Internship Report

20-Week Professional Internship in
Hochschule für Technik Rapperswil, Switzerland

Liew Dar Win

Nanyang Technological University, Singapore

2019-06-21

Contents

List of Figures **2**

List of Tables **2**

1 Introduction **3**
- 1.1 Organization . . . 3

2 Acknowledgement **3**

3 Background **3**
- 3.1 Problem . . . 3
- 3.2 Polypropylene . . . 3
  - 3.2.1 Warpage . . . 4
  - 3.2.2 Adhesion . . . 4
- 3.3 Software . . . 4
  - 3.3.1 Digimat . . . 4
  - 3.3.2 Siemens NX . . . 4
  - 3.3.3 Simplify3D . . . 5

4 Experimental Research Process **6**
- 4.1 Polypropylene Filament Selection . . . 6
- 4.2 Polypropylene Material Extrusion . . . 6
- 4.3 Polypropylene Filament Extrusion . . . 6
- 4.4 Bed Adhesion Tests . . . 9
- 4.5 First Layer Adhesion Tests . . . 10
- 4.6 Temperature Tests . . . 10
- 4.7 Print Tests . . . 11

5 Simulation Research Process **14**
- 5.1 Digimat AM Study . . . 14
- 5.2 Acquisition of Polypropylene Thermomechanical Properties . . . 15
- 5.3 Warpage Compensation Workflow . . . 15
- 5.4 Mesh Sensitivity Analysis . . . 17
  - 5.4.1 Findings . . . 17

6 Crystallization Modelling **19**
- 6.1 Avrami Equation . . . 21
- 6.2 Nakamura Fitting . . . 23
- 6.3 Differential Scanning Calorimetry . . . 24

7 Conclusion **25**

8 References **26**

A Thermomechanical Properties of BC245MO Polypropylene **27**

B MATLAB Code for Avrami Fitting **30**

C MATLAB Code for Nakamura Fitting **31**

List of Figures

1 Heading and Cooling of Semi-Crystalline Polymers [3] 4
2 Thermal Moment due to Differential Cooling [3] 4
3 Software 5
4 Extrusion Facility 7
5 Components of an Extrusion System 7
6 Filament Feeder Mechanism 8
7 Noncircular Extrudant Cross Section 8
8 Laser Measurement Results of Extruded Filaments 8
9 Polypropylene Tape 9
10 Custom Adhesive 9
11 First Layer Adhesion Tests 10
12 Single Model Temperature Test 11
13 Print Results Top Row: Print Samples from first few tests. Bottom Row: Print Samples 1-5 from Figure 5. 13
14 Front View of Print Results from Figure 13 13
15 Front View of Print Results from Figure 13 13
16 Digimat Simulation Workflow 14
17 Digimat Workflow Visualized 15
18 Pressure, Volume, Temperature Diagram of BC245MO Polypropylene on Cadmould Database 15
19 Digimat Simulation Results: Warped PP Mesh, Colour Indicates Severity of Distortion 16
20 Digimat Simulation Results: Counter Warped PP .STL Model 16
21 Digimat Simulation Results: Mesh and Exported STL Model of Demobauteil V2 16
22 Illustrated Dimensions of Mesh Sensitivity Analysis 18
23 Top View of Simulation Mesh Results at 0.2, 0.25, 0.5, 1, 2 mm Resolutions 19
24 Coloured Superimposed Top Views on Critical Areas (Left: Rounded Edge, Right: Hole) . . . 19
25 Digimat Crystallization Interface 20
26 Linearized Plot and Avrami Fitting 22
27 Plot and Avrami Fitting 22
28 Nakamura Fitting Interface 24
29 Temperature History for two Nodes (Left: Node in a Middle Layer, Right: Node on a Top Layer) 25
30 Plot of BC245MO Polypropylene Thermomechanical Properties 29

List of Tables

1 Polypropylene Filament Comparison 6
2 Polypropylene Feedstock Comparison 6
3 Percentage Count of Samples within Tolerance 6
4 Optimized First Layer Parameters 9
5 Print Test Parameters (Highlighted numbers are deviations from Sample 1) 12
6 Print Test Results 12
7 Mesh Sensitivity Analysis Parameters 17
8 Mesh Sensitivity Analysis Details 17
9 Mesh Sensitivity Analysis Results 18
10 Values of Avrami Exponent n for Various Types of Nucleation and Growth Acts [4] 20
11 Thermomechanical Properties of BC245MO Polypropylene at $0MPa$ 27
12 Thermomechanical Properties of BC245MO Polypropylene 28

1 Introduction

This report presents the different aspects of a research project undertaken by the author as part of his 20-week industrial attachment programme in Hochschule für Technik Rapperswil (HSR), in Rapperswil-Jona, Switzerland. The project encompasses the characterization of shrinkage and warpage of printed semi-crystalline prototypes, specfically polypropylene, and also development of guidelines for warp-free production of 3D printed polypropylene parts, by means of experiments and simulations. The report also describes the various stages of the project leading up to the completion as well as the results obtained.

1.1 Organization

Hochschule für Technik Rapperswil is a technical university in Eastern Switzerland. Located next to Lake Zurich in Rapperswil, the university consists of 16 institutes that spans within the field of application-oriented research and development. The author is assigned to the Institute of Materials and Plastics Processing (IWK), headed by Professor Frank Ehrig, who is also supervising the author for this internship. The institute is involved in the research and development of materials, process technologies, joining technologies, and computer aided engineering for materials processing.

2 Acknowledgement

The author is deeply grateful to Hochschule für Technik Rapperswil for offering the students from Nanyang Technological University such a rare privilege to conduct research in Switzerland. It was truly an enriching experience to explore the research field whilst being immersed in a rich European culture and surrounded by an incredibly beautiful scenery.

The author would like to express his gratitude to his supervisor, Professor Frank Ehrig, for his guidance and input for the direction of the project, as well as his school supervisor, Professor Chen I-Ming for coordinating the formalities required for this internship.

The author is also thankful for the guidance of Wick Curdin and Brändli Hedlund Karin for their guidance and support throughout the duration of the project.

Finally, the author would like to thank all the staff in the institute who have assisted him in the various stages of research, such that he is able to complete the project smoothly.

3 Background

3.1 Problem

Fused Filament Fabrication (FFF) is an additive manufacturing technique that involves heating and extruding a polymer feedstock, which are commonly amorphous plastics such as ABS and PLA. However, some series production parts are not produced using these plastics and thus, functional tests on these prototypes cannot be accurately conducted due to a difference in material properties. These products are usually made out of semi-crystalline polymers such as polypropylene, but such polymers are notoriously difficult to print as they tend to have serious warping and shrinkage issues. In this project, the focus will be on optimization of printing polypropylene parts.

3.2 Polypropylene

Polypropylene (PP) is a thermoplastic polymer frequently used in various consumer and industrial products. Due to its high fatigue resistance, it is commonly used in living hinges, which are extremely thin pieces of plastics that bend repeatedly without breaking. Some other properties includes good chemical resistance, high relative toughness and good impact resistance.

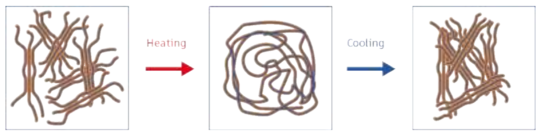


Figure 1: Heating and Cooling of Semi-Crystalline Polymers [3]

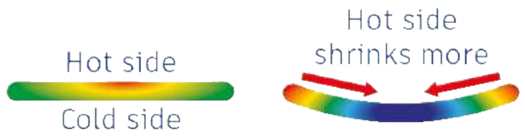


Figure 2: Thermal Moment due to Differential Cooling [3]

3.2.1 Warpage

When a semi-crystalline polymer cools, they shrink significantly as the loose molecules rearranges during recrystallization (Figure 1). Following the extrusion of a layer, shrinkage strains accumulate as it cools, causing a thermal moment formed around the edge of the part, which results in warpage (Figure 2). This effect is compounded when another layer is deposited and shrinks over the already cooled layer, which further increases the moment formed. The significant amount of shrinkage and warping causes severe part distortion and in the worst case, print failure, resulting from misaligned prints.

3.2.2 Adhesion

The low surface energy is an advantage when it comes to applications requiring a surface that is slippery and resistant to absorbing water. However, this poses an issue during printing as there is poor adhesion between the extruded filament and the bed. This inhibits accurate printing as it causes the printed part to shift in the z axis as the part lifts up due to the aforementioned warping.

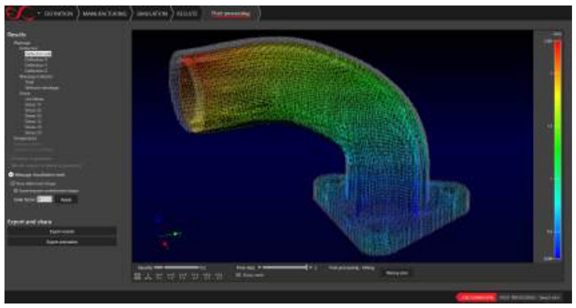
3.3 Software

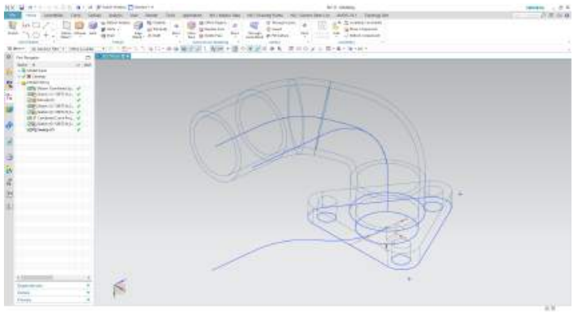
3.3.1 Digimat

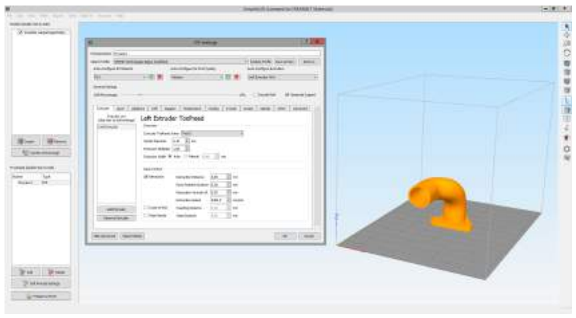
Digimat is a multiscale modelling program and one of its simulation chain, Digimat Additive Manufacturing (Digimat AM), allows for numerical simulation of additive manufacturing processes such as FFF, fused deposition modelling and selective laser sintering. It supports simulation of plastics, composites and metals. There are two main simulation approaches: one being a less computationally intensive multiscale workflow where a thermomechanical analysis is first performed to identify the warpage behavior of the material at a micro-level, which is described by inherent strain values. The values are then used in the second macro-level analysis, which involves a mechanical layer-by-layer finite element analysis. The other approach, simulates a transient workflow where portions of filament are incrementally activated for analysis, taking into account the printing deposition speed. The underlying mathematical model governing the polymer crystallization simulation is the Nakamura model, where the degree of crystallinity is a function of time and temperature. The degree affects the thermal contraction, which causes part distortion and ultimately shrinkage.[2]

3.3.2 Siemens NX

Siemens NX is a software that supports Computer Aided Design (CAD), Computer Aided Manufacturing and Engineering. In this project, only the CAD system is utilized for parametric modelling of test pieces.


(a) Digimat AM


(b) Siemens NX


(c) Simplify3D in Process Modification Mode

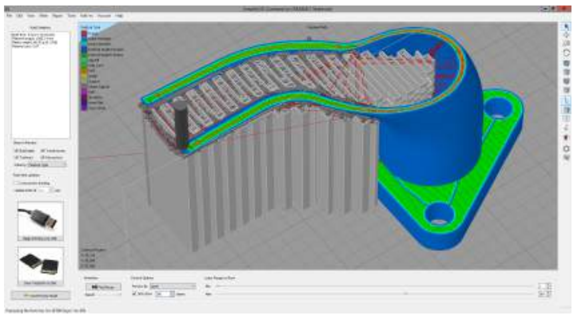

(d) Simplify3D in Process Simulation Mode

Figure 3: Software

3.3.3 Simplify3D

Simplify3D is a slicing software that converts 3D models to coded FFF instruction for the 3D printer. This specific software allows for the creation of processes with finely tuned parameters and offers features that would improve the print process, which are essential for printing complex parts.

4 Experimental Research Process

4.1 Polypropylene Filament Selection

One of the initial stages of the experimental project was to source for polypropylene filaments available on the market. However, during the research it was found that all of the products contained some form of additives that would minimize the warpage, which defeats the purpose of the project as the goal was to minimize warpage by means of adjusting process parameters and the model.

Table 1: Polypropylene Filament Comparison

Product	Filament Type	CHF/100g	Tensile Strength (MPa)	Flexural Modulus (MPa)
Verbatim PP Transparent	PP	8,08	14 (ISO 527)	350 (ISO 178)
3DWare.ch Handelsware White	PP	7,86		
Centaur PP – Natural	PP	6,47	12 (ASTM D638)	402 (ASTM D790)
Innofil3D PP	PP	6,38	8.5 (ISO 527)	2465.5±428.4 (ISO 178)
Smart Materials 3D	PP	4,71		
XSTRAND GF30-PP	PP + 30% GF	17,09	60 (ISO 527)	4300 (ISO 178)

4.2 Polypropylene Material Extrusion

Next, it was decided to source for the polypropylene material instead, as the institute will extrude the filament. The different polypropylene feedstocks were then sourced online and it was narrowed down to several choices with the lowest shrink rates. BC245MO was ultimately decided as the institute had this feedstock.

