Process parameters of 4D printing which affect the shape memory effect of PLA

A master thesis by

I.S. Best

4484509 2024.MME.892

Supervisor: Jovana Jovanova

Date: 2024-03-21

Abstract

This masters thesis investigates the influence of various printing parameters on the shape memory effect of 3D printed objects, with a focus on fixity and recovery rates. Through a series of experimental tests, it was observed that printing temperature had minimal impact on fixity and recovery rates. However, correlations were identified between fixity rate and layer height, as well as between percentage infill and recovery rates. Lengthwise shrinkage, particularly prominent in samples with 0% infill, was attributed to printing speed the formation of voids within the structure and layer height. Higher printing speeds were found to compromise mechanical properties while facilitating enhanced shape transformation responses. Additionally, changes in layer height led to observable alterations in the printed object's geometry, including bending and bulging, due to retained shape memory of the filaments original form. Moreover, certain 100% infill samples exhibited an unexpected hardening phenomenon akin to annealing. These findings underscore the intricate interplay of printing parameters in determining shape memory properties, mechanical properties and highlight potential avenues for optimization in 3D printing processes.

List of Symbols

The next list describes several symbols that will be later used within the body of the document

Abbreviations

R_F Fixity Rate

R_R Recovery Rate

T_g Glass Transition Temperature

°C Degrees Celsius

AM Additive Manufacturing

CAD Computer Aided Design

dB decibels

DIW Direct Ink Writing

DLP Digital Light Processing

FDM Fused Deposition Modeling

FE Finite Elements

FEM Finite element method

HDT Heat Deflection Temperature

LCE Liquid Crystal Elastomer

LNG Liquid Natural Gas

MIT Molecular imprinting technology

ML Machine Learning

MPa Mega Pascals

MSLA Masked sterolithography

PETG Polyethylene Terephthalate Glycol

PLA Poly(Lactic Acid)

SLA Stereolithograpy

SLS Selective Laser Sintering

SMA Shape Memory Alloy

SMM Shape Memory Material

SMP Shape Memory Polymer

Contents

Chapter 1

Introduction

In this research, we will design an experiment in order to investigate how the shape memory effect(SME) of Poly-Lactic-Acid(PLA) can be influenced through adjusting basic printing parameters, such as layer height, printing temperature, and nozzle velocity. We will begin with an introduction to 3D and 4D printing for the reader and the SME of PLA in particular. Then we will quantify the shape memory effect through the use of the fixity and recovery rates. We will then use a design of experiments(DoE) to set up the experiment, proposing a systematic method for stimulating the SME of PLA, how the parameters will be varied, the ranges in which these parameters will be varied, and designing a method to record the necessary data, the fixity and recovery rates. The data will then be analyzed through the use of Python to investigate if any significant relation exists within the parameters and which parameter is the largest influence on the SME.

1.1 Additive manufacturing

3D printing fall in the category of additive manufacturing(AM), a defined ISO Term. It is defined as: "The process of joining materials to make parts from 3D model data, usually layer upon layer, as opposed to subtractive manufacturing and formative manufacturing methodologies"[1]. A brief introduction to 3D printing will help with developing a better understanding of the concept of 4D printing. 3D printing started gaining traction when a patent was filed for the stereolithographic process and was met with huge interest[2]. The 3D printing technology is currently used by consumer communities and for rapid prototyping with various AM technologies such as, fused deposition modeling (FDM), stereolithography (SLA), masked stereolithography (MSLA), selective laser sintering (SLS), selective laser melting (SLM), jet 3D printing, direct ink writing, etc. [3]. 3D has a significant benefit because it is possible to construct an object with high precision in a short time without extra required tools, steps, and with little or no waste material[4]. 3D printing has attracted immense interest from both industrial giants and academic institutions due to these features. 3D printing is a multidisciplinary field and involves collaboration from fields of science such as material engineering, mechanical engineering, medicine, etc. The developments in recent years have resulted in the ability to create intricate and complex structures, that were impossible or challenging to construct before using conventional techniques. [5][6][7][8][9][10][11]. The most commonly used techniques in 3D printing are FDM and (M)SLA. FDM printers deposit molten material on a bed, layer by layer using a nozzle, with layer heights commonly ranging from 0.05mm to 0.6mm. (M)SLA printers emit UV radiation with high accuracy on liquid polymers which then harden in a controlled environment on a vertically moving build plate with layer heights commonly ranging from 100μm to 5μm. 4D printing is the process of 3D printing a programmed form or structure which is able to either change or retain its form through external influence[12]

1.2 Shape memory effect

The shape memory effect(SME) is a fascinating phenomenon exhibited by certain polymers, which allows them to change their shape in response to external stimuli and return to their original shape upon removal of the stimulus. These polymers are known as shape memory polymers (SMPs) and have garnered significant interest in various fields due to their unique properties and potential applications.

SMPs possess the ability to "remember" and recover their original shape from a temporary shape induced by an external trigger, such as heat, light, moisture, or electrical current. This shape recovery occurs due to the presence of two main components in SMPs: a temporary shape and a permanent shape. The temporary shape is set by deforming the polymer above its transition temperature, which allows it to be easily manipulated and fixed into a new shape. When the SMP is exposed to the triggering stimulus, it undergoes a phase transition and reverts back to its permanent shape. The shape memory effect in polymers is primarily attributed to the presence of cross-linked networks within the material. These networks enable the SMPs to exhibit a dual-phase behavior, with a rigid and glassy state at low temperatures and a rubbery and elastic state at higher temperatures. This unique characteristic allows the material to be deformed and recover its original shape by switching between these two states.

