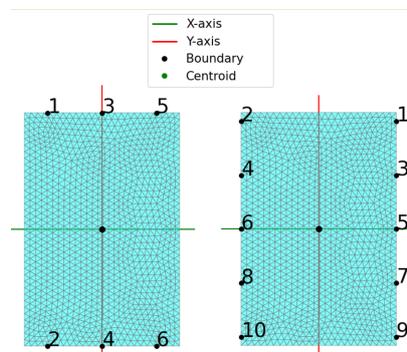


Figure 2.14: Flowchart: Boundary Intersection

The primary output of the code is the localization of the intersection points. To differentiate the various strategies (S, C, O), the numbering of the intersections can be observed. The Figures in 2.15 present a solution for a rectangular flat panel with the Standard lay-up strategy at 90° and 0° angles. Additional examples are provided for different lay-up strategies, such as the center-and-out approach (2.16) and with other angles ranging between 0° and 180° (2.17).

Figure 2.15: Comparison of S strategy at 90° - left and 0° - right

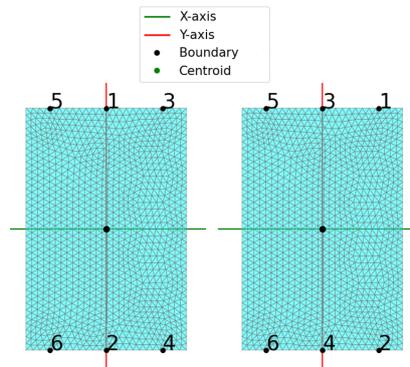


Figure 2.16: Comparison of C strategy at 90° - left and O strategy at 90° - right

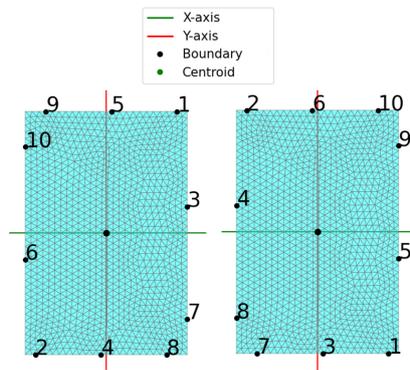


Figure 2.17: Comparison of C strategy at 60° - left and 120° - right

Being an iterative process based solely on the boundary points of the mesh, this method does not require any information regarding the surface definition formulas or the geometric properties of the analyzed shape. It is sufficient to know the maximum and minimum limits in order to define the number of planes to be used for the different tapes, thus covering the entire tool. Therefore, this approach is applicable even to rounded shapes, which do not necessarily have to be circles, ellipses, or other ruled shapes, but can be arbitrary in nature.

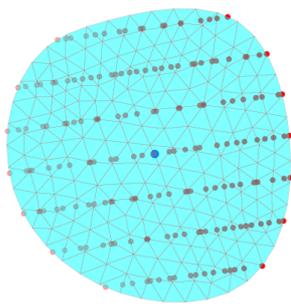
This represents another fundamental element in the choice of this method, as it allows the implementation of the simulation even for geometries not precisely defined by parametric coordinates X , Y , and Z , Figure 2.18.

However, it should be noted that there are limitations regarding the handling of certain shapes. In fact, very high curvatures or excessive concavities are difficult to manage, making the process less effective for geometries with such

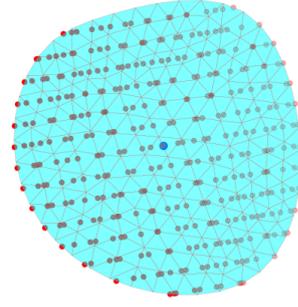
features.

Nevertheless, for applications in the AFP process, where the tools typically maintain defined shapes and are not extremely complex or unusual, this approach is still suitable.

The examples shown below illustrate the cases discussed Figures 2.19 and 2.20.



(a) Round panel with tape width of 200 mm and 45°



(b) Round panel with tape width of 100 mm and 45°

Figure 2.18: Comparison of Round panels with different tape widths and 45° angle (200 mm - left and 100 mm - right)

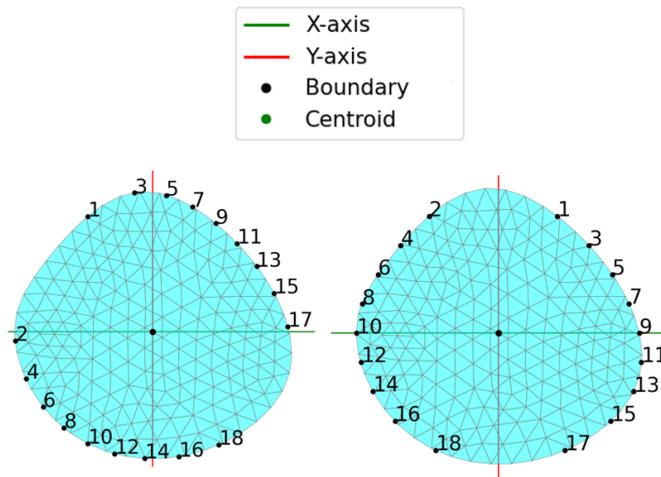


Figure 2.19: Comparison S strategy at 60° - left and 0° - right with tape width 150mm

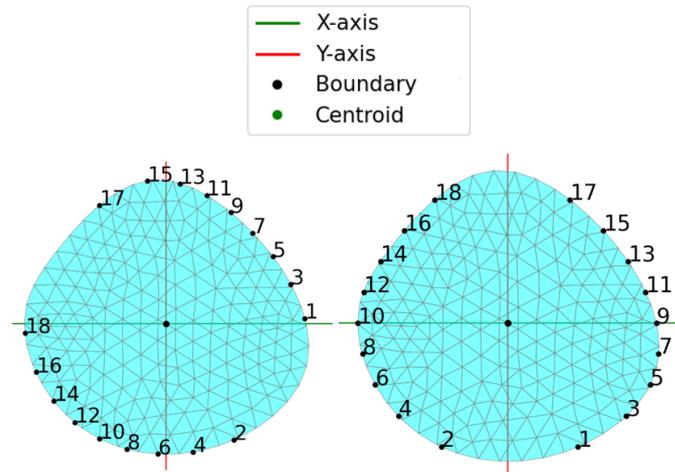


Figure 2.20: Comparison O strategy at 60° - left and 0° - right with tape width 150mm

2.2.2.3 Extension of the lengths of the paths

Once the coordinates of the intersection points have been obtained and the extrema of the trajectories have been determined, one might be led to believe that the path modelling process is complete. In fact, when considering flat-panel structures, once the starting and ending points of the trajectory are defined, it may seem sufficient to connect these two points with a straight line, without considering curvature or other complex geometric features, Figure 2.21. Nevertheless, it is necessary to add extensions to the endpoints, in order to achieve satisfactory results in the modelling of the associated tapes surfaces.

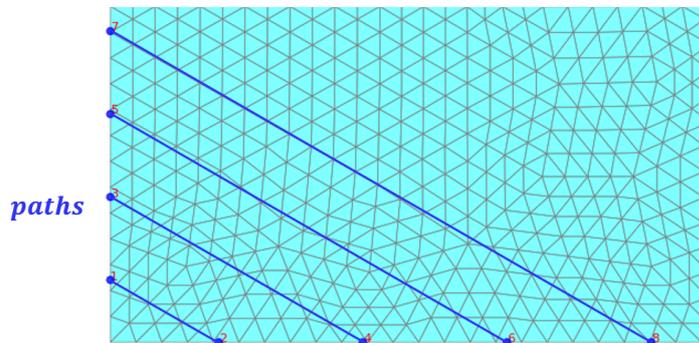


Figure 2.21: Paths on the mesh

This step is crucial to ensure that the resulting surfaces are consistent and properly connected to the surrounding models, thus ensuring adequate adaptation to the constraints imposed by the design context. As previously explained, the coordinates of the trajectory extrema are determined by the intersections between the planes positioned at the midpoints of the tapes, which are perpendicular to the mesh plane and oriented according to the lay-up angle of the layers. To model the tape, understood as a rectangular surface that covers the tool's outer layer, it is sufficient to draw two lines. These lines must have a length equal to the tape width and should be centred at the trajectory endpoints, perpendicular to the intersection plane, and aligned with the direction of the trajectory itself. By connecting the various lines, a rectangular surface is obtained, similar to the tape that the AFP machine deposits onto the component layers, Figure 2.22. A problem emerges with respect to the insufficient coverage of the surface layer of the tool, as demonstrated in Figure 2.23.

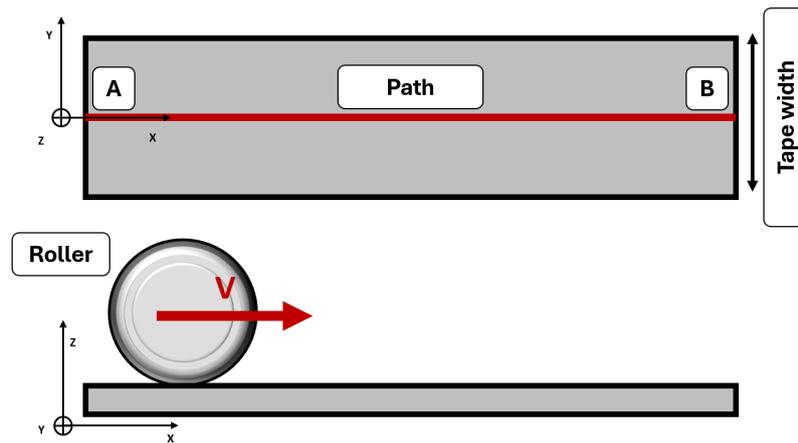


Figure 2.22: Schematic representation of the tape model

When the extrema of the rectangles modeled as tapes coincide with the intersections at the edges of certain elements, some areas remain "uncovered" with some gaps along the boundaries.

This partial coverage creates an incomplete surface representation that may lead to inaccuracies in the modelling process, particularly when adapting to the constraints of the design or manufacturing context.

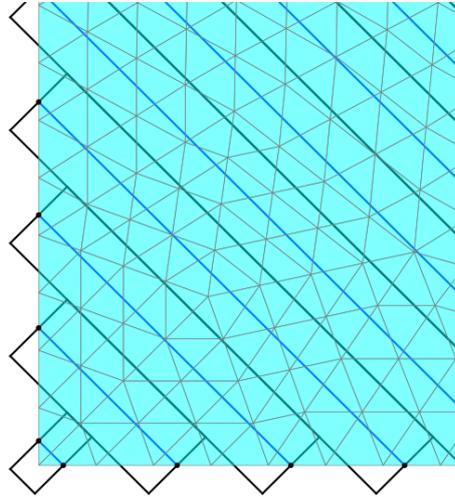


Figure 2.23: Tapes with endpoints coincident with the boundary intersections

The function that implements the extension addition is also based on a mathematical formulation. The initial idea to include all the gaps along the contour is based on translating the edge of the rectangular box that simulates the tape, such that it covers the entire area within the deposition space of the tape, on both sides. Through geometric observations, it can be deduced that the added area corresponds to a right triangle, in both directions.

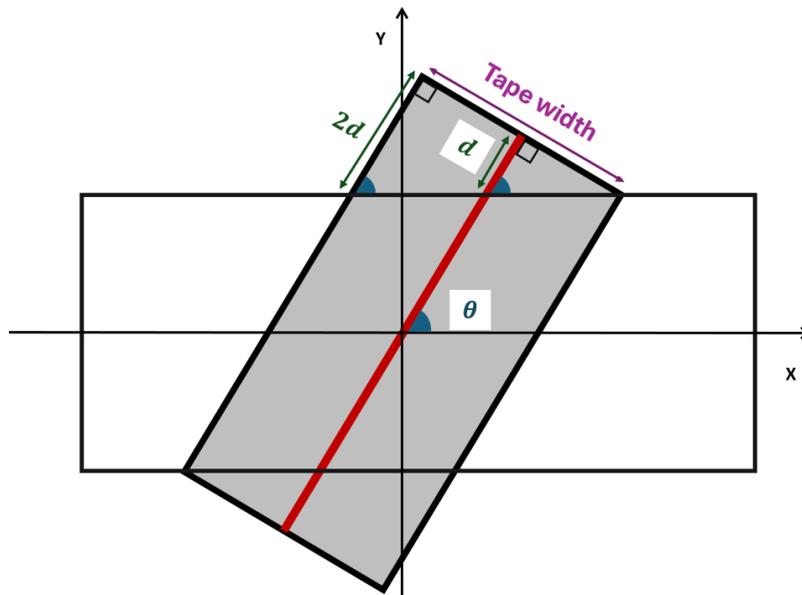


Figure 2.24: Trigonometric Formulation for extensions added to the path

Therefore, by applying the criteria of triangle similarity, the trigonometric principles of the right triangle, and the properties of the angles, the minimum distance to be added can be calculated using the following formula:

$$d = \frac{\text{tape_width}}{2} \cdot \tan \theta \quad (2.13)$$

The new position of the points in space is determined using the previously stated principle through the parameterization of the segment 2.9.

Similarly, the intersection point with the boundary is taken as a reference, and the coordinates of the extension point along the plane direction are then identified using d as a scaling factor.

The plane direction coincides with the normalized normal to the plane, ensuring a unit magnitude.

Since the geometry is positioned in a three-dimensional space with a reference frame centered at the centroid, the point coordinates can take both positive and negative values relative to the axes and it is necessary to determine the extension points at both ends of the various trajectories.

For each intersection, two extension points are computed as follows:

$$\vec{p}_{\text{extension1}} = \vec{p}_{\text{boundary}} + d \cdot \vec{\text{dir}}_{\text{plane}} \quad (2.14)$$

This point is obtained by translating the intersection point along the path direction by a distance d . The addition (+) operation shifts the point along the direction vector, effectively extending it in that direction.

$$\vec{p}_{\text{extension2}} = \vec{p}_{\text{boundary}} - d \cdot \vec{\text{dir}}_{\text{plane}} \quad (2.15)$$

In the same way, this second point is obtained by shifting the point in the opposite direction by the same distance, effectively extending it accordingly. This approach ensures that the extension is applied symmetrically along the trajectory, and therefore, for each intersection, there are two new shifted positions, Figure 2.25. For each intersection plane, four new points along the direction are created, consequently to obtain results consistent with the real process, only two new coordinates are necessary, which are the ones located at the extreme outer positions of the mesh.

When the wrong points are selected, such as those located inside the mesh, the results worsen, leading to poorer element selection and leaving even more gaps. So, by selecting the groups of points for each plane, the maximum Euclidean distance between them is calculated. With the values being referenced to the same coordinate system, this way, the most extreme points are definitely selected.

If $p_{\text{extension1}}(x_1, y_1, z_1)$ and $p_{\text{extension2}}(x_2, y_2, z_2)$:

$$\text{distance_max} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (2.16)$$

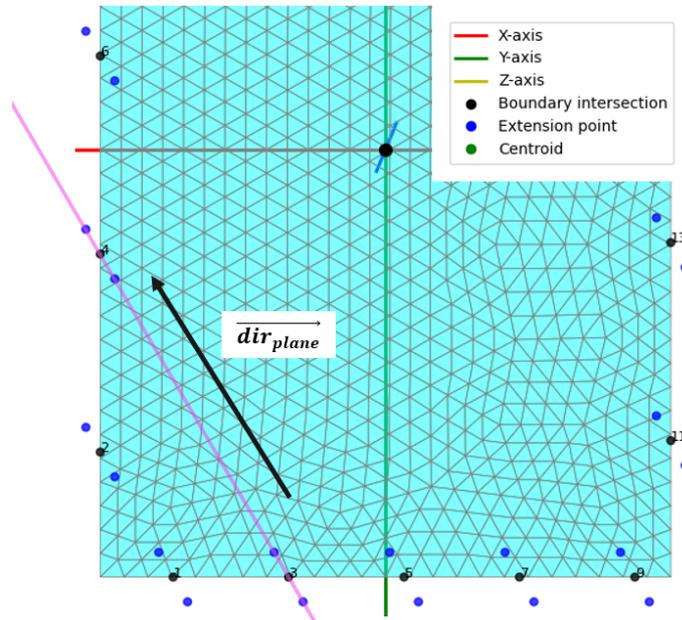


Figure 2.25: Extension points details

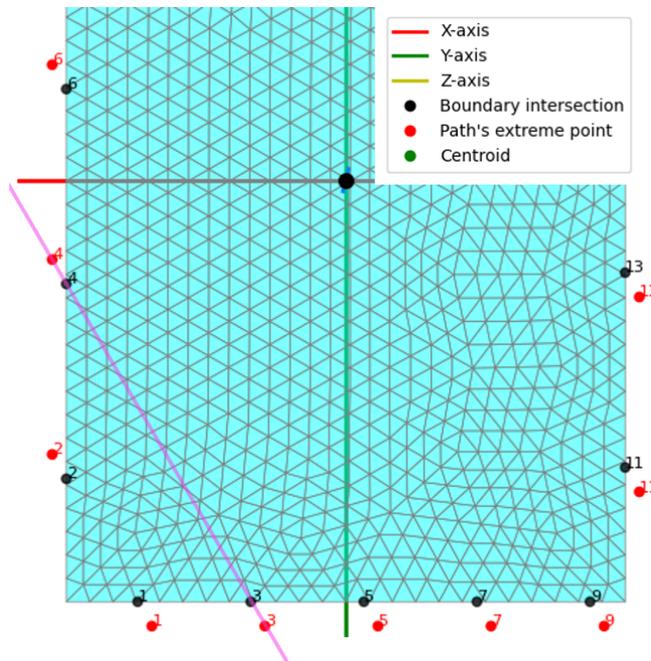


Figure 2.26: Extreme point of the path details

Refer to Flowchart 2.32 for a summary of the iterations.

By comparing Figures 2.23 and 2.27 with the addition of the extensions, it is evident that the calculation is accurate in managing the empty spaces during the tape placement.

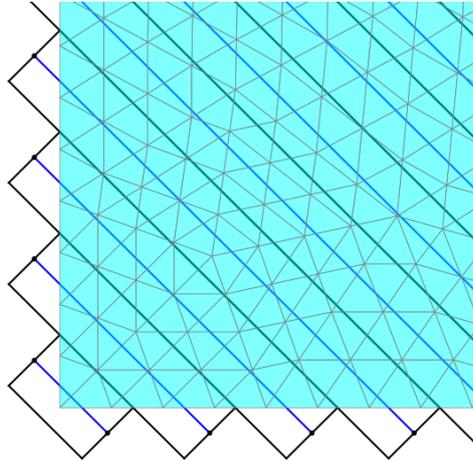


Figure 2.27: Tapes with endpoints coincident with the boundary intersections plus extensions

In the case of the round shape, it can be observed that the applied mathematical principle ensures complete filling of the layer. The only difference lies in the irregularity of the added areas, caused by the variation in the curvature of the shape.

Furthermore, in the presence of drastic changes in shape or when certain parts of the surface boundary align with the angle of the path direction, Figure 2.29, it may occur that complete gap filling is not achieved.

However, considering the tapes as strips with a width in the range of tens of millimeters, it is generally observed that the coverage is satisfactorily achieved. Increasing the width of the tapes or choosing more complex shapes may therefore require the addition of extra distance, by appropriately modifying the calculation formula.

In any case, given these two limitations and the ultimate goal of this work, a thermal simulation, another way to solve this issue could be the choose of larger element sizes in the simulation model. This ensures a more accurate and precise selection, allowing elements that are not entirely within the tape to still be included through selection.

In Figure 2.28, for example, it can be observed that the compromise between the width parameter and the surface boundary is well balanced, allowing full coverage.

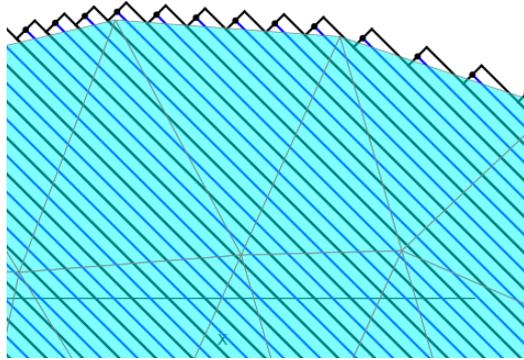


Figure 2.28: Extended tapes on a round shape

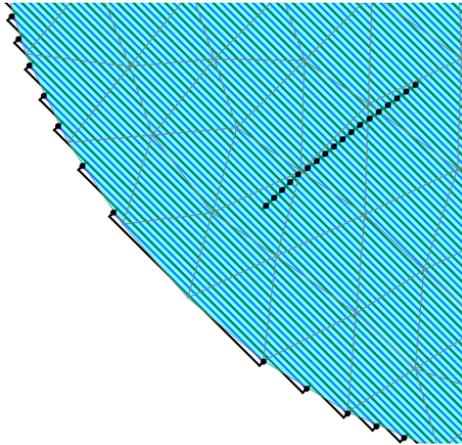


Figure 2.29: Round shape with tapes width of 12 mm

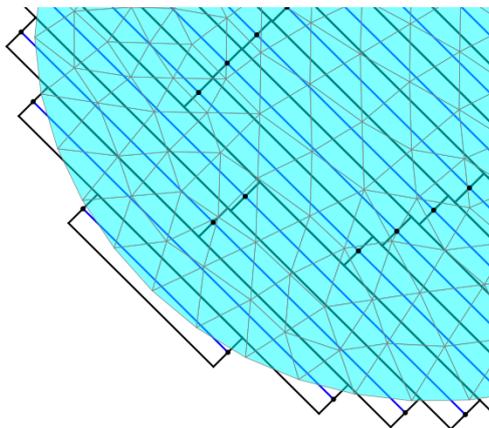


Figure 2.30: Round shape with tapes width of 75 mm

From the comparison between Figures 2.29 and 2.30, it is confirmed that the tape width parameter is crucial. In fact, by using the same mesh, with the same geometry, surface, and orientation angle, but with one case having a tape width of 12 mm and the other 75 mm, it can be observed how the layer filling deteriorates, especially in critical areas.

In the code, two separate functions are defined: one for shifting the points and one for calculating the Euclidean distance.

For the shifting of points, the formula for the new coordinates is based on the angle θ , but it is important to avoid the angle ranges where the cotangent values become problematic. In these ranges, $\tan(\theta)$ becomes either very high or infinite, making the extension undefined.

Thus, for angles in the range $(0^\circ \leq \theta \leq 5^\circ)$ or $(85^\circ \leq \theta \leq 90^\circ)$, the extension cannot be defined properly, and 45° is considered ad reference angle, Figure 2.31.

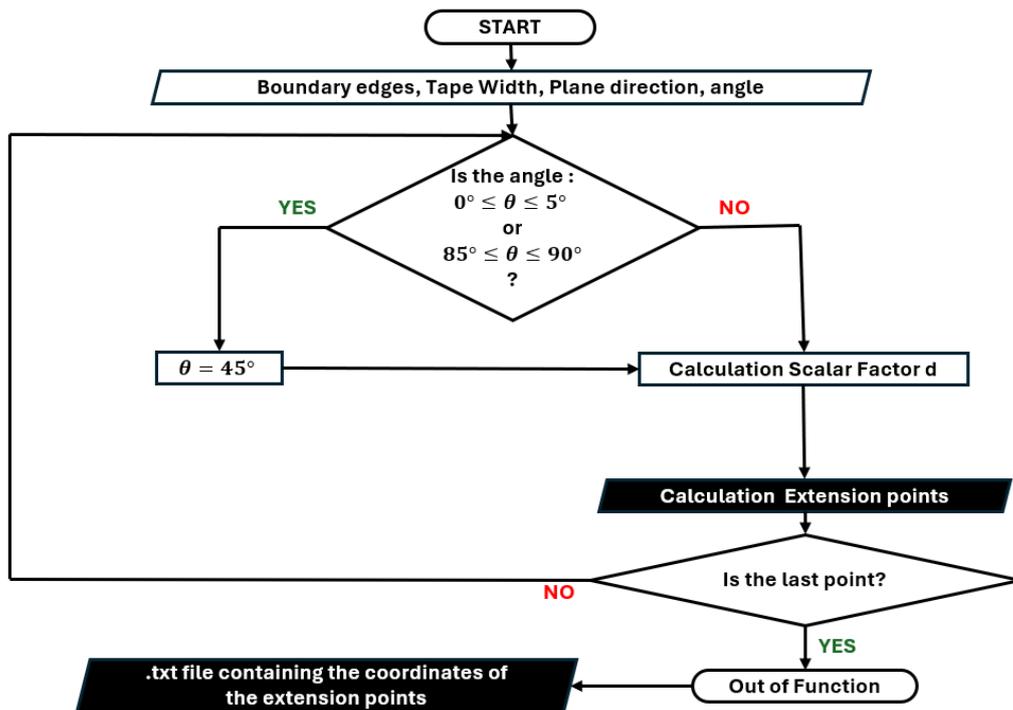


Figure 2.31: Flowchart: Extension Points Calculation

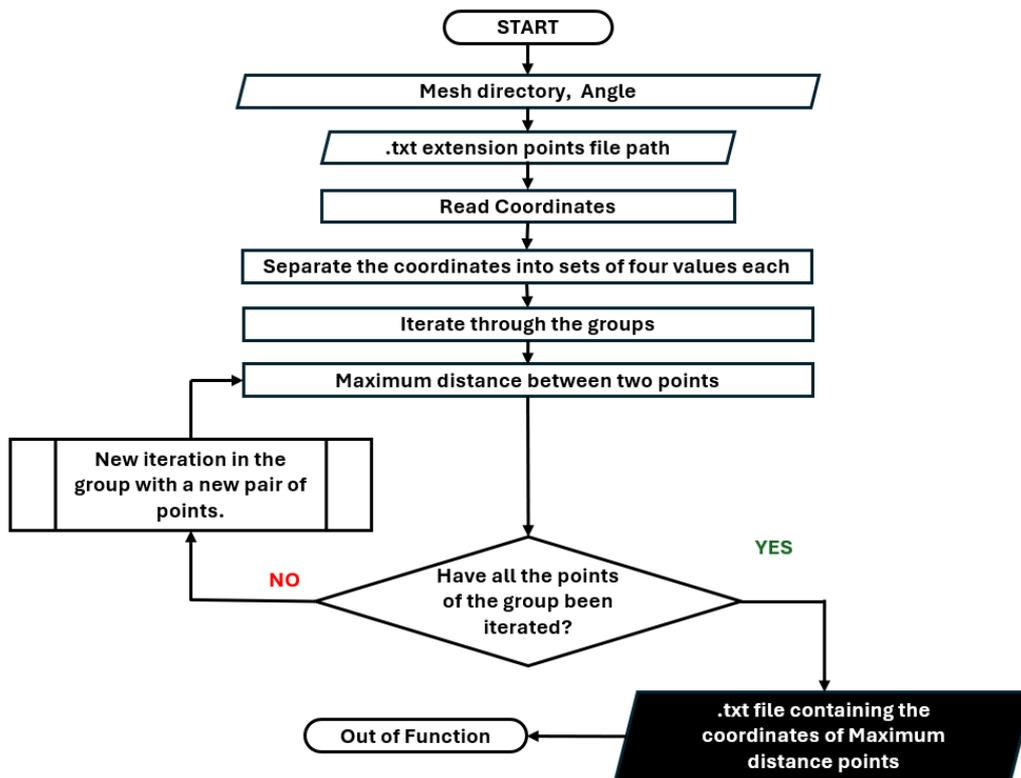


Figure 2.32: Flowchart: Max Distance Couple Points

2.2.3 Data Output

In addition to the aforementioned functions, there are three auxiliary functions whose primary purpose is data output. These functions are categorized into three main types: coordinate saving, file reading, and visualization.