Table 2: Polypropylene Feedstock Comparison

Properties	BC245MO	BJ380MO	VG621	HF136MO
Melt Flow Rate (230 °C/2,16 kg)	3,5 g/10min	80 g/10min	5 g/10min	20 g/10min
Tensile Modulus	1.350 MPa	1.300 MPa	7.000 MPa	1.500 MPa
Shrinkage	1 - 2 %	1 - 2 %	Not stated	1 - 2 %
Specs avail in CMDB	Yes	No	No	Yes

4.3 Polypropylene Filament Extrusion

To create the filament, the BC245MO polypropylene pellets are extrusion moulded at the institute's manufacturing facility off-campus. In order to understand the filament manufacturing as well as the issues with extruding polypropylene, the author went to observe and assist Marcus, who was in-charge of the extrusion. For the sake of brevity, the extrusion process is briefly described. Instead, the difficulties and issues of producing polypropylene filament will be explained in detail.

Table 3: Percentage Count of Samples within Tolerance

Filament	% Samples Within Desired Tolerance			
	Axis 1 ⌀ ± 0.05 mm	Axis 2 ⌀ ± 0.05 mm	Mean ⌀ ± 0.05 mm	Roundness <0.1mm
2.85 mm	19.98262	21.80712	86.35969	20.41703
1.75 mm	46.98484	37.24605	38.37472	69.00997

The pellets are fed into the extrusion moulder (Figure 5a) where it is melted, extruded then cooled in a bed of water (Figure 5b). The dryer then rids the moisture and the extrudant is clamped by two belts

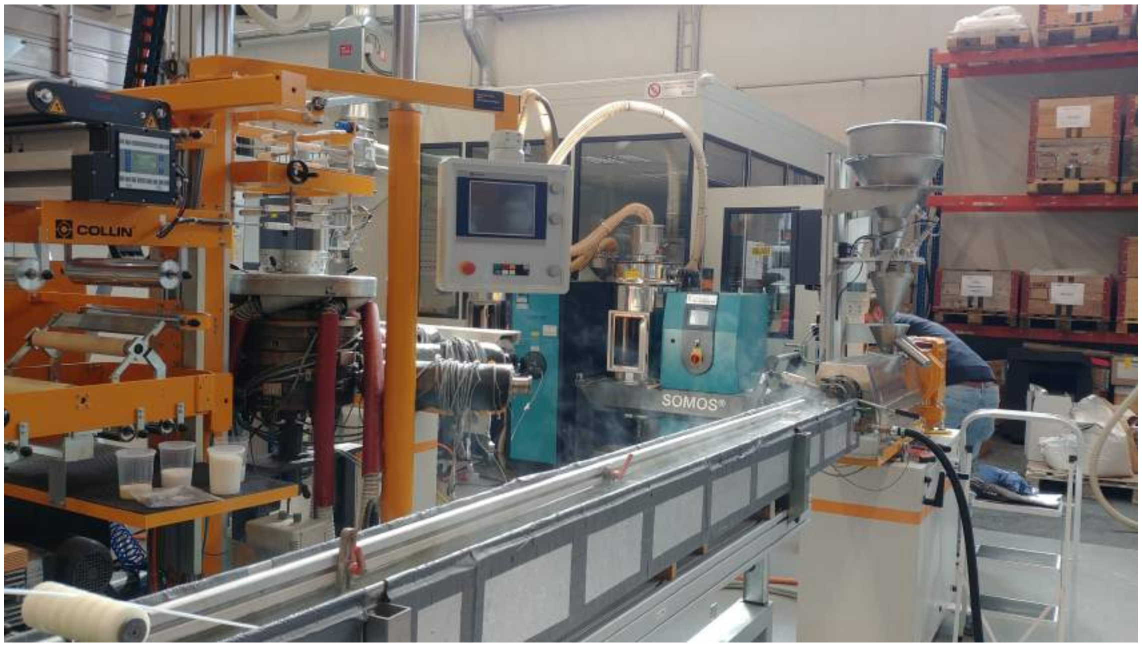
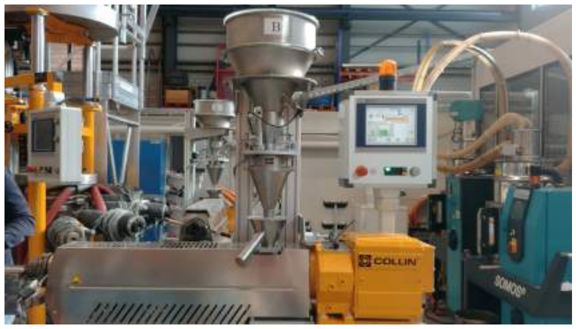
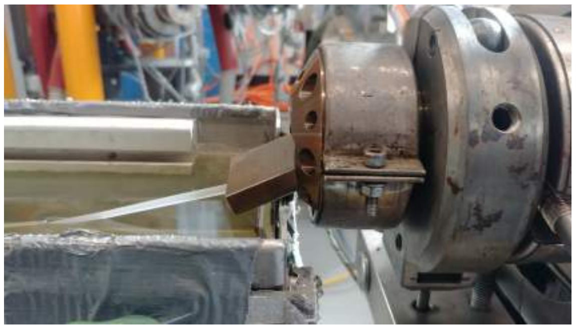


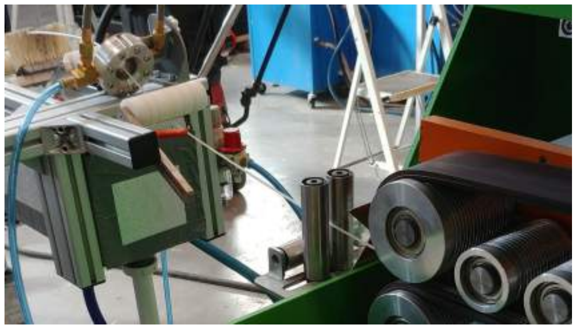
Figure 4: Extrusion Facility



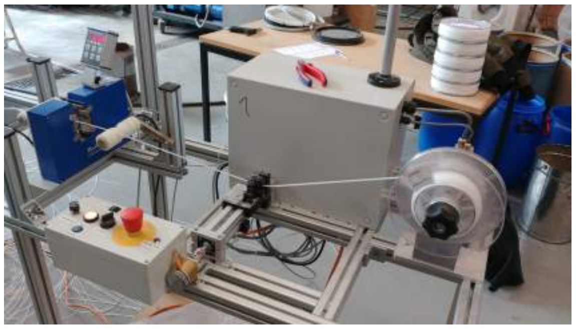
(a) Extrusion Moulder



(b) Cooling Bed



(c) Dryer and Profile Puller



(d) Laser Measuring Device and Spooler

Figure 5: Components of an Extrusion System

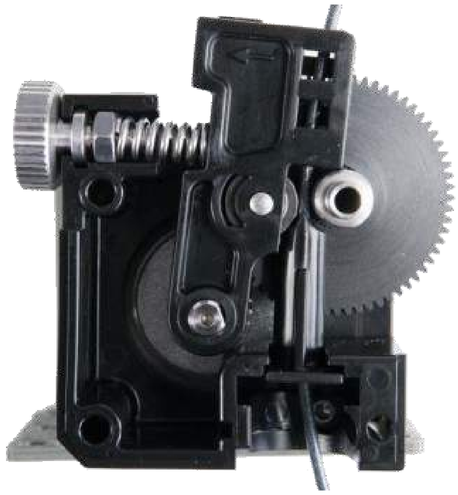


Figure 6: Filament Feeder Mechanism

Figure 7: Noncircular Extrudant Cross Section

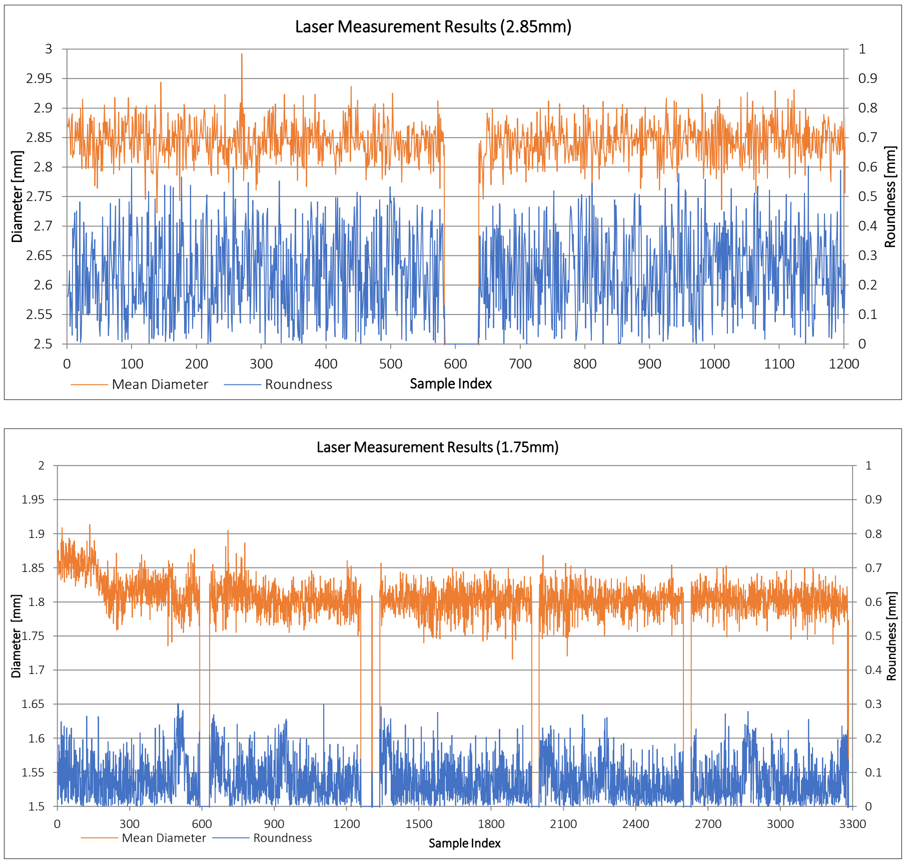


Figure 8: Laser Measurement Results of Extruded Filaments

and drawn by a profile puller (Figure 5c), where the drawing speed affects the diameter upstream. The laser measuring device measures the filament diameter from two axes, and it is finally spooled into a reel (Figure 5d).

In order to create a filament optimal for 3D printing, the tolerances for diameter and circularity have to be very small, as the filament feeder of a printer is designed for a fixed diameter, with a spring mechanism to allow for minute variations (Figure 6). If the filament is too thin, the gripper bolt might not be able to catch and drive the filament; conversely, too thick of a filament would cause jams. An inconsistent filament diameter would result in inconsistent volumetric flow rates at the nozzle end, ultimately affecting the print quality. There are numerous variables that affects the diameter and the circularity, some of the major factors are: extrusion die diameter, height of extrusion unit, angle of entry into cooling bed, cooling bed temperature, speed of the profile puller, etc. As the polypropylene pellets are designed for injection molding, it is not optimal for extrusion due to its dimensional instability. Coupled with the inhomogeneous cooling from the angled entry into the cooling bed (Figure 5b), an ellipsoidal filament section is formed (Figure 7), which definitely did not meet the required tolerances of $\pm 0.05mm$ (Figure 8). Thus, every mentioned variable of the process had to be finely tuned to attain the required diameter and tolerances. However, due to the physical limitations of the polypropylene material and the equipment, it was still not possible to achieve the desired properties after several hours of modifying the process and it was decided to accept the filament exceeding the desired tolerances (Table 3).

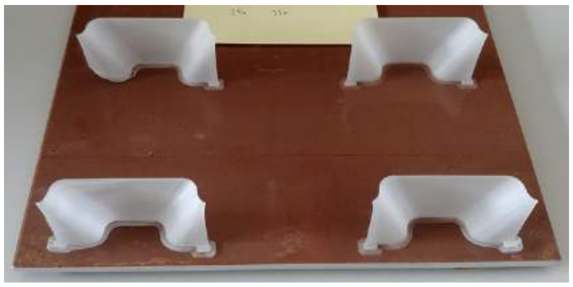


Figure 9: Polypropylene Tape

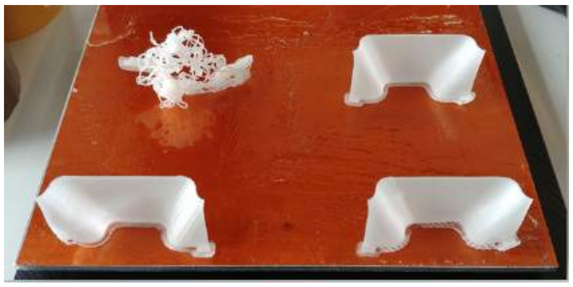


Figure 10: Custom Adhesive

4.4 Bed Adhesion Tests

In order to minimize warping during the print process, the cohesive forces between the bed and the adhesive must be higher than the shrinkage forces. A test was carried out to compare the adhesion between a glue and polypropylene tape. The glue is created by dissolving a soluble PP support material in water. The test specimen printed is a part with minimal contact to the unheated bed, such that the part will warp easily. From the results in Figures 9 and 10, it can be seen that PP tape is slightly better than the glue as one of the prints in the glue test failed due to severe warping in the initial layers. Thus, the tests will continue using PP tape as the bed adhesion method.