1.3 4D printing

4D printing is an emerging technology that builds upon the principles of 3D printing by incorporating the element of time as an additional dimension. It refers to the process of creating objects that can self-transform or change their shape over time when subjected to external stimuli, such as heat, moisture, light, or mechanical force. The "4D" in 4D printing represents the fourth dimension, which signifies the temporal aspect of the printed objects' behavior. This dynamic behavior is achieved by printing with smart materials that possess inherent properties allowing them to respond and adapt to specific environmental conditions or triggers. These materials are typically programmable or possess shape memory characteristics. The process of 4D printing involves designing and fabricating a structure using materials with the desired properties. These materials may include shape memory polymers, hydrogels, or composites that can undergo reversible changes in their shape, size, or stiffness. The printed object is then subjected to the external stimulus, which activates the material's response and causes the intended transformation. Applications of 4D printing span across various fields, including engineering, architecture, biomedicine, and aerospace. In architecture and construction, 4D printing enables the creation of structures that can adapt to changing environmental conditions or have self-assembling capabilities. In biomedicine, it holds the potential for developing advanced tissue engineering scaffolds, drug delivery systems that can respond to specific physiological cues, cardiovascular implants[13], self-bending stents and self-shrinking/tightening staples[14]. The aerospace industry can benefit from 4D printing by creating lightweight components that can self-repair or adapt to different flight conditions. Despite its potential, 4D printing is still in its early stages, and there are challenges to overcome. These include refining the materials and printing techniques, ensuring reliable and precise control over shape transformations, and developing robust design methods to optimize the printed objects' performance.

1.3.1 Shape memory polymers

Some shape memory polymers can behave in different ways, some are commercially hard to acquire or require extremely specific printing parameters in order to properly print a structure. Some of the commercially available polymers, their activation temperatures and their main features are listed in the table below1.1, from the work of Sun et al.[15]. Based on the properties of the SMP's, earlier research, availability of the material and costs, PLA will be the subject of this study.

1.4 Problem statement

A lot of SME research has been done with PLA polymer blends in order to exert control over the effect, however, most research that is done with FDM printing of SMP all uses the same standard printing settings. We are interested in how much influence the printer itself can have on the SME in order to be able to control the SME to a higher degree since these findings could be applicable to other materials as well.

Polymer	Activation Temperature	Main Features
ABS	Tg: 105 °C	Thermo-plastic, Excellent heating-responsive SME
EVA	Tg: 60 °C	Thermo-plastic, Excellent heating/chloroform-responsive SME
PC	Tg: 142 °C	Thermo-plastic, Excellent heating-responsive SME
PCL	Tm:55 °C	Thermo-plastic, Bio-degradable, Excellent heating-responsive SME, Programmed at low temperatures
PEEK	Tg: 155 °C	Thermo-plastic, Excellent heating-responsive SME
PLA	Tg: 65 °C	Thermo-plastic, Bio-degradable, Good heating-responsive SME
PMMA	Tg: 115 °C	Thermo-plastic, Good heating/ethanol-responsive SME
PS	Tg: 65 °C	Thermo-set; Good heating/acetone-responsive SME
PTFE	Tg: 65 °C °C	Thermo-plastic, Excellent heating-responsive SME, Two-step recovery upon heating
PU	Tg: 35 65 °C	Thermo-plastic/thermo-set, Biocompatible, Excellent heating/ethanol/water-responsive SME
PVA	Tg: 30 °C	Thermo-plastic, Excellent heating-responsive SME, Water-responsive SME
TPU	Tm: 55 °C	Thermo-plastic/vitrimer, Excellent heating-responsive SME

Table 1.1: Summary of typical commercial shape memory polymers (SMPs)[15]

1.5 Research Questions

The research questions that will be answered are:

- How to control the SME of PLA through printing parameters?

This will be answered through the following sub questions:

- What are the mechanical and chemical properties of PLA and at what temperature does the SME occur?
- What are the printing parameters of an FDM printer and how will they be modified?
- How do we qualitatively describe the SME parameters?
- What test setup will be used to deform the test structure?
- What relations can be found between parameters?

Chapter 2

Materials and Methods

Poly(lactic acid) (PLA), also referred to as polylactic acid or polylactide, is a thermoplastic polyester known for its sustainable properties. Its backbone formula is represented as $(C_3H_4O_2)_n$ or $[-C(CH_3)HC(=O)O-]_n$. The structure of PLA exhibits a polymeric helix arrangement with an orthorhombic unit cell, as depicted in Figure 2.1. It is worth noting that the name "polylactic acid" is commonly used but does not adhere to the IUPAC standard nomenclature, which specifies "poly(lactic acid)"[16]. This distinction is important as PLA is a polyester rather than a polyacid (polyelectrolyte) [17].

The increasing popularity of PLA stems from its ability to be economically produced from renewable resources. In fact, in 2021, PLA achieved the highest consumption volume among all bioplastics worldwide [18]. One prominent application of PLA is in the field of 3D printing, where it stands as the most widely used plastic filament material. This preference arises from its advantageous characteristics, including a low melting point, high strength, low thermal expansion, excellent layer adhesion, and notable heat resistance when annealed. These properties collectively contribute to making PLA an ideal material for 3D printing applications.

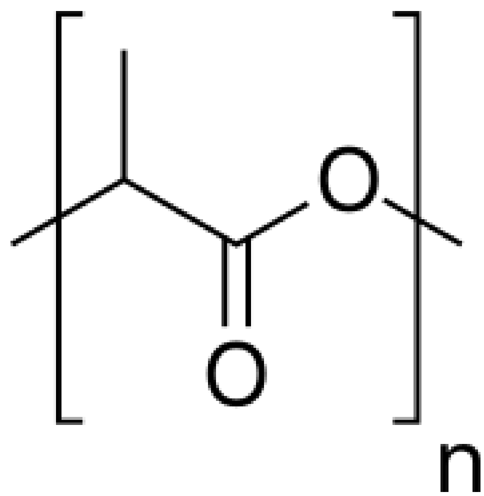

Figure 2.1: PLA molecule, where "n" denotes the chain length, O represents Oxygen and and every undefined part is Carbon

PLA belongs to the category of thermo-responsive shape memory polymers(SMPs) and as such possess the SME when subjected to high temperatures. The underlying mechanism for the SME in SMPs is the dual segment/domain system. The mechanism for the SME of a thermo-responsive SMP is simplified and illustrated in Fig. 2.2. The SMP is usually much softer at high temperatures than that at low temperatures and as such can be deformed. This deformation is also called "programming" of the structure.