The first two functions are also useful within the script execution process, as they ensure redundancy and clarity in the extracted information from plane intersections, while also providing greater control and debugging capabilities. Each time points are extracted, they are saved in a TXT file at every computed step.

As explained previously, these steps include:

1. **Boundary Intersection:** Computation and saving of boundary intersection points.
2. **Extension Points:** Generation and saving of extension points derived from boundary points.
3. **Max Distance Couple Points:** Computation of the point pair with the maximum Euclidean distance.

Steps 1 and 2 follow an iterative process: once the boundary points are obtained, new variables are created to store the extension points identified using the appropriate function.

Regarding Step 3, which represents the final and most significant output of the script, based on the Euclidean distance calculation, it is necessary to open the file containing the extension points, extract the relevant information, and proceed with the computation.

For each category of coordinates stored, a specific file name is assigned, each of which is updated with the angle of the process.

For example, with an angle of 60° :

```
Output 1: boundary_intersection_points_angle_60.txt
Output 2: extension_points_angle_60.txt
Output 3: max_distance_points_angle_60.txt
```

The saving format, instead, divides the points by each intersection plane, providing a visual distinction. The general structure is as follows:

- Each pair of points is represented by two lines in the file.
- The first line contains the coordinates X, Y, Z of the first point.
- The second line contains the coordinates X, Y, Z of the second point.

- An empty line separates each pair of points.

Example:

For two intersection planes, the saved file could appear as:

```
x1, y1, z1
x2, y2, z2

x3, y3, z3
x4, y4, z4
```

The plot function provides a clear, interactive 3D graphical representation of the mesh and the analysed points related to different stages of analysis (boundary intersections, extensions, and maximum distance point pairs).

For each desired angle:

1. **Initializes the visualization:** Creates a 3D figure and sets axis limits.
2. **Draws the mesh:** Adds the mesh with triangles in cyan and the centroid in black.
3. **Reads the data:** Extracts points from files and visualizes them:
 - Boundary points;
 - Extension points;
 - Maximum distance points.
4. **Numbering and colouring:** Assigns a number to each point and cycles through colours.
5. **Adds the legend:** Identifies the point types with the legend.
6. **Displays the visualization:** Uses `plt.show()` to display the result.

2.2.4 Core function

In conclusion, each function, including auxiliary ones, is invoked within the core function of the script, which serves as a structured workflow guiding the entire process.

Below the resume and the code lines:

1. Iteration over Angles

- For each angle, it defines the output file paths.
- Computes the centroid, plane normal, and offset plane points via `process_mesh()`.
- Identifies the mesh boundary edges using `find_boundary_edges()`.

2. Output 1: Plane-Boundary Intersection Computation

- For each offset plane, it determines intersection points with the mesh edges using `intersect_segment_plane()`.
- Sorts the intersection points along the in-plane direction.
- Saves the results in a TXT file.

3. Output 2: Extension Point Calculation

- Generates offset points for each intersection using `add_length()`.
- Stores the computed point pairs.

4. Output 3: Maximum Distance Computation

- Calls `calculate_max_distance()` to determine the pairs of points with the greatest Euclidean separation.

5. Final Visualization

- Uses `plot()` to graphically represent the mesh along with computed intersection and extension points.

2.2.5 Flowchart : Full First Section

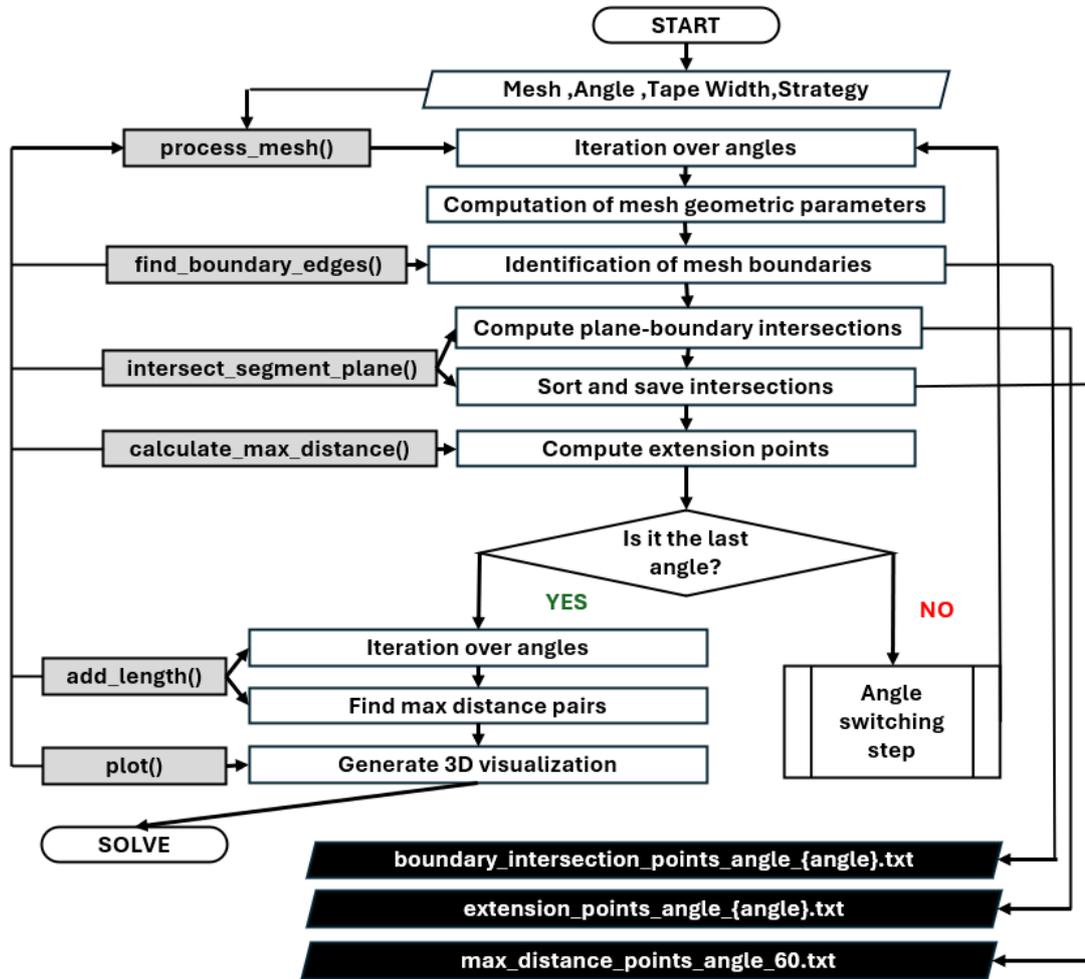


Figure 2.33: Flowchart: First Section mesh_intersection.py

2.3 Second Section: `boxes_steps_mode.py`

The `boxes_steps_mode.py` script serves three main purposes:

- To enable the visualization of tapes modeled as boxes on the surface of the layers as a control mechanism before the next step, taking as input the coordinate values of the path endpoints produced as the final output of the first section of the code.
- To select the placement modalities for the entire composite structure.
- To export all necessary data for the subsequent simulation process in ANSYS by generating output files in CSV format .

2.3.1 Thermal Strategy Modes

In addition to the different deposition order modes of the tapes on the layer such as standard, out, and center other modes exist related to the thermal strategy adopted in the AFP process to complete all layers. These strategies are schematically represented in Figure 2.34.

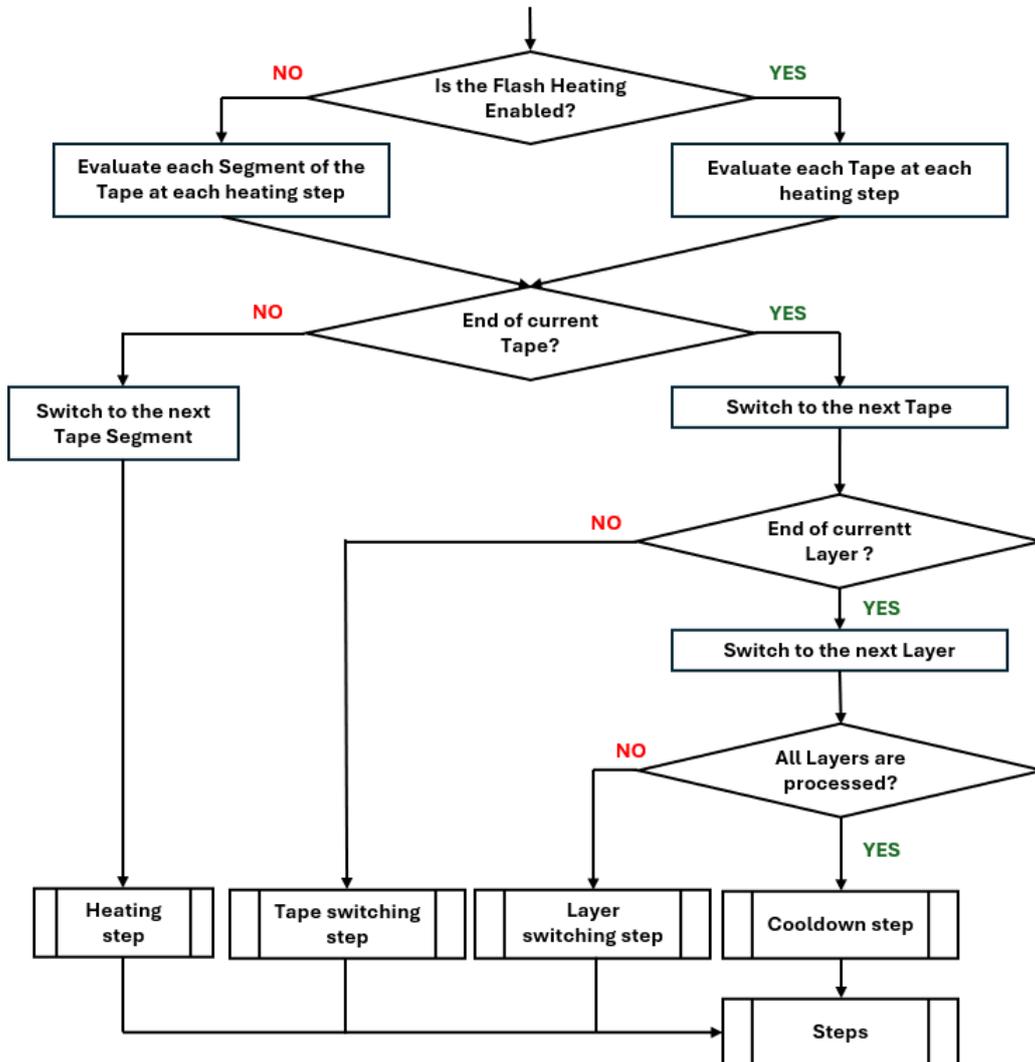


Figure 2.34: Thermal Strategy Modes Scheme

To ensure consistent results, the simulation process must accurately replicate AFP production conditions. This requires implementing functions capable of reproducing reality as precisely as possible.

To this end, all boundary conditions and convection criteria must be defined, layer by layer, down to the element and node level.

As previously mentioned, the process is based on the sequential deposition of tapes, which are heated to the required temperature by a laser positioned on the machine head. Therefore, in the step-by-step simulation, it is necessary to select not only the elements based on their coordinates but also the activation times, the corresponding boundary conditions, and the associated convection criteria [22].

Geometric information is calculated by the first section of the code, while the necessary temporal data is determined in this section. Consequently, it is essential to differentiate between the different types of steps.

The first criteria concerns **Flash Heating**:

- **Flash heating disabled:** The tape is modeled as a sequence of segments, whose length depends on the laser's action area and its speed.

$$Heating_time = \frac{Laser_Thickness}{Process_Speed} \quad (2.17)$$

In this case, the laser heats each segment for a predefined time before moving to the next one, continuing until the end of the tape.

- **Flash heating enabled:** The entire tape is activated in a single step, and heating is applied uniformly across the entire structure in one phase.

The use of flash heating for modelling the segments does not provide an accurate real-world representation of the process. Instead, it serves as a means to expedite the process in order to reduce computational costs and processing time.

Once the tape is fully activated, a time step is incorporated to allow the robotic head to reposition itself at the new end of the tape.

Upon completion of each layer, a designated waiting period is applied for the optical inspection of the layer, which is estimated based on the average time required by an operator.

Finally, after the entire process has been completed and the full thickness of the composite has been achieved, a cooldown period is implemented.

Under the examined conditions, as presented in Table in the Appendix A6, the various process time parameters are:

Heating Time [s]	0.014
Robot Time [s]	10
Layer Time [s]	100
Cooldown Time [s]	10000

Table 2.1: Different Time Steps in Seconds

The mode described above represents the complete mode; however, additional modes have been implemented that modify the waiting times between different process stages.

These modes adjust the waiting times between tape placement, layer formation, and the final cooldown phase.

This provides greater flexibility in the thermal strategy, allowing the waiting periods to be tailored to the specific requirements of each stage.

The modes differ in the following aspects:

- **Waiting T:**
 - If flash heating is disabled, after each segment, the heating time is observed. Then, after each tape, the robot time is applied to allow the machine head to reposition itself. Finally, after the last tape, the layer time is applied, followed by the cooldown period.
 - If flash heating is enabled, for each tape, only the heating time is applied, followed by the robot time, then the layer time, and finally the cooldown period.
- **Waiting L:** No waiting time is allocated for the robot's movement between tapes and the optical time inspection is applied.
- **Waiting C:** Both the waiting time for the robot's movement between tapes and the waiting time between layers are eliminated.
- **Waiting N:** Only the heating time is applied. It is noted that this mode has no practical applications; however, it is still considered for verification and potential alternative applications.

Each type of waiting step has an ID, which is a code assigned to differentiate at both the variable and process levels, as does flash heating.

Heating Time	1111
Robot Time	2222
Layer Time	3333
Cooldown Time	4444
Flash Heating YES	0.1
Flash Heating NO	0

Table 2.2: Different Time Steps in Seconds

An example of a two-layer laminate is provided, with the first layer oriented at 0° relative to the x-axis and the second at 90° . Additionally, very wide tapes are modelled, with exceptionally long segments that are not feasible in practice; however, this is done solely for explanatory purposes in this example. In this scenario, the first layer consists of 5 tapes, each comprising 2 segments, while the second layer consists of 3 tapes, each comprising 3 segments, as shown in Figure 2.36. Figure 2.37 illustrates the sequential time steps for this specific example under different modes, with flash heating either enabled or disabled. Figure 2.35 presents a three-dimensional visualization of the movement of the laser head on the tool surface, illustrating the tape deposition process according to the generated model. The movement follows a defined sequence, starting from position A, where the deposition of the first tape begins. The process then continues to position C, marking the completion of the first segment. From there, the head moves to position D, where the deposition of the second tape starts. This sequence is repeated iteratively until the entire layer is fully deposited (last tape). The entire process can be tracked through the numbering of positions and trajectories.

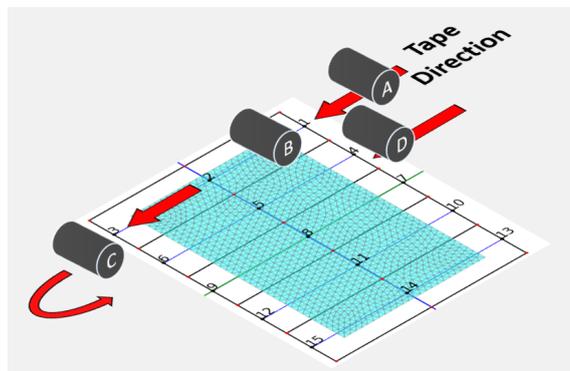


Figure 2.35: Modelling the Movement of the Laser Head

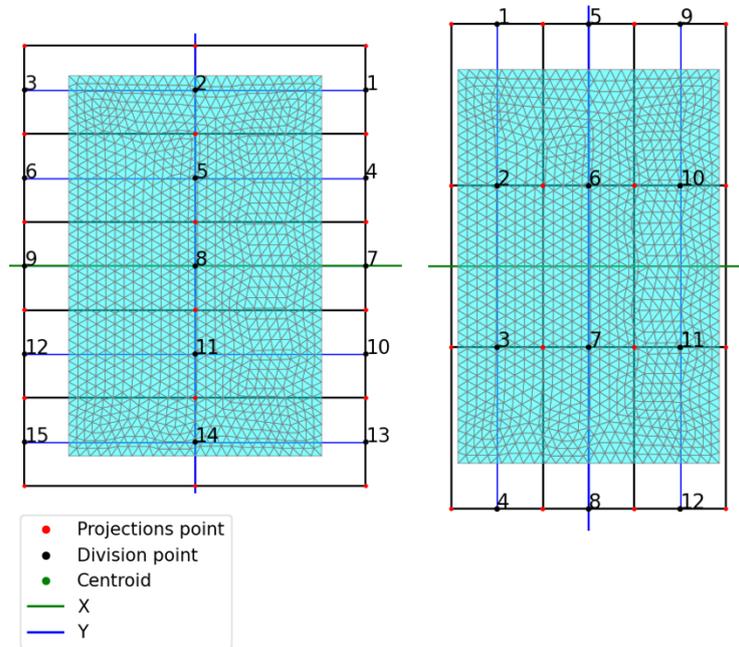


Figure 2.36: Laminate Example Model

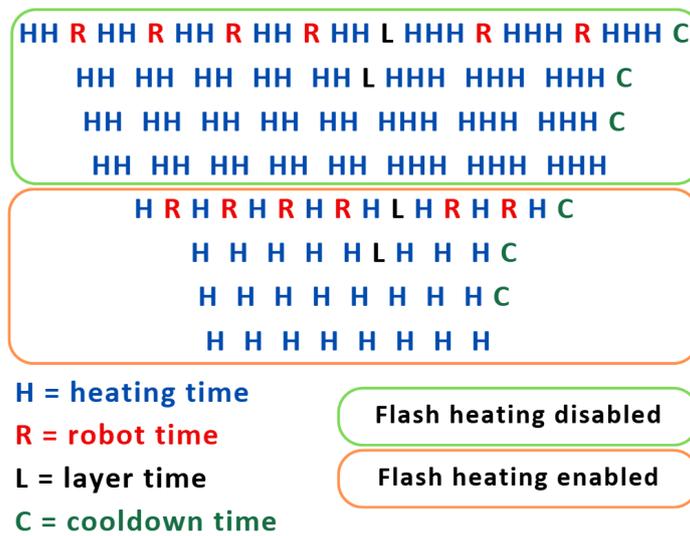


Figure 2.37: Sequence of Time Steps for the Example

2.3.2 Input Data

This section of the Python script defines the parameters required for the thermal simulation process in the ANSYS environment. Alongside the variables needed for step differentiation and the geometric implementation of the tapes, it is also essential to specify those related to the process itself, the complete laminate, and the boundary conditions to be applied later. Specifically, these parameters relate to the properties of the laminate, the tape and process parameters, and the mesh of the surface used in the script. The starting layer is also defined, allowing the simulation to begin from a layer that is not necessarily the outermost one in the laminate. These parameters are specifically used for the material selected for the laminate, CF/LM-PAEK. In the following chapter, detailed information regarding the material will be provided.

Laminate Size [mm]	100 × 100
Layers Number	8
Laminate Thickness [mm]	1.584
Layup	$[90^\circ / -45^\circ / 0^\circ / 45^\circ]_{2s}$

Table 2.3: Laminate Properties: Square Panel

Tape Thickness [mm]	0.198
Tape Width [mm]	12.7
Segment Length [mm]	20

Table 2.4: Tapes Parameters

Heating Time [s]	0.014
Robot Time [s]	10
Layer Time [s]	100
Cooldown Time [s]	10000

Table 2.5: Different Time Steps in Seconds

Pressure Roll [Atm]	600000
Laser Thickness [mm]	0.001
Process Speed [$\frac{mm}{s}$]	72
Process Temperature [°C]	470
Tool Temperature [°C]	20
Cooldown Temperature [°C]	20
Tool Heat Transfer Coefficient [W, m^{-2}, K^{-1}]	9.363
Air Heat Transfer Coefficient [W, m^{-2}, K^{-1}]	10

Table 2.6: Process Parameters

2.3.3 Process of Functions

As with the previous section, this one outlines various supporting functions that complement the core function, which orchestrates the entire sequence of operations. The subsequent analysis delves into the key characteristics, required inputs, and expected outputs of these functions.

2.3.3.1 Division of the paths into discrete segments

The first step is the discretisation of the trajectories into distinct segments, based on the area of action of the laser, carried out by an auxiliary function. The discretisation process is relatively straightforward: the starting and ending points of the trajectories, along with the established segment length, are required, after which the "line" is divided according to the defined length. It should be noted that this approach is consistent and valid exclusively for flat-panel structures. Returning to the previous section, the starting and ending points of the trajectories coincide with the points of maximum distance, that is, those of extension added to the intersection points with the mesh boundary, determined by the maximum Euclidean distance.

For this reason, a supporting procedure is necessary for this new function, which allows for reading data from the previously generated output file, containing the required coordinates. This function reads pairs of points in sequence and associates with them the endpoints of a path of a tape, iterating over all the coordinates contained in the file .

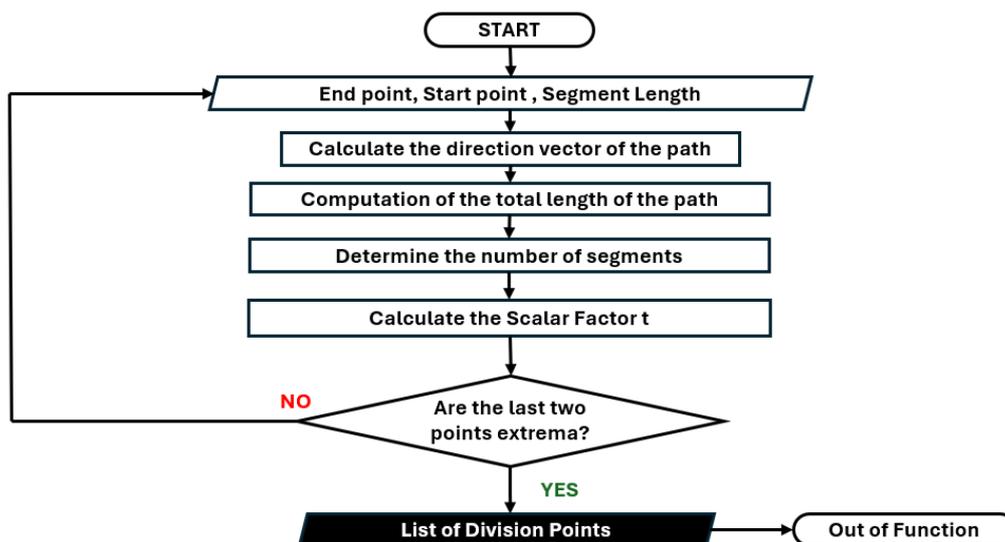


Figure 2.38: Flowchart: Division paths in segments

Another function performs the geometric and mathematical operations required for the correct discretisation of the segments, Figure 2.38. Iteratively, for each pair of points, the directional vector of the path is first calculated by subtracting the coordinates of the starting point from those of the end-point, and the Euclidean distance between the points is determined using the Euclidean norm. Knowing the segment length, the distance value is then divided into integer numbers. Additionally, a condition is set so that if the length is smaller than the specified segment length, at least one segment is created. This is particularly useful for trajectories near the edges of the surface. To define the coordinates of the division points, the parametrization rule is applied in this case as well.

Each division point is associated with a scaling factor that ranges from 0 to 1, positioned at a precise distance relative to the length of the segment, Figure 2.39.

As discussed in the subsection 2.2.2.2:

$$\vec{point} = \text{start} + t \cdot \vec{line_vector} \quad (2.18)$$

- When $t = 0$, the returned point will be the starting point (**start**).
- When $t = 1$, the returned point will be the endpoint (**end**).
- For intermediate values of t , the points will be calculated along the line.