Table 4: Optimized First Layer Parameters

Parameters	Value
Height	75% of 0.2 mm
Width	75% of 0.6 mm
Speed	75% of 2700 mm/s
No. of Layers	2
Extrusion Temperature	240 C
Fanspeed	0%

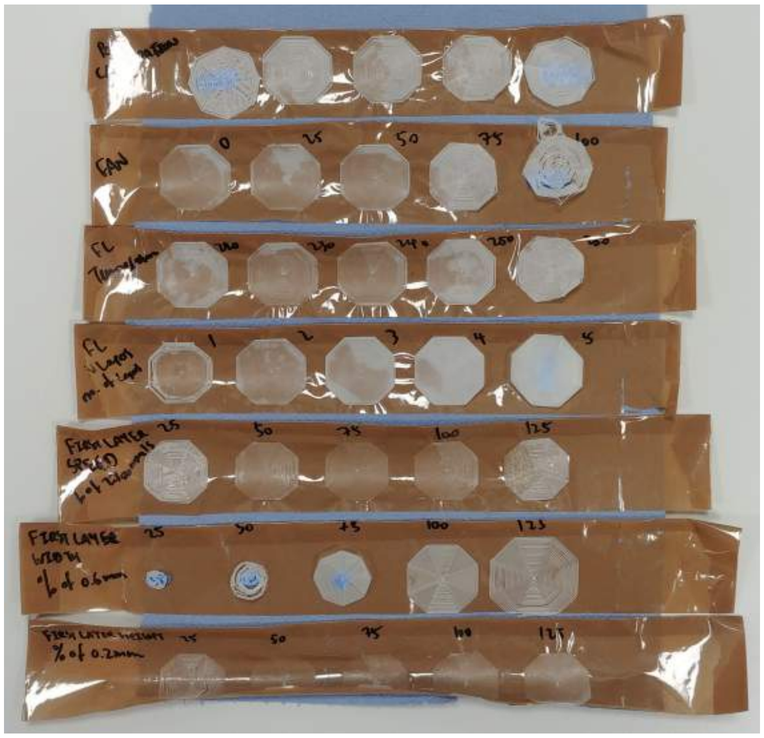


Figure 11: First Layer Adhesion Tests

4.5 First Layer Adhesion Tests

After achieving the right adhesion method, tests were conducted on the optimum settings for printing the first layer to achieve maximum cohesive forces. The extruded filament has to maximize contact with the bed adhesive, in this case the PP tape, and with its neighboring filaments. In the first test, one variable is tweaked while the others remain as default. The next test will then utilize the best setting from the previous test and another variable is tweaked. Such variables include layer width, layer height, printing speed, number of layers, temperature and fan speed. Finally, the most optimum variables (See Table 4) for the first layer is determined and will be used in further tests.

4.6 Temperature Tests

Temperature is a key parameter in the printing process. Too high of a temperature will cause the filament to be less viscous, which will result in drooping when the support is absent, or stringing as the filament refuses to seperate as intended. It also causes an increased warping as the part experiences a higher cooling rate for a prolonged period. If the temperature is too low, the layers might not fuse together properly, stringing might occur due to higher viscosity, and the filament might not extrude properly and will cause an uneven printing or even print failure. Thus, by printing a part with layers of different temperature, through determining the layer fusion and stringing, it is possible to determine an optimal temperature (See Figure 12).

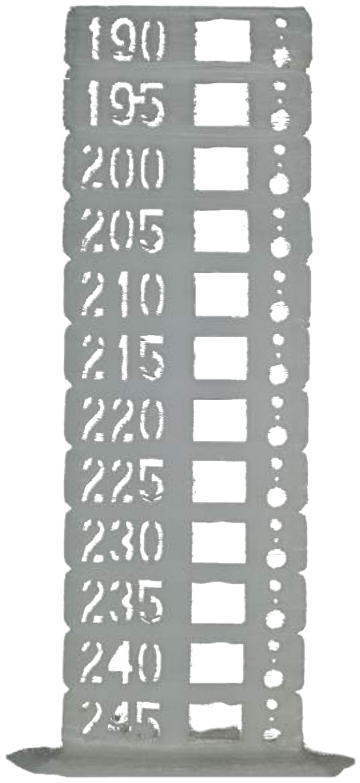


Figure 12: Single Model Temperature Test

4.7 Print Tests

Tests were then conducted by tuning various temperatures in order to achieve the best print quality and least distortion. The first sets of tests conducted yielded prints of poor quality and severe warping until when Karin and Florian provided parameters that were optimized for polypropylene. Thereafter, the prints experienced a significant improvement of quality and thus, the final set of tests with the best results are as shown.

Table 5: Print Test Parameters (Highlighted numbers are deviations from Sample 1)

Sample		1	2	3	4	5
Nozzle	Nozzle Diameter (mm)	0.6	0.6	0.6	0.6	0.6
	Extrusion Multiplier	1.1	1.1	1.1	1.1	1.1
	Print Width (mm)	0.6	0.6	0.6	0.6	0.6
Retraction	Distance (mm)	6	6	6	6	6
	Extra Restart Distance (mm)	0	0	0	0	0
	Veritcal Lift (mm)	0.11	0.11	0.11	0.11	0.11
	Speed (mm/s)	2400	2400	2400	2400	2400
Layer	Height	0.2	0.2	0.2	0.2	0.2
	Top Solid Layers	0	0	0	0	0
	Bottom Solid Layers	0	0	0	0	0
	Perimeter Outlines	1	1	1	1	1
Temperature	Extruder ($^{\circ}C$)	240	240	240	240	240
	Bed ($^{\circ}C$)	0	100	0	0	0
First Layer	Height (%)	100	100	100	100	100
	Width (%)	100	100	100	100	100
	Underspeed	0.67	0.67	0.67	0.67	0.67
Skirt	Layers	1	1	1	2	2
	Offset (mm)	0	0	0	0	0
	Outlines	5	5	5	25	40
Cooling	Layer 1 (%)	0	0	0	0	0
	Layer 2 (%)	50	50	50	50	50
	Layer 5 (%)	100	100	100	100	100
Bed	PP/PA Blended	Yes	Yes	No	No	No
	PP Tape	No	No	Yes	Yes	Yes
Speed	Default (mm/s)	2700	2700	2700	2700	2700
	Outline	1	1	1	1	1
	Solid Infill	0.85	0.85	0.85	0.85	0.85
	Support	0.9	0.9	0.9	0.9	0.9
	RapidXY (mm/s)	12000	12000	12000	12000	12000
	RapidZ (mm/s)	1002	1002	1002	1002	1002

Table 6: Print Test Results

Sample	Results
1	No stringing, small warpage at base
2	Severe warping at layer 8, causing printhead to catch on distorted part, print aborted
3	Severe warping mid-print, print aborted
4	No stringing, warpage slightly lower than sample 1
5	No stringing, least warping amongst all samples so far

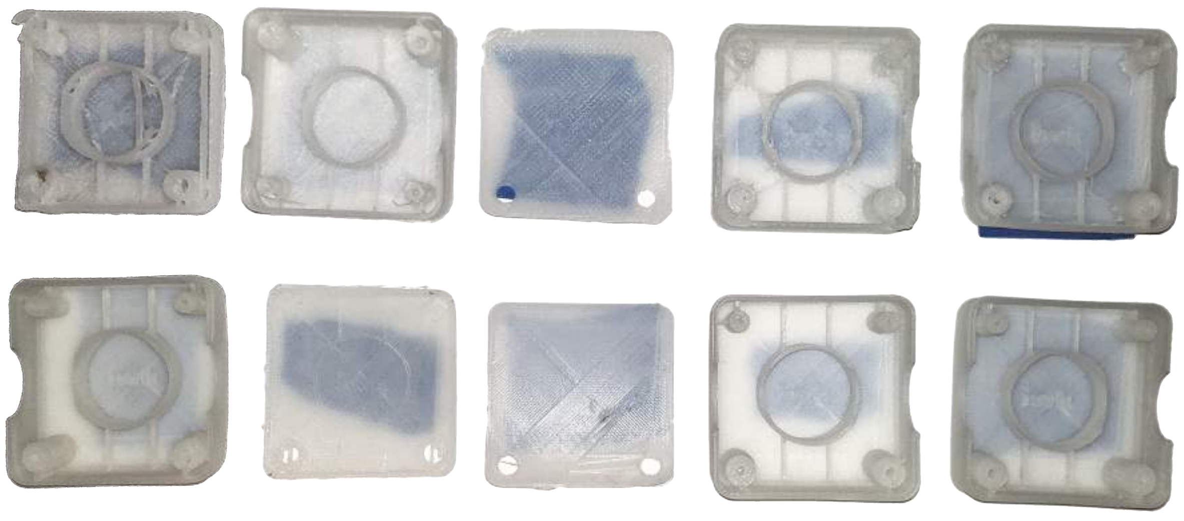


Figure 13: Print Results
Top Row: Print Samples from first few tests. Bottom Row: Print Samples 1-5 from Figure 5.

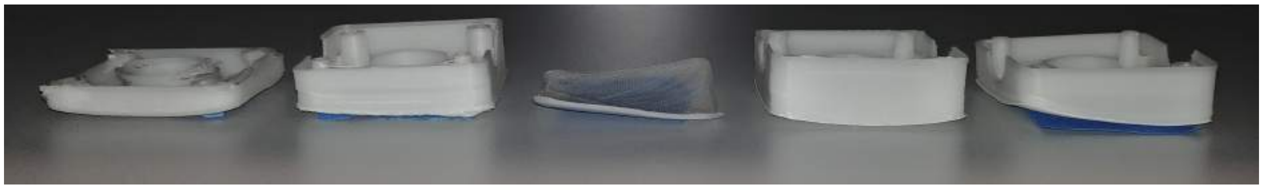


Figure 14: Front View of Print Results from Figure 13

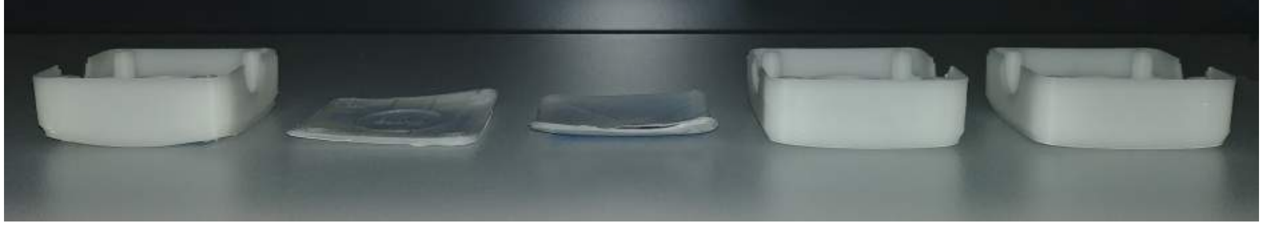


Figure 15: Front View of Print Results from Figure 13

5 Simulation Research Process

5.1 Digimat AM Study

To be able to utilise the software fully, the Digimat AM manual was first studied in detail and the relevant details were presented to the IWK staff. This consisted of the software capabilities, limitations, simulation stages, necessary files and data required for specific simulations. A summarized illustration of the Digimat Simulation Workflow is provided in Figures 16 and 17.

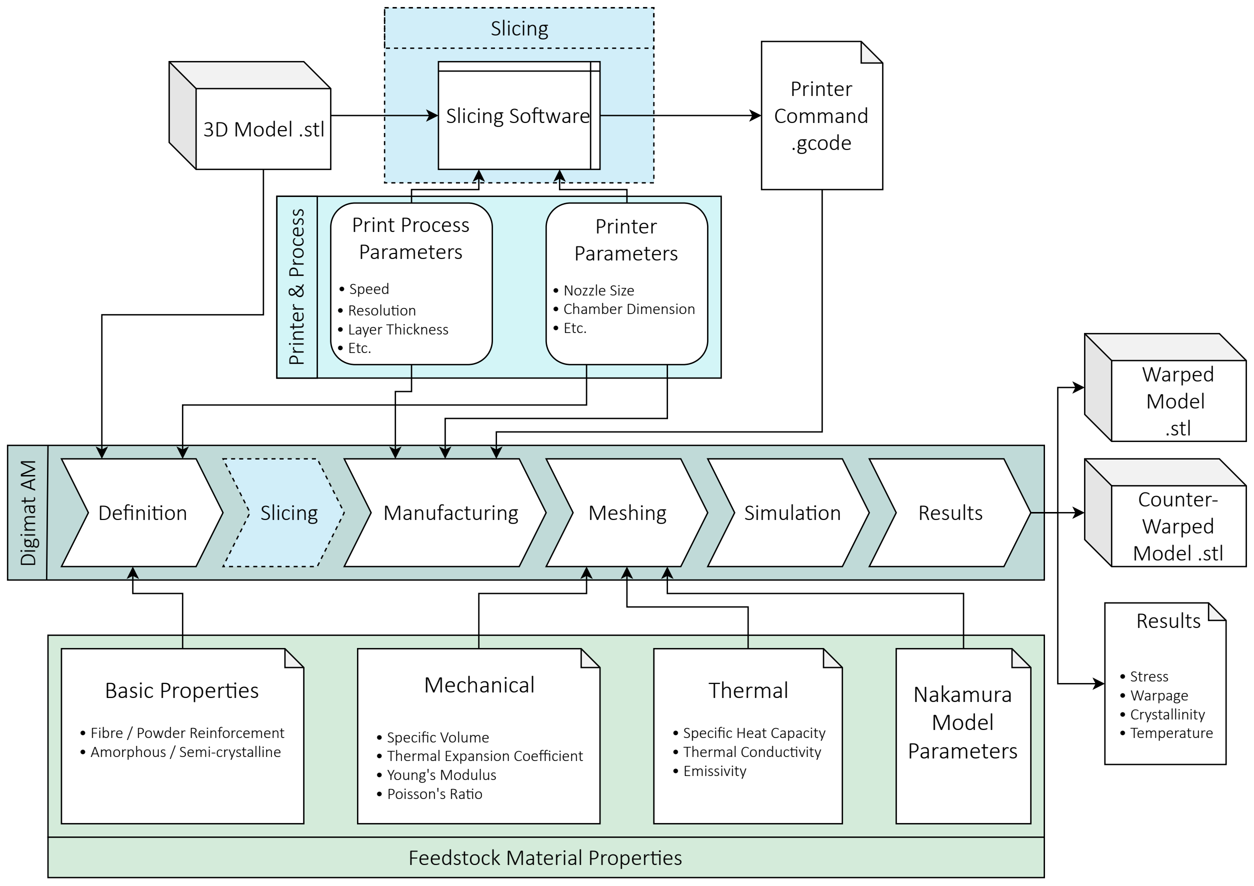


Figure 16: Digimat Simulation Workflow

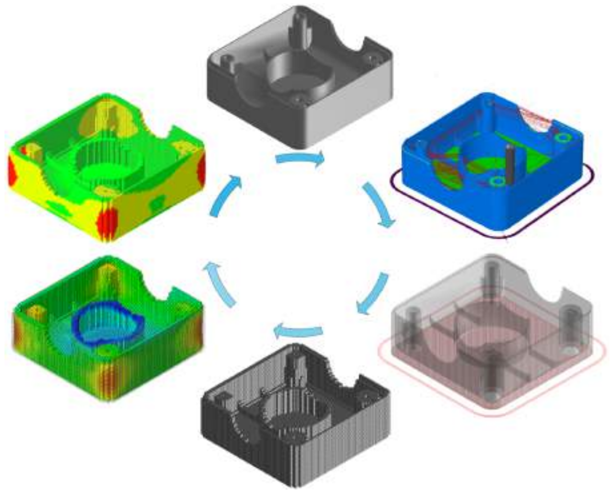


Figure 17: Digimat Workflow Visualized
Clockwise From Top: Definition, Slicing, Manufacturing, Meshing, Simulation, Results

5.2 Acquisition of Polypropylene Thermomechanical Properties

Prior to performing an FFF simulation on Digimat, the material properties, 3D model of the prototype and the gcode for the printer are required. Unfortunately, the Digimat material database does not include polypropylene. However, it does allow for the definition of new materials but extensive properties of the material are required, such as temperature dependent variables: specific volumes, thermal expansion coefficients, young's modulus; and constants such as poisson's ratio, thermal conductivity, emissivity, etc. Some of these properties were recorded and imported into the Cadmould Database (Figure 18) for moldflow simulation so it was extracted and used for the Digimat AM simulation. For those not available, they were either extrapolated or sourced online. A list of properties acquired is shown in Tables 11 and 12 in the annex.