The dual-segment/domain system of a thermoresponsive SMP, which is responsible for its SME, mentioned earlier consists of shape fixing and switching parts. Fig.2.3. The shape-fixing parts are the structural feature or domain i.e. the SMP network's net points responsible for dimensional solidity during the deformation and recovery cycles, as shown in Fig.2.3. They can be chemically cross-linked due to covalent bonds or formed through physical entanglement due to intermolecular interactions. The

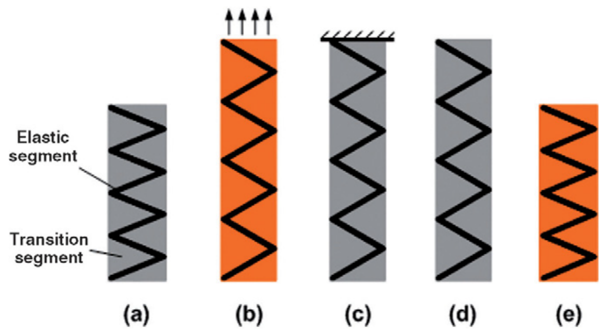

Figure 2.2: Illustration of the mechanism of the SME in thermo-responsive SMP. (a) Hard at low temperature; (b) easily deformed at high temperature; (c) hard again after cooling; (d) temporary (deformed) shape after constraint removed; (e) shape recovery upon heating.[19]

shape-swapping parts, which are relevant when deformation occurs, are often long chains of the polymer existing between the shape-fixing parts (net points) and aids in storing the elastic strain that is exerted during the deformation. This enables elastic recoiling during the recovery cycle [20]. Conformational entropy drives the transitions within the SMP networks to the chemical/physical cross-links known as junction density and chain entanglement/stretching[21]. In Fig.2.3, at the start of the programming cycle, that is, the conformational entropy is high at low temperatures. when the temperature rises over the glass transition temperature (T_g) during the deformation phase the chain mobility increases, and the polymer softens. After that, the entanglement/stretching is generated, and during cooling, the entropic energy decreases. During the recovery cycle, when the deformed sample is reheated, the polymeric chain mobility increases again. The polymer regains its original shape by returning to a thermodynamically favored state by releasing the stored entropy, showing the SME.

Below are some commonly discussed mechanical properties of PLA[23][24][25][26][27][28]:

- Tensile Strength: PLA exhibits good tensile strength, which refers to its ability to withstand pulling forces without breaking. The tensile strength of PLA typically ranges from 50 to 70 megapascals (MPa). However, it should be noted that the tensile strength of PLA can vary depending on the specific grade, processing conditions, and testing methods.

- Young's Modulus: Young's modulus, also known as the elastic modulus, characterizes a material's stiffness or resistance to deformation under an applied force. PLA has a relatively high Young's modulus, typically ranging from 3 gigapascals (GPa). This property makes PLA relatively rigid and less prone to elongation.

- Flexural Strength: The flexural strength of PLA refers to its resistance to deformation when subjected to bending forces. PLA generally exhibits good flexural strength, typically averaging 100 MPa. This property is crucial in applications where PLA components need to withstand bending or flexing without fracturing.

- Elongation Break: the elongation at break of PLA falls in the range of 2% to 7%. This means that PLA can typically stretch or deform up to 2% to 7% of its original length before breaking under tensile stress at room temperature.

It's important to note that both the mechanical and thermal properties of PLA can be influenced by various factors, including molecular weight, processing techniques, blending with other materials, and post-processing treatments like annealing. PLA annealing is a known heat treatment process in which the internal stress of the material caused by production on a 3D printer is minimized, and thus higher strength of printed parts is achieved[29]. Additionally, different manufacturers may offer PLA grades with specific mechanical properties tailored for different applications.