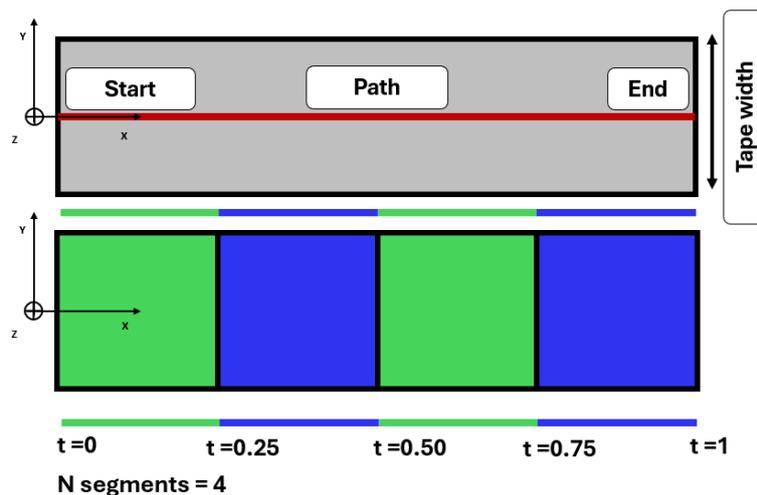


Figure 2.39: Subdivision of Tapes into Segments

Several examples outputs are presented in Figures 2.40

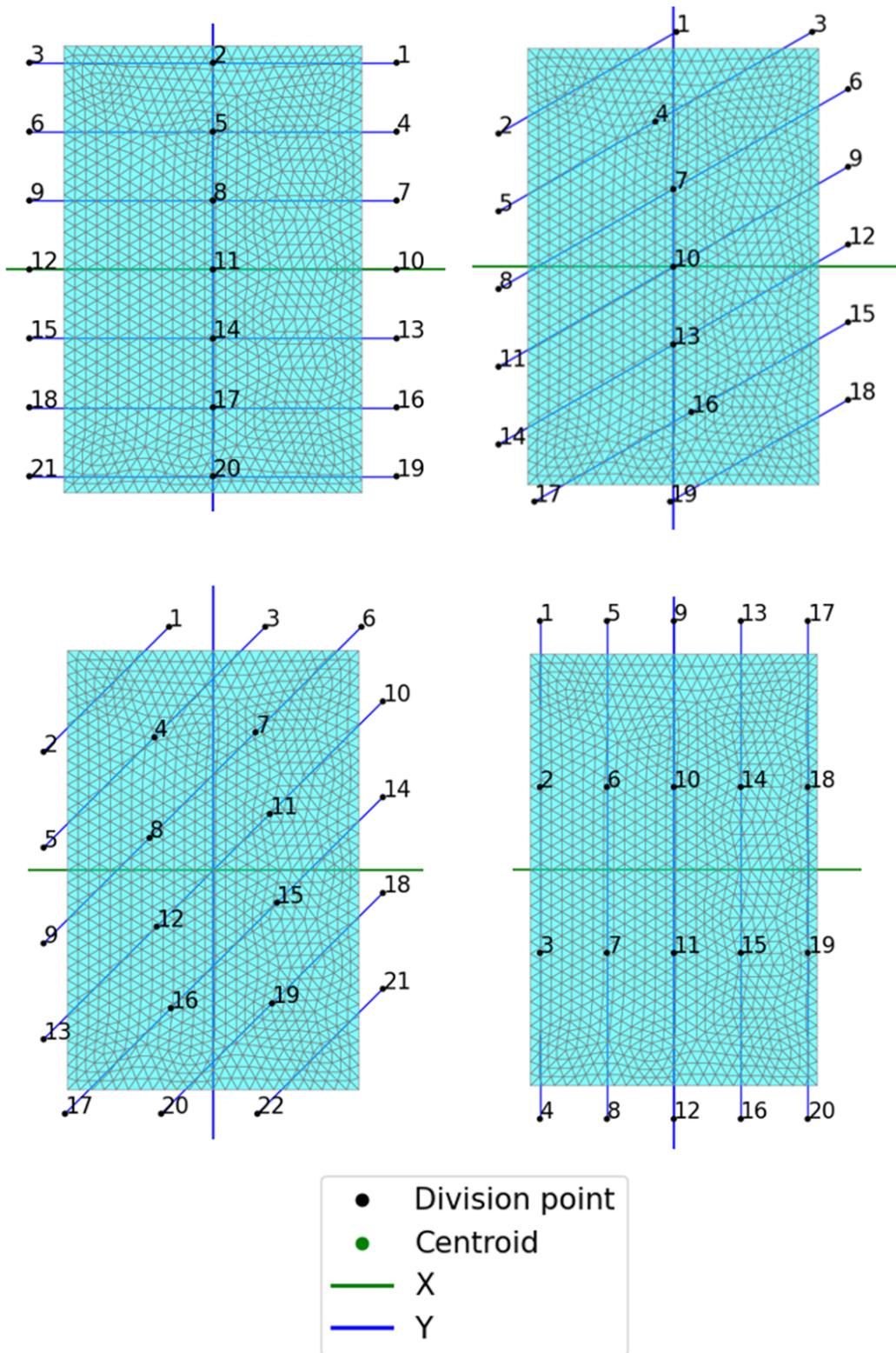


Figure 2.40: Examples of Division Points for Square Panel: 0° (top left), 30° (top right), 45° (bottom left), 90° (bottom left)

2.3.3.2 Projection Points

The next step, after obtaining the coordinates of the division points of the various segments, is the final modelling of the tape, represented as "boxes". To achieve this, these points must be projected along axes perpendicular to the path in the midline, at a distance equal to half the width of the tape on each side. A new function is implemented, which computes two points projected perpendicularly from a given point along the direction of the centre-line, Figure 2.43. Since this is a flat panel, only the XY plane is considered, disregarding the Z direction. The considerations regarding the thicknesses of the tapes are handled by other scripts written directly in ANSYS. The directional vector of the path is taken, and a perpendicular vector in the XY plane is computed by rotating it by 90° , Figure 2.41:

$$\text{perp_vector} = \vec{v} = (-y, x, 0) \quad (2.19)$$

Subsequently, the length (norm) of the perpendicular vector is calculated:

$$\|\vec{v}\| = \sqrt{(-y)^2 + x^2} \quad (2.20)$$

It is then normalised to obtain a unit vector (of length 1) in the same direction:

$$v_unit = \frac{(-y, x, 0)}{\|\vec{v}\|} \quad (2.21)$$

Finally, two points displaced along the perpendicular direction are obtained:

$$\text{projection_1} = \text{point} + \frac{\text{tape_width}}{2} \cdot v_unit \quad (2.22)$$

$$\text{projection_2} = \text{point} - \frac{\text{tape_width}}{2} \cdot v_unit \quad (2.23)$$

Some examples are shown with different setup angles in Figure 2.42.

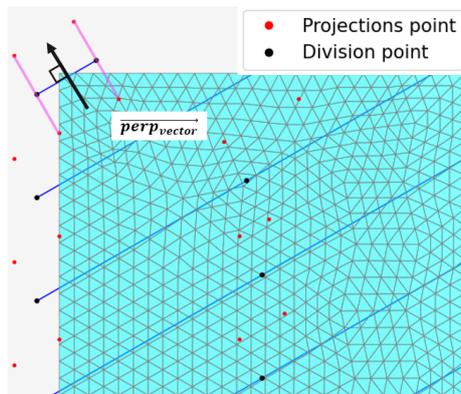


Figure 2.41: Projection mathematical formulation

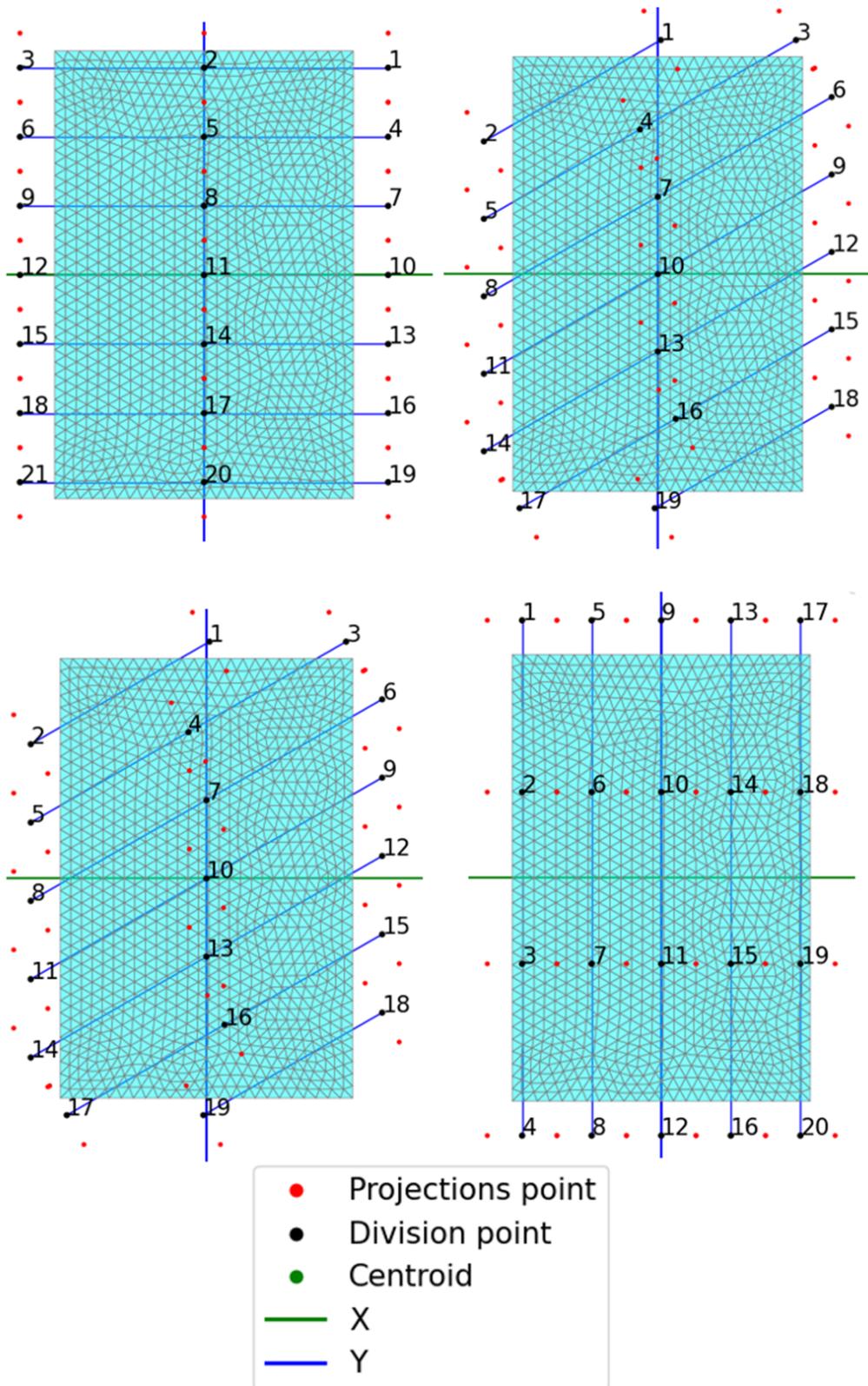


Figure 2.42: Examples of Projection Points for Square Panel: 0° (top left), 30° (top right), 45° (bottom left), 90° (bottom right)

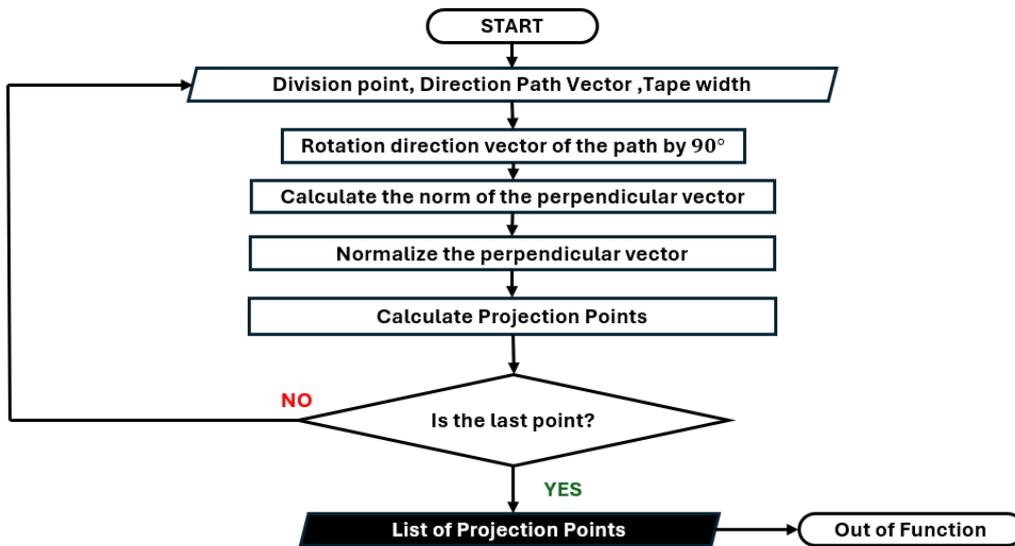


Figure 2.43: Flowchart: Calculation of the Projection Points

2.3.4 Core Process

The script does not include a function that allows for calling and execution, as is the case in the `mesh_intersection.py` file. Instead, it is structured as a well-defined process that, starting from a set of desired angles, allows for data processing, visualisation generation, and the production of structured output, all in a sequential and detailed manner.

The process begins with an iteration over a predefined list of angles, each representing a scenario requiring the processing of distance data on a mesh. For each angle, the script dynamically generates paths for output files and prepares to process the associated data. The necessary data is read from input files, while variables to store the results are initialized in order to track all the information to be processed.

After the data has been read, the script proceeds by dividing it into smaller segments. These segments are then stored in a list, ready to be further processed throughout the loop.

Thereafter, the code handles the generation of 3D plots using the **matplotlib library**. The plots not only depict the segments that were just created but also display the division points, improving the understanding of the mesh structure.

Moreover, for each segment, the projection points are calculated and visualised as boxes in the plot.

This provides a visual representation of the projections, essential for verifying the correctness of both the geometry and calculations.

Once the segments and projection points have been calculated, they are saved to files. The writing process includes the application of a scaling factor to the data to enhance visibility and comprehension when the results are later analysed or used.

In addition to the geometric aspect, the script also integrates a temporal component for thermal analysis, managing the different thermal modes for the creation of layers for each segment and phase of the process. The times required for various operations, such as system heating, robot movement along the segments, and transition between layers, are calculated. These are recorded in detail, allowing for real-time tracking of process efficiency and progress.

The management of layers and tapes is another central aspect of the code. Each layer, which may consist of multiple tapes, is carefully monitored. The starting indices of the tapes are calculated and updated during processing. Furthermore, the transition from one layer to another is handled meticulously, with the times required for movement between layers being calculated. Finally, at the end of the cycle, a final cooling phase is incorporated to sim-

ulate its completion, with particular attention given to the last phase of the process.

All this work is organised in a **DataFrame** using the **pandas library**. This allows for the systematic collection of all information related to segments, times, and other parameters. In the end, the data is exported into a `.csv` file, representing a structured and easily readable version of the results, which is useful for further analysis or reporting.

Summary of Main Actions:

1. **Divide the line into segments.** Divide the entire line into smaller, manageable segments for the further processing.
2. **Calculate the projection points for each segment.** For each segment, compute the corresponding projection points to represent their positions.
3. **Generate a 3D plot with segments, projection points, and boxes.** Create a 3D visualisation showing the segments, their projection points, and bounding boxes for clear representation.
4. **Write the division points and projection points to an output file.** Store the calculated division and projection points in a text file for later use. Here is an example with 60° :

Output: `.txt` file: `coordinates_boxes_angle_{60}.txt`

The saving nomenclature follows the same structure as the first section, [2.2.3](#).

5. **Track and calculate the time for each step in the process.** Monitor and record the time taken for each step in the process to track efficiency.
6. **Manage the transitions between layers and tapes, updating indices and times accordingly, and save all in dataframes.** Handle the movement between layers and tapes, adjusting the indices and times, and save this data for later analysis.
7. **Save the output data in CSV files.** Export the final data into `.csv` format for further processing or review.

2.3.4.1 Step Times

The thermal strategies and waiting types used in this process are defined as follows:

- **Flash Heating Enabled:**
 - Yes: Y
 - No: N
- **Waiting Types:**
 - T - Tape Waiting
 - L - Layer Waiting
 - C - Cooling Waiting
 - N - No Waiting

There are 8 possible modes, each representing a combination of the flash heating strategy and waiting type:

1. **Mode 1:** Flash Heating N, Waiting Type T
2. **Mode 2:** Flash Heating N, Waiting Type L
3. **Mode 3:** Flash Heating N, Waiting Type C
4. **Mode 4:** Flash Heating N, Waiting Type N
5. **Mode 5:** Flash Heating Y, Waiting Type T
6. **Mode 6:** Flash Heating Y, Waiting Type L
7. **Mode 7:** Flash Heating Y, Waiting Type C
8. **Mode 8:** Flash Heating Y, Waiting Type N

Furthermore, each step is assigned a specific code, as outlined in the table [2.2](#). In the case where the flash heating strategy is not used, the heating time must be added for each individual segment, and the segments must be updated for the different tapes, as well as the total number. On the other hand a generic configuration number is used, specifically the code 0.1.

To correctly set up the replication of the real T-AFP process, it is also necessary to properly configure the thermal transient analysis, which determines the temperatures and the needed thermal quantities that vary over time.

Therefore, appropriate setting times must be defined as input for the ANSYS environment.

As a result, the formulation of the times defined in Table 2.1 should be modified in order to optimise the simulation process.

Since the analysis involves time steps of variable duration in seconds, it is inevitable to apply the AUTOTS mode, which specifies whether to use automatic time stepping. The necessary subsequent action is to define the time settings for the DELTIM command. This function specifies the time step sizes to be used for the current load step.

The configuration in APDL Commands Scripting consists of the following parameters, as described in *Ansys Mechanical APDL Command Reference* (pp. 207-208, 517-518) [11]:

```
AUTOTS,ON
DELTIM, DTIME, DTMIN, DTMAX,Carry
```

DELTIM times load definitions are taken from the ANSYS APDL manual:

- **DTIME:** This represents the time step size for the current step. When automatic time stepping is enabled this defines the starting time sub-step.
- **DTMIN:** This defines the minimum allowable time step.
- **DTMAX:** This specifies the maximum allowable time step.

Time intervals defined as Min, Medium, and Max time are introduced for each type step.

DTIME	heating_time/2
DMIN	heating_time/100
DMAX 1111	heating_time
DMAX 2222	robot_time/2
DMAX 3333	layer_time/2
DMAX 4444	cooldown_time/2

Table 2.7: Different Time Steps in Seconds

In Figure 2.44, a schematic flowchart explains the step-by-step changes for each type of mode.

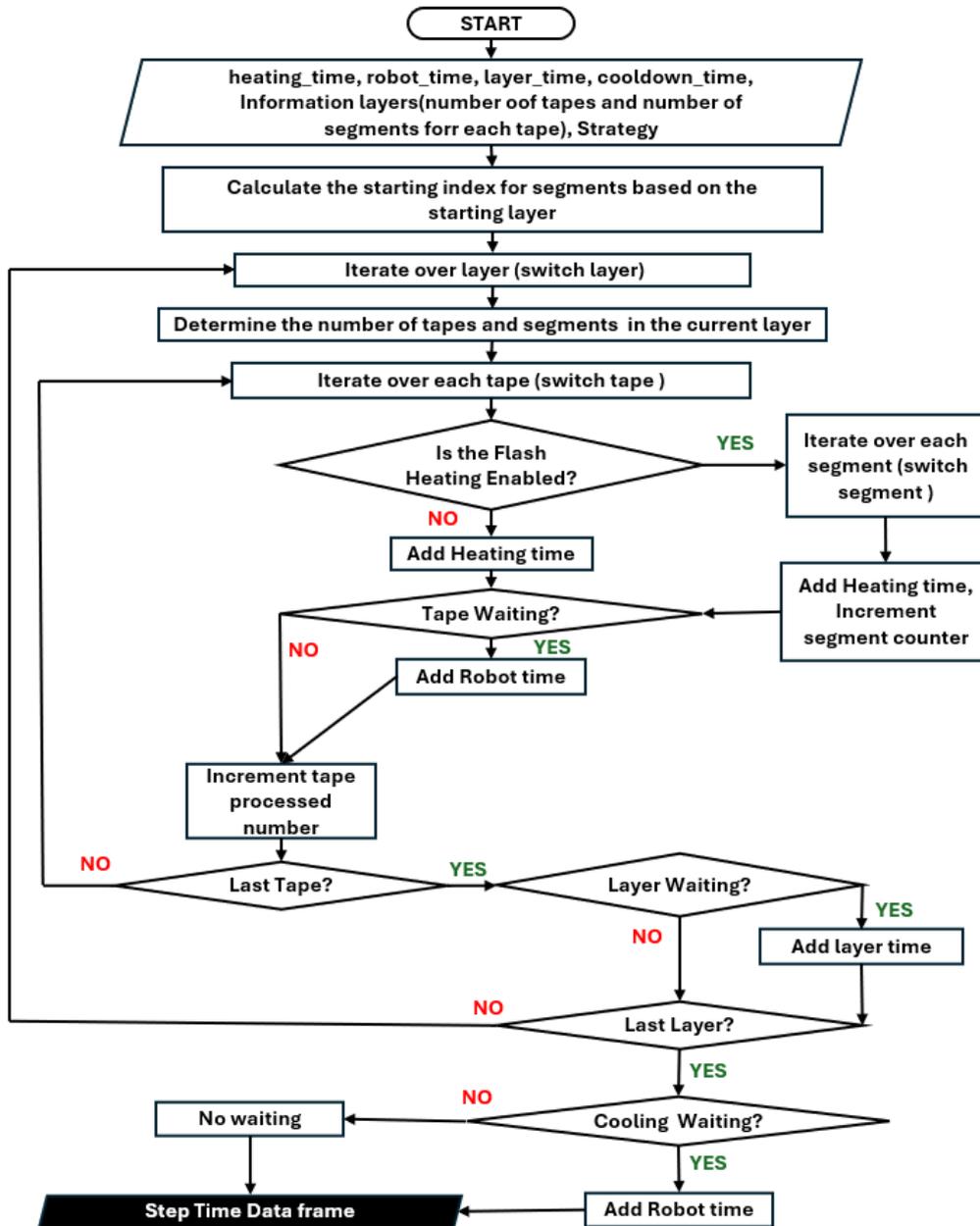


Figure 2.44: Flowchart: Step Time Modes

Building upon the example shown in Figure 2.36, the following outputs are presented for modes 1 and 5.

Time Added	Time	Time Min	Time Med	Time Max	Step Code	Layer	Segment
0.01388889	0.01389	0.00014	0.0069444	0.013889	1111	1	1
0.01388889	0.02778	0.00014	0.0069444	0.013889	1111	1	2
10	10.0278	0.00014	0.0069444	5	2222	1	0
0.01388889	10.0417	0.00014	0.0069444	0.013889	1111	1	3
0.01388889	10.0556	0.00014	0.0069444	0.013889	1111	1	4
10	20.0556	0.00014	0.0069444	5	2222	1	0
0.01388889	20.0694	0.00014	0.0069444	0.013889	1111	1	5
0.01388889	20.0833	0.00014	0.0069444	0.013889	1111	1	6
10	30.0833	0.00014	0.0069444	5	2222	1	0
0.01388889	30.0972	0.00014	0.0069444	0.013889	1111	1	7
0.01388889	30.1111	0.00014	0.0069444	0.013889	1111	1	8
10	40.1111	0.00014	0.0069444	5	2222	1	0
0.01388889	40.125	0.00014	0.0069444	0.013889	1111	1	9
0.01388889	40.1389	0.00014	0.0069444	0.013889	1111	1	10
100	140.139	0.00014	0.0069444	50	3333	1	0
0.01388889	140.153	0.00014	0.0069444	0.013889	1111	2	11
0.01388889	140.167	0.00014	0.0069444	0.013889	1111	2	12
10	150.167	0.00014	0.0069444	5	2222	2	0
0.01388889	150.181	0.00014	0.0069444	0.013889	1111	2	13
0.01388889	150.194	0.00014	0.0069444	0.013889	1111	2	14
10	160.194	0.00014	0.0069444	5	2222	2	0
0.01388889	160.208	0.00014	0.0069444	0.013889	1111	2	15
0.01388889	160.222	0.00014	0.0069444	0.013889	1111	2	16
10000	10160.2	0.00014	0.0069444	5000	4444	2	0

Figure 2.45: Example Mode 1 : Flash Heating N, Waiting Type T

Time Added	Time	Time Min	Time Med	Time Max	Step Code	Layer	Segment
0.01388889	0.013889	0.000139	0.006944	0.013889	1111	1	0.1
10	10.01389	0.000139	0.006944	5	2222	1	0.1
0.01388889	10.02778	0.000139	0.006944	0.013889	1111	1	0.1
10	20.02778	0.000139	0.006944	5	2222	1	0.1
0.01388889	20.04167	0.000139	0.006944	0.013889	1111	1	0.1
10	30.04167	0.000139	0.006944	5	2222	1	0.1
0.01388889	30.05556	0.000139	0.006944	0.013889	1111	1	0.1
10	40.05556	0.000139	0.006944	5	2222	1	0.1
0.01388889	40.06944	0.000139	0.006944	0.006944	1111	1	0.1
100	140.0694	0.000139	0.006944	50	3333	1	0.1
0.01388889	140.0833	0.000139	0.006944	0.013889	1111	2	0.1
10	150.0833	0.000139	0.006944	5	2222	2	0.1
0.01388889	150.0972	0.000139	0.006944	0.013889	1111	2	0.1
10	160.0972	0.000139	0.006944	5	2222	2	0.1
0.01388889	160.1111	0.000139	0.006944	0.006944	1111	2	0.1
10000	10160.11	0.000139	0.006944	5000	4444	2	0.1

Figure 2.46: Example Mode 5 : Flash Heating Y, Waiting Type T

2.3.5 Output Data

Regarding the output files, there are five, classified as follows. These files divide and collect the information, parameters, and process outputs for the simulation in ANSYS.

These will be the transient data files, which are provided as input.

```

step_time.csv
info_process.csv
info_layer.csv
info_tape.csv
info_center_line.csv
info_selection_boxes.csv

```

Several output files related to the example in figure 2.36 are presented to illustrate the information contained in each file, except for those related to point coordinates, which consist of three columns for the X, Y, and Z coordinates.

Cooling ^C	Laser ^C	Tool ^C	Robot Speed	Roll Pressure	Air heat coefficient	Tool heat coefficient	Air ^C
22	470	22	0.072	600000	10	9.363	22

Figure 2.47: info_process.csv

Layer	Tapes	Angle	Total Width	Total Length	Total Thickness	Laser Thickness	Tool Thickness
1	5	0	0.648	0.428	0.01	0.001	1
2	3	90	0.648	0.428	0.01	0.001	1

Figure 2.48: info_layer.csv

Tape	Segments	Tape width	Tape thickness	Segment length
1	2	0.15	0.000198	0.25
2	2	0.15	0.000198	0.25
3	2	0.15	0.000198	0.25
4	2	0.15	0.000198	0.25
5	2	0.15	0.000198	0.25
1	2	0.15	0.000198	0.25
2	2	0.15	0.000198	0.25
3	2	0.15	0.000198	0.25

Figure 2.49: info_tape.csv

Another type of output is generated on the terminal, where the primary information obtained from the process leading to the modelling of the tapes

is printed and also information about the executed file writings. Two examples are shown below with the corresponding plot visualisations: one for the considered square flat panel and one for a round panel with 2 layers, Figures 2.50 and 2.51.