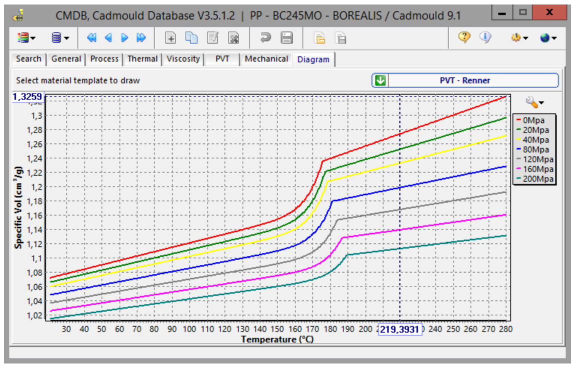


Figure 18: Pressure, Volume, Temperature Diagram of BC245MO Polypropylene on Cadmould Database

5.3 Warpage Compensation Workflow

One of Digimat AM's proposed offerings is the warpage compensation workflow which allows for the export of the counter-warped shaped resulting from the warping simulation. Ideally, the user is capable of circumventing the common additive manufacturing issue of warpage by printing a simulated counter-warped shaped. However, after numerous tests on this workflow, the software is deemed incapable of performing this workflow on non-simple geometries. Investigation of the different stages of the workflow yields a discovery of

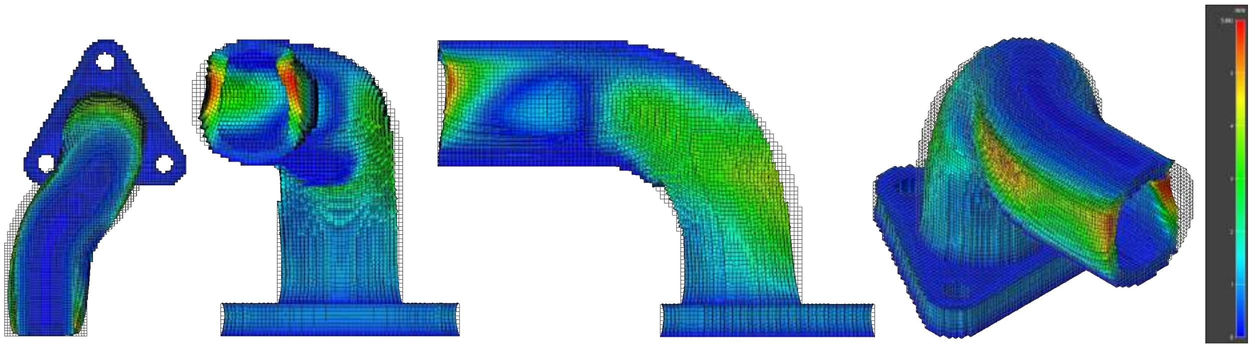


Figure 19: Digimat Simulation Results: Warped PP Mesh, Colour Indicates Severity of Distortion

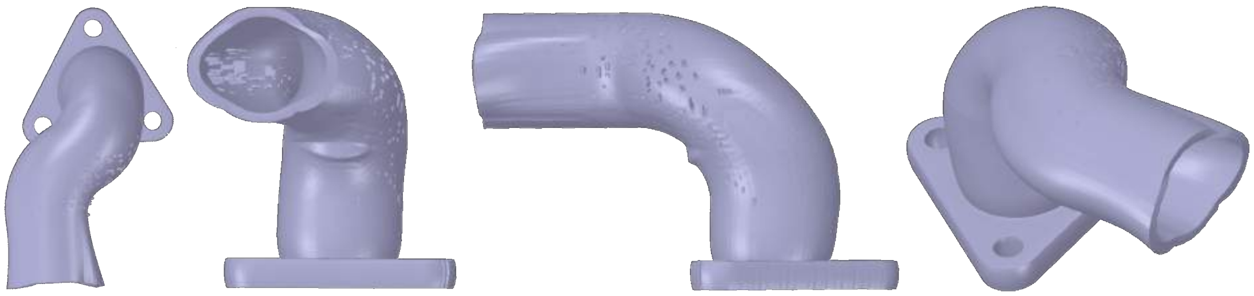


Figure 20: Digimat Simulation Results: Counter Warped PP .STL Model

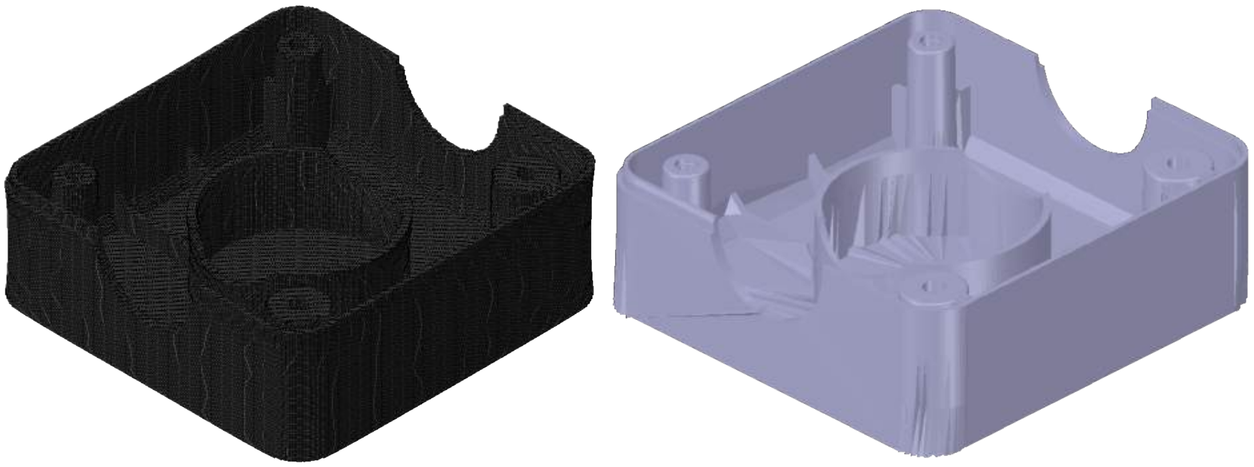


Figure 21: Digimat Simulation Results: Mesh and Exported STL Model of Demobauteil V2

a major bottleneck at the result export stage; the counter-warped .stl model is corrupted. This is caused by the software attempting to re-create the model using the polygonal mesh, which would be inaccurate given the loss in resolution (Figure 20). Re-meshing this counter-warped model would lead to a corrupt mesh and would eventually lead to inaccurate results. Despite attempts to mesh the model in its initial stage at the lowest possible resolution, the software is still incapable of exporting a smooth and printable .STL model. Figure 20 shows a rendering of the exported STL of a model (60 x 60 x 22 mm) at 0.2 mm voxel resolution. There are many unwanted faces present on the surface and this would ultimately result in a print failure.

5.4 Mesh Sensitivity Analysis

Due to the substantial computation intensity of this kind of simulations, a mesh sensitivity analysis is conducted to determine the resolution at which the results would converge. This allows for collection of the most accurate data while minimizing the computational resources required. For this analysis the simulations will be carried out with the fixed parameters found in Table 7, while the voxel resolutions will vary from 2mm down to 0.2mm (2, 1, 0.5, 0.25, 0.2 mm).

Table 7: Mesh Sensitivity Analysis Parameters

Manufacturing Process	FFF
Analysis Type	Warpage - Thermomechanical
Printer	Generic FFF Printer, 210 x 297 x 210 mm
Part	AM-Demobauteil V2 S3D Reoriented.STL
Material	e-X_GENERIC_FFF_ABS
Manufacturing Steps	Printing, Cooling, Support Removal
Warpage Compensation	NA
Max Refined Element Size	$66.127mm$
Toolpath	AM-Demobauteil V2.gcode
Chamber Temperature Type	Constant
Chamber Temperature	$25°C$
Extrusion Temperature	$245°C$
Heated Build Plate	Present
Build Plate Temperature	$100°C$
Bead Width	$0.5mm$
Convection Coefficient	$0.086mW/(mm^2 \cdot^\circ C)$

5.4.1 Findings

1. At 2 mm resolution, the hole is not meshed (Figure 23).
2. Simulations at 0.25mm and 0.2mm resolutions show no visible differences in results and computation timing, as seen in Figure 24, the cyan pixels representing the 0.25mm simulation are not visible.
3. Only the simulations running at the two finest resolutions is capable of achieving an accuracy better than just importing the original STL file. (See Table 8 and Figure 22)

Table 8: Mesh Sensitivity Analysis Details

Resolution (mm)	0.2	0.25	0.5	1	2
Voxel Count	1949245	1015050	136043	21504	3167
Simulation Time (Hours)	49	40	1.5	0.15	0.02

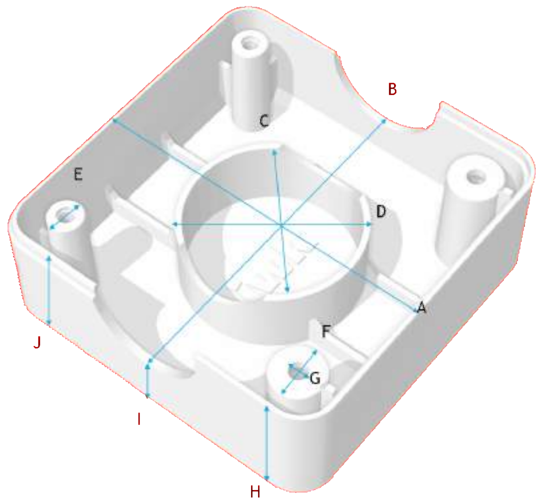


Figure 22: Illustrated Dimensions of Mesh Sensitivity Analysis

Table 9: Mesh Sensitivity Analysis Results

Measured Dimensions (mm)							
	STL	Actual Print	Simulation				
			0.2 mm	0.25 mm	0.5 mm	1 mm	2 mm
A	60.000	59.800	59.068	59.068	58.983	59.153	59.322
B	60.000	59.400	58.644	58.644	58.644	58.729	58.814
C	28.000	28.200	27.966	27.966	27.966	27.627	27.627
D	30.000	29.700	30.169	30.169	30.254	29.831	31.610
E	6.500	6.400	6.610	6.610	7.119	8.051	7.966
F	9.000	9.090	9.322	9.322	9.915	9.915	11.949
G	3.000	2.800	2.881	2.881	2.966	1.949	NA
H	21.500	20.500	21.265	21.265	21.402	21.905	21.951
I	10.000	9.700	10.015	10.015	10.015	9.970	11.982
J	20.000	19.100	19.939	19.939	19.893	19.848	19.893
Difference From Actual Print (%)							
A	0.334	0.000	-1.224	-1.224	-1.366	-1.083	-0.799
B	1.010	0.000	-1.273	-1.273	-1.273	-1.130	-0.987
C	-0.709	0.000	-0.829	-0.829	-0.829	-2.031	-2.031
D	1.010	0.000	1.581	1.581	1.866	0.439	6.432
E	1.562	0.000	3.284	3.284	11.229	25.794	24.470
F	-0.990	0.000	2.553	2.553	9.079	9.079	31.454
G	7.143	0.000	2.906	2.906	5.932	-30.387	NA
H	4.878	0.000	3.733	3.733	4.402	6.856	7.079
I	3.093	0.000	3.250	3.250	3.250	2.778	23.523
J	4.712	0.000	4.393	4.393	4.153	3.914	4.153
Avg	2.544	0.000	2.502	2.502	4.338	8.349	10.093