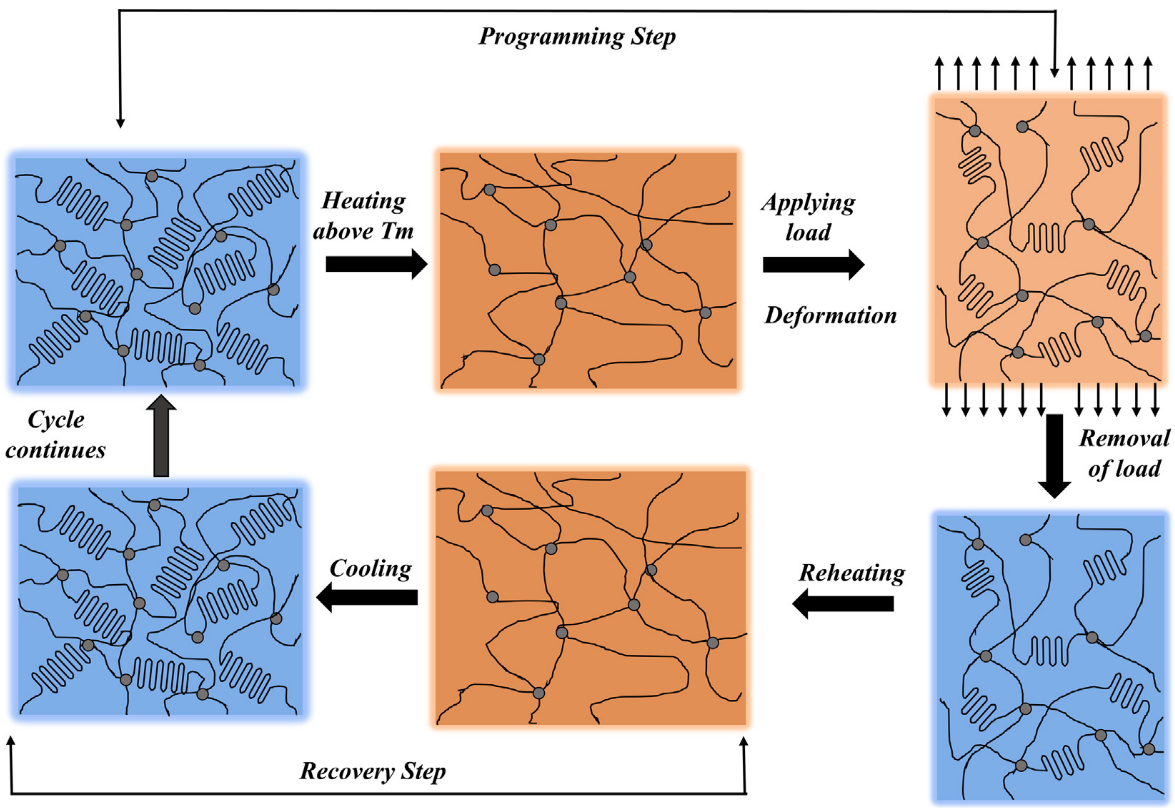

Figure 2.3: Mechanism of shape memory effect in thermoresponsive shape memory polymer[22].

Below are some commonly reported thermal properties of PLA[23][28]:

- Melting Point: The melting point of PLA is typically about 180°C. However, it is worth noting that PLA is a semi-crystalline polymer, and its melting behavior can be influenced by factors such as molecular weight and crystallinity.
- Glass Transition Temperature: The glass transition temperature (Tg) of PLA is usually around 55 to 60°C. The Tg represents the temperature at which PLA transitions from a rigid, glassy state to a softer, rubbery state.
- Heat Deflection Temperature: HDT refers to the temperature at which a material deforms under a specified load. It is an indicator of the heat resistance of the material.
  - At 0.46 MPa the HDT averages 82.9 °C
  - At 1.8 MPA the HDT averages 76.1 °C

2.1 Experiment Design

For this experiment the theory of the design of experiments will be used. Design of experiments (DOE) is a systematic and structured approach used to plan, conduct, analyze, and optimize experiments. It involves carefully designing the experimental conditions and variables to gather meaningful data and draw reliable conclusions. The main goals of DOE are to efficiently explore the relationships between factors and their effects on a response variable, identify significant factors, and optimize the process or system under investigation. Using this method we will investigate the effect the initial conditions, i.e. the printing parameters, also called the independent variable, has on the dependent variable, i.e. the fixity and recovery rate. Three parameters are expected to influence the SME of the material, which are the printing temperature, the infill percentage and the layer height. Our variables are as follows:

- Dependent variables:
  - fixity rate
  - recovery rate
- Independent variables:
  - Printing temperature
  - Infill percentage
  - Layer height

Three important criteria for the experimental plan have to be satisfied: validity, reliability and replicability:

- In the context of Design of Experiments (DOE) refers to the extent to which the experimental results accurately represent the true effects of the factors being studied. It is essential to ensure that the experimental design and procedures used in DOE produce valid and reliable results. Validity can be further broken down into three key points:
  - Internal validity refers to the degree to which the observed effects can be attributed to the manipulated factors and not to other extraneous factors or sources of bias. To enhance internal validity, it is important to control or eliminate potential confounding variables that could influence the response variable. Randomization, blocking, and the use of control groups or baseline measurements are some strategies to enhance internal validity in DOE.

  - External validity refers to the generalizability of the experimental findings to the larger population or real-world scenarios. It is important to design experiments that are representative of the target population or relevant operating conditions. However, it is often challenging to achieve high external validity due to the need for controlled experimental conditions. Therefore, researchers should carefully consider the trade-off between internal and external validity when designing experiments.
  - Construct validity refers to the extent to which the experimental design and procedures accurately measure or manipulate the intended factors or constructs of interest. This involves ensuring that the chosen factors and their levels are conceptually and operationally valid. Construct validity can be enhanced by using well-established measurement techniques, validated instruments, and ensuring the appropriate manipulation of factors.

- Reliability is about repeatability. If someone else would run these experiments similar results have to be found.

- The term replicability means that someone else has to be able to perfectly replicate this experiment through the documentation provided.

These three criteria have to be satisfied in order to have a proper experimental plan. Through this plan answers are sought to the following questions:

- Is there a relation between the printing parameters used during the operation of an FDM printer and the SME of PLA:
  - For different temperatures
  - For different percentages of infill
  - For different layer heights

The implementation of a full factorial design in scientific investigations confers notable advantages. This approach systematically evaluates all conceivable combinations of factor levels, leading to an extensive comprehension of the main effects and interactions of each factor on the response variable. Through this comprehensive evaluation, no factor or interaction is overlooked during the analysis process, ensuring a thorough investigation. Moreover, the full factorial design enables effective detection and estimation of interactions between factors, which is crucial for comprehending the complexity of the investigated system. This aspect contributes significantly to understanding the interplay between factors and their combined impact on the response variable. From a statistical perspective, the full factorial design demonstrates a high level of efficiency compared to other experimental designs. By evaluating all potential factor combinations, it facilitates precise estimation of main effects, interactions, and response variable variability, resulting in more accurate statistical inferences and reliable conclusions. Additionally, the full factorial design exhibits robustness and generalizability in the presence of uncontrolled variables or variations in experimental conditions. The exploration of the entire factor space enhances the assessment of the effects of factors, ensuring that the obtained results are representative and applicable across a broader range of operating conditions. This characteristic enhances the reliability and validity of the findings derived from the full factorial design approach.

A full factorial analysis of the parameters which are used by the FDM printer will be done with the following ranges:

- Printing Temperature in [°C]
  - 190, 200, 210

- Infill percentage in [%]
  - 0, 50%, 100%

- Layer height in [mm]
  - 0.1, 0.2, 0.3

The resulting amount of experiments are shown in table 2.1.