For the square flat panel:

```
Flash heating disabled
Pause between each tape
Total number of files: 2
Tapes each layer: [3, 5]
Segments per tape: [2, 2, 2, 2, 2, 2, 2, 2]
excel_time_step_saved
process_excel
excel_layer
excel_tape
excel_center_line
excel_selection_boxes
```

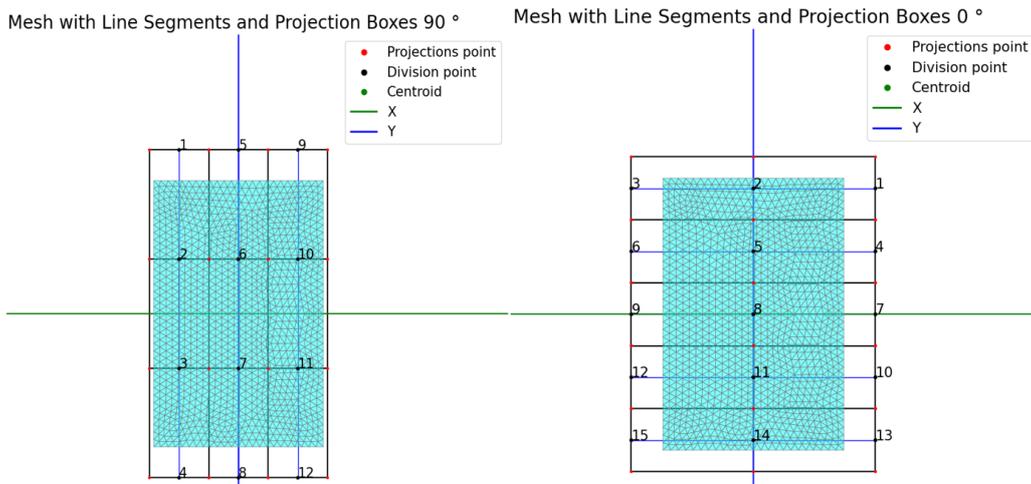
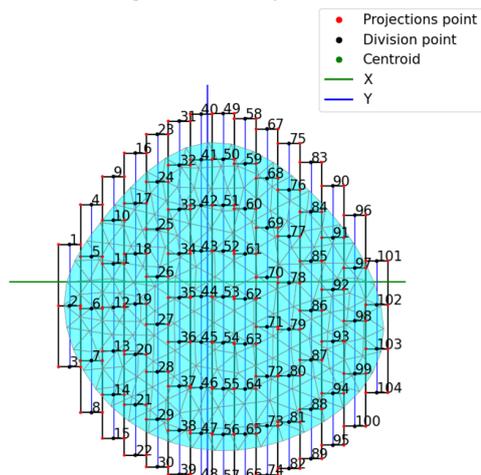


Figure 2.50: Example: Plot Output Square Panel

For the round flat panel:

```
Flash heating enabled
Pause between each tape
Total number of files: 2
Tapes each layer: [14, 15]
Segments per tape: [3, 4, 5, 6, 7, 7, 8, 8, 8, 8, 8, 7, 6, 4,
2, 4, 6, 6, 7, 8, 8, 8, 8, 7, 7, 6, 5, 4, 3]
excel_time_step_saved
process_excel
excel_layer
excel_tape
excel_center_line
excel_selection_boxes
```

Mesh with Line Segments and Projection Boxes 90 °



Mesh with Line Segments and Projection Boxes 0 °

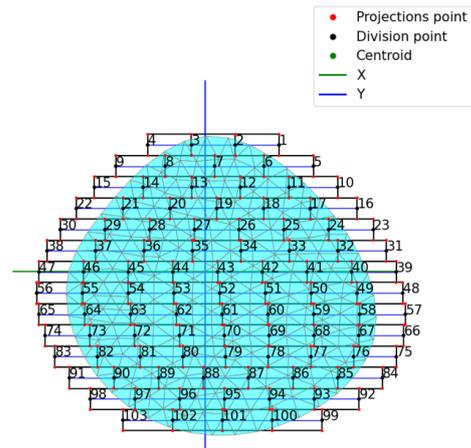


Figure 2.51: Example: Plot Output Round Panel

2.3.6 Flowchart : Full Second Section

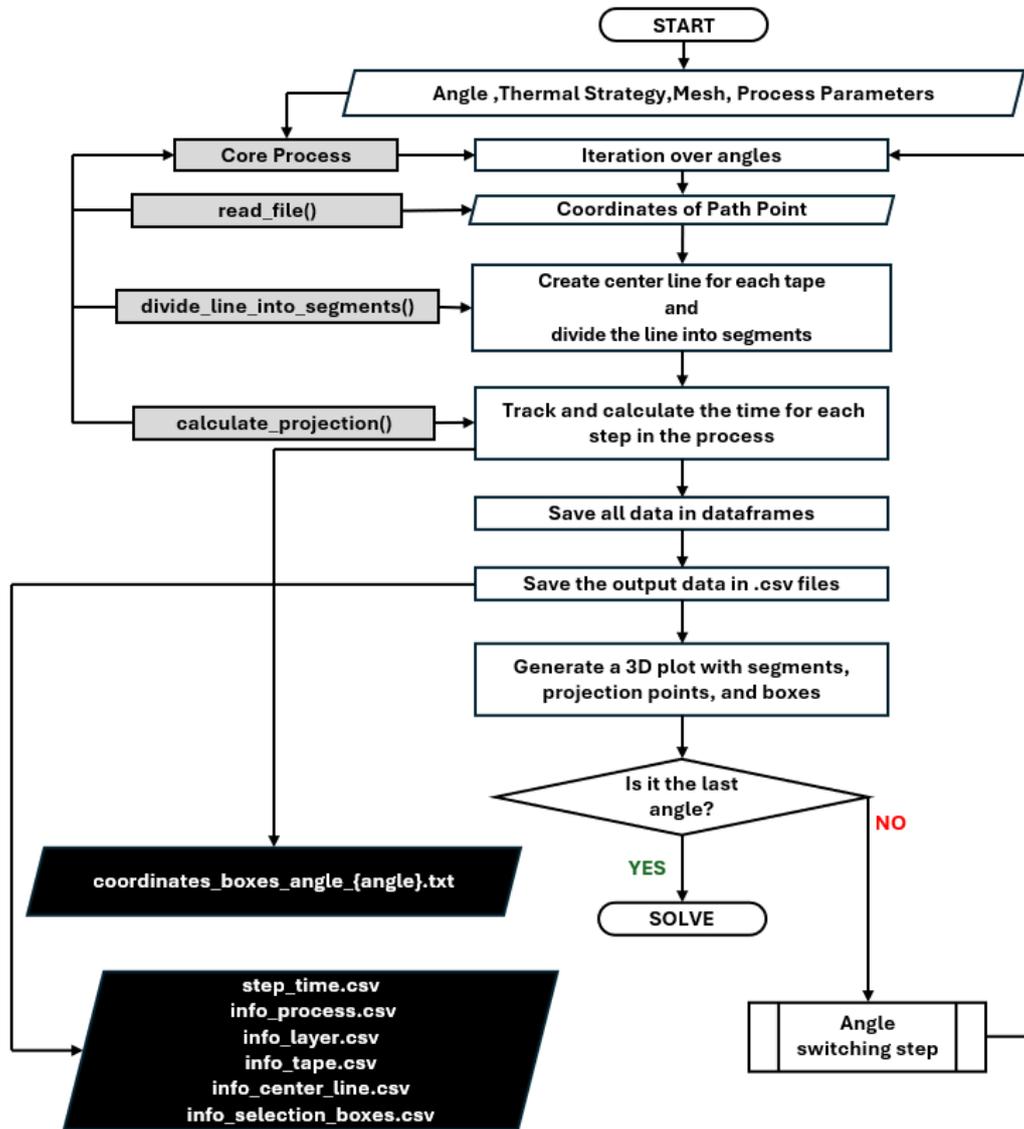


Figure 2.52: Flowchart: boxes_steps_mode.py

2.4 Curved Panel

After completing the modelling of the tapes on flat panels, represented as rectangular boxes, a possible solution for the tapes on curved panels has also been explored. This approach is based on a concept analogous to the one previously presented but involves the use of geodesic curves to define the trajectory.

In the case of flat surfaces, the trajectory for each individual tape can be determined simply by drawing a straight line between the endpoints of the path, without requiring further considerations. Conversely, for a curved surface, it is necessary to follow its variation in curvature, which requires detailed spatial information.

Several strategies exist for determining the optimal trajectory, as outlined in Section 2 of Chapter 1. In the literature, approaches based on parametric curves, numerical formulations, or differential equations can be found. However, these solutions often require detailed data on the domain, such as surface equations or control points. In the context of this research, geodesic curves were selected as they allow for trajectory determination without the need for extensive surface-specific information, since they can be directly identified through the mesh, making them particularly suitable for the present study. The model under consideration is an "irregular" surface composed of multiple subsurfaces with different curvatures.

Specifically, it represents a section of an airliner fuselage, as illustrated in Figure 2.53.

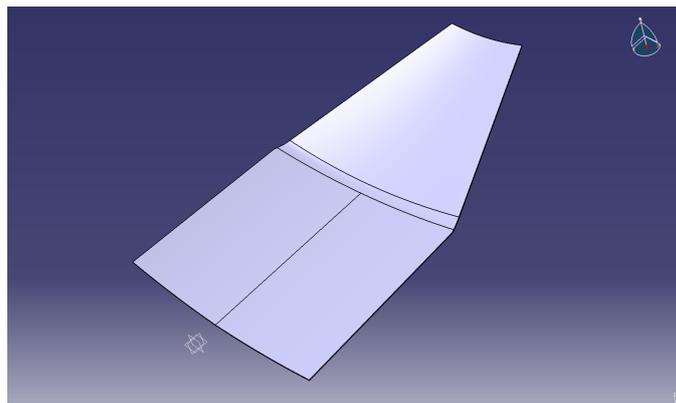


Figure 2.53: Curved Panel Model : section of airliner fuselage

A "geodesic path" refers to the shortest trajectory between two points on a surface [37]. The underlying idea of the study is to first identify the intersection points of the planes, oriented according to the lay-up directions,

with the boundaries of the mesh, as done for flat panels. Subsequently, the shortest trajectory between these points is traced, following the curvature of the surface and respecting the offset between each line. This approach fits perfectly within the context of trajectory modelling for the AFP process and is conceptually similar to the setup proposed in the previous model.

Using a code similar to that for flat surfaces, the intersections between the parametric plane, oriented according to the established direction, and the offset planes (based on the chosen strategy) are calculated. Subsequently, the coordinates of the intersection points with the mesh boundaries are determined and saved. The output files for the case with a tape width of 200 mm and an angle of 120° are shown in the following Figures 2.54 and 2.55.

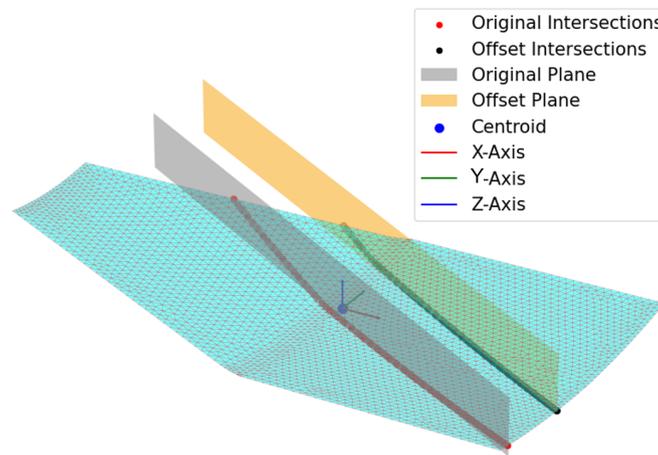


Figure 2.54: Representation of the intersection of the two planes with the tool surface mesh

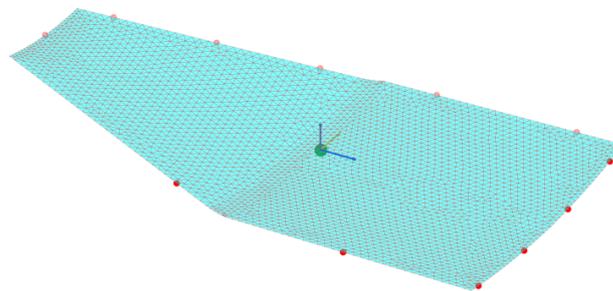


Figure 2.55: Intersections at the boundary between offset planes and the mesh

For the implementation of geodesic curves, the **FlipOut algorithm**, developed by Nicholas Sharp in the Python Library `potpourri3d` ([43]) and presented in [42], was used.

This algorithm implements the piecewise-Dijkstra approach, which divides the calculation of the shortest path into distinct segments, adapting to complex surfaces and variations within the domain.

FlipOut operates by iteratively adjusting the edges in the mesh to reduce the path length, gradually refining the geodesic path until it reaches a local minimum.

The described approach is particularly effective for navigating irregular surfaces, as it does not require explicit knowledge of the underlying geometry, but only the initial and final points of the path, as in the case at hand, the endpoints of the tapes.

The issue encountered in the application is that the algorithm, as originally conceived by N. Sharp, requires initial and final nodes for the path definition. Therefore, another algorithm was implemented which, given the coordinates of the endpoints as input, performs two functions:

- **Node Presence Verification:** Checks whether a node is already present at the given coordinates or if a nearby node exists within a very tight tolerance.
- **Node Handling:** If the node is not present at the specified location, identifies the closest node and relocates it to the desired position, Figure 2.56.



Figure 2.56: Node position verification

In this way, the dynamic addition of the necessary nodes for the path definition can be managed, solving the problem of the need to have predefined

nodes.

Some tests carried out are presented in Figures 2.57 and 2.58, where the tape dimensions are varied. In the first case, the width is 150 mm, and in the second, it is 12.7 mm, which is the standard width for the AFP process with a lay up angle of 120° .

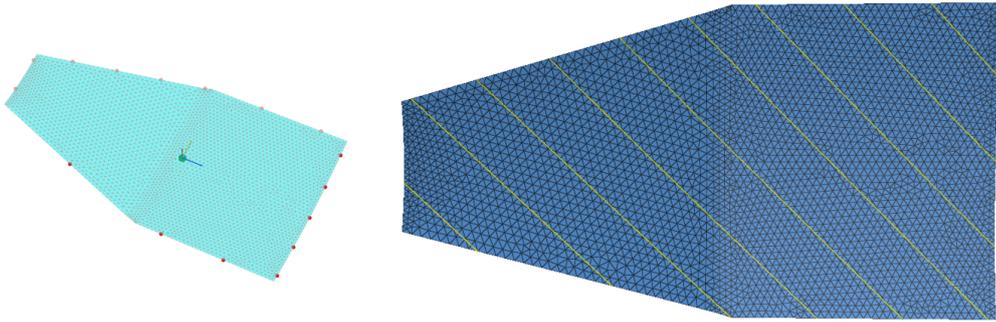


Figure 2.57: Paths on the tool surface with a tape width of 150 mm

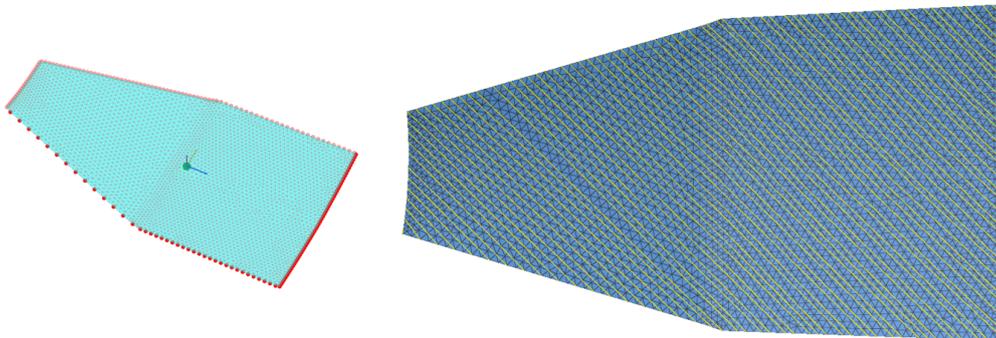


Figure 2.58: Paths on the tool surface with a tape width of 12.7 mm

It is immediately noticeable from the images (Figures 2.57 and 2.58), in a very clear manner, that if the mesh is not optimized or the surface has different curvatures, the offset between one path and another is not respected. It is also evident from Figure 2.59 that even with a simpler surface, the offset may not be respected.

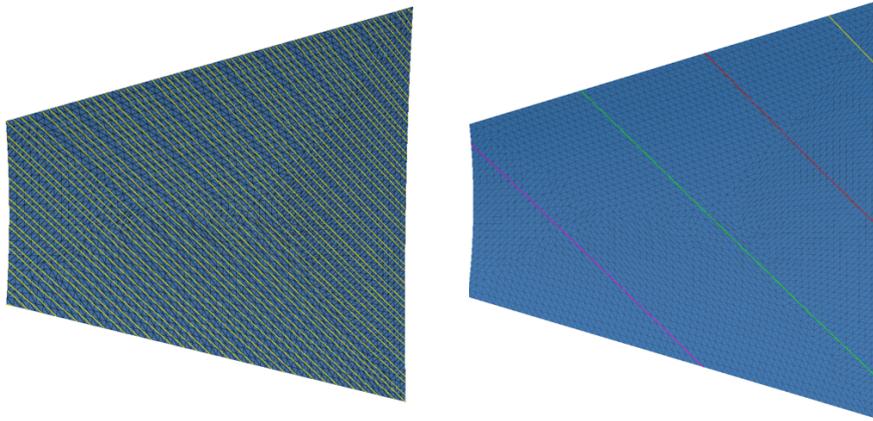


Figure 2.59: Test with the geometry of one part of the tool surface

Due to the issue encountered, the study and in-depth analysis of trajectory tracing on curved surfaces would have required significant time and resources, as well as a more detailed investigation that exceeds the primary objectives of this research. As a result, work on this aspect has been suspended. However, it is hoped that the observations and findings presented may provide useful insights for future studies and the initiation of subsequent projects, which could further integrate and develop the proposed approaches.

Chapter 3

Thermal Transient Analysis

The analysis implemented in ANSYS for the present study is a transient thermal analysis, aimed at determining the variation in temperature and other thermal quantities over time, as well as observing the resulting thermal response of the system. This analysis follows procedures similar to those of a stationary thermal analysis, with the fundamental difference that it involves the application of time-varying loads.

Consequently, it is necessary to define time-dependent boundary conditions and thermal loads, depending on the specific simulation to be conducted.

Since this project requires functionality that is not available in the standard software interface, both in terms of geometric configuration and time management, customisation of the simulation process is necessary. To this end, ANSYS Mechanical APDL is used within the ANSYS Workbench platform, as a support tool for advanced simulation control.

The use of APDL allows a specific workflow to be defined, ensuring more accurate management of the selection of elements in the different process steps and the correct application of boundary conditions.

To implement this customisation, two input scripts in INPUT (.inp) format were developed and integrated into the transient thermal analysis through the **Commands extension** of ANSYS Workbench.

- The first input file is intended to initialise the simulation, defining variables, parameters, settings required to start the analysis and convert the CSV files to values.
- The second, on the other hand, forms the core of the element selection management during the analysis. This script is executed at each time step of the simulation, repeating the required operations for each increment. The progressive advancement of the simulation is ensured

by increasing the number of steps, allowing a controlled temporal evolution of the process.

The ANSYS Workbench interface is used for defining and importing the geometry, generating and editing the mesh, assigning materials and analysing the results.

This combination of Workbench and APDL makes it possible to optimise the transient thermal simulation and ensure its correct execution.

3.1 Ansys Parametric Design Language

Ansys Parametric Design Language (APDL) is a structured scripting language used to interact with the Ansys Mechanical solver, enabling the automation of simulations, parameterisation of models, and customisation of numerical analyses, particularly in highly complex cases where the required interactions extend beyond the capabilities of the standard GUI.

Mechanical APDL (MAPDL), a finite element analysis programme, is driven by APDL. It can run Stress Analysis, Spectrum Analysis, Vibro-Acoustic Analysis and Thermal-Heat Transient Analysis [8].

Some of the operations that can be performed with APDL scripting include:

- **Geometry and Meshing**
- **Solver Controls**
- **Pre-/Post-Processing**
- **Vector and Math operations**
- **Process Control**
- **Create and Use Macros**

The main Features of APDL are :

- **Analysis Automation:** Allows simulations to be executed without using the graphical interface, improving efficiency and repeatability.
- **Parameterization:** Enables the definition of variables to control geometries, materials, loads, and boundary conditions, facilitating model modification.
- **Working with Mesh and Finite Elements:** Provides commands for mesh generation and management, supporting different types of finite elements.
- **Loop Structure and Logical Conditions:** Supports loops (*DO), conditions (*IF), functions (*FUNCTION), and subroutines to create complex scripts.
- **Input and Output Management:** Can read, write, and modify text files for data processing.
- **Integration with ANSYS Solver:** Enables the control of structural, thermal, fluid dynamics, and electromagnetic analyses.

- **Useful for Companies:** APDL enables the modification of settings, workflow customisation, and automation, supporting the development of sector-specific applications. It also facilitates the execution of advanced mathematical operations and provides advanced post-processing capabilities.

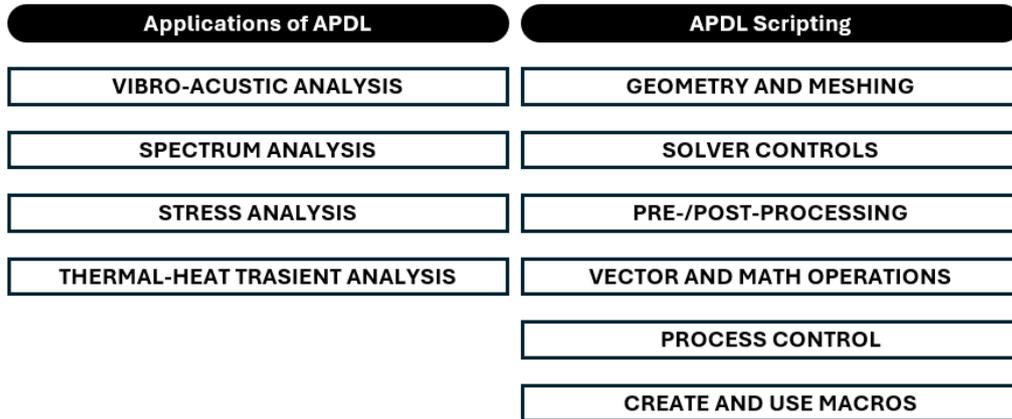


Figure 3.1: APDL Applications and Scripting Functions

3.2 Analysis Setting

The aim of this simulation, as mentioned above, is to reproduce the thermal behaviour of the Automated Fibre Placement process with thermoplastic material, in order to analyse the temperature trend during deposition and obtain information on the crystallisation of the material.

In order to obtain reliable results that can be compared with experimental data, it is essential to pay attention to various aspects in the configuration of the model.

One crucial aspect concerns the construction of a suitable geometric model, in which the mesh must be detailed enough to capture the physical phenomena involved and not too computationally heavy; therefore, it must be optimised.

In the real application, a support tool is usually used for the deposition of the tapes, made of materials such as aluminium or steel; therefore, two objects are analysed: the multilayer laminate, which represents the progressively deposited material, and the support tool, which acts as a base and influences the overall thermal behaviour.

During the process, these two elements are in direct thermal contact, which means that the simulation must take into account the thermal conduction effects between them.

In the real T-AFP process, the material is heated by a laser, which directly melts the matrix at the nip point between the tape and the laminate beneath, welding both together.

Consequently, the second aspect to be observed is the correct definition of the boundary conditions and/or heat exchanges between the elements. Therefore, the simulation must consider the distribution of the energy supplied by the laser, the progressive deposition of the tapes over time and the thermal exchange between the laminate and the support tool.

To manage these aspects dynamically, APDL commands are used, which allow the model configuration to be updated at each simulation step.

To ensure greater flexibility, the project is organised into modules within ANSYS Workbench, figure 3.7.

3.2.1 Model Geometry and Mesh

Since the composite material is assumed to be laminated over a tool for the simulation, certain geometric characteristics must be observed to ensure the simulation and elements selection are correct, based on the outputs of the Python code. These include:

1. **Coordinate Systems:** The coordinate system is aligned with the

laminate, with its origin positioned at the Center of the laminate and at $Z = 0$. This point represents the contact plane between the tool and the laminate. The tool and laminate are positioned relative to this reference point in the Z -axis.

2. Tool and Laminate Placement:

- The tool is generated with its thickness extending in the negative direction of the vertical Z -axis, positioning it below the contact plane ($Z < 0$).
- The laminate extends in the opposite direction, above the contact plane, towards the positive Z -axis ($Z > 0$).

3. **Contact Surface Continuity:** The tool and the laminate must maintain a continuous contact surface to ensure accurate simulation of the interaction between the two entities, Figure 3.3.

4. **Tapes Reference System:** The reference system for each tape is placed at the bottom of the tape, with the origin positioned at the midpoint of the midline of the tape, Figure 3.4

For the design of the mesh, it is essential to respect the specifications regarding the width of each tape, which must be divided into at least three elements in the spatial directions, and composed of a single element for each UD layer along the Z -direction, Figure 3.5.

For the representation of the tool, the simplification of the mesh involves the use of a single element in the Z -direction, with a thickness corresponding to that of the tool, in order to reduce computational complexity.

Linear elements with 8 nodes (**SOLID 278 type**) are used, which reduces the total number of nodes, as they eliminate the mid-side nodes characteristic of conventional quadratic elements. In addition, contact elements are defined between the two adjacent surfaces of the two models, these are of the **CONTA174 type** and **TARGE170 type**.