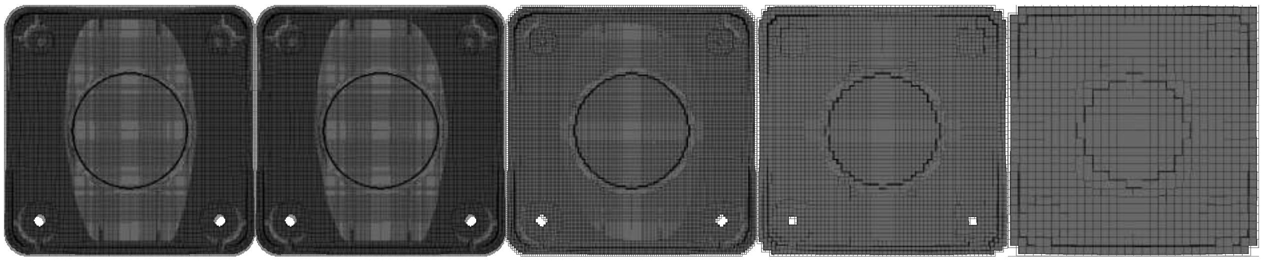


Figure 23: Top View of Simulation Mesh Results at 0.2, 0.25, 0.5, 1, 2 mm Resolutions

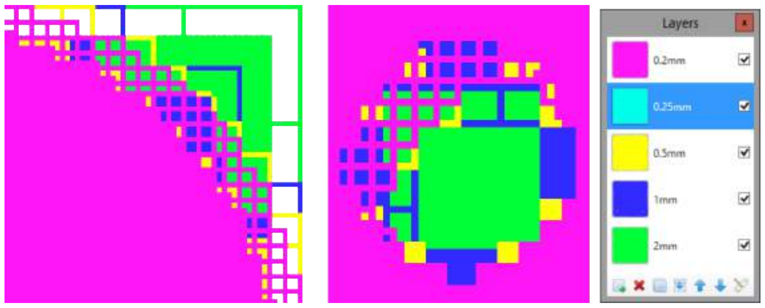


Figure 24: Coloured Superimposed Top Views on Critical Areas (Left: Rounded Edge, Right: Hole)

6 Crystallization Modelling

When a semi-crystalline polymer cools to a region between the melting temperature and glass transition temperature, the loose molecules rearrange to form chains (Figure 1) which fold into amorphous regions, creating large spheroidal structures called spherulites. These spherulites begin forming during the nucleation stage, the primary nuclei then experiences a fast spherulitic growth, and upon impingement to neighbouring spherulites the growth slows down and eventually stops. When simulating this type of polymers, it is imperative to account for the evolution of crystallinity as the part cools since this directly impacts the thermal contraction.

$$X(t) = 1 - \exp(-kt^n) \tag{1}$$

where:

X	= relative degree of crystallinity
k	= crystallization rate parameter
n	= avrami index
t	= time

The three different growth stages can be characterized into the Avrami equation (1), which is typically used to model the crystallization of a polymer, where it assumes that nucleation occurs randomly and homogeneously with a uniform omni-directional growth. The crystallization rate parameter k characterizes the nucleation and crystal growth whereas the avrami index n is dependent on the geometry and dimensionality of the growth shown in Table 10. However, it has to be noted that this equation only applies to quiescent isothermal crystallization. Thus, the Nakamura model, a modified form of the Avrami model, is utilised to perform an isokinetic non-isothermal approximation. [4]

$$X(t) = 1 - \exp\left(-\left(\int_0^t K'(T)d\tau\right)^n\right) \tag{2}$$

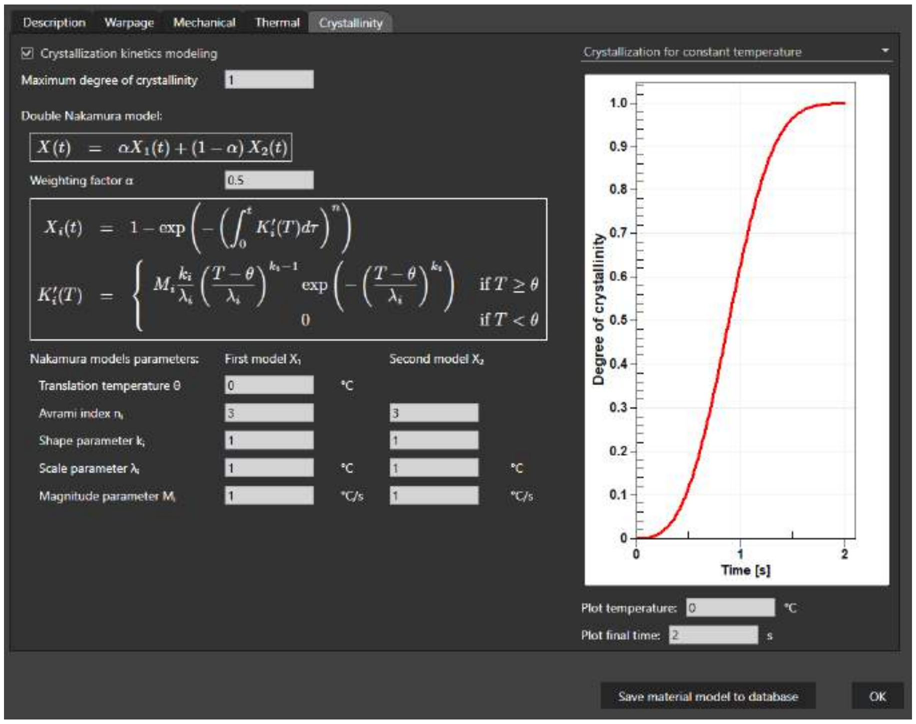
Table 10: Values of Avrami Exponent n for Various Types of Nucleation and Growth Acts [4]

Growth habit	Homogeneous Nucleation				Heterogeneous Nucleation
	Linear Growth		Diffusion Controlled Growth		Linear Growth
	Steady State	t=0	Steady State	t=0	
Sheaf-like	6	5	3.5	2.5	$5 \leq n \leq 6$
Three-dimensional	4	3	2.5	1.5	$3 \leq n \leq 4$
Two-dimensional	3	2	2	1	$2 \leq n \leq 3$
One-dimensional	2	1	1.5	0.5	$1 \leq n \leq 2$

$$K'(T) = \begin{cases} M\frac{k}{\lambda}\left(\frac{T-\theta}{\lambda}\right)^{k-1}\exp\left(-\left(\frac{T-\theta}{\lambda}\right)^{k}\right) & \text{if } T \geq \theta \\ 0 & \text{if } T < \theta \end{cases} \tag{3}$$

where:

K'(T)	= nakamura constant
n	= avrami index
θ	= translation temperature
k	= shape parameter
λ	= scale parameter
M	= magnitude parameter


Description Warpage Mechanical Thermal Crystallinity

☑ Crystallization kinetics modeling

Maximum degree of crystallinity: 1

Double Nakamura model:

$$X(t) = \alpha X_1(t) + (1-\alpha)\, X_2(t)$$

Weighting factor α: 0.5

$$X_i(t) = 1 - \exp\left(-\left(\int_0^t K_i'(T)d\tau\right)^n\right)$$

$$K_i'(T) = \begin{cases} M_i\frac{k_i}{\lambda_i}\left(\frac{T-\theta}{\lambda_i}\right)^{k_i-1}\exp\left(-\left(\frac{T-\theta}{\lambda_i}\right)^{k_i}\right) & \text{if } T \geq \theta \\ 0 & \text{if } T < \theta \end{cases}$$

Nakamura models parameters:	First model X_1		Second model X_2	
Translation temperature θ	0	°C		
Avrami index n_i	3		3	
Shape parameter k_i	1		1	
Scale parameter λ_i	1	°C	1	°C
Magnitude parameter M_i	1	°C/s	1	°C/s

Crystallization for constant temperature

Plot temperature: 0 °C

Plot final time: 2 s

Figure 25: Digimat Crystallization Interface

The software provides the double Nakamura model to predict the relative degree of crystallinity during a transient simulation (Figure 25). This model has a higher fidelity and is more suitable for materials with specific crystallization kinetics such as PEEK which displays slow secondary crystallization. By setting the weighting parameter to 1, the double Nakamura model is simplified to the single Nakamura model.[2]

$$X(t) = \alpha X_1(t) + (1-\alpha)X_2(t) \tag{4}$$

$$X_i(t) = 1 - \exp\left(-\left(\int_0^t K_i'(T)d\tau\right)^n\right) \tag{5}$$

$$K_i'(T) = \begin{cases} M_i \frac{k_i}{\lambda_i}\left(\frac{T-\theta}{\lambda_i}\right)^{k_i-1} \exp\left(-\left(\frac{T-\theta}{\lambda_i}\right)^{k_i}\right) & \text{if } T \geq \theta \\ 0 & \text{if } T < \theta \end{cases} \tag{6}$$

where:

α = weighting parameter

6.1 Avrami Equation

To measure the rate of crystallization of a sample, a differential scanning calorimeter (DSC) is used to track the relative crystallization with respect to time. It works by measuring the amount of heat energy required to raise the temperature of a sample and reference with a defined heat capacity, then comparing the difference of heat flows to determine the phase transition of the sample. Prior to performing a DSC experiment, it is first determined if it is possible to recreate the model using the results from a DSC reading. Thus, an Avrami fitting is first performed on a sample polypropylene DSC reading retrieved from experiments conducted by Maziar D [1]. First, the isothermal crystallization at $t = 125°C$ is utilized to determine the crystallization rate parameter and avrami index. A curve obtained from the DSC reading is imported into MATLAB to extract the points (Extracted points in Figure 27). These points are then linearized in order for MATLAB to perform an exponential curve fitting, as this installation of MATLAB in use does include the fitting toolbox.

Rearranging equation (1) the linearized form is

$$\ln\left[-\ln\left(1 - X(t)\right)\right] = \ln k + n \ln t \tag{7}$$

For simplicity, the following terms are substituted:

$$X'(t) = \ln\left[-\ln\left(1 - X(t)\right)\right], K' = \ln k, t' = \ln t \tag{8}$$

This allows the formation of the matrix:

$$\begin{bmatrix} X'(t_1) \\ X'(t_2) \\ X'(t_3) \\ \dots \\ X'(t_n) \end{bmatrix} = \begin{bmatrix} 1 & t_1' \\ 1 & t_2' \\ 1 & t_3' \\ .. & \dots \\ 1 & t_n' \end{bmatrix} \begin{bmatrix} K' \\ n \end{bmatrix} \tag{9}$$

Which is represented as

$$\vec{\boldsymbol{A}} = \vec{\boldsymbol{B}}\vec{\boldsymbol{C}} \tag{10}$$

Thus, to find $\vec{\boldsymbol{A}}$, inserting the inverse of $\vec{\boldsymbol{B}}$ in both sides of the equation gives

$$\vec{\boldsymbol{C}} = \vec{\boldsymbol{B}}^{-1}\vec{\boldsymbol{A}} \tag{11}$$

Which allows the calling of ***backslash*** in MATLAB, as this function returns the least-squares solution to the system of equations and thus computing a regressed K' and n. These values are then substituted back into the Avrami Equation to plot the fitted curve (Figure 23). See Appendix for the MATLAB code.

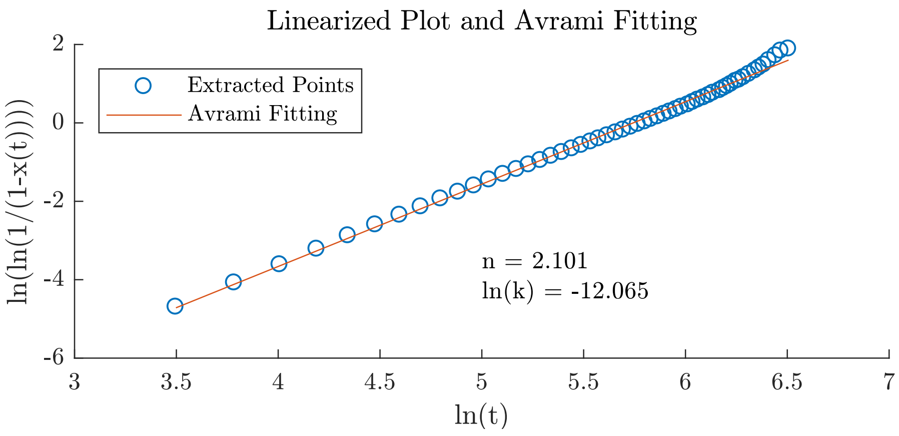


Figure 26: Linearized Plot and Avrami Fitting

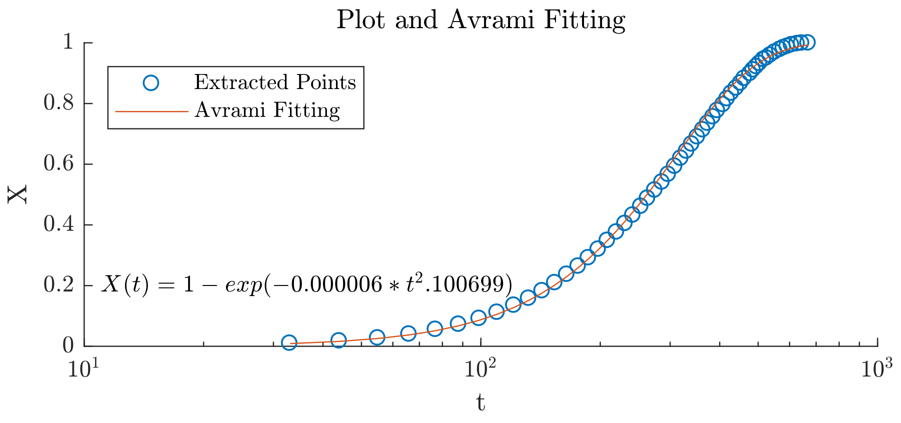


Figure 27: Plot and Avrami Fitting

6.2 Nakamura Fitting

After contacting the Digimat software support it was determined that an optimization tool has to be created to identify the weibull distribution parameters. Initially, it was planned to develop a script capable of performing regression to find the parameters with the best fit, by means of an iterative approach that cycles through all parameters. However, this script involved integrating multiple arrays of anonymous functions with limits defined by a vector of at least 1000 evenly spaced numbers, which was too computationally expensive to iterate. Thus, a script that generated a user interface to visually fit the model to the experimental results was created instead, which still required code optimization as MATLAB still took a considerable amount of time to update the plots. The optimization was ultimately disregarded due to time constraints. To create the script, the underlying mathematics of the model has to be understood:

The temperature function is first defined:

$$T(t) = a - b \cdot t \tag{12}$$

where:

T = temperature at time t
a = initial temperature
b = rate of temperature change

For clarity, the piecewise function in equation (3) is re-written to include the temperature function:

$$K'(T(t)) = \begin{cases} M\frac{k}{\lambda}\left(\frac{T(t)-\theta}{\lambda}\right)^{k-1}\exp\left(-\left(\frac{T(t)-\theta}{\lambda}\right)^{k}\right) & \text{if } T(t) \geq \theta \\ 0 & \text{if } T(t) < \theta \end{cases} \tag{13}$$

On MATLAB this can be represented in a single anonymous function. The reader is advised that in the following code snippets the functions and variables might be arbitrarily represented, such as $K'(T(t))$ being $ST.f$ and $X'(t)$ as $ST.Q$.

```
ST.f = @(t, m, k, theta, lamda, a, b) ((m * (k / lamda) * ...
(((ST.Tvar(t, a, b) - theta) / lamda) .^ (k - 1))) .*...
exp(- ((ST.Tvar(t, a, b) - theta) / lamda) .^ k)) .* (ST.Tvar(t, a, b) > theta);
```

The integral function from equation (2) is extracted:

$$X'(t) = \int_0^t K'(T(t))d\tau \tag{14}$$

This is expressed as two anonymous functions; the former outputing the results of integrating the aforementioned anonymous function with limits 0 and tau, and the latter outputting an array of the former function with tau being a vector of evenly spaced numbers spanning from zero to the time at which the line is required to be plotted.

```
ST.q = @(tau, m, k, theta, lamda, a, b)...
         (integral(@(t) ST.f(t, m, k, theta, lamda, a, b ), 0, tau));
ST.Q = @(tau, m, k, theta, lamda, a, b)...
         arrayfun(ST.q, tau, m, k, theta, lamda, a, b);
```

Finally, the Nakamura equation is defined, with reference from equation (2):

$$X(t) = 1 - \exp(-(X'(t))^n) \tag{15}$$

Which is simply represented in the following anonymous function:

```
ST.X = @(t, t2, m, k, theta, lamda, a, b)...
        (1 - exp(- (ST.Q(t, m, k, theta, lamda, a, b)) .^ t2));
```

Using a series of callback functions in the user interface control, it is possible to update the plot according to the values of the parameters in the sliders within the figure. (Figure 28) The full code is available in the Appendix for reference.

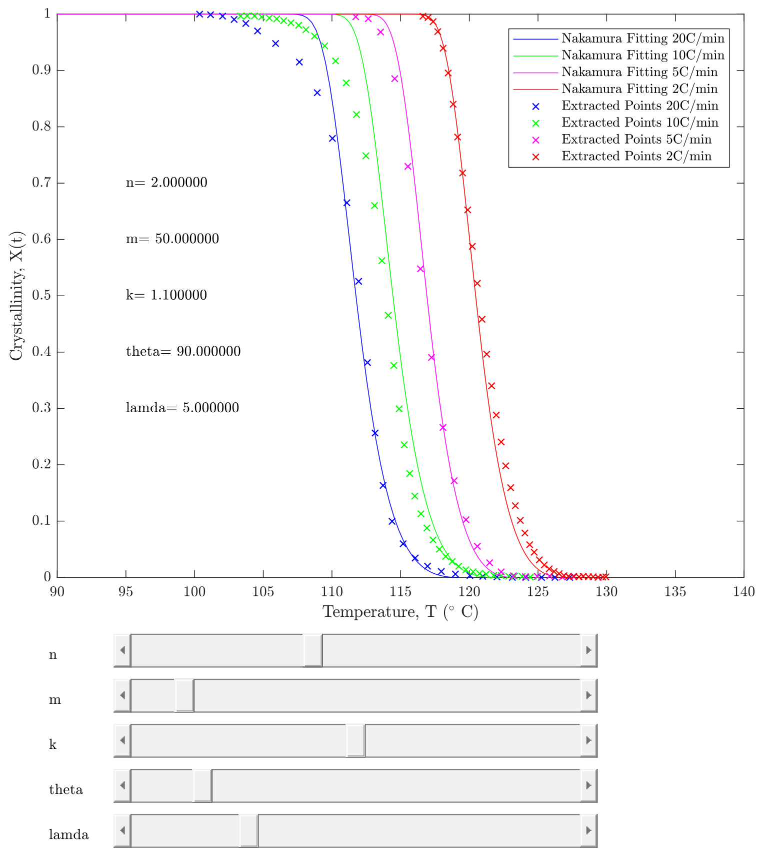


Figure 28: Nakamura Fitting Interface

6.3 Differential Scanning Calorimetry

As mentioned earlier, a DSC analysis would provide the necessary data for reconstructing the crystallisation model in simulation. In this case, there should be DSC tests conducted for isothermal and constant temperature-drop crystallisation. Another test should be conducted in order to determine the maximum absolute crystallization of the material.

For these tests to be most accurate, the tests have to be conducted close to the simulated rate of change of temperature during the print process. In the case for this material, the average and maximum temperature drop is approximated to be 300 & 1200$°C/min$ respectively (Figure 29), far exceeding the limit of the DSC instrument available in the campus of 10$°C/min$, thus a flash DSC instrument overseas has to be used as it supports for heating and cooling rates up to $30,000°C/min$. After discussion with Daniel, it was decided that due to time constraints, the DSC tests will not be conducted as the lead time was beyond the project's remaining duration.

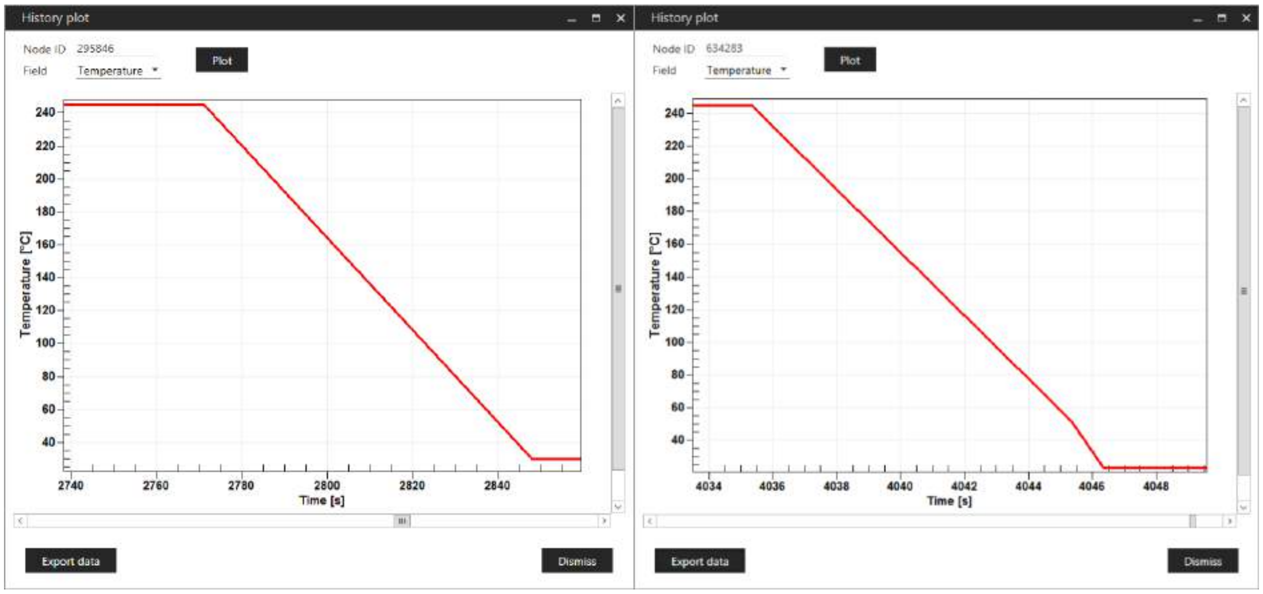


Figure 29: Temperature History for two Nodes (Left: Node in a Middle Layer, Right: Node on a Top Layer)

7 Conclusion

This project has explored the possibility of implementing the print simulation into a workflow to improve print quality. While the software is capable of predicting the part distortion, it is incapable of providing a model that compensates for the warping. Thus, the results can only be used as a reference to design a counterwarped model that will distort to its original shape upon printing.

Unfortunately, due to time constraints, the accurate simulation of polypropylene printing cannot be achieved as the crystallization modelling was not conducted. Nevertheless, the code for modelling the crystallisation was written and documented in the appendex should there be a need to extract the parameters accurately. However, the code does not perform fitting for the double Nakamura function and thus modification is required if necessary.

The optimal parameters for printing the first layer, the model has been determined and is documented in Tables 4 and 5 respectively. These parameters can be used as a guideline for future prints, and should be modified accordingly as some geometries might require adjusted parameters in order to print optimally.

8 References

[1] Maziar Derakhshandeh. *Crystallization of polypropylene: experiments and modeling.* PhD thesis, The University of British Columbia (Vancouver), nov 2015.

[2] e Xstream Engineering. *Digimat User's Manual.* MSC Software, Mont-Saint-Guibert, Belgium.

[3] Autodesk Inc. The causes of warpage, 2019.

[4] Rahul Patki, Khaled Mezghaniy, and Paul J. Phillips. *Physical Properties of Polymers Handbook, Crystallization Kinetics of Polymers*, chapter 39. Springer New York, 2007.

A Thermomechanical Properties of BC245MO Polypropylene

Table 11: Thermomechanical Properties of BC245MO Polypropylene at $0MPa$

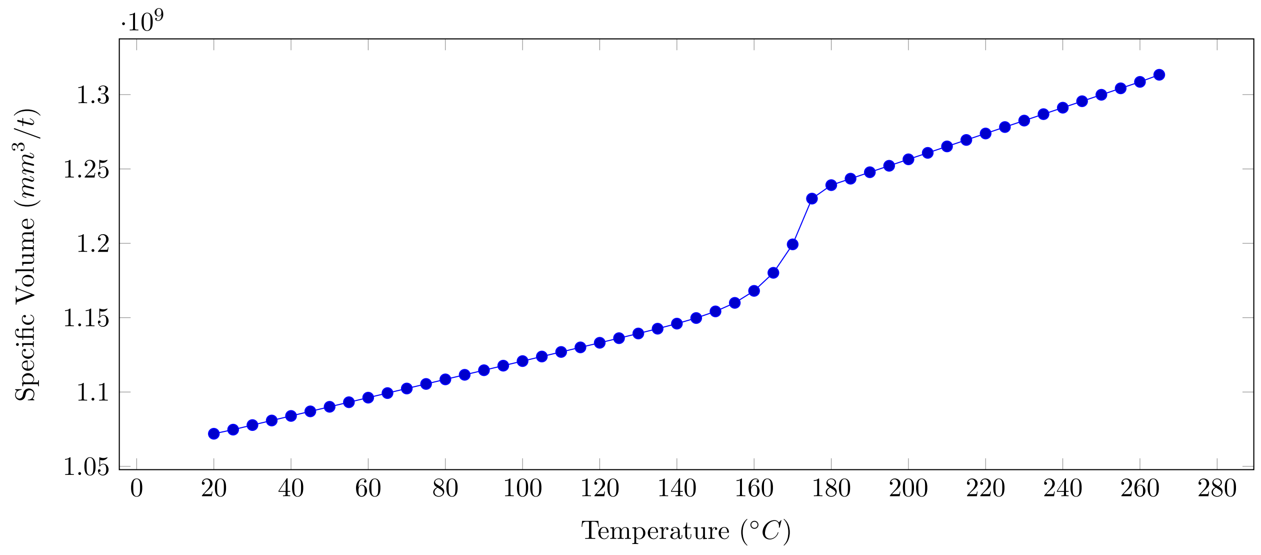
Temp ($^{\circ}C$)	Specific Volume (mm^3/t)	Coefficient of Thermal Expansion ($1/^{\circ}C$)	Young's Modulus (MPa)
20	1.0719E+09	1.7186E-04	1.37E+03
25	1.0747E+09	1.9045E-04	1.25E+03
30	1.0778E+09	1.8990E-04	1.14E+03
35	1.0808E+09	1.8936E-04	1.04E+03
40	1.0839E+09	1.8883E-04	9.52E+02
45	1.0870E+09	1.8829E-04	8.70E+02
50	1.0901E+09	1.8776E-04	7.95E+02
55	1.0931E+09	1.8724E-04	7.26E+02
60	1.0962E+09	1.8671E-04	6.63E+02
65	1.0993E+09	1.8619E-04	6.06E+02
70	1.1024E+09	1.8567E-04	5.53E+02
75	1.1054E+09	1.8516E-04	5.06E+02
80	1.1085E+09	1.8464E-04	4.62E+02
85	1.1116E+09	1.8413E-04	4.22E+02
90	1.1146E+09	1.8363E-04	3.85E+02
95	1.1177E+09	1.8313E-04	3.52E+02
100	1.1208E+09	1.8265E-04	3.22E+02
105	1.1239E+09	1.8222E-04	2.94E+02
110	1.1269E+09	1.8189E-04	2.68E+02
115	1.1300E+09	1.8183E-04	2.45E+02
120	1.1331E+09	1.8237E-04	2.24E+02
125	1.1362E+09	1.8421E-04	2.05E+02
130	1.1394E+09	1.8876E-04	1.87E+02
135	1.1426E+09	1.9869E-04	1.71E+02
140	1.1460E+09	2.1891E-04	1.56E+02
145	1.1498E+09	2.5811E-04	1.43E+02
150	1.1542E+09	3.3115E-04	1.30E+02
155	1.1600E+09	4.6250E-04	1.19E+02
160	1.1680E+09	6.9088E-04	1.09E+02
165	1.1802E+09	1.0746E-03	9.93E+01
170	1.1993E+09	1.6958E-03	9.07E+01
175	1.2301E+09	4.8992E-04	8.28E+01
180	1.2391E+09	2.3323E-04	Not Available
185	1.2435E+09	2.3241E-04	
190	1.2478E+09	2.3161E-04	
195	1.2521E+09	2.3080E-04	
200	1.2565E+09	2.3001E-04	
205	1.2608E+09	2.2922E-04	
210	1.2652E+09	2.2843E-04	
215	1.2695E+09	2.2765E-04	
220	1.2738E+09	2.2688E-04	
225	1.2782E+09	2.2611E-04	
230	1.2825E+09	2.2534E-04	
235	1.2869E+09	2.2458E-04	
240	1.2912E+09	2.2383E-04	