Table 2.1: Experiments created by applying full factorial method using pythons doepy library

Experiment nr.	Print Temperature	Infill	Layer Height
1	190.0	0.0	0.1
2	200.0	0.0	0.1
3	210.0	0.0	0.1
4	190.0	0.5	0.1
5	200.0	0.5	0.1
6	210.0	0.5	0.1
7	190.0	1.0	0.1
8	200.0	1.0	0.1
9	210.0	1.0	0.1
10	190.0	0.0	0.2
11	200.0	0.0	0.2
12	210.0	0.0	0.2
13	190.0	0.5	0.2
14	200.0	0.5	0.2
15	210.0	0.5	0.2
16	190.0	1.0	0.2
17	200.0	1.0	0.2
18	210.0	1.0	0.2
19	190.0	0.0	0.3
20	200.0	0.0	0.3
21	210.0	0.0	0.3
22	190.0	0.5	0.3
23	200.0	0.5	0.3
24	210.0	0.5	0.3
25	190.0	1.0	0.3
26	200.0	1.0	0.3
27	210.0	1.0	0.3

For the experiment, a CAD model has been made for both the testing structure and the sample to be tested. The testing structure will be printed in PETG, which is a robust polymer that will be printed at 260 degrees Celsius and has a T_G of 75 degrees Celsius [30]. As such when the PLA is heated above the glass transition temperature the testing structure will be unaffected, while the sample will have transitioned to its rubbery state allowing it to be bent. The structure has a blue moving part which is guided by a slot in the bottom plate. It is the only part which is printed separately. The entire gray structure will be a single print. A fixed angle of 45 degrees is the endpoint of this arc. This ensures that every experiment will always have the same angle in it. When measuring the sample the left most inner corners will be used. Both the top and bottom side of the sample will be measured and then averaged. The angle the sample will take when deformed will be 38.5 from these points. The testing sample will have the dimensions of 150mm x 20mm x 10mm, see Fig. 2.4 and will be printed 27 times, with different settings as according to Table 2.1. The other print parameters which will be constant can be found in Table 2.2.The load which is applied on the sample by the testing structure will be perpendicular to the z-axis of the 3D print, which means it is perpendicularly loaded on the layer

Printing parameters	
Nozzle diameter	0.4 [mm]
Perimeter printing speed	45 [mm/s]
Infill printing speed	80 [mm/s]
Bed temperature	60 [°C]
Minimum shell thickness	0.7 [mm]
Number of perimeter layers	3
First layer height	0.1 [mm]
Number of bottom layers	7
Number of top layers	9
seam position	nearest

Table 2.2: Printing parameters

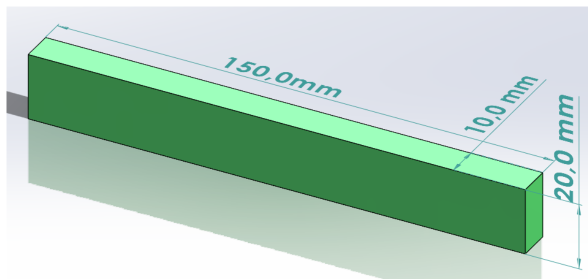

Figure 2.4: The sample which will be tested

lines. The heating will take place with a induction plate which can maintain a constant temperature above the glass transition point. Boiling water cannot exceed a temperature of 100 degrees celsius and has been chosen as the heating medium.

The FDM printer that will be used is a Prusa MK3 with a 0.4mm extrusion nozzle. The brand of PLA used for this experiment is "MTB3D PLA filament 1.75mm 1kg". A camera will be used to take pictures of the samples before programming, after programming, and after recovery.

Yu et al.[31] has done a similar experiment,as shown in Fig. 2.6, and used the following SM cycle: In the programming step, the SMP is stretched to a target strain $\dot{e}_{max}$ (20%) with a constant loading rate $(0.01 * s^{-1})$ at the programming temperature T_d, followed by a specified holding time at T_d before being cooled to the shape-fixing temperature T_L (20 °C) at a rate of q $(2.5 * C * min^{-1})$. Once T_L is reached, the specimen is held for 1 hour and then the tensile force is removed. In the free recovery step, the temperature is increased to the recovery temperature T_r at the same rate of cooling and subsequently stabilized for another 50 min. They discovered that the shape fixity is improved when increasing the holding time and at higher temperatures. Usually there is a small bouncing back of the material upon unloading, as shown in Fig. 2.6, which causes some loss in shape fixing. The shape fixity R_f and shape recovery R_r are defined as follows[32][33]:

$$R_f = \epsilon_u / \epsilon_m \tag{2.1}$$

$$R_r = \epsilon_r / \epsilon_m \tag{2.2}$$

Where ϵ_u is the angle the programmed sample has when cooled down, ϵ_m is the desired programmed angle, and ϵ_r is the recovered angle

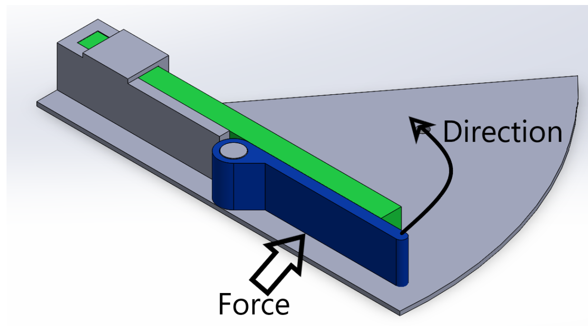

Figure 2.5: Isometric view of the testing structure. The gray parts are 1 print. The blue part will be printed separately and slotted in place afterward. The green part is the testing sample. Direction of the applied force and resulting movement have been highlighted.