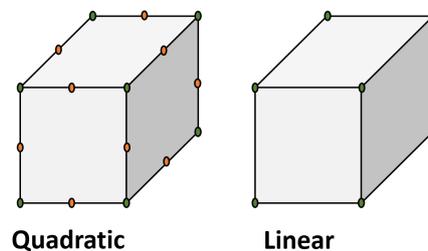


Figure 3.2: Quadratic and Linear Elements

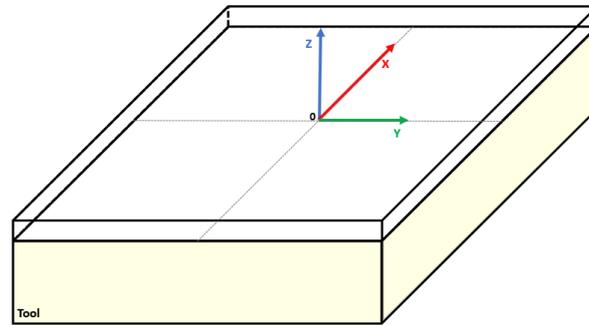


Figure 3.3: Entire Model: Tool (Yellow) and Laminate (White)

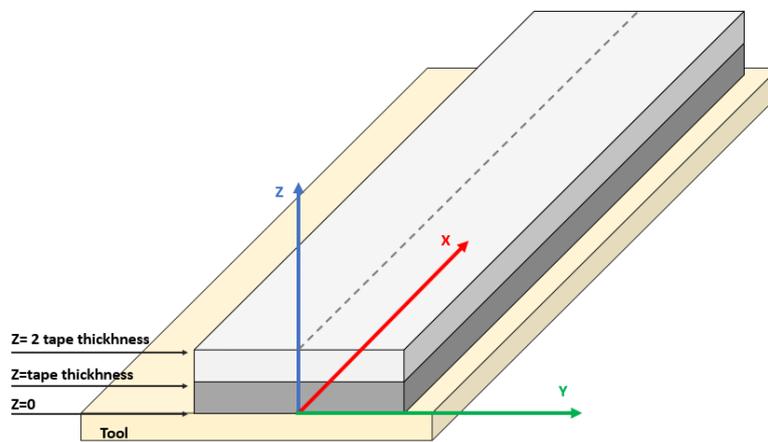


Figure 3.4: Tape Reference System

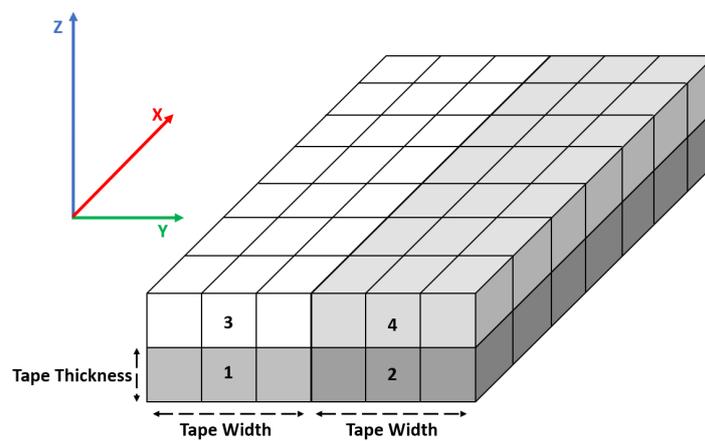


Figure 3.5: Depiction of the geometric constraints of the mesh

3.2.2 Material and Boundary Conditions

The material used in the simulation for the laminate construction is CF/LM-PAEK, a thermoplastic polymer reinforced with carbon fibres, renowned for its excellent mechanical and thermal properties. Due to these characteristics, CF/PAEK is widely employed in numerous industrial applications, particularly in the aerospace sector. The model parameters required for the simulation have been derived from the work of D. Fricke [22], which provides experimental data related to the material's properties, adapted to the operating conditions anticipated within the simulation context. A summary of these parameters is presented in Table 3.1.

Regarding the thermal conductivity, only the value in the X -direction, which corresponds to the direction of tape placement and fibre alignment, is used. This is more than adequate for the current simulation, given its focus on the macroscopic behaviour of the laminate. However, if a more detailed model of the laminate were to be developed, considering other directions would be necessary to account for the anisotropic thermal properties of the material. The adopted lamination sequence, $[90^\circ / -45^\circ / 0^\circ / 45^\circ]_{2s}$, results in a laminate that is symmetric with respect to the mid-plane with a variation of $\pm 45^\circ$ between the layers. This stacking configuration contributes to enhancing the mechanical properties of the laminate. Specifically, a symmetric laminate allows the uncoupling of membrane and bending behaviours.

From an analytical perspective, this results in the cancellation of the $[B]$ matrices within the $[ABD]$ matrix. Moreover, the use of these angles for the lay-up of the individual layers ensures a certain degree of in-plane isotropy, improving resistance to secondary loads and limiting excessive thermal expansion coefficients, while also reducing stresses at the free edges of the laminate.

Density $[\text{Kg mm}^{-3}]$	1800
Isotropic thermal conductivity $[\text{W m}^{-1} \text{C}^{-1}]$	4.57
Specific heat at constant pressure, C_p $[\text{J Kg}^{-1} \text{C}^{-1}]$	997.8

Table 3.1: Material parameters for CF/LM-PAEK

In order to represent the tape deposition effect during the simulation of the T-AFP process, it is necessary to model the moving the laser in the machine head. The calculation of the different time steps is crucial for determining the selection of element nodes and the application of the heat source. The duration of each time step is determined by the speed of the machine head

and the size of the elements.

The Python script implements the distinction between the following steps:

- **Heating time steps**, during which heat is applied at the strip deposition;
- **Tape switching steps**, which is the time required for the robot to move to the next starting position. During this phase, the simulation can neglect thermal and mechanical effects as no deposition takes place;
- **Layer switching steps**, which are necessary to allow for the examination and optical inspection of the newly deposited layer;
- **Cooling-down step**, which precedes the de-moulding step.

During the simulation, the laminate is subject to the action of the heat source and thermally interacts with the surrounding environment and the support tool. The ambient air is considered to be at an average temperature of 20°, while the tool is maintained at a constant temperature on the surface opposite to that of contact with the laminate (bottom) through a permanent boundary condition. In the last phase, the final cooling, this condition is replaced by convection at room temperature.

The heat exchanges between the laminate, the tool and the environment are modelled by means of standard convection coefficients, which allow the heat dissipation and boundary conditions of the system to be adequately represented.

To prevent elements not yet deposited from influencing the process during the step sequence, their temporary deactivation is implemented. Elements in this state, termed DEATH, remain present in the model but with zero stiffness and thermal conductivity values, and with cancelled thermal and mechanical properties. When a tape is processed, the corresponding elements are reactivated with all associated properties and the thermal boundary condition is applied. As a result, thermal conduction between active and inactive elements is eliminated. The effect of air is simulated by applying a convective boundary condition to the surfaces of the active elements of the deposited webs, at regions in contact with inactive elements or without adjacent elements. For the last deposited layer, a specific selection of boundary conditions is applied to ensure proper modelling of the thermal system.

The boundary conditions applied during the simulation are schematically summarised in Figure 3.6, while the values of the coefficients and temperatures in Appendix A6.

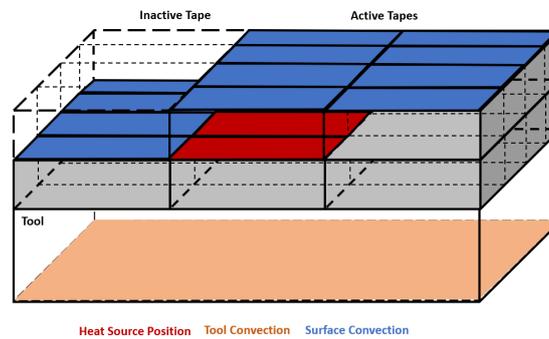


Figure 3.6: BCs Applied

3.2.3 Workbench Project

It is decided to divide the project into several modules in order to work more precisely on the individual components. This approach allows changes to be made more easily and efficiently, ensuring a more flexible and controllable simulation overall. Furthermore, it allows different models and geometries to be processed without having to replace the module dedicated to transient thermal analysis. In particular, two Mechanical Model modules have been created, one for the tool and one for the laminate, which allow the thermophysical properties of the materials and the discretisation of the mesh to be managed separately. The laminate can also be defined with ACP (Ansys Composite Prepost), which allows for easier setup of FRP parts. In addition, there is a Transient Thermal module, which represents the central element of the thermal simulation and communicates with the mechanical models to define the global thermal behaviour. Two APDL script commands must be added to the analysis module and activated at different times: the Commands (ADPL) only for the first step, relating to the setting that reads the first input file, and the Commands (ADPL) 2 for all subsequent steps. This input file, repeated for all steps, simulates the movement of the AFP machine head on the tool and the laminate. The set-up is shown in Figure 3.7.

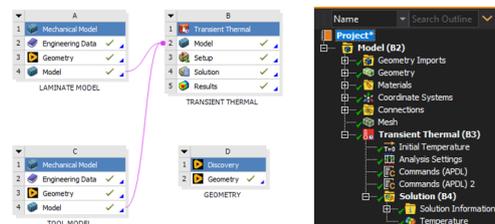


Figure 3.7: Workbench Project and Commands (APDL) Details

3.3 Commands (APDL)

As mentioned in the section on the outputs of the second part of the Python document, the final outputs consist of CSV files, which contain various DataFrames necessary for importing the data into ANSYS.

Therefore, the first step from a programming perspective is to import these data and transform them into a format that is acceptable and usable within ANSYS for the purposes of the project. These data must be converted into arrays and variables, allowing them to be recalled when necessary, creating consistent workflows for the final application, and saving them into the project memory in ANSYS.

3.3.1 Data Acquisition

For the conversion of data into arrays and variables, the `*TREAD` command is used, which enables the reading of multiple data formats, such as those in UTF-8 encoding and delimited by tabs, spaces, or commas. In the case at hand CSV files fall under the comma-delimited category.

In addition to this, the `*DEL` and `*DIM` commands are used to delete and create tables and arrays, respectively, while the `*VFUN` command is employed for moving data into arrays.

Here is the step-by-step explanation:

- 1. Skipping the header and counting the lines in the file:** First, the code sets a variable to indicate that the first line in the CSV file is informational, which should be skipped. It then uses the `/INQUIRE` command to check how many lines are in the file, storing that value. The number of lines to be read (excluding the header) is calculated by subtracting the skipped line from the total number of lines.
- 2. Deleting any previous table data:** This is important to avoid any conflicts or leftover data from previous runs.
- 3. Creating a table to hold the CSV data:** Next, a table is created with a number of rows determined by the line count of the file and the needed columns, which are necessary to adapt to the format of the data in the CSV. This table will serve to store the data read from the file, facilitating its use in the subsequent stages of analysis.
- 4. Reading the CSV file into the table:** The `*TREAD` command reads the data from the relative file into the new table. It is guaranteed that the first row (the header) is skipped, while subsequent data is correctly processed and loaded into the table.
- 5. Moving the data into an array:** The code then prepares to transfer the data from the table into an array for further processing. First, a check is made and then any previous arrays are deleted to free up space. Next, a new array is defined using the command `*DIM`, and an extra column is added to handle moving data from the table. The data from the table is then transferred to the array using the `*VFUN` command, which essentially shifts the data by one position to one column left to fill the Zeroth column of a Table Array.

3.3.2 Variables Set-Up

Once the data has been transferred to ANSYS, the next step involves setting up the variables as they are stored in the data frames of the CSV files. This applies to all data except for those related to the coordinates of the characteristic points of the tapes used for element selection. Therefore, only the parameters and process properties, as well as the characteristics of the laminate dimensions and tapes, are considered.

Additionally, other empty variables are introduced and initialised for defining control indices and values to be used in subsequent cycles.

3.3.3 Analysis Initialization

Before running the second input file for the selection of elements and the application of boundary conditions, it is necessary to configure the initial conditions for a finite element analysis in ANSYS APDL.

This process focuses on the preprocessing phase, nonlinear simulation settings, and solution parameters.

Several lines of code are written to achieve this configuration.

- `/NERR`: It controls the level of error reporting during the execution of the analysis.
- `/PREP7`: This command enters the preprocessing phase of ANSYS, where the model geometry, boundary conditions, materials, and mesh are defined. It is a crucial step prior to solving the system of equations during the simulation.
- `NROPT,FULL`: The `NROPT` command defines the nonlinear solution options for the simulation. By setting it to `FULL`, the analysis is configured to account for full nonlinear effects, including material nonlinearity, large deformations, and possibly contact nonlinearity.
- `SELTOL,0.00001`: The `SELTOL` command sets the tolerance for selecting nodes and elements within the model, improving the precision of the model, which is crucial for high-accuracy simulations. It is necessary to modify this parameter because the standard selection tolerance is larger than the tape thickness, leading to incorrect selections
- `/SOLU`: This command transitions to the solution phase, where ANSYS begins solving the system of equations defined in the preprocessing step, and is essential for obtaining the final analysis outcomes.

3.4 Flowchart: Commands (APDL)

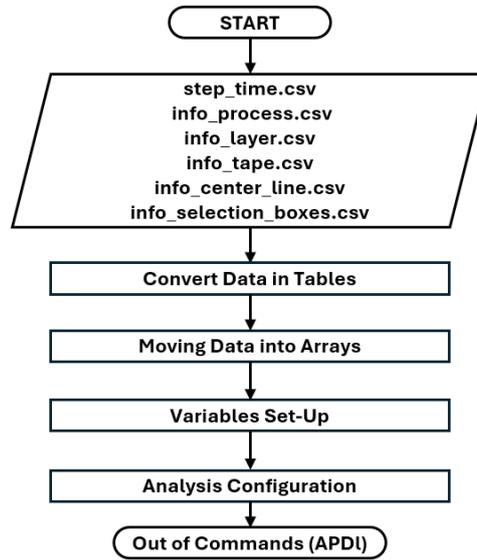


Figure 3.8: Flowchart: Commands (APDL)

3.5 Commands (APDL) 2

The second input file is utilised to access this information and finalise the analysis framework established in the preceding step. This process ensures the correct utilisation of previously stored variables and enables the precise selection of elements based on geometric parameters, defined according to angular configurations and layup strategies (standard, centre, out).

Furthermore, it ensures the appropriate definition of boundary conditions and waiting times, in accordance with the specified thermal analysis parameters (flash heating enabled or disabled, and the various waiting modes: T, L, C, N).

This input script remains consistent across all time steps defined within the tabular data, while the workflow is structured to allow step-by-step differentiation.

Following this, the focus will shift towards the APDL Script, specifically in terms of its programming structure and implementation.

3.5.1 Differentiation of the Steps

For the correct utilisation of the variables, the `step_time.csv` file is used as the reference file, as it is crucial for differentiating between steps. Simply examining the data it contains allows for the identification of both the thermal mode and the chosen lay-up strategy (Figure 3.9).

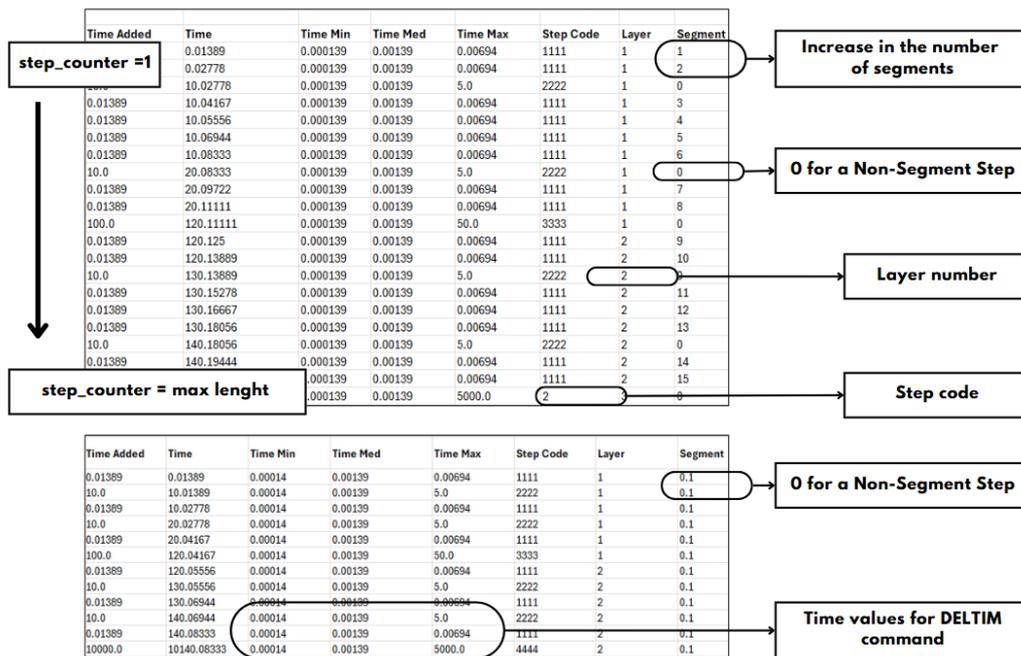


Figure 3.9: Detailed Functioning of `step_counter`

The control variable for the entire workflow in ANSYS is defined as `step_counter`. This variable enables navigation along the array `step_a` and is initially set to 1 in the input script. Then it is updated at each step, ensuring a correspondence between the extracted data and its usage. The maximum value it can reach corresponds exactly to the length of the array.

3.5.2 Selection of Elements

The selection of elements is also governed by the `step_counter` variable, which is closely linked to the identification of the various steps in the process. This variable enables the selection of the midline endpoints of the tape and/or different segments, thereby ensuring the correct application of the appropriate boundary conditions.

However, before delving into the details of the programming, it is essential to understand the element selection process by analysing the coordinate systems used and the underlying reasoning behind this procedure.

In ANSYS, various selection methods exist, distinguishing between nodes, elements, areas, volumes, groups, and lines, each characterised by its own commands and subcommands. Nevertheless, there is no built-in selection option that allows direct operation based on the modelling of tapes, which is the primary objective of this project.

To overcome this limitation, an alternative strategy is adopted, ensuring a selection process that aligns with the progression of steps and meets the requirements of the simulation process.

Following the modelling of the paths and the extraction of the defining points for each segment of the tapes and their endpoints via Python code, the approach developed relies on creating a series of selection boxes containing the elements to be activated during each step, according to the conditions that need to be met.

Consequently, for each point, whether a division point or an endpoint, a new local reference system is defined according to the chosen strategy, with the point itself as the origin and Z as the vertical axis. Along the X -axis in this new reference system, selected as the direction of the machine head movement and hence the direction of material deposition, the coordinate of the next endpoint is calculated in order to select the appropriate portion of elements for that particular segment or tape. The X -axis follows the orientation angle.

The identification of elements along the width occurs along the Y -axis using the standard selection command in Ansys, imposing as limits half of the maximum width in both directions.

For the vertical Z -direction, an iterative loop is performed under conditions that allow the identification of elements corresponding to the thickness of the tape for each layer to be deposited, with the first layer set at $Z=0$.

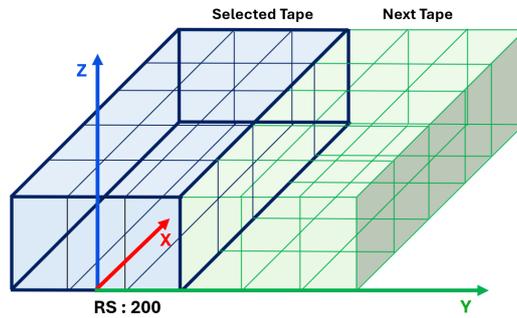


Figure 3.10: Element Selection Boxes

Each "starting" point is located in the central reference system (RS 0), and at this point, the local reference system (RS 200) is defined with the X -axis oriented at an angle equal to that of the lay-up for that layer. In contrast, for the "destination" points, a mathematical transformation is performing using a translation first and then a rotation matrices in 2D plane and in a counterclockwise sense .

$$y = y_{\text{destination}} - y_{\text{starting}} \quad (3.1)$$

$$x = x_{\text{destination}} - x_{\text{starting}} \quad (3.2)$$

$$x' = -x \cdot \cos(\theta) - y \cdot \sin(\theta) \quad (3.3)$$

$$y' = y \cdot \cos(\theta) - x \cdot \sin(\theta) \quad (3.4)$$

Once the point has been rotated, the distance between the original point and the rotated point is calculated, which corresponds to the length of the segment or tape in the new system.

$$\text{distance} = \sqrt{x'^2 + y'^2} \quad (3.5)$$

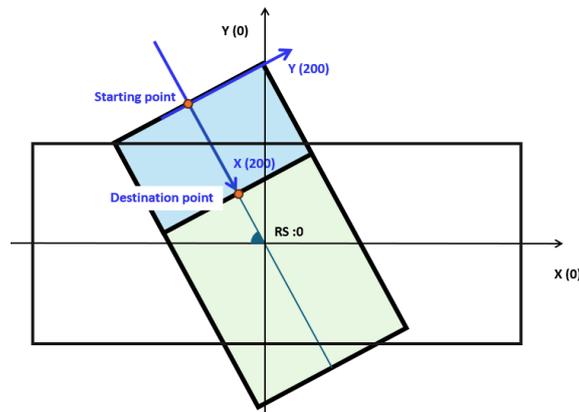


Figure 3.11: Central and Local Reference Systems

3.5.3 Process of the Script

In terms of programming, this process is based on a series of conditional control loops, executed by the `*IF` conditional function, associated with sub-conditions of equality (`EQ`), inequality (`NE`), and combination (`AND`), which compare multiple sets of data organised into levels. The iteration follows along the array related to the overview and temporal steps, the array which contains the coordinates of the midpoint path points, and the array which holds information on orientation degrees, layer numbers, and tapes.

3.5.3.1 Step setting

At the beginning of the simulation, when the control variable is set to 1, all laminate elements are deactivated using the ANSYS commands, while only those of the tool remain active. This replicates the real-life scenario of the process. For subsequent stages, a reset of all previously activated boundary conditions is performed, and the reference system is defined as being set to the centre of the model (`RS 0`), in order to establish an initial state for each iteration of the code.

For each iteration, automatic time stepping is enabled, and the time step sizes to be used for the current load step are defined through `DELTIM`, following the logic previously described in section 2.3.4.1.

3.5.3.2 Angle identification

The identification of the orientation angle is carried out by comparing the extracted layer number data with the corresponding time, as well as defining the vertical coordinate. Two states are possible:

- If the `step_counter` is at its first step, the angle corresponds to the first value recorded in the layer information file, and $Z = 0$.
- If the layer value at the current time differs from that at the previous time, 1 is added to the layer number variable (initialised to 1), the angle is selected, and an additional tape thickness is added to the vertical position.

Figure 3.12 shows the flowchart for the identification of the correct layer angle.

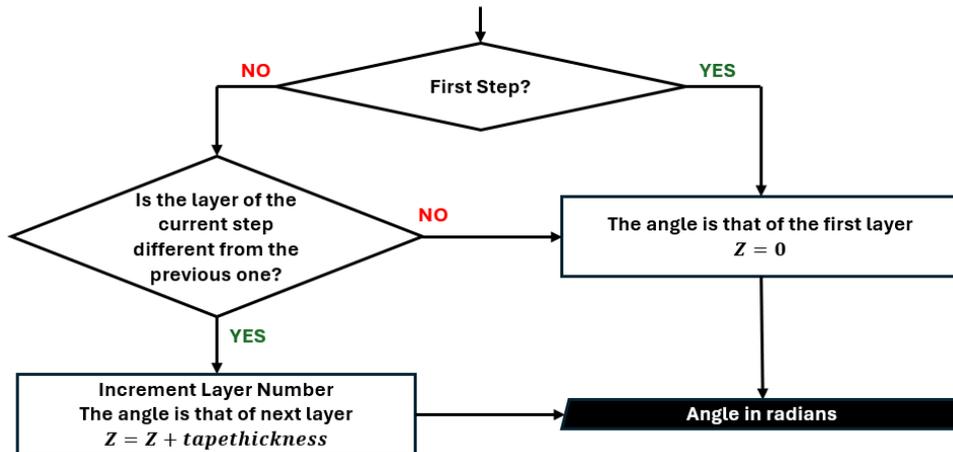


Figure 3.12: Flowchart: Angle Identification

3.5.4 Activation of elements and application of BCs

Only when the step is related to a heating step (code 1111), that is, during the deposition time of material for each layer, is the activation of the elements carried out. This is done by setting the new coordinate system and performing the transformation and translation of the coordinates as previously described.

However, there are two different methods for the selected flash heating mode, as it alters the identification of the point coordinates:

- If the **flash heating is deactivated**, for each step that satisfies this condition, the starting point is selected based on its position in the array containing the coordinates, which initially coincides with the beginning. Subsequently, it is incremented by 1, and the endpoint of the segment is set to the next position. This process is iterated.
- If the **flash heating is activated**, the process proceeds in the same manner for the starting point, which is where the local reference system is set. However, for the endpoint of the tape, the position in the coordinate array corresponds to the sum of the position of the starting point and the number of segments that make up the tape. Finally, this new point is updated as the starting point for the next iteration.

Considering a simplified case, i.e., a single layer with a lay-up angle of 0° and consisting of three tapes, each made up of two segments (Figure 3), the process is analysed with or without flash heating for each heating step.

In Figure 3.13, which represents the plot of the tape modelling, each black point indicates the viewing points of the tapes, to which the coordinates in the array shown in Figure 3.14 are associated. The order of counting is related to the deposition strategy adopted, in this case, the standard one.

In the case without flash heating and two successive iterations that complete the tape, in the first iteration, the local system is set at point 1, and the selection end point coincides with the second point. In the second iteration, point 2 becomes the starting point for the selection, and point 3 becomes the endpoint. This way, all elements of the tape are selected, following the operation for each segment.

When flash heating is enabled, the entire tape is selected in a single step. The origin point of the local system remains point 1, while the end point is point 3, calculated as 1 (starting point) + 2 (number of segments for that tape).

At the end of each tape, the position of the last point is updated by incrementing it by one so that, for example, in this case, the starting point of the new tape becomes point 4, in both thermal strategy cases.