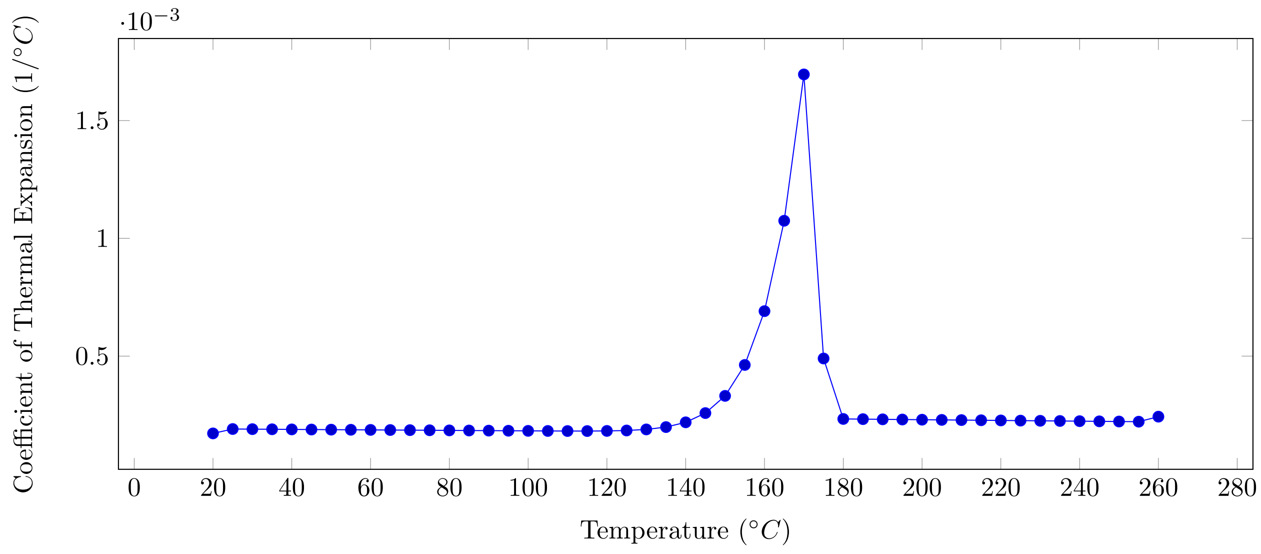
245	1.2955E+09	2.2308E-04
250	1.2999E+09	2.2234E-04
255	1.3042E+09	2.2160E-04
260	1.3086E+09	2.4292E-04
265	1.3133E+09	Not Available

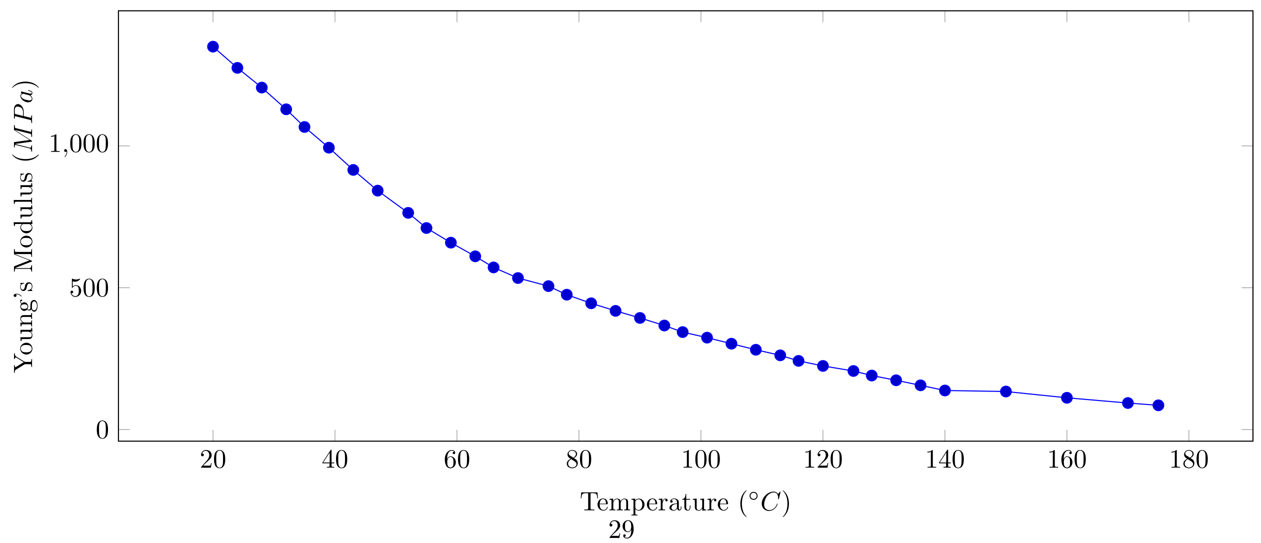
Table 12: Thermomechanical Properties of BC245MO Polypropylene
N.b Values retrieved are an approximation of a general polypropylene polymer as some properties are temperature-dependent.

Constants	Values
Poisson's Ratio	0.4
Thermal Conductivity	$0.2W/m \cdot k$
Melting Temperature	$175°C$
Crystallization Temperature	$165°C$
Emissitivity Ratio	0.97
Specific Heat Capacity	$2.80E + 9mJ/t \cdot^{\circ} C$

Figure 30: Plot of BC245MO Polypropylene Thermomechanical Properties







B MATLAB Code for Avrami Fitting

Listing 1: MATLAB Code for Avrami Fitting

```matlab
clf; clear; close all
% Import Crystallinity Data
load("Data002.mat");
    logExtractedT = Data002(:,1);
    extractedX = Data002(:,2);
% Convert Log Scale to Linear Scale
   extractedT = 10.^logExtractedT;
% Linearize
linearizedX = log(-log(1-extractedX));
linearizedT = log(extractedT);
% Create Matrix M
M = [ones(numel(linearizedT),1), linearizedT(:)];
% Linear Regression via Backslash
Result = M \ linearizedX;
% Extracting Results
fittedKprime = Result(1,1);
fittedK = exp(fittedKprime);
fittedN = Result(2,1);
t = linspace(min(linearizedT),max(linearizedT),15);
% Plotting the Linearized Curve and its Fitting
fittedLinearizedX = fittedKprime + fittedN*t;
figure;
scatter(linearizedT,linearizedX);
hold on
plot(t,fittedLinearizedX);
    title('Linearized Plot and Avrami Fitting','Interpreter','Latex')
    legend({'Extracted Points','Avrami Fitting'},'Location','northwest','Interpreter','Latex')
    caption1 = sprintf('n = %f, ln(k) = %f', fittedN, fittedKprime);
    text(4.5,-4,caption1,'Interpreter','Latex')
    xlabel('ln(t)','Interpreter','Latex')
    ylabel('ln(ln(1/(1-x(t))))','Interpreter','Latex')
    set(gca,'TickLabelInterpreter','latex')
% Plotting the Original Curve and its Fitting
computedX = 1 - exp(-fittedK .* extractedT.^ fittedN);
figure;
scatter(extractedT,extractedX);
hold on
plot(extractedT,computedX);
    title('Plot and Avrami Fitting','Interpreter','Latex')
    legend({'Extracted Points','Avrami Fitting'},'Location','northwest','Interpreter','Latex')
    xlabel('t','Interpreter','Latex')
    ylabel('X','Interpreter','Latex')
    set(gca,'TickLabelInterpreter','latex')
    set(gca, 'XScale', 'log')
    caption2 = sprintf('$X(t) = 1 - exp (-%f * t^%f)$', fittedK, fittedN);
    text(11,0.2,caption2,'Interpreter','Latex');
```