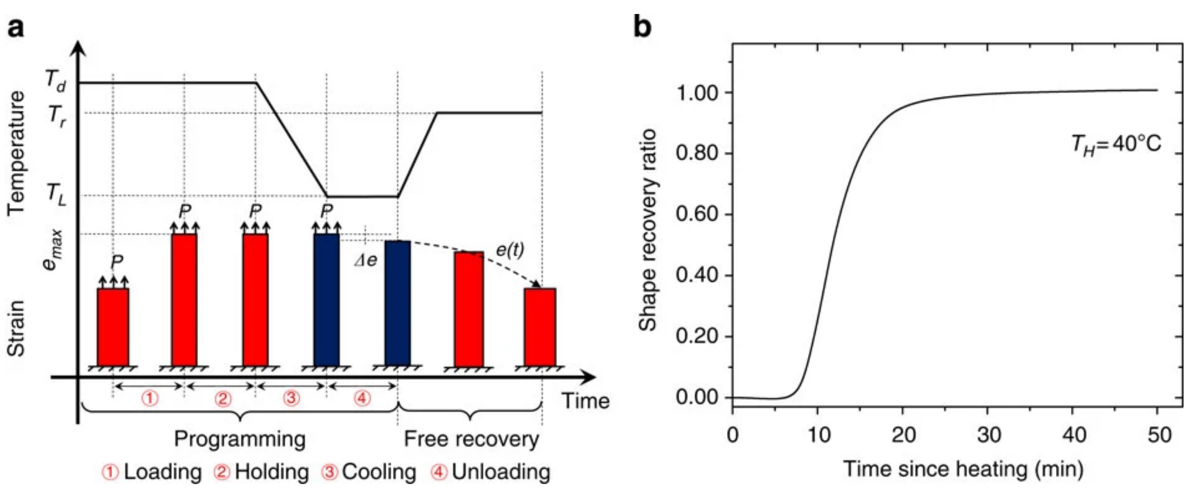

Figure 2.6: (a) A schematic illustration of the thermomechanical history of the programming and free recovery process in an SM cycle. (b) A typical free recovery curve as a function of time.[31]

When combining all the earlier mentioned materials and methods the experiment will run as follows:

- The sample will be FDM printed with parameters according to table 2.1.
- A picture will be taken of the sample.
- The sample's dimensions will be measured with calipers
- The sample will be heated to 100 degrees Celsius.[28]
- The sample will be slotted into the testing structure and be bent under a 45-degree angle.
- The sample will be held in place until cooled down sufficiently[31]
- The sample will be left to cool down to room temperature.
- A picture will be taken of the cooled-down sample.
- The sample will be heated to 100 degrees Celsius and a timer is set to track the time to recovery.
- A picture will be taken of the recovered sample when it has been cooled down to room temperature.

The time required for the sample to heat up above and to cool down below its glass transition temperature is largely dependent on the percentage of infill. Times used in this experiment are recorded in the table below2.3.

Infill [%]	Heating time [min]	Cooling time [min]	Recovery Time [min]
100	2	4	2
50	1	2	1
0	0.33	0.67	0.33

Table 2.3: critical time steps in the experiment

2.2 Data Analysis

Data has to be extracted from the pictures taken during the experiment. This process consists of manually measuring the relevant angles and any other irregular deformations that may have occured. The data obtained from the experiment will then be analyzed and investigated using Python and the pandas, glob, matplotlib, numpy, scipy, IPython, and statistics libraries. For analysis of the experiments, interest lies in the degree to which the independent variable changes the dependent variable. Variables can be related by a linear relationship that is additive across the two data samples. Such a relationship is called covariance. First all independent variables will be examined in order to determine whether these are significantly correlated though the Pearson's correlation. The Pearson's correlation coefficient (r) is calculated as the covariance of the dependent and independent divided by the product of the standard deviation of each data sample. It is the normalization of the covariance between the two variables to give an interpretable score with a range of [-1,1]. A value of -1 indicates a negative correlation, when the independent variable increases the dependent variable decreases, whereas a value of 1 indicates a positive correlation. When the independent variable increases the dependent variable also increases[34][35].

$$cov(X, Y) = (sum(x - mean(X)) * (y - mean(Y))) * 1/(n - 1) \tag{2.3}$$

$$r = \frac{covariance(X, Y)}{stdv(X) * stdv(Y)} \tag{2.4}$$

If the Pearson's correlation indicates correlation, a linear regression, also known as ordinary least squares, will be made for the independent variables to compare with the experiment results. Regression analysis is used in order to predict a continuous dependent variable from a number of independent variables [36]. A linear regression allows us to estimate how the dependent variable changes as the independent variable is changed. This analysis estimates parameters by minimizing the the sum of the squared errors following equation 2.5. Where equation 2.6 is form of the desired linear fit to the observed data [36].

$$SSD = \sum_{i=1}^{n} [y_i - y_{fi}]^2 = \sum_{i=1}^{n} [y_i - (ax_i + b)]^2 \tag{2.5}$$

$$y = a + bx \tag{2.6}$$

Where 'y' is the dependent variable, 'b' is gradient of the independent variable and 'a' which is a fixed number.

In order to test the accuracy of the regression line a t-test is performed in python between the observed data and the regression line. A t-test can be used to determine if two sets of data are significantly different from each other. This hypothesis test will produce a p-value and a t-value. A high p-value indicates that the two sets of data have the same mean. A high t-value indicates that large differences exists between the two sets.[37]

Following from the Pearson's correlation coefficient an coefficient of determination(R^2) is formulated. it is a means of accuracy for a regression line which ranges from 0-1. R^2 is a statistical measure of fit that indicates how much variation of a dependent variable is explained by the independent variable in a regression model. An R^2 of 1, indicates that 100% of the outcomes can be explained by the regression model.[38]

Additionally a root mean squared deviation (RMSD), also known as the root mean squared error(RMSE), is calculated to further investigate the accuracy of the model. The RMSD is a measure of the differences between values of a model and observed values. The lower the value, the better the accuracy is. It has to be noted that the RMSD is scale dependant. It is calculated by taking the difference between each predicted and observed value, squaring each difference, summing them up and dividing the by the number of observed values.[39] This becomes equation 2.7.

$$RMSD = \sqrt{(\frac{\sum(x_e - x_o)^2}{n})} \tag{2.7}$$

If the Pearson's correlation coefficients of two or three of the data sets are significant, a multiple linear regression will be applied to further investigate the relationship between variables[40]. When comparing two independent variables with one dependent variable this multiple linear regression method will construct an trendline with an equation in the form of:

$$q = a + bx + cy + dz \tag{2.8}$$

Where 'q' is the dependent variable, 'b', 'c', and 'd' the gradients of the independent variables, and 'a' which is a fixed number.