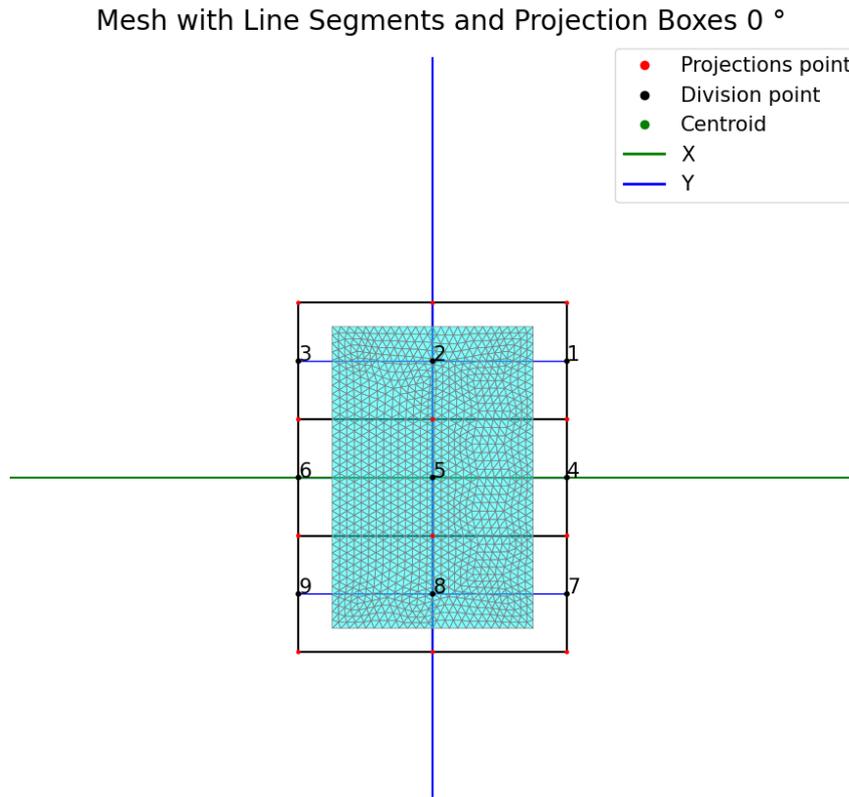


Figure 3.13: Example: Selection of elements and application of BCs

Position	X	Y	Z			
1	0.286169	0.25	0	1° Segment	2° Segment	1° Tape
2	0	0.25	0			
3	-0.28617	0.25	0			
4	0.286169	0	0	1° Segment	2° Segment	2° Tape
5	0	0	0			
6	-0.28617	0	0			
7	0.286169	-0.25	0	1° Segment	2° Segment	3° Tape
8	0	-0.25	0			
9	-0.28617	-0.25	0			

Figure 3.14: Example: Coordinates List List of coordinates of the points of the segments and tapes

Given the information regarding the position of the tape points and/or segments to be activated in the current step within the array of midline coordinates, and thus the geometric positions on the surface in the *XY* plane

relative to the central reference system (RS 0), the creation of the local reference system (RS 200) is carried out, along with the application of translation and rotation transformations. Subsequently, the activation of the elements is performed, including the selection in thickness along Z -axis.

The programming in APDL is carried out using classic selection commands such as `ESEL`, `ALLSEL`, reference system setup commands like `LOCAL` and `CSYS`, `DSYS`, and the activation command `EALIVE` [11].

For this type of step, it is necessary to model the simulation of the heating of the matrix of the tape by the heat source, namely the laser, at the moment of deposition onto the underlying layer or, in the case of the first step, directly onto the tool. In this context, the aim is to define a boundary condition that represents the temperature change of the elements as they are activated over time.

To achieve this, a table is inserted that, at the start time of the step, associates the temperature of the tool, when the laser has not yet heated the tape. Subsequently, at the end time of the step, the temperature of the laser is associated, modulating the heating process.

In this way, a temperature variation is obtained during the heating process at the nodes of the elements that are just activated, located at the bottom of the tape.

Also, the definition of the start time and end time follows an iteration with the variable `step _counter`. However, in the case of the first step, the start time coincides with the time at the beginning of the array. In the other cases, the start time corresponds to the time of the subsequent step, and the end time corresponds to the current step.

Start time [s]	Tool Temperature [$^{\circ}C$]
End time [s]	Laser Temperature [$^{\circ}C$]

Table 3.2: Start and End Time Definitions

If considering the example in Figure 2.45 in the figure for segment 3 with the process parameters in the Appendix A6:

10.0278 s	20 $^{\circ}C$
10.0417 s	470 $^{\circ}C$

Table 3.3: Example: Start and End Time Definitions

For all types of steps, except for the final cooldown step, a convective boundary condition is applied to the nodes at the bottom of the tool model. This

condition simulates the maintenance of the tool at a constant temperature. The **SF** command is used with a convective heat transfer condition [11]. In the case of step 4444, the convection temperature is replaced with the cooldown temperature.

Furthermore, the influence of the surrounding air around the entire model is considered. This is simulated through a surface convection boundary condition, applied only to the two-dimensional surface elements of the active elements in that specific step, using **SFE** command [11].

Figure 3.15 shows the Flowchart for the elements activation and application BCs.

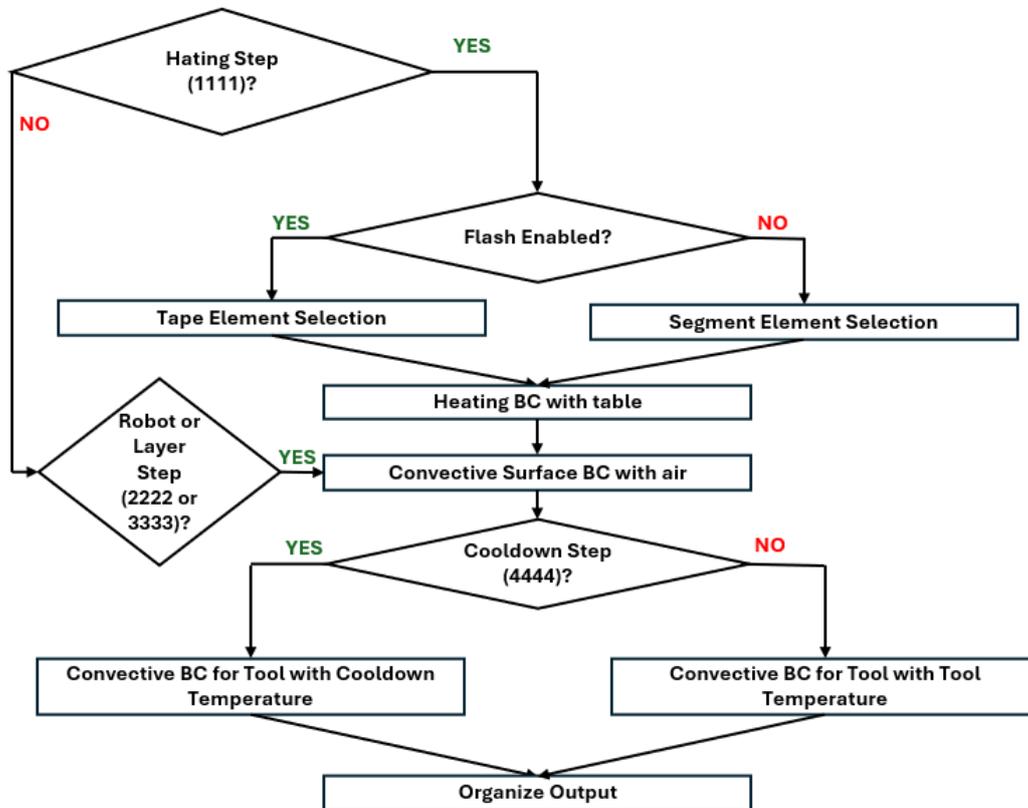


Figure 3.15: Flowchart: Elements Activation and Application BCs

3.5.5 Flowchart : Commands (APDL) 2

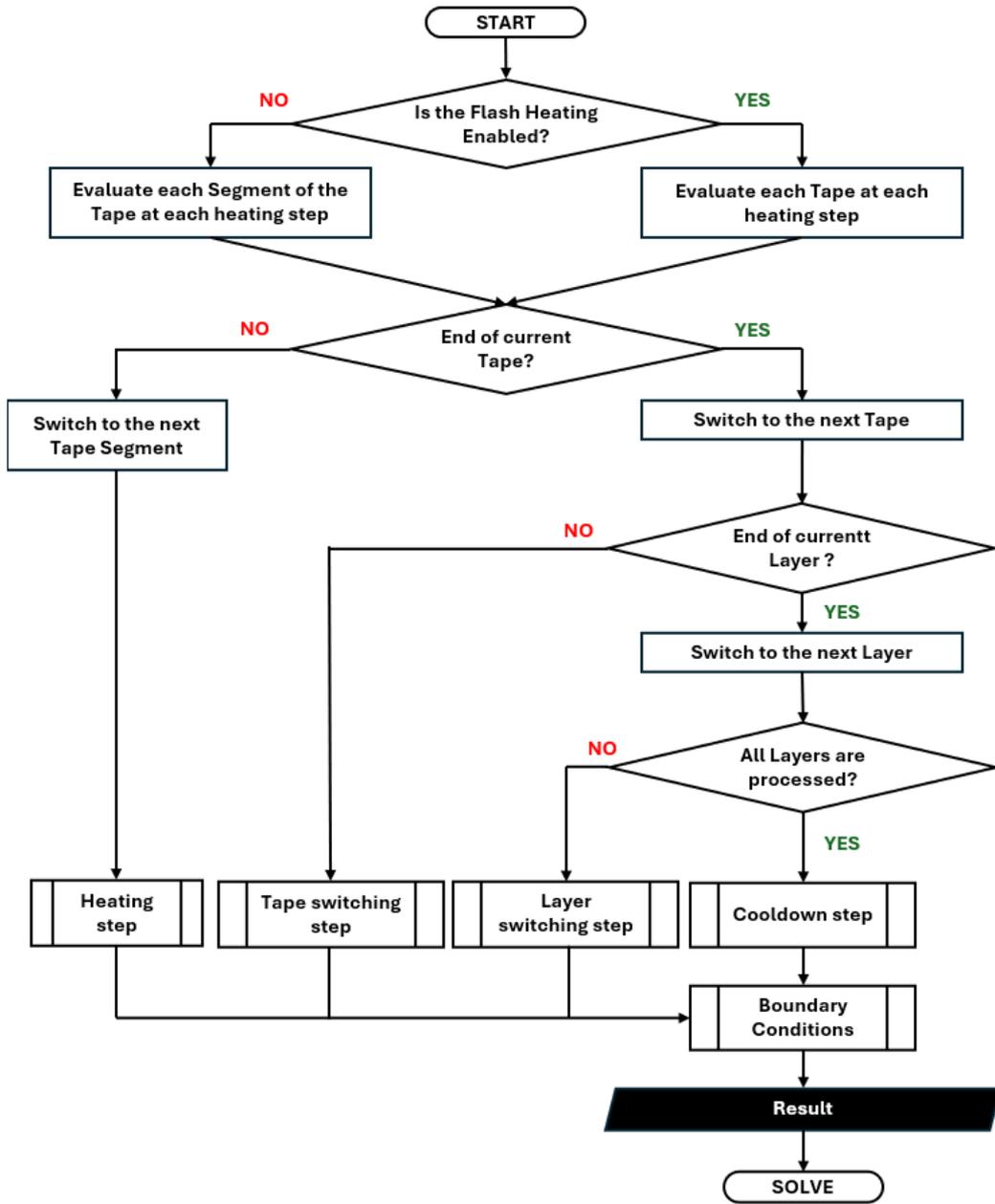


Figure 3.16: Flowchart: Commands (APDL) 2

Chapter 4

Results

4.1 Test Conducted

The tests conducted aim to validate the implementation of the path creation method and the modeling of tapes as boxes using Python codes, as well as the selection of elements and the application of boundary conditions through ANSYS and ANSYS Mechanical APDL codes to replicate the real thermal process experienced by the material during the deposition of layers. Square and round flat panels were chosen as final models in order to validate the application of the method for curved shapes, one of the initial objectives of the simulation.

Despite the fast solution time of the Python code for extracting the final coordinates of the tape trajectories and saving the data in the input files for thermal simulation, the analysis in ANSYS requires significant computational time, as the various layers are processed tape by tape or segment by segment, requiring many steps and time increments. Therefore, the models were reduced to test panels, but they remain functional for verifying the validity of the entire work, as they contain a sufficient number of tapes, based on the characteristics defined in this work. Reducing the dimensions results in fewer steps for the transient analysis, improving processing efficiency and reducing computational time.

The dimensions of the square and round panels are within the same millimeter range in order to obtain more consistent results. The same process parameters were used for each model, such as the laser temperature, the air temperature in the deposition chamber, the temperature of the tool, and the processing times between one step and the next, as well as the same thermal characteristics of the material, CF/LM-PAEK. Details of these data are provided in [Appendix A6](#).

In brief, the modelling characteristics of the tapes in the thermal analysis model, as detailed in the previous chapter, are repeated here. Each tape is modelled with 3 elements along the width of 12.7 mm, each with a dimension of 4.233 mm, and one element along the thickness, equal to 0.000198 mm. The tool is made of steel and has a thickness of 20 mm. Regarding the laminate configuration, the final 8-layers panels, in both cases, is developed symmetrically, with deposition orientation according to the sequence $[90^\circ / -45^\circ / 0^\circ / 45^\circ]_2s$.

However, results are also compared with 4-layers panels that have a single lamination sequence $[90^\circ / -45^\circ / 0^\circ / 45^\circ]$.

The analysis aims to evaluate the real application of the thermal transformation of the material. Although all thermal process modes are presented in 2.3.4.1, only the complete configuration, with and without flash heating enabled (Mode 1 and Mode 5), is considered here. All process times are taken into account, including the waiting time for the robot head positioning after each tape, the inspection time required for layer optical inspection, and the final cooling time before complete removal of the laminate. In addition, the strategies of standard, central, and out lay-up are compared.

During each step, the appropriate boundary conditions are applied. For deposition steps, the laser heating (heat source) is set to 470°C , along with convection between the tool and active elements, as well as convection with the air at 20°C . In other steps, laser heating is deactivated, and only convection remains active. During the deposition of the last layer, surface convection with the air is added. Finally, during the cooling process, material cooling is applied.

In Figures 4.4, 4.3, 4.2, and 4.1, the meshes modelled in ANSYS for the square flat panel and the round panel are shown, also displaying the lamination in detail.

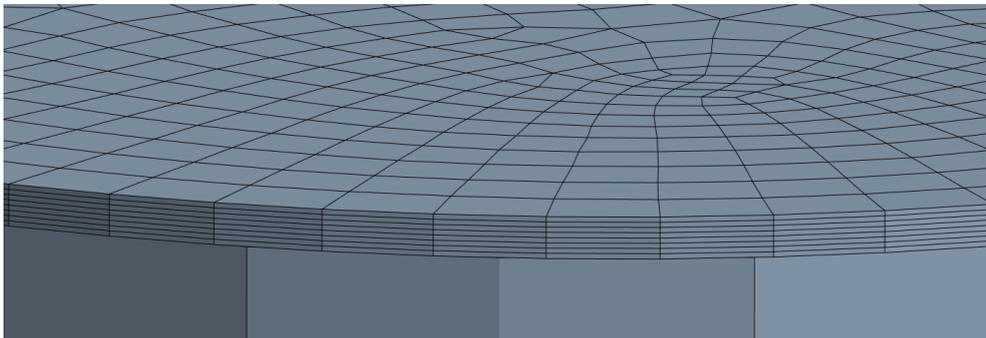


Figure 4.1: Lamination of an 8-layer flat round panel.

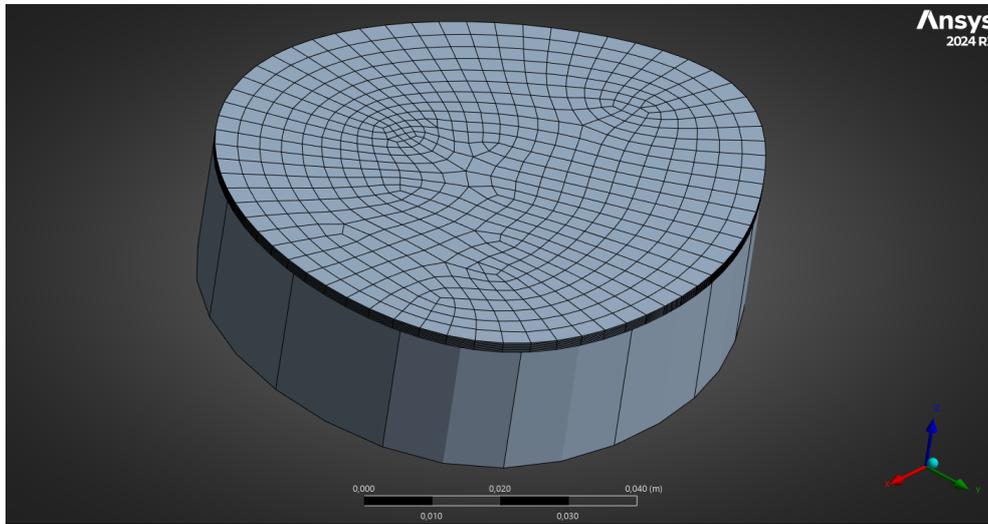


Figure 4.2: Model and mesh of the 8-layer flat round panel and tool

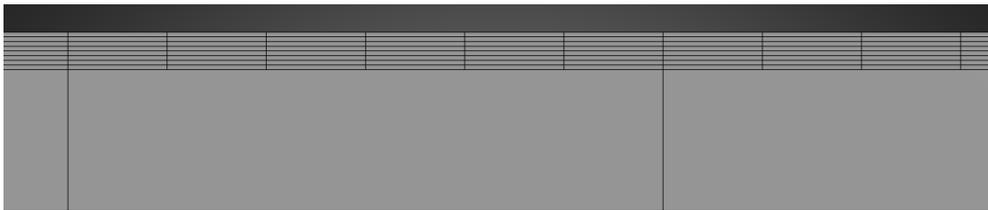


Figure 4.3: Lamination of an 8-layer flat square panel

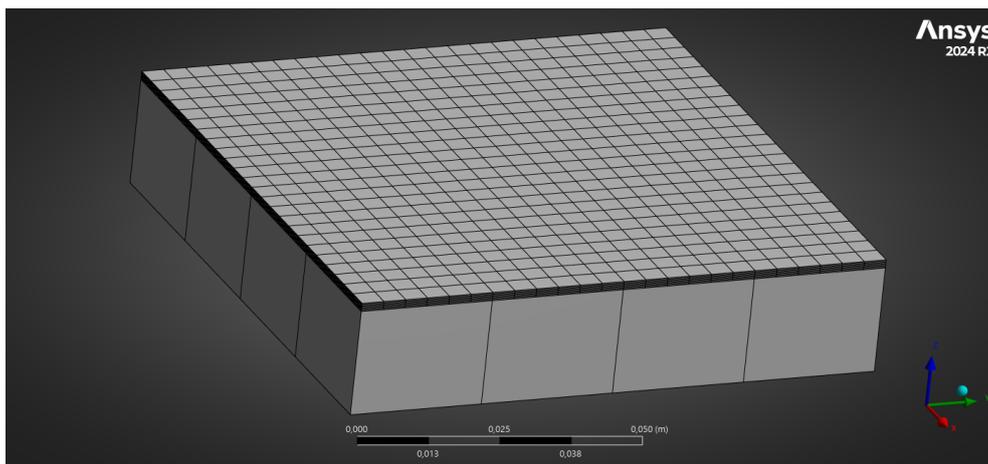


Figure 4.4: Model and mesh of the 8-layer flat square panel and tool

4.2 Flat Square Panel

The selected square panel has a $100 \text{ mm} \times 100 \text{ mm}$ quadrilateral shape and consists of approximately 9 tapes for 0° and 90° angles and 11 tapes for $\pm 45^\circ$ orientations.

As previously highlighted, this number of tapes is sufficient for result validation and to ensure the overall consistency of the model. In simulations where flash heating is disabled, the segments measure 20 mm in length, meaning each tape is divided into 2 to 5 segments, depending on its orientation.

In the following three figures, the modelling of the tapes for each layer and each orientation angle is shown, with the Standard "S" strategy in the 4.5 Figure, the Center "C" strategy in the 4.6 Figure, and the Out "O" strategy in the 4.7 Figure. The difference in lay-up strategies can be observed from the tape numbers, which represent the deposition order.

However, for better visualization, the segments are not displayed, but each tape is subdivided into segments, as shown in 3.13 .

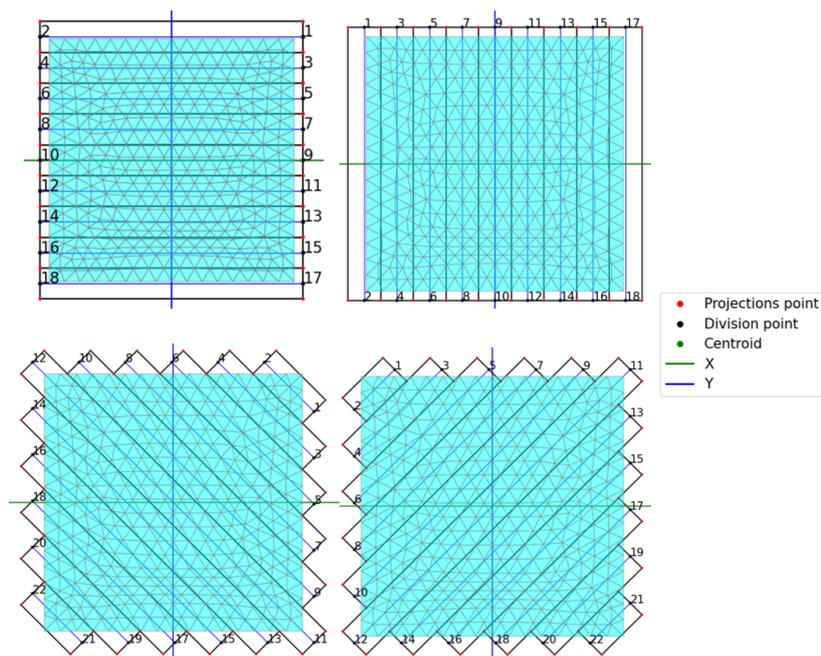


Figure 4.5: S Mode. Angles: 0° at the top right, 90° on the left, -45° at the bottom left, and 45° at the bottom right

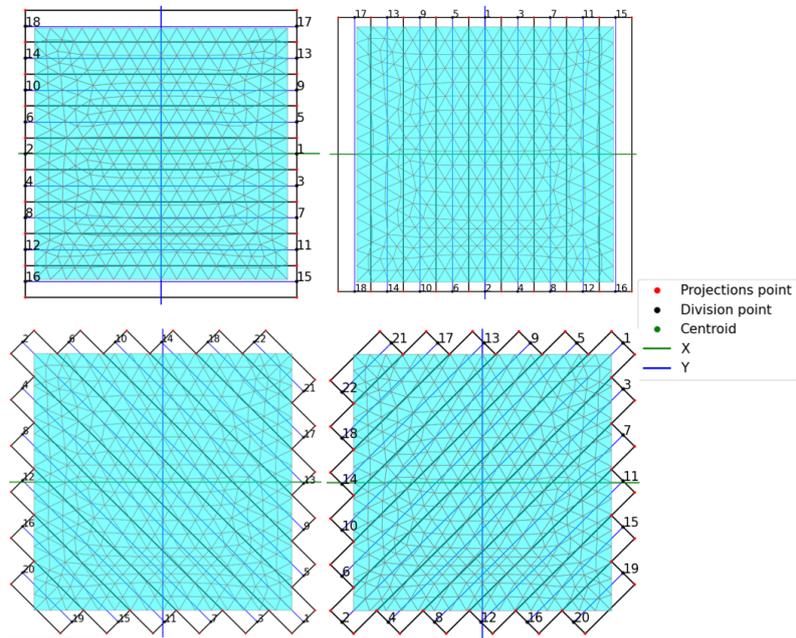


Figure 4.6: C Mode. Angles: 0° at the top right, 90° on the left, -45° at the bottom left, and 45° at the bottom right

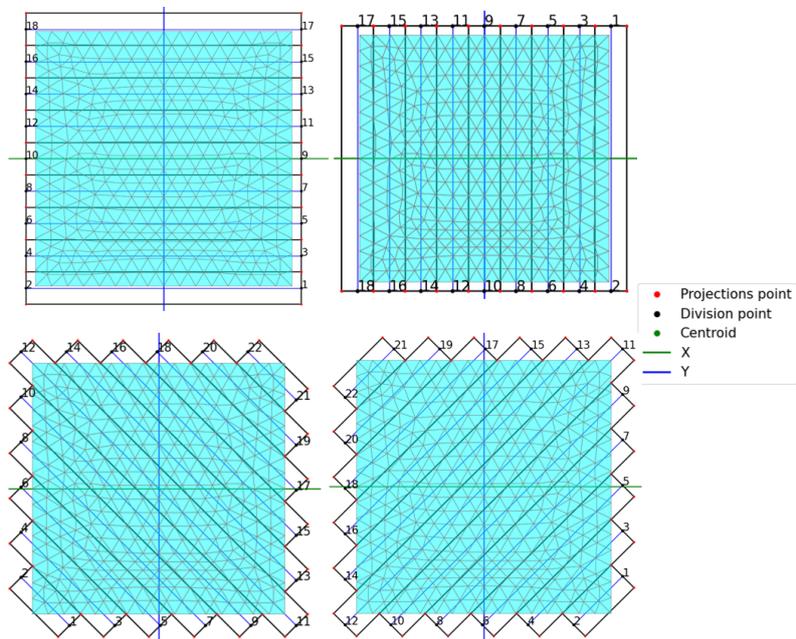


Figure 4.7: O Mode. Angles: 0° at the top right, 90° on the left, -45° at the bottom left, and 45° at the bottom right

The following different tests were produced, in sequence:

- 4 layers ($[90^\circ / -45^\circ / 0^\circ / 45^\circ]$) with/without flash heating enabled for the Standard strategy.
- 8 layers ($[90^\circ / -45^\circ / 0^\circ / 45^\circ]2s$) with/without flash heating enabled for the Standard strategy.
- 8 layers ($[90^\circ / -45^\circ / 0^\circ / 45^\circ]2s$) with flash heating enabled for the Center strategy.

The Out strategy is not detailed further, as it would yield the same results as the standard approach, with a different tape placement direction.

4.2.1 4-Layer Laminates Results

To confirm the success of the developed method, for each heating step, the activated elements in that step must be heated with the laser temperature at the bottom of the laminate. In fact, the laser acts at the nip-point, and convection occurs on the surface with the air and beneath with the tool. The figure 4.8 shows the positioning of the second tape of the laminate, segment by segment. It is clear that the process is sequential, with heating applied to each section of the tape. The images presented represent the exact moment of deposition, and it is noticeable that the newly activated elements have a temperature slightly lower compared to the previously selected elements. This is because the visualization is instantaneous at the moment of deposition. However, the actions are set correctly in the system, as in the immediately following step, the heating can be visualized. The data is also confirmed by the temperature values extracted. Meanwhile, in Figure 4.9, the difference is evident as the entire tape is deposited in one single step, with no segments involved.

Figure 4.10, on the other hand, refers to the subsequent step, which is the waiting time for the robot's positioning for the placement of the new tape, without any laser heating action. It can be observed that after approximately 8 seconds, the temperature becomes more uniform. It remains hotter in the tapes that were previously heated, while it decreases in the surrounding areas, indicating the evolution of convection between the various components of the model.