C MATLAB Code for Nakamura Fitting

Listing 2: MATLAB Code for Nakamura Fitting

```
%% Initialization
clear; close all;

%Initialize Values
ST.n = 2;
ST.theta = 90;
ST.k = 1.1;
ST.lamda = 5;
ST.M = 50;
ST.alpha = 0.5;

ST.t = linspace(0, 500, 100);

ST.initN = ST.n;
ST.initTheta = ST.theta * ones(1, size(ST.t, 2));
ST.initK = ST.k * ones(1, size(ST.t, 2));
ST.initLamda = ST.lamda * ones(1, size(ST.t, 2));
ST.initM = ST.M * ones(1, size(ST.t, 2));

ST.Ta = 119.2;
ST.Tb = 0.333;
ST.T2a = 121.8;
ST.T2b = 0.167;
ST.T3a = 123.2;
ST.T3b = 0.0833;
ST.T4a = 126.9;
ST.T4b = 0.0333;
ST.TaAR = ST.Ta * ones(1, size(ST.t, 2));
ST.TbAR = ST.Tb * ones(1, size(ST.t, 2));
ST.T2aAR = ST.T2a * ones(1, size(ST.t, 2));
ST.T2bAR = ST.T2b * ones(1, size(ST.t, 2));
ST.T3aAR = ST.T3a * ones(1, size(ST.t, 2));
ST.T3bAR = ST.T3b * ones(1, size(ST.t, 2));
ST.T4aAR = ST.T4a * ones(1, size(ST.t, 2));
ST.T4bAR = ST.T4b * ones(1, size(ST.t, 2));

ST.t2 = linspace(0,10000,size(ST.t, 2));
ST.Tc = 120;
ST.Tc2 = 125;
ST.Tc3 = 130;
ST.Tc4 = 135;
ST.TcB = 0;
ST.TcAR = ST.Tc * ones(1, size(ST.t2, 2));
ST.Tc2AR = ST.Tc2 * ones(1, size(ST.t2, 2));
ST.Tc3AR = ST.Tc3 * ones(1, size(ST.t2, 2));
ST.Tc4AR = ST.Tc4 * ones(1, size(ST.t2, 2));
ST.TcBAR = ST.TcB * ones(1, size(ST.t2, 2));

%Import Values from DSC - CryVTemp
load("CryVTemp20DegMin.mat");
ST.extractedT1 = CryVTemp20DegMin(:, 1);
ST.extractedX1 = CryVTemp20DegMin(:, 2);
```

```
load("CryVTemp10DegMin.mat");
ST.extractedT2 = CryVTemp10DegMin(:, 1);
ST.extractedX2 = CryVTemp10DegMin(:, 2);
load("CryVTemp5DegMin.mat");
ST.extractedT3 = CryVTemp5DegMin(:, 1);
ST.extractedX3 = CryVTemp5DegMin(:, 2);
load("CryVTemp2DegMin.mat");
ST.extractedT4 = CryVTemp2DegMin(:, 1);
ST.extractedX4 = CryVTemp2DegMin(:, 2);

%Import Values from DSC - CryVTime
load("CryVTime120C.mat");
ST.extractedT1t = 10.^CryVTime120C(:, 1);
ST.extractedX1t = CryVTime120C(:, 2);
load("CryVTime125C.mat");
ST.extractedT2t = 10.^CryVTime125C(:, 1);
ST.extractedX2t = CryVTime125C(:, 2);
load("CryVTime130C.mat");
ST.extractedT3t = 10.^CryVTime130C(:, 1);
ST.extractedX3t = CryVTime130C(:, 2);
load("CryVTime135C.mat");
ST.extractedT4t = 10.^CryVTime135C(:, 1);
ST.extractedX4t = CryVTime135C(:, 2);

%Temperature Plot Set-Up
ST.Tvar = @(t, a, b) (a - b * t);

%% Nakamura Model

%Piecewise Function
ST.f = @(t, m, k, theta, lamda, a, b) ((m * (k / lamda) * ...
(((ST.Tvar(t, a, b) - theta) / lamda) .^ (k - 1))) .*...
exp(- ((ST.Tvar(t, a, b) - theta) / lamda) .^ k)) .* (ST.Tvar(t, a, b) > theta);

%Generating Array of Integral Functions
ST.q = @(tau, m, k, theta, lamda, a, b) (integral(@(t) ST.f(t, m, k, theta, lamda, a,
    b ), 0, tau));
ST.Q = @(tau, m, k, theta, lamda, a, b) arrayfun(ST.q, tau, m, k, theta, lamda, a, b)
    ;

%Exponential Function
ST.X = @(t, t2, m, k, theta, lamda, a, b) (1 - exp(- (ST.Q(t, m, k, theta, lamda, a,
    b)) .^ t2));

%% Plotting ancillary figures
% Plot crystallization vs time
figure;
plot(ST.t, ST.X(ST.t, ST.initN, ...
ST.initM, ST.initK, ST.initTheta, ST.initLamda, ST.TaAR, ST.TbAR), 'b');
hold on;
plot(ST.t, ST.X(ST.t, ST.initN, ...
ST.initM, ST.initK, ST.initTheta, ST.initLamda, ST.T2aAR, ST.T2bAR), 'g');
hold on;
plot(ST.t, ST.X(ST.t, ST.initN, ...
ST.initM, ST.initK, ST.initTheta, ST.initLamda, ST.T3aAR, ST.T3bAR), 'm');
hold on;
```

```
plot(ST.t, ST.X(ST.t, ST.initN, ...
ST.initM, ST.initK, ST.initTheta, ST.initLamda, ST.T4aAR, ST.T4bAR), 'r');
xlabel('Time, t (s)','Interpreter','Latex');
ylabel('Crystallinity, X(t)','Interpreter','Latex');
legend('Nakamura Fitting ','Nakamura Fitting ',...
'Nakamura Fitting 5C/min','Nakamura Fitting 2C/min','Location','southeast');

% Plot crystallization vs time (isothermal)
ST.fig2 = figure;
ST.TLN = plot(ST.t2, ST.X(ST.t2, ST.initN, ...
ST.initM, ST.initK, ST.initTheta, ST.initLamda, ST.TcAR, ST.TcBAR), 'b');
hold on;
ST.TLN2 = plot(ST.t2, ST.X(ST.t2, ST.initN, ...
ST.initM, ST.initK, ST.initTheta, ST.initLamda, ST.Tc2AR, ST.TcBAR), 'g');
hold on;
ST.TLN3 = plot(ST.t2, ST.X(ST.t2, ST.initN, ...
ST.initM, ST.initK, ST.initTheta, ST.initLamda, ST.Tc3AR, ST.TcBAR), 'm');
hold on;
ST.TLN4 = plot(ST.t2, ST.X(ST.t2, ST.initN, ...
ST.initM, ST.initK, ST.initTheta, ST.initLamda, ST.Tc4AR, ST.TcBAR), 'r');
hold on;
scatter(ST.extractedT1t, ST.extractedX1t,'bx');
hold on;
scatter(ST.extractedT2t, ST.extractedX2t,'gx');
hold on;
scatter(ST.extractedT3t, ST.extractedX3t,'mx');
hold on;
scatter(ST.extractedT4t, ST.extractedX4t,'rx');
set(gca, 'XScale', 'log')
xlabel('Time, t (s)','Interpreter','Latex');
ylabel('Crystallinity, X(t)','Interpreter','Latex');
legend('120C','125C','130C','135C','Extracted Points 120C','Extracted Points 125C',...
'Extracted Points 130C','Extracted Points 135C','Location','southeast');

% Plot temperature vs time
figure;
plot(ST.t, ST.Tvar(ST.t, ST.Ta, ST.Tb), 'b');
hold on;
plot(ST.t, ST.Tvar(ST.t, ST.T2a, ST.T2b), 'g');
hold on;
plot(ST.t, ST.Tvar(ST.t, ST.T3a, ST.T3b), 'm');
hold on;
plot(ST.t, ST.Tvar(ST.t, ST.T4a, ST.T4b), 'r');
xlabel('Time, t (s)','Interpreter','Latex');
ylabel('Temperature, T ($^{\circ}$ C)','Interpreter','Latex');
legend('20C/min','10C/min','5C/min','2C/min','Location','southwest');

%% User Interface for Manual Fitting of Nakamura Model
% Plot different plots according to initial parameters
ST.fig = figure('units', 'pixels', ...
  'position', [500 100 700 900], ...
  'menubar', 'none', ...
  'name', 'Nakamura slider plot', ...
  'numbertitle', 'off', ...
  'resize', 'off');
ST.ax = axes('unit', 'pix', 'position', [50 350 610 500]);
ST.LN = plot(ST.Tvar(ST.t, ST.Ta, ST.Tb), ST.X(ST.t, ST.initN, ...
ST.initM, ST.initK, ST.initTheta, ST.initLamda, ST.TaAR, ST.TbAR), 'b');
hold on;
ST.LN2 = plot(ST.Tvar(ST.t, ST.T2a, ST.T2b), ST.X(ST.t, ST.initN, ...
ST.initM, ST.initK, ST.initTheta, ST.initLamda, ST.T2aAR, ST.T2bAR), 'g');
hold on;
ST.LN3 = plot(ST.Tvar(ST.t, ST.T3a, ST.T3b), ST.X(ST.t, ST.initN, ...
ST.initM, ST.initK, ST.initTheta, ST.initLamda, ST.T3aAR, ST.T3bAR), 'm');
hold on;
ST.LN4 = plot(ST.Tvar(ST.t, ST.T4a, ST.T4b), ST.X(ST.t, ST.initN, ...
ST.initM, ST.initK, ST.initTheta, ST.initLamda, ST.T4aAR, ST.T4bAR), 'r');
xlabel('Temperature, T ($^{\circ}$ C)','Interpreter','Latex');
ylabel('Crystallinity, X(t)','Interpreter','Latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');
xlim([90 140])
ylim([0 1])
hold on
scatter(ST.extractedT1, ST.extractedX1,'bx');
scatter(ST.extractedT2, ST.extractedX2,'gx');
scatter(ST.extractedT3, ST.extractedX3,'mx');
scatter(ST.extractedT4, ST.extractedX4,'rx');
legend('Nakamura Fitting 20C/min','Nakamura Fitting 10C/min',...
'Nakamura Fitting 5C/min','Nakamura Fitting 2C/min',...
'Extracted Points 20C/min','Extracted Points 10C/min',...
'Extracted Points 5C/min','Extracted Points 2C/min');

ST.text1 = text(95, 0.7, sprintf('n= %f', ...
ST.initN), 'Interpreter', 'Latex');
ST.text2 = text(95, 0.6, sprintf('m= %f', ...
ST.initM(1)), 'Interpreter', 'Latex');
ST.text3 = text(95, 0.5, sprintf('k= %f', ...
ST.initK(1)), 'Interpreter', 'Latex');
ST.text4 = text(95, 0.4, sprintf('theta= %f', ...
ST.initTheta(1)), 'Interpreter', 'Latex');
ST.text5 = text(95, 0.3, sprintf('lamda= %f', ...
ST.initLamda(1)), 'Interpreter', 'Latex');

annotation('textbox', [0.05, 0.24, 0, 0], 'string', 'n','Interpreter','Latex')
annotation('textbox', [0.05, 0.19, 0, 0], 'string', 'm','Interpreter','Latex')
annotation('textbox', [0.05, 0.14, 0, 0], 'string', 'k','Interpreter','Latex')
annotation('textbox', [0.05, 0.09, 0, 0], 'string', 'theta','Interpreter','Latex')
annotation('textbox', [0.05, 0.04, 0, 0], 'string', 'lamda','Interpreter','Latex')

% Plot different plots according to slider parameters
ST.sl = uicontrol('style', 'slide', ...
  'unit', 'pix', ...
  'position', [100 170 430 30], ...
  'min', 0, 'max', 5, 'val', ST.n, ...
  'sliderstep', [1 / 20 1 / 20], ...
  'callback', {@sl_call, ST}, ...
  'Tag', 'slider1');
ST.s2 = uicontrol('style', 'slide', ...
  'unit', 'pix', ...
  'position', [100 130 430 30], ...
  'min', 0, 'max', 500, 'val', ST.M, ...
  'sliderstep', [1 / 20 1 / 20], ...
  'callback', {@sl_call2, ST}, ...
  'Tag', 'slider2');
ST.s3 = uicontrol('style', 'slide', ...
  'unit', 'pix', ...
  'position', [100 90 430 30], ...
  'min', 0.9, 'max', 1.3, 'val', ST.k, ...
  'sliderstep', [1 / 20 1 / 20], ...
  'callback', {@sl_call3, ST}, ...
  'Tag', 'slider3');
ST.s4 = uicontrol('style', 'slide', ...
  'unit', 'pix', ...
  'position', [100 50 430 30], ...
  'min', 80, 'max', 150, 'val', ST.theta, ...
  'sliderstep', [1/20 1/20], ...
  'callback', {@sl_call4, ST}, ...
  'Tag', 'slider4');
ST.s5 = uicontrol('style', 'slide', ...
  'unit', 'pix', ...
  'position', [100 10 430 30], ...
  'min', 0, 'max', 20, 'val', 5, ...
  'sliderstep', [1 / 20 1 / 20], ...
  'callback', {@sl_call5, ST}, ...
  'Tag', 'slider5');

saveas(gcf,'myfiguretest1.pdf');

function [] = sl_call(varargin)
    [h, ST] = varargin{[1, 3]};
    newN = (get(h, 'value'));
    h = findobj('Tag', 'slider2');
    newM = (get(h, 'value') * ones(1, size(ST.t, 2)));
    h = findobj('Tag', 'slider3');
    newK = (get(h, 'value')) * ones(1, size(ST.t, 2));
    h = findobj('Tag', 'slider4');
    newTheta = (get(h, 'value')) * ones(1, size(ST.t, 2));
    h = findobj('Tag', 'slider5');
    newLamda = (get(h, 'value')) * ones(1, size(ST.t, 2));
    updatePlot(ST,ST.X,ST.t, newN, newM, newK, newTheta, newLamda);
    set(ST.text1, 'String', sprintf('new n = %f', newN));
end

function [] = sl_call2(varargin)
    [h, ST] = varargin{[1, 3]};
    newM = (get(h, 'value')) * ones(1, size(ST.t, 2));
    h = findobj('Tag', 'slider1');
    newN = (get(h, 'value'));
    h = findobj('Tag', 'slider3');
    newK = (get(h, 'value')) * ones(1, size(ST.t, 2));
    h = findobj('Tag', 'slider4');
    newTheta = (get(h, 'value')) * ones(1, size(ST.t, 2));
    h = findobj('Tag', 'slider5');
    newLamda = (get(h, 'value')) * ones(1, size(ST.t, 2));
    updatePlot(ST,ST.X,ST.t, newN, newM, newK, newTheta, newLamda);
    set(ST.text2, 'String', sprintf('new M = %f', newM(1)));
end
```

```

function [] = sl_call3(varargin)
    [h, ST] = varargin{[1, 3]};
    newK = (get(h, 'value')) * ones(1, size(ST.t, 2));
    h = findobj('Tag', 'slider1');
    newN = (get(h, 'value'));
    h = findobj('Tag', 'slider2');
    newM = (get(h, 'value') * ones(1, size(ST.t, 2)));
    h = findobj('Tag', 'slider4');
    newTheta = (get(h, 'value')) * ones(1, size(ST.t, 2));
    h = findobj('Tag', 'slider5');
    newLamda = (get(h, 'value')) * ones(1, size(ST.t, 2));
    updatePlot(ST,ST.X,ST.t, newN, newM, newK, newTheta, newLamda);
    set(ST.text3, 'String', sprintf('new K = %f', newK(1)));
end

function [] = sl_call4(varargin)
    [h, ST] = varargin{[1, 3]};
    newTheta = (get(h, 'value')) * ones(1, size(ST.t, 2));
    h = findobj('Tag', 'slider1');
    newN = (get(h, 'value'));
    h = findobj('Tag', 'slider2');
    newM = (get(h, 'value') * ones(1, size(ST.t, 2)));
    h = findobj('Tag', 'slider3');
    newK = (get(h, 'value')) * ones(1, size(ST.t, 2));
    h = findobj('Tag', 'slider5');
    newLamda = (get(h, 'value')) * ones(1, size(ST.t, 2));
    updatePlot(ST,ST.X,ST.t, newN, newM, newK, newTheta, newLamda);
    set(ST.text4, 'String', sprintf('new theta = %f', newTheta(1)));
end

function [] = sl_call5(varargin)
    [h, ST] = varargin{[1, 3]};
    newLamda = (get(h, 'value')) * ones(1, size(ST.t, 2));
    h = findobj('Tag', 'slider1');
    newN = (get(h, 'value'));
    h = findobj('Tag', 'slider2');
    newM = (get(h, 'value') * ones(1, size(ST.t, 2)));
    h = findobj('Tag', 'slider3');
    newK = (get(h, 'value')) * ones(1, size(ST.t, 2));
    h = findobj('Tag', 'slider4');
    newTheta = (get(h, 'value')) * ones(1, size(ST.t, 2));
    updatePlot(ST,ST.X,ST.t, newN, newM, newK, newTheta, newLamda);
    set(ST.text5, 'String', sprintf('new lamda = %f', newLamda(1)));
end

function [] = updatePlot(varargin)
    ST = varargin{1};
    [ST.X,ST.t, newN, newM, newK, newTheta, newLamda] = varargin{2:8};
    set(ST.LN, 'ydata', ((ST.X(ST.t, newN, newM, newK, newTheta, newLamda, ST.TaAR,
        ST.TbAR))));
    set(ST.LN2, 'ydata', ((ST.X(ST.t, newN, newM, newK, newTheta, newLamda, ST.T2aAR,
        ST.T2bAR))));
    set(ST.LN3, 'ydata', ((ST.X(ST.t, newN, newM, newK, newTheta, newLamda, ST.T3aAR,
        ST.T3bAR))));
    set(ST.LN4, 'ydata', ((ST.X(ST.t, newN, newM, newK, newTheta, newLamda, ST.T4aAR,
         ST.T4bAR))));
    set(ST.TLN, 'ydata', ((ST.X(ST.t2, newN, newM, newK, newTheta, newLamda, ST.TcAR,
         ST.TcBAR))));
    set(ST.TLN2, 'ydata', ((ST.X(ST.t2, newN, newM, newK, newTheta, newLamda, ST.
        Tc2AR, ST.TcBAR))));
    set(ST.TLN3, 'ydata', ((ST.X(ST.t2, newN, newM, newK, newTheta, newLamda, ST.
        Tc3AR, ST.TcBAR))));
    set(ST.TLN4, 'ydata', ((ST.X(ST.t2, newN, newM, newK, newTheta, newLamda, ST.
        Tc4AR, ST.TcBAR))));
end
```