Chapter 3

Results and Discussion

3.1 Results

Using the pictures that were taken during the experiment the angles have been measured, as seen in Fig. 3.1-3.2, and converted to fixity- and recovery rates. After the data of the experiment had been extracted scatter plots were made with groups of the independent variables vs the dependent variable for visual inspection which can be found below. Both 2D (fig.[3.3, ..., 3.14]) and 3D plots have been created in order to visually inspect the found data. The 3D plots are accompanied by a regression plane in order to better visualize the data. The covariance has been calculated for all independent variables and can be found in table 3.1 below. A linear regression has been made based on Pearson's correlation coefficient and visual inspection of the scatter plots. Based on the Pearson's correlation coefficients found in table 3.1 only the relation between the fixity rate and layer height, and recovery rate and infill could correlate with values of 0.57 and -0.83 respectively. The temperature does not seem to have any correlation associated with it. The following equations 3.1 and 3.2 have been constructed through linear regression. These linear regression lines have been plotted with their corresponding data points in fig. 3.22 and 3.21 below.

$$FixityRate = 0.889 + 0.307x \tag{3.1}$$

Where x stands for the Layer Height in millimeters

$$RecoveryRate = 0.949 - 0.001x \tag{3.2}$$

Where x stands for the Infill in %

Both these have been analyzed through the statistical t-test and p-test, root mean squared error, and the coefficient of determination. The results can be found below in table 3.2. Multiple linear regression has not been deemed useful for the acquired data with regards to the statistical data relating to the fixity rate and the layer height.

Independent Variable	Dependent Variable	Pearson's correlation coefficient
Layer Height	Fixity Rate	0.57
Layer Height	Recovery Rate	-0.01
Printing Temperature	Fixity Rate	0.15
Printing Temperature	Recovery Rate	-0.10
Infill	Fixity Rate	0.39
Infill	Recovery Rate	-0.83

Table 3.1: Pearson's correlation coefficients

Regression of	t-test	p-test	RMSE	Coefficient of determination
Fixity Rate	43.82	0.00	0.035	0.33
Recovery Rate	5.55	0.00	0.027	0.69

Table 3.2: Statistical analysis of the regression fit

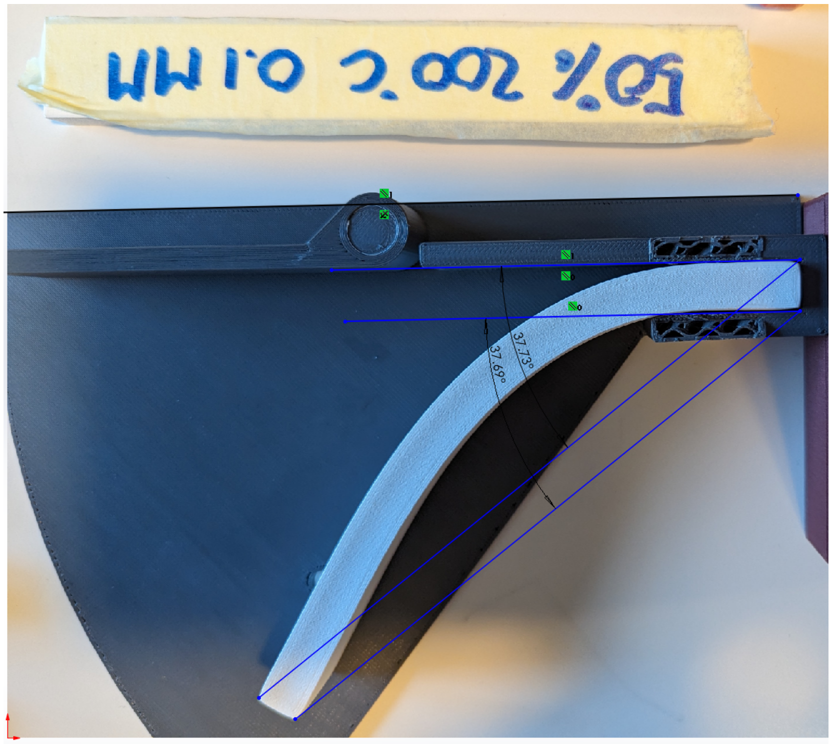

Figure 3.1: Measuring the fixed angle the sample has during the experiment using SolidWorks