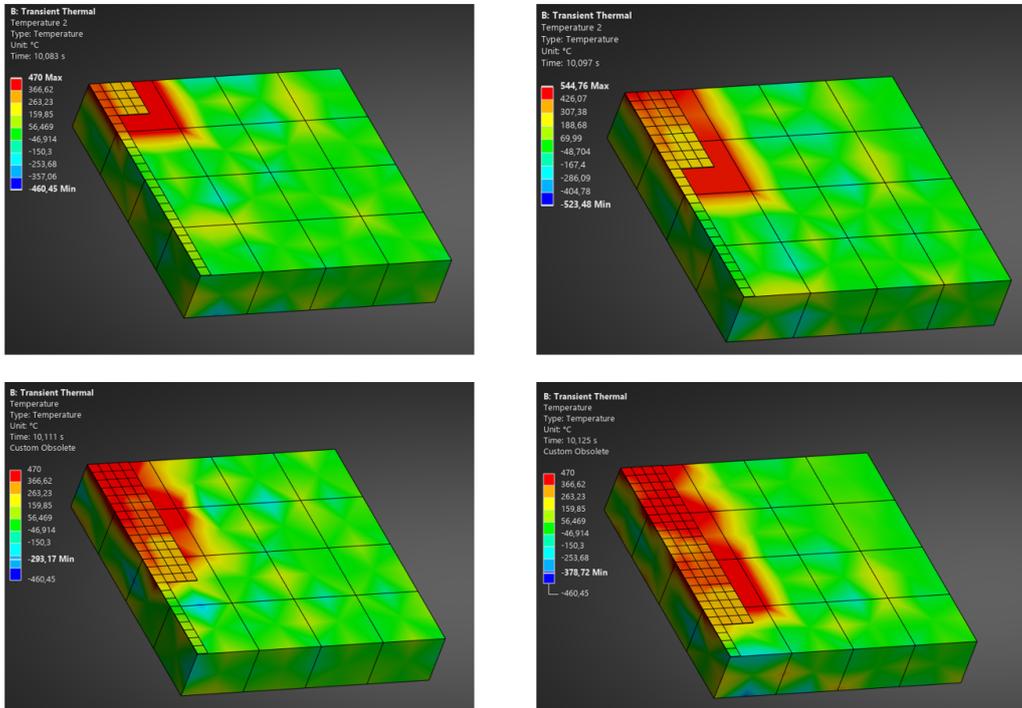


Figure 4.8: Detail of the placement of the second tape on the first layer, segment by segment

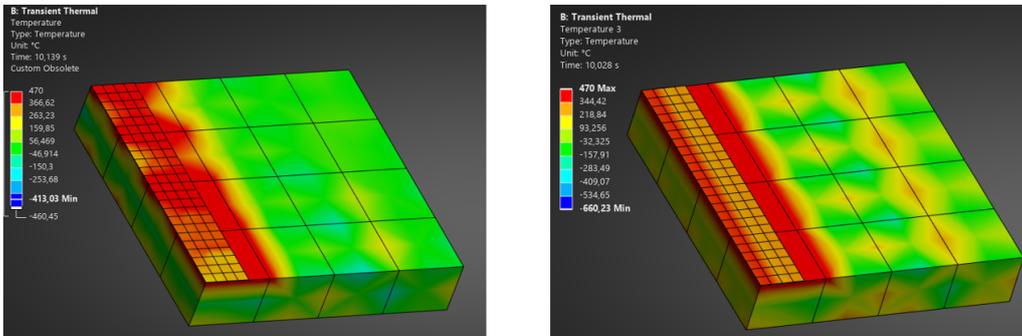


Figure 4.9: Comparison of the second tapes of the first layer, deposited with the flash heating strategy (right) and without it (left)

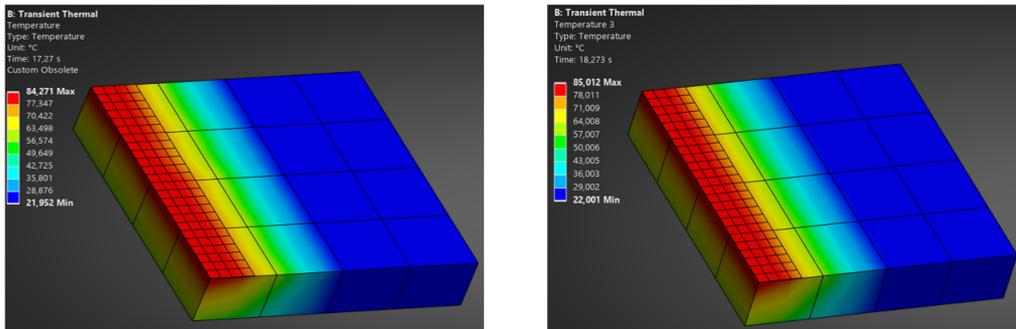


Figure 4.10: A comparison of tape temperatures, with flash heating deactivated on the left and activated on the right, during the robot's waiting time

In Figure 4.11, the deposition, with/without flash heating (right/left), of the central tape of the second layer, oriented at -45° , is shown according to the two modes. A noticeable difference between the layers is observed, with the red zone confirming the application of heating to the new tape. Additionally, the tape appears green as it is cooler on the surface, due to the convective heat exchange with the surrounding air.

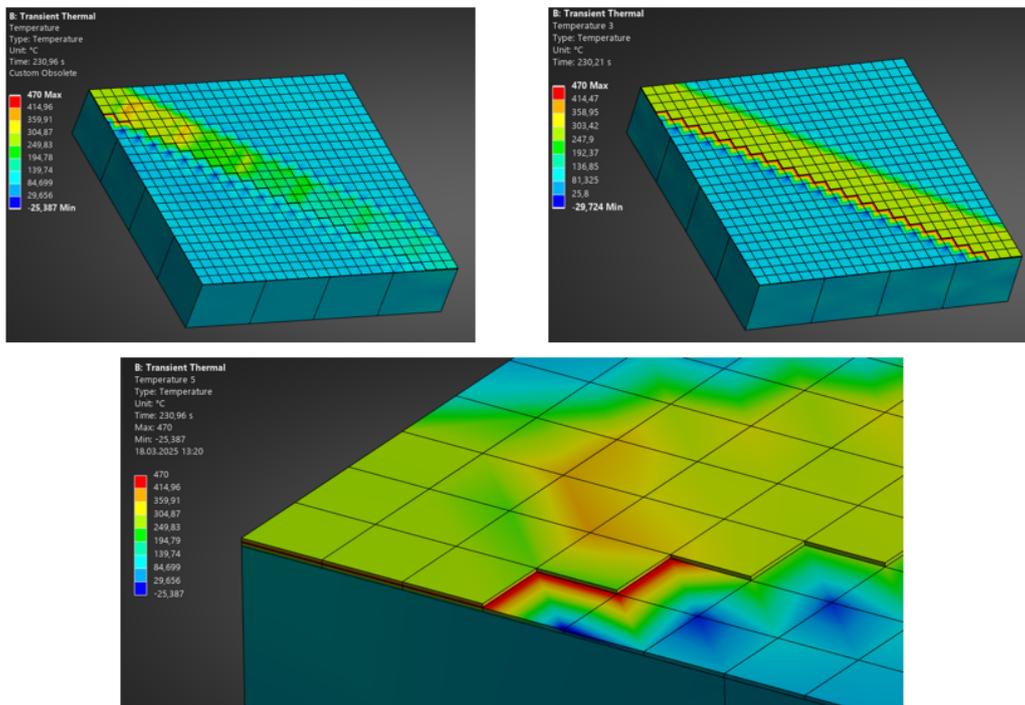


Figure 4.11: Deposition, with/without flash heating (right/left), of the central tape of the second layer, oriented at -45°

Figure 4.12 particularly illustrates the thermal exchange occurring between layers during the deposition of the tape, in this case, between the third and second layers, as well as the first. As shown, heat propagates through the thickness of the material, while the surface interacts with the surrounding environment.

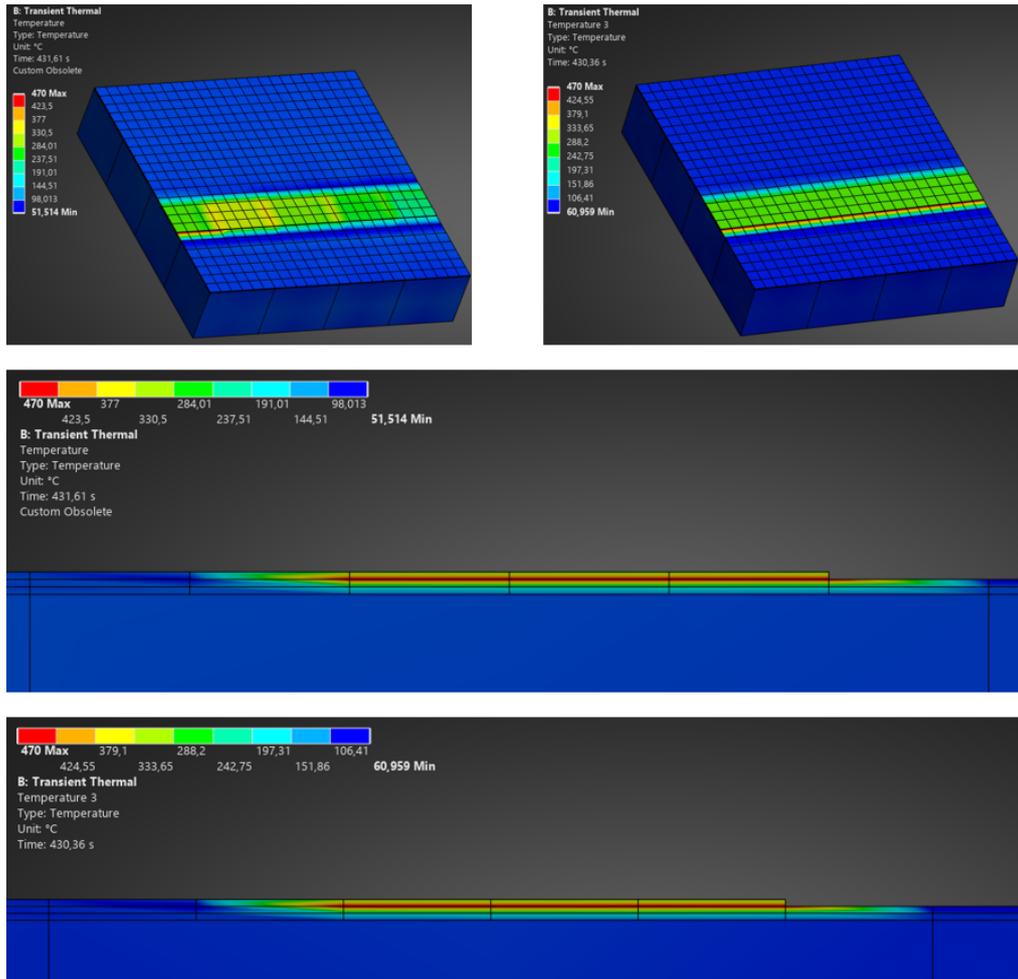


Figure 4.12: Heat diffusion along the various layers

Figures 4.13 and 4.14 compare the average temperatures extracted from each individual substep of the process in ANSYS. It can be observed that the overall temporal temperature trend is almost identical in both modes, with and without flash heating.

Furthermore, the average temperature reached at the end of the cooling period is nearly the same: with flash heating, it is 24.356°C , while without it,

it is 24.289°C.

The four temperature clusters are associated with the deposition of the tapes in the four layers. Each group corresponds to the number of peaks equivalent to the number of tapes deposited in the layer, and the time between one cluster and the next represents the waiting time for the optical inspection, during which no heating is applied, thus the temperature remains almost constant, although it slightly decreases.

When deposition occurs with the flash heating strategy, the tape is fully deposited, and the covered and heated area is larger, resulting in a more significant temperature peak compared to the other thermal deposition strategy. In fact, in this case, the temperature peaks reached are lower and more constant.

In the first case, longer tapes, especially in the case of lamination at $\pm 45^\circ$ (2nd and 4th layers), cause very different behaviour for the central tapes compared to the outer ones, leading to a higher increase in temperature. Furthermore, by counting the peaks for each group, the number of tapes per layer can be identified. The first temperature group corresponds to the first layer, which shows a different trend compared to the others, as discontinuities in conditions applied to the nodes and surfaces occur at the beginning of the analysis. This issue is ignored for this type of simulation.

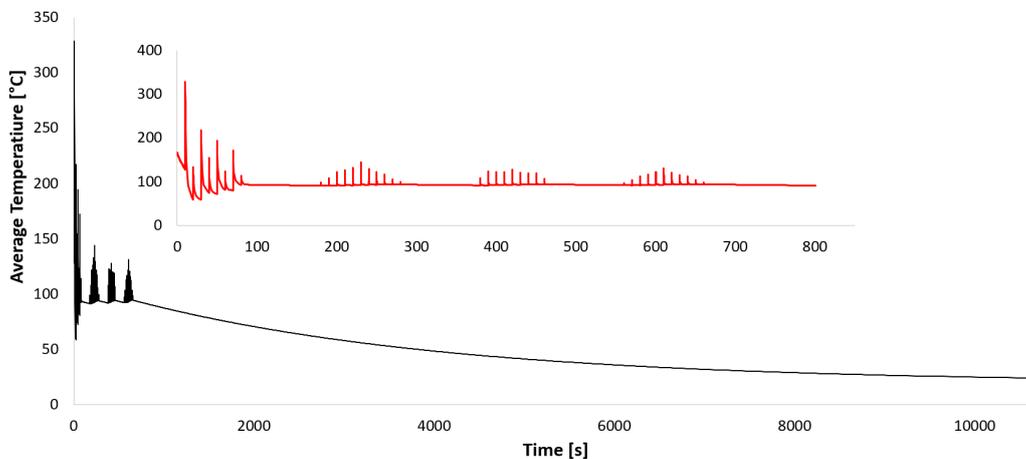


Figure 4.13: Average temperature over time of the 4-layer model placed with Flash heating, shown in black, with a detailed view in red

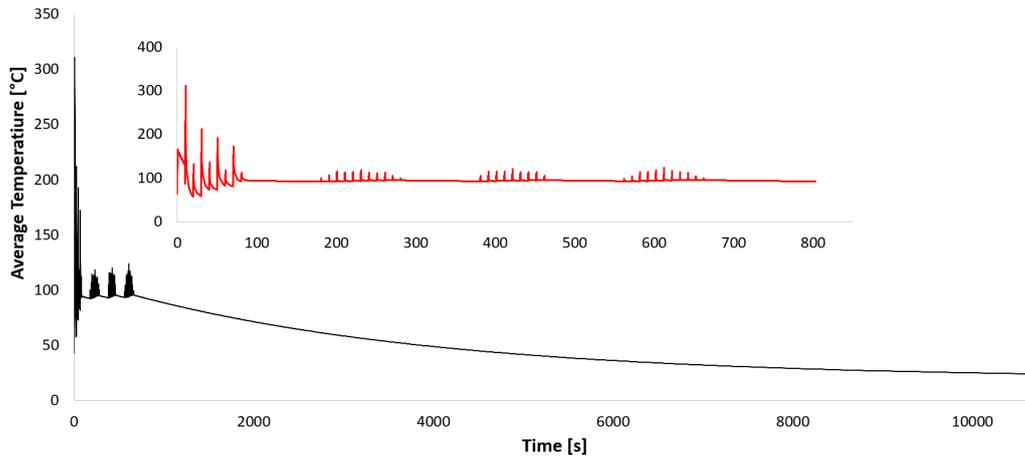


Figure 4.14: Average temperature over time of the 4-layer model placed without Flash heating, shown in black, with a detailed view in red

4.2.2 8-Layer Laminate Results

The functionality of the analysis is detailed by showing the boundaries applied to the final layer of the 8-layer panel in Figure 4.15. The temperature discretisation is more clearly visible in this case.

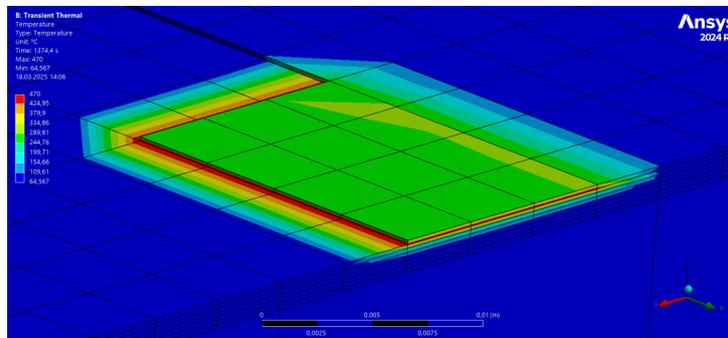


Figure 4.15: Heat transfer along the thickness of the laminate

Even for the case of the 8-layer laminate, the results are similar to those of the simple laminate with a sequential deposition in the Standard Mode. There is a perfect correspondence, also in the final temperature data, with the average temperatures reached: with flash heating, it is 24.669°C, while without it, it is 24.591°C.

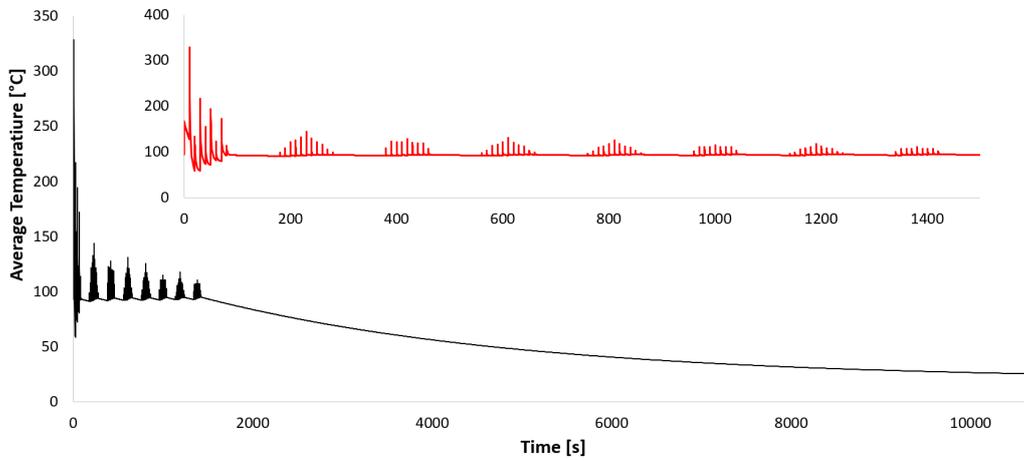


Figure 4.16: Average temperature over time of the 8-layer model placed with Flash heating, shown in black, with a detailed view in red

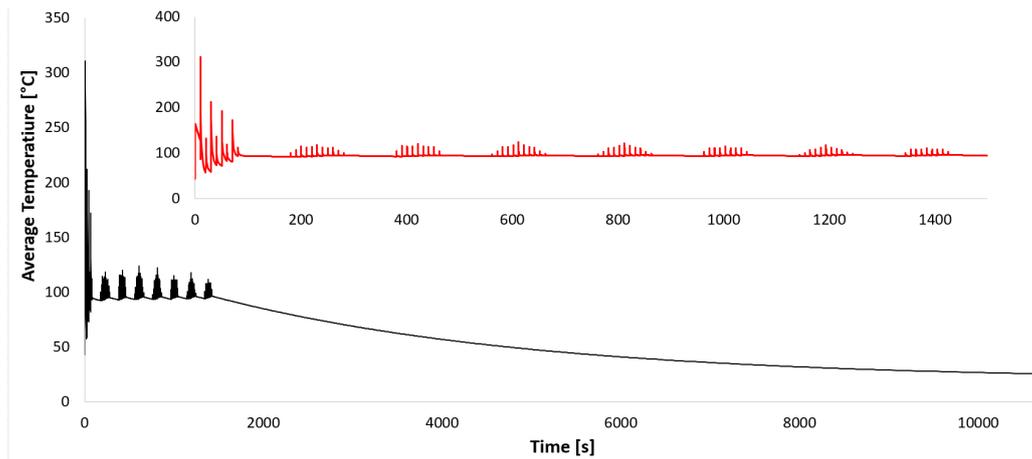


Figure 4.17: Average temperature over time of the 8-layer model placed without Flash heating, shown in black, with a detailed view in red

By comparing the two graphs in Figures 4.16 and 4.18, it can be observed that the distribution of temperature peaks differs in the new graph related to the placement of the tapes according to the Center layup strategy. Therefore, referring to Figures 4.5 and 4.6, which show the modelling of the tapes on each layer and the deposition sequence, the only difference between the two methods is that in the second case, the layup starts from the central tape, i.e., for the $\pm 45^\circ$ strategy, the longest tape. Thus, by analogy to what occurs, the highest peaks are those of the longer tapes, i.e., the central ones, which in this case are the first to be deposited. For the 0° and 90° layers, the heat-

ing remains almost constant as in the previous case, but the smaller peaks, related to the outer tapes, appear at the end of the temperature groups, precisely due to this change in deposition order. This once again confirms the validity of the implemented simulation model.

The average temperature reached at the end of the cooldown period is 24.606°C , very close to that of the standard strategy.

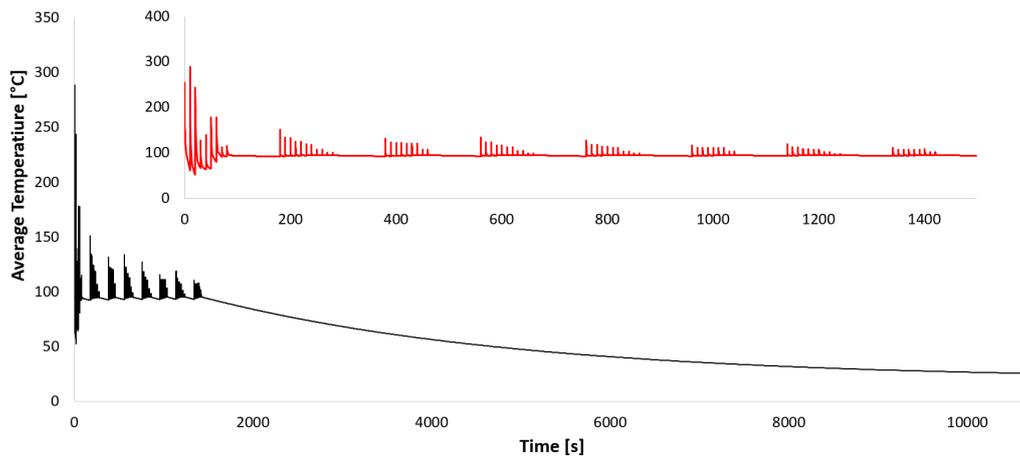


Figure 4.18: Average temperature over time of the 8-layer model placed without Flash heating, shown in black, with a detailed view in red, Center lay-up Strategy

4.3 Flat Round Panel

To test the simulation concepts, the flat round panel has dimensions similar to those of the square panel. The round panel has a maximum diameter of 90 mm, with maximum and minimum radii of 55 mm and 45 mm, respectively. The curvature of the panel is not highly pronounced and is convex. This is reflected in the section of the model described in Chapter 2, where the models produced using the AFP manufacturing process, or at least those considered in this model implementation, do not present overly complex surfaces. The complete model consists of an 8-layer laminate, with layer orientations of 0° , $\pm 45^\circ$, and 90° . The number of tapes varies between 7 and 8 depending on the chosen orientation angle. Figures 4.19 and 4.20 show the tape lay-up modelling for the Standard and Center strategies, with the Out configuration involving the positioning of the tapes in the direction opposite to the standard strategy.

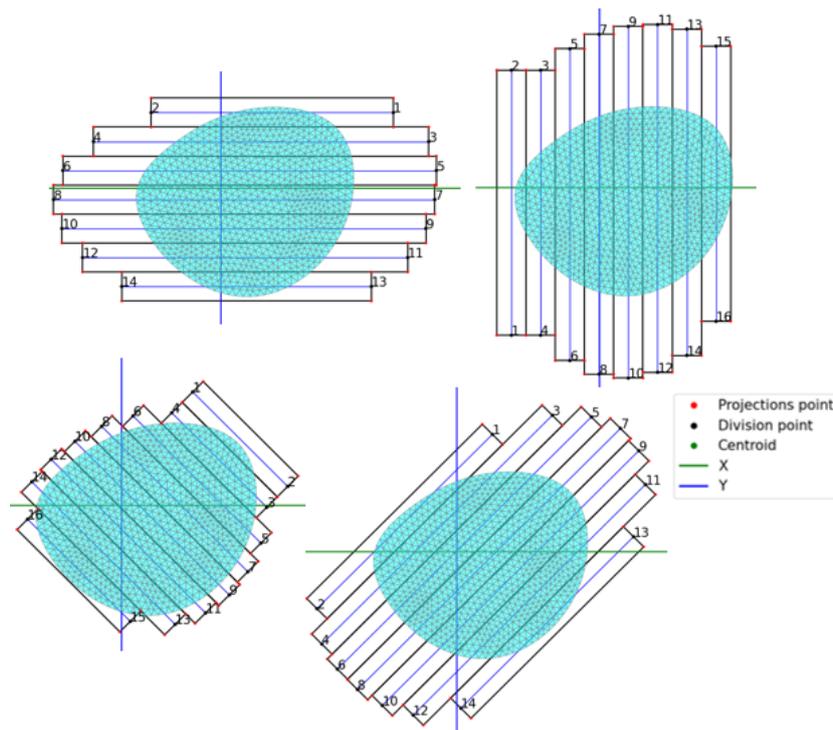


Figure 4.19: S Mode. Angles: 0° at the top right, 90° on the left, -45° at the bottom left, and 45° at the bottom right

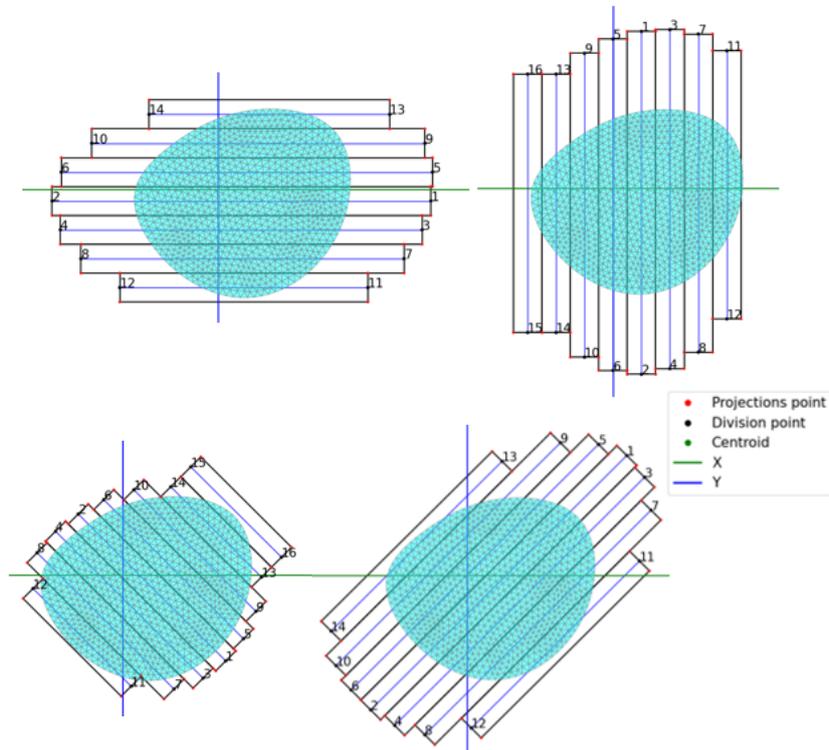


Figure 4.20: C Mode. Angles: 0° at the top right, 90° on the left, -45° at the bottom left, and 45° at the bottom right

For the flat round panel, only two cases are reported and compared:

- 8 layers ($[90^\circ / -45^\circ / 0^\circ / 45^\circ]2s$) with flash heating enabled for the Standard strategy.
- 8 layers ($[90^\circ / -45^\circ / 0^\circ / 45^\circ]2s$) with flash heating enabled for the Center strategy.

In these cases as well, the complete modality is analysed, with the application of all the boundary conditions and waiting times for each step of the process. Figure 4.23 shows the detailed central tapes of the model.

As for the flat square panel, the dimensions of the tapes are different. In this case, for all the lay-up configurations, the dimensions are smaller at the edges and larger in the center.

Therefore, the results follow the same analysis logic explained earlier. It can be observed that the results are consistent: for the longer tapes, higher peaks of average temperature are observed. Additionally, the temperatures reached at the end of the process are 23.683°C for the Standard strategy, Figure 4.21, and 23.596°C for the Center strategy, Figure 4.22.

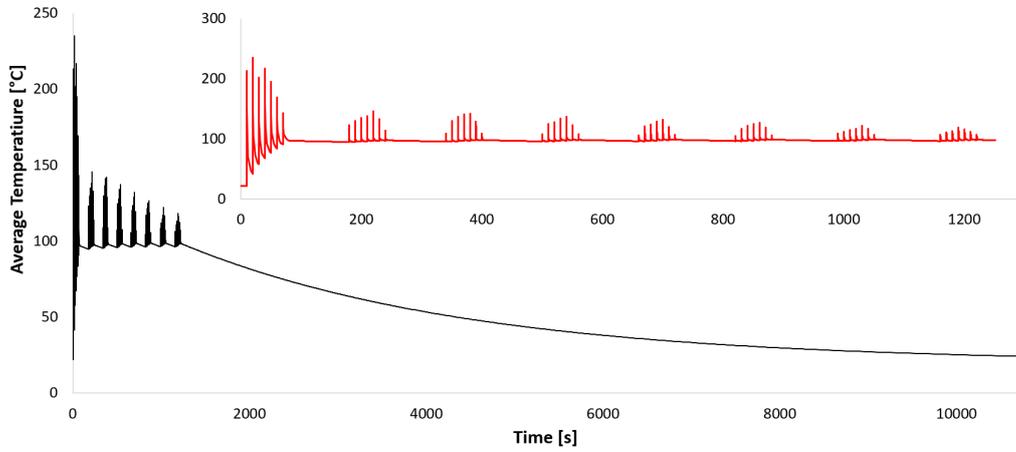


Figure 4.21: Round panel: Average temperature over time of the 8-layer model placed with Flash heating, shown in black, with a detailed view in red, Standard lay-up Strategy

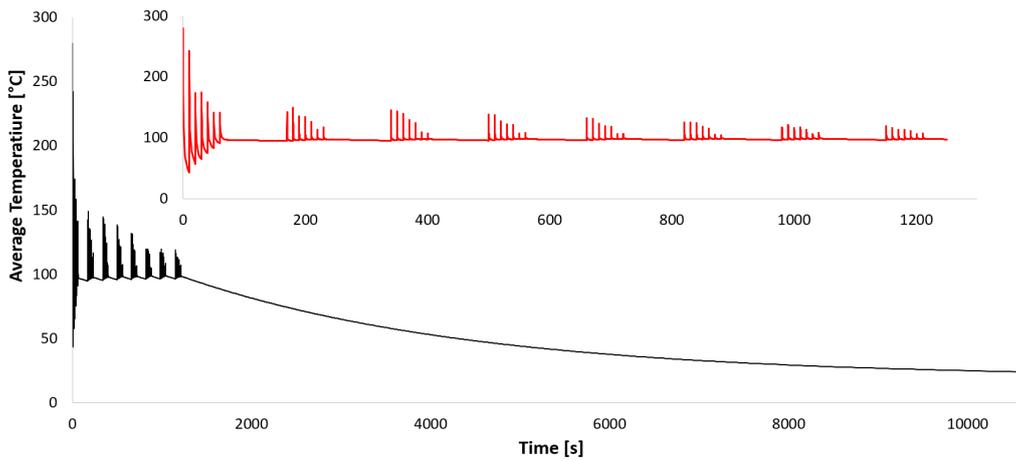


Figure 4.22: Round panel: Average temperature over time of the 8-layer model placed with Flash heating, shown in black, with a detailed view in red, Center lay-up Strategy

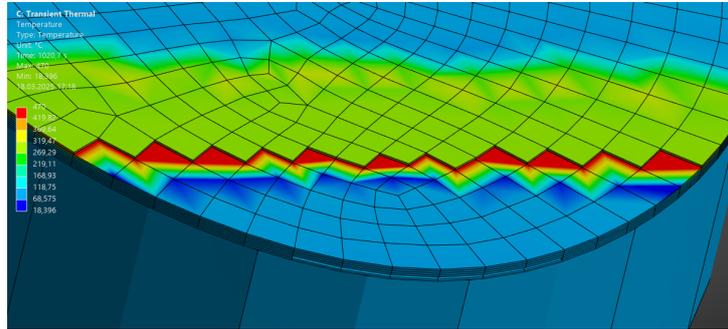


Figure 4.23: Heating step for a central tape of a round flat panel

Chapter 5

Conclusions

The data derived from the tests conducted in Chapter 4 demonstrated a strong correlation between the logic of the analysis and the simulation itself. The implemented path planning model, developed through Python code, enabled accurate trajectory tracking and proper tape modelling, ensuring adequate coverage of the entire tool surface, whether flat or square, and consequently, the laminate produced. Furthermore, the standard, centre, and out strategies were successfully implemented without errors in overlap or incorrect lay-up sequence. The extension of the intersection plane/mesh coordinates to the boundary had a positive impact, allowing for complete coverage, including the edges, without excluding any area of the tool geometry from the simulation. Regarding the Python code implementation, no processing or data export errors were encountered, and the entire process proceeded smoothly and without interruptions.

In terms of the transient thermal analysis in ANSYS, the approach developed for selecting the tape elements, modelled as "boxes", showed proper handling of the change in reference systems at the segment/tape starting points and the variation in the orientation angles of the paths, with correct transformation and translation. This allowed the various lay-up angles in the defined sequence, as well as the associated variation steps, to be followed accurately. The laminate elements, initially deactivated, were correctly selected during the heating stages, enabling the simulation of the thermoplastic matrix melting process, both for continuous laminate application and for segments. In addition to the heating management, the other boundary conditions (BCs) were applied without errors or delays, ensuring that the simulation accurately replicated the real T-AFP process. The average temperatures recorded at the end of the cooldown process and their development throughout the entire simulation followed the material's thermal history in a linear manner, as expected from the AFP machine in all the cases examined.

All the initial requirements proposed at the start of the work have been met, thus demonstrating the versatility of trajectory tracking, the increase in automation, and the reduction in processing times. Automation has been achieved by inputting the process parameters exclusively into the Python file, eliminating the need for redundancies in the supporting APLD code or the ANSYS environment. This approach has been successfully implemented, improving the process efficiency, particularly in the extraction of coordinates and the generation of CSV files containing data related to times, information, and details of the tape elements. Compared to the previous simulation, the processing speed in ANSYS has been increased by supplying data externally, avoiding the system from working solely with internal data. This has made the simulation process faster and more efficient, optimising both time and computational resources. These improvements not only fully meet all the initial requirements but also contribute to strengthening the overall effectiveness of the model, making the workflow smoother and broadening the applicability of the simulation in real-world processes.

5.1 Future Work

This section outlines several recommendations for future research aimed at further improving the accuracy of the proposed method.

Firstly, the path planning strategy could be improved by expanding the possibility of adapting it for curved panels. A preliminary approach was presented in Section 2 of Chapter 4, but further studies are necessary to establish a constant offset across the entire surface. Additionally, since this part of the code is written in Python, secondary open-source libraries could be incorporated to support new lay-up strategies, new functionalities, or new output requirements.

Another possible direction for improvement could involve the use of artificial intelligence (AI) to increase the automation of subprocesses and enhance the overall simulation, optimising both processing time and result output dataframes. Furthermore, AI could potentially be used to calculate the correct extension of the trajectories without relying on the fixed parameters typically used as reference for angles close to 90° in this work.

An important addition to this analysis could also be the integration of data acquisition from current AFP machine monitoring software, in order to create a database aimed at optimising the process with a view towards achieving complete coverage of the surface areas.

Alternatively, the same trajectory tracking methodology could be expanded to work directly on 3D models, eliminating the need to control the currently

deposited layer and thereby simplifying the code. These improvements would also likely lead to a reduction in the potential for errors or code crashes.

Currently, a transient thermal analysis is conducted, where structural checks are not implemented. However, the setup of the simulation in ANSYS, related to the element selection, could easily allow for the incorporation of structural checks, which could either replace or supplement the thermal analysis, without requiring overly complicated operations.

Lastly, further developments could focus on improving the accuracy of thermal and structural simulations by refining the coupling between these two domains, leading to more realistic predictions of material behaviour under AFP processing conditions. For example, a real-time laser model or another heat source could be developed in Python to simulate more realistic temporal and temperature profiles for each heating step, or even implement a secondary heat source for more complex heat transfer modelling.

Chapter 6

Data Appendix

Density [Kg mm ⁻³]	1800
Isotropic thermal conductivity [W m ⁻¹ C ⁻¹]	4.57
Specific heat at constant pressure, C_p [J Kg ⁻¹ C ⁻¹]	997.8

Table 6.1: Material parameters for CF/LM-PAEK

DTIME	heating_time/2
DMIN	heating_time/100
DMAX 1111	heating_time
DMAX 2222	robot_time/2
DMAX 3333	layer_time/2
DMAX 4444	cooldown_time/2

Table 6.2: Different Time Steps in Seconds

Laminate Size [mm]	100 × 100
Layers Number	8
Laminate Thickness [mm]	1.584
Layup	[90° / - 45° / 0° / 45°] _{2s}

Table 6.3: Laminate Properties: Square Panel

Maximum Diameter [mm]	90
Maximum Radius [mm]	55
Minimum Radius [mm]	45
Layers Number	8
Laminate Thickness [mm]	1.584
Layup	$[90^\circ / -45^\circ / 0^\circ / 45^\circ]_{2s}$

Table 6.4: Laminate Properties: Round Panel

Tape Thickness [mm]	0.198
Tape Width [mm]	12.7
Segment Length [mm]	20

Table 6.5: Tapes Parameters

Heating Time [s]	0.014
Robot Time [s]	10
Layer Time [s]	100
Cooldown Time [s]	10000

Table 6.6: Different Time Steps in Seconds

Pressure Roll [Atm]	600000
Laser Thickness [mm]	0.001
Process Speed [$\frac{mm}{s}$]	72
Process Temperature [°C]	470
Tool Temperature [°C]	20
Cooldown Temperature [°C]	20
Tool Heat Transfer Coefficient [W, m^{-2}, K^{-1}]	9.363
Air Heat Transfer Coefficient [W, m^{-2}, K^{-1}]	10

Table 6.7: Process Parameters

Bibliography

- [1] URL: <https://docs.python.org/3/library/os.html>.
- [2] URL: <https://numpy.org/>.
- [3] URL: <https://pandas.pydata.org/docs/index.html#>.
- [4] URL: <https://trimesh.org/>.
- [5] URL: https://matplotlib.org/stable/api/pyplot_summary.html.
- [6] URL: https://matplotlib.org/stable/api/_as_gen/mpl_toolkits.mplot3d.art3d.Poly3DCollection.html.
- [7] URL: <https://docs.python.org/3/library/itertools.html#itertools.combinations>.
- [8] URL: <https://www.ansys.com/blog/what-is-apdl#:~:text=By%20inserting%20a%20specific%20sequence%20of%20APDL%20instructions%2C%204%20Complex%20math%20operations%20%20Custom%20post%20processing>.
- [9] M.M. Abdalla and Z. Gürdal. “Optimization of course locations in fiber-placed panels for general fiber angle distributions”. In: *Compos. Sci. Technol.* 70.4 (2010), pp. 564–570. DOI: [10.1016/j.compscitech.2009.12.003](https://doi.org/10.1016/j.compscitech.2009.12.003). URL: <https://doi.org/10.1016/j.compscitech.2009.12.003>.
- [10] Mazen A. Albazzan et al. “Efficient design optimization of nonconventional laminated composites using lamination parameters: A state of the art”. In: *Composite Structures* 209 (2019), pp. 362–374. ISSN: 0263-8223. DOI: <https://doi.org/10.1016/j.compstruct.2018.10.095>. URL: <https://www.sciencedirect.com/science/article/pii/S0263822318321664>.
- [11] Inc. ANSYS. *Ansys Mechanical APDL Command Reference*. 2025. URL: <https://www.ansys.com/>.

- [12] M. Assadi and T. Field. *AFP processing of dry fiber carbon materials (DFP) for improved rates and reliability*. Tech. rep. SAE Technical Paper, 2020. DOI: [10.4271/2020-01-0030](https://doi.org/10.4271/2020-01-0030). URL: <https://doi.org/10.4271/2020-01-0030>.
- [13] N. Bakhshi and M. Hojjati. “Effect of compaction roller on layup quality and defects formation in automated fiber placement”. In: *Journal of Reinforced Plastics and Composites* 39.1-2 (2019), pp. 3–20. DOI: [10.1177/0731684419868845](https://doi.org/10.1177/0731684419868845).
- [14] Adriana W. Blom et al. “Fiber path definitions for elastically tailored conical shells”. In: *Composites Part B: Engineering* 40.1 (2009), pp. 77–84. ISSN: 1359-8368. DOI: <https://doi.org/10.1016/j.compositesb.2008.03.011>. URL: <https://www.sciencedirect.com/science/article/pii/S1359836808001029>.
- [15] Alex Brasington et al. “Automated fiber placement: A review of history, current technologies, and future paths forward”. In: *Composites Part C: Open Access* 6 (2021), p. 100182. ISSN: 2666-6820. DOI: <https://doi.org/10.1016/j.jcomc.2021.100182>. URL: <https://www.sciencedirect.com/science/article/pii/S2666682021000773>.
- [16] Michaël Bruyneel and Samih Zein. “A modified Fast Marching Method for defining fiber placement trajectories over meshes”. In: *Computers Structures* 125 (2013), pp. 45–52. ISSN: 0045-7949. DOI: <https://doi.org/10.1016/j.compstruc.2013.04.015>. URL: <https://www.sciencedirect.com/science/article/pii/S0045794913001454>.
- [17] Stefan Carosella et al. “A short review on recent advances in automated fiber placement and filament winding technologies”. In: *Composites Part B: Engineering* 287 (2024), p. 111843. ISSN: 1359-8368. DOI: <https://doi.org/10.1016/j.compositesb.2024.111843>. URL: <https://www.sciencedirect.com/science/article/pii/S1359836824006553>.
- [18] H. Chen, T. Fuhlbrigge, and X. Li. “A review of CAD-based robot path planning for spray painting”. In: *Industrial Robot, An International Journal* 36.1 (2009), pp. 45–50.
- [19] Q. Chu et al. “Placeability restricted by incomplete contact between laying roller and mould in an automated fiber placement process”. In: *Journal of Reinforced Plastics and Composites* 37.7 (2018), pp. 475–489. DOI: [10.1177/0731684417752871](https://doi.org/10.1177/0731684417752871).

- [20] CompositesWorld. “Consolidating Thermoplastic Composite Aerostructures In-Place, Part 1”. In: *CompositesWorld* (2025). URL: <https://www.compositesworld.com/articles/consolidating-thermoplastic-composite-aerostructures-in-place-part-1>.
- [21] R. Flynn, T. Rudberg, and J. Stamen. “Automated fiber placement machine developments: modular heads, tool point programming and volumetric compensation bring new flexibility in a scalable AFP cell”. In: *2011* ().
- [22] Daniel Fricke, Lukas Raps, and Ines Schiel. “Prediction of warping in thermoplastic AFP-manufactured laminates through simulation and experimentation”. In: *Advanced Manufacturing: Polymer & Composites Science* 7.1 (2021), pp. 89–101. DOI: [10.1080/20550340.2021.2015212](https://doi.org/10.1080/20550340.2021.2015212). URL: <https://doi.org/10.1080/20550340.2021.2015212>.
- [23] P. G. de Gennes. “Reptation of a polymer chain in the presence of fixed obstacles”. In: *The Journal of Chemical Physics* 55.2 (1971), pp. 572–579.
- [24] Wouter J.B. Groupe. “Weld Strength of Laser-Assisted Tape-Placed Thermoplastic Composites”. PhD Thesis. Faculty of Engineering Technology: University of Twente, 2015. URL: <https://research.utwente.nl/en/publications/weld-strength-of-laser-assisted-tape-placed-thermoplastic-composi>.
- [25] S.M. Grove. “Thermal modelling of tape laying with continuous carbon fibre-reinforced thermoplastic”. In: *Composites* 19.5 (1988), pp. 367–375. ISSN: 0010-4361. DOI: [https://doi.org/10.1016/0010-4361\(88\)90124-3](https://doi.org/10.1016/0010-4361(88)90124-3). URL: <https://www.sciencedirect.com/science/article/pii/0010436188901243>.
- [26] Z. Gürdal, B.F. Tatting, and K.C. Wu. “Tow-placement technology and fabrication issues for laminated composite structures”. In: *Proceedings of the Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 4*. 2005, pp. 2716–2733. DOI: [10.2514/6.2005-2017](https://doi.org/10.2514/6.2005-2017). URL: <https://doi.org/10.2514/6.2005-2017>.
- [27] R. Harik et al. *Automated fiber placement defect identity cards: cause, anticipation, existence, significance, and progression*. Accessed: Mar. 25, 2020. [Online]. Available: <https://www.researchgate.net/publication/326464139>. 2018.

- [28] Clément Hély, Lionel Birglen, and Wen-Fang Xie. “Feasibility study of robotic fibre placement on intersecting multi-axial revolution surfaces”. In: *Robotics and Computer-Integrated Manufacturing* 48 (2017), pp. 73–79. ISSN: 0736-5845. DOI: <https://doi.org/10.1016/j.rcim.2017.02.005>. URL: <https://www.sciencedirect.com/science/article/pii/S0736584516301661>.
- [29] B.C. Kim et al. “Limitations of fibre placement techniques for variable angle tow composites and their process-induced defects”. In: (2011).
- [30] Konrad Kozaczuk. “AUTOMATED FIBER PLACEMENT SYSTEMS OVERVIEW”. In: *Transactions of the Institute of Aviation* 245 (Dec. 2016), pp. 52–59. DOI: [10.5604/05096669.1226355](https://doi.org/10.5604/05096669.1226355).
- [31] M. A. Lamontia et al. “Manufacturing flat and cylindrical laminates and built up structure using automated thermoplastic tape laying, fiber placement, and filament winding”. In: *Sampe Journal* 39.2 (2003), pp. 30–43.
- [32] Lina Li et al. “An accurate Path Planning Algorithm based on triangular Meshes in robotic fibre Placement”. In: *Int. J. Robotics Autom.* 32 (2017). URL: <https://api.semanticscholar.org/CorpusID:3877312>.
- [33] Anis Limaiem and François Trochu. “Geometric algorithms for the intersection of curves and surfaces”. In: *Computers Graphics* 19.3 (1995), pp. 391–403. ISSN: 0097-8493. DOI: [https://doi.org/10.1016/0097-8493\(95\)00009-2](https://doi.org/10.1016/0097-8493(95)00009-2). URL: <https://www.sciencedirect.com/science/article/pii/0097849395000092>.
- [34] P.K. Mallick. *Processing of Polymer Matrix Composites*. 1st. CRC Press, 2017. DOI: [10.1201/9781315157252](https://doi.org/10.1201/9781315157252).
- [35] S. C. Mantell and G. S. Springer. “Manufacturing process models for thermoplastic composites”. In: *Journal of Composite Materials* 26.16 (1992), pp. 2348–2377.
- [36] M.N. Ghasemi Nejhad, R.D. Cope, and S.I. Güceri. “Thermal Analysis of in-situ Thermoplastic Composite Tape Laying”. In: *Journal of Thermoplastic Composite Materials* 4.1 (1991), pp. 3–24. DOI: [10.1177/089270579100400102](https://doi.org/10.1177/089270579100400102).
- [37] Nicholas Patrikalakis and Takashi Maekawa. *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer Science & Business Media, 2001.

- [38] Z. Qureshi et al. “In situ consolidation of thermoplastic prepreg tape using automated tape placement technology: Potential and possibilities”. In: *Compos. Part B Eng.* 66 (2014), pp. 255–267. DOI: [10.1016/j.compositesb.2014.05.025](https://doi.org/10.1016/j.compositesb.2014.05.025). URL: <https://doi.org/10.1016/j.compositesb.2014.05.025>.
- [39] L. Raps et al. “CF/LM-PAEK: Characterisation and sensitivity to critical process parameters for automated fibre placement”. In: *Composite Structures* 284 (2022), p. 115087. ISSN: 0263-8223. DOI: <https://doi.org/10.1016/j.compstruct.2021.115087>. URL: <https://www.sciencedirect.com/science/article/pii/S0263822321015026>.
- [40] D. Regiec and P. Vxaeji. “High-speed fibre placement on large complex structures”. In: *JEC Composites Magazine* 49 (Jan. 2012), pp. 41–43.
- [41] G. Rousseau et al. “Automated fiber placement path planning: a state-of-the-art review”. In: *Computers & Graphics* 16.2 (2019), pp. 172–203. DOI: [10.14733/cadaps.2019.172-203](https://doi.org/10.14733/cadaps.2019.172-203).
- [42] Nicholas Sharp and Keenan Crane. “You can find geodesic paths in triangle meshes by just flipping edges”. In: *ACM Trans. Graph.* 39.6 (Nov. 2020). ISSN: 0730-0301. DOI: [10.1145/3414685.3417839](https://doi.org/10.1145/3414685.3417839). URL: <https://doi.org/10.1145/3414685.3417839>.
- [43] Nicholas M. W. Sharp. *Potpourri3D*. Accessed: 2025-03-11. 2021. URL: <https://github.com/nmwsharp/potpourri3d.git>.
- [44] Bijan Shirinzadeh et al. “Trajectory generation for open-contoured structures in robotic fibre placement”. In: *Robotics and Computer-Integrated Manufacturing* 23.4 (2007), pp. 380–394. ISSN: 0736-5845. DOI: <https://doi.org/10.1016/j.rcim.2006.04.006>. URL: <https://www.sciencedirect.com/science/article/pii/S0736584506000408>.
- [45] F. O. Sonmez and H. T. Hahn. “Modeling of heat transfer and crystallization in thermoplastic composite tape placement process”. In: *Journal of Thermoplastic Composite Materials* 10.3 (1997), pp. 198–240.
- [46] Fazil O. Sonmez and Mustafa Akbulut. “Process optimization of tape placement for thermoplastic composites”. In: *Composites Part A: Applied Science and Manufacturing* 38.9 (2007), pp. 2013–2023. ISSN: 1359-835X. DOI: <https://doi.org/10.1016/j.compositesa.2007.05.003>. URL: <https://www.sciencedirect.com/science/article/pii/S1359835X07000814>.
- [47] J. Tierney. “Modeling of in situ strength development for the thermoplastic composite tow placement process”. In: *Journal of Composite Materials* 40.16 (2006), pp. 1487–1506.

- [48] K. Wang et al. “A general method of trajectory generation based on point-cloud structures in automatic fibre placement”. In: *Composite Structures* 314 (2023), p. 116976. DOI: [10.1016/j.compstruct.2023.116976](https://doi.org/10.1016/j.compstruct.2023.116976). URL: <https://doi.org/10.1016/j.compstruct.2023.116976>.
- [49] R. Wehbe et al. “Geometrical modeling of tow wrinkles in automated fiber placement”. In: *Composites Structures* 246 (2020), p. 112394. DOI: [10.1016/j.compstruct.2020.112394](https://doi.org/10.1016/j.compstruct.2020.112394).
- [50] Wang Xiaoping et al. “Uniform Coverage of Fibres over Open-contoured Freeform Structure Based on Arc-length Parameter”. In: *Chinese Journal of Aeronautics* 21.6 (2008), pp. 571–577. ISSN: 1000-9361. DOI: [https://doi.org/10.1016/S1000-9361\(08\)60176-4](https://doi.org/10.1016/S1000-9361(08)60176-4). URL: <https://www.sciencedirect.com/science/article/pii/S1000936108601764>.
- [51] Long Yan et al. “An accurate approach to roller path generation for robotic fibre placement of free-form surface composites”. In: *Robotics and Computer-Integrated Manufacturing* 30.3 (2014), pp. 277–286. ISSN: 0736-5845. DOI: <https://doi.org/10.1016/j.rcim.2013.10.007>. URL: <https://www.sciencedirect.com/science/article/pii/S0736584513000884>.
- [52] Samih Zein, Freddie Colsoul, and Michael Bruyneel. “A Novel Approach for Fiber Placement Trajectories and Fabric Draping in CAE”. In: *SAMPE Technical Conference*. Liège Science Park, Rue des Chasseurs-Ardennais 8, Angleur, B-4031, Belgium. Baltimore, USA: LMS Samtech, 2013, 6–9 May.
- [53] Xuesong Zhang, Yongjun Chen, and Junling Hu. “Recent advances in the development of aerospace materials”. In: *Progress in Aerospace Sciences* 97 (2018), pp. 22–34. ISSN: 0376-0421. DOI: <https://doi.org/10.1016/j.paerosci.2018.01.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0376042117301483>.