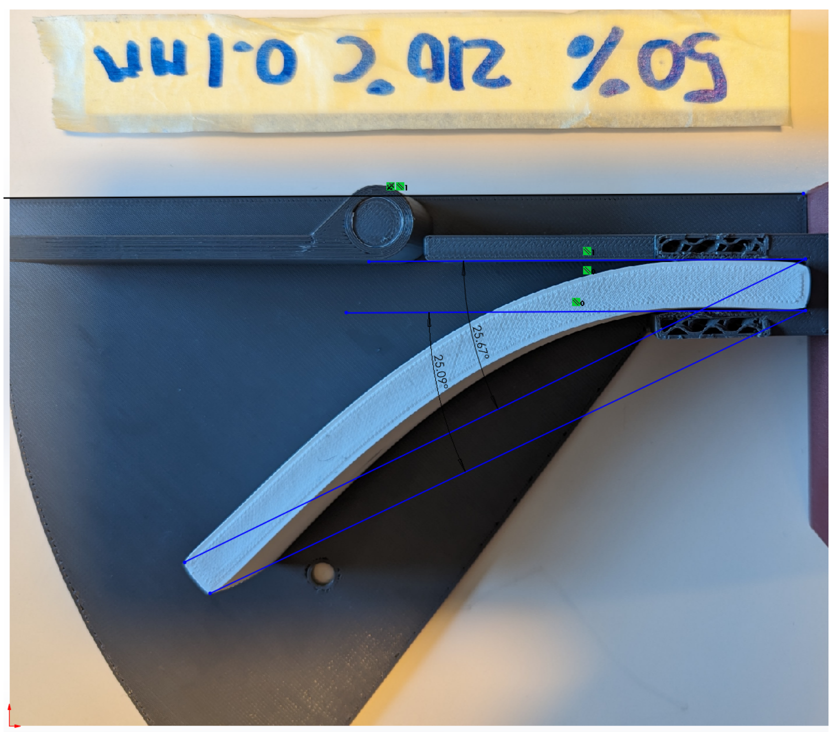

Figure 3.2: Measuring the recovered angle the sample has during the experiment using SolidWorks

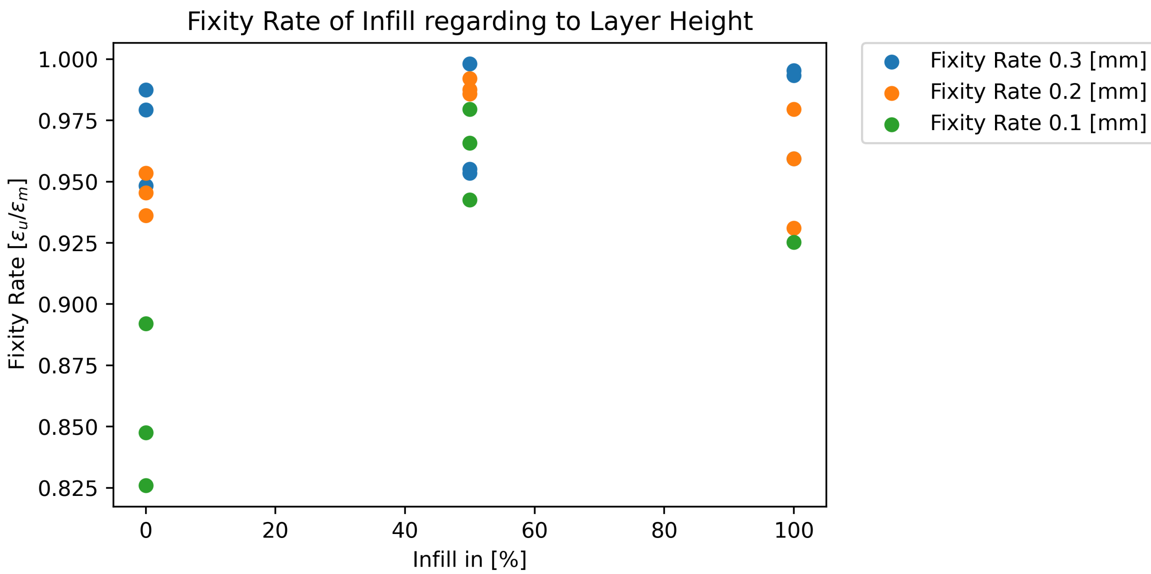

Figure 3.3: Fixity Rate of Infill regarding to Layer Height. The 0.1 mm samples have, on average, a lower fixity rate than the 0.3 mm samples

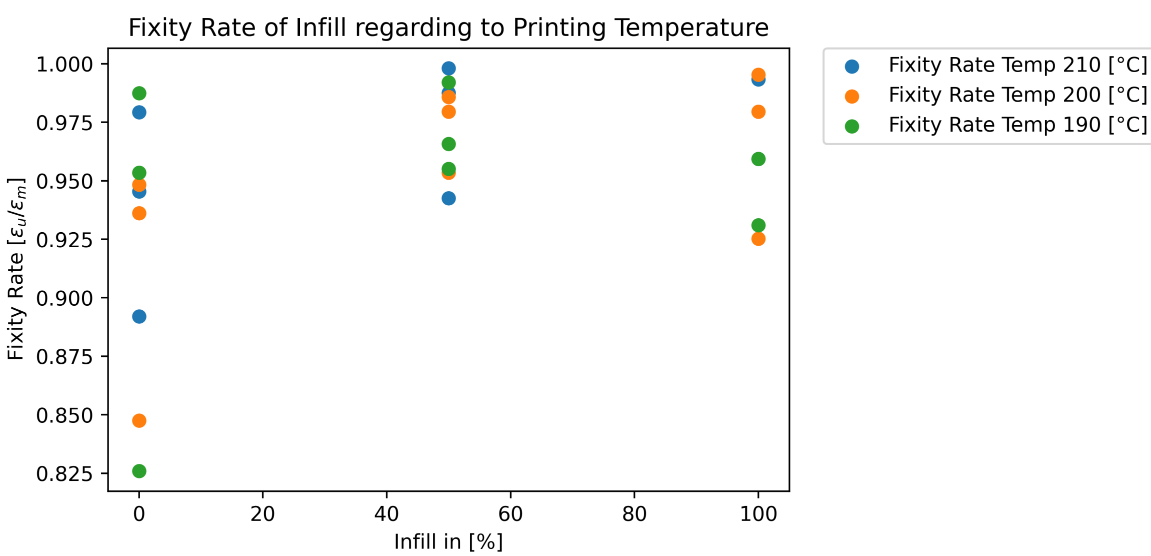

Figure 3.4: Fixity Rate of Infill regarding to Printing temperature. No relation between the points seem to be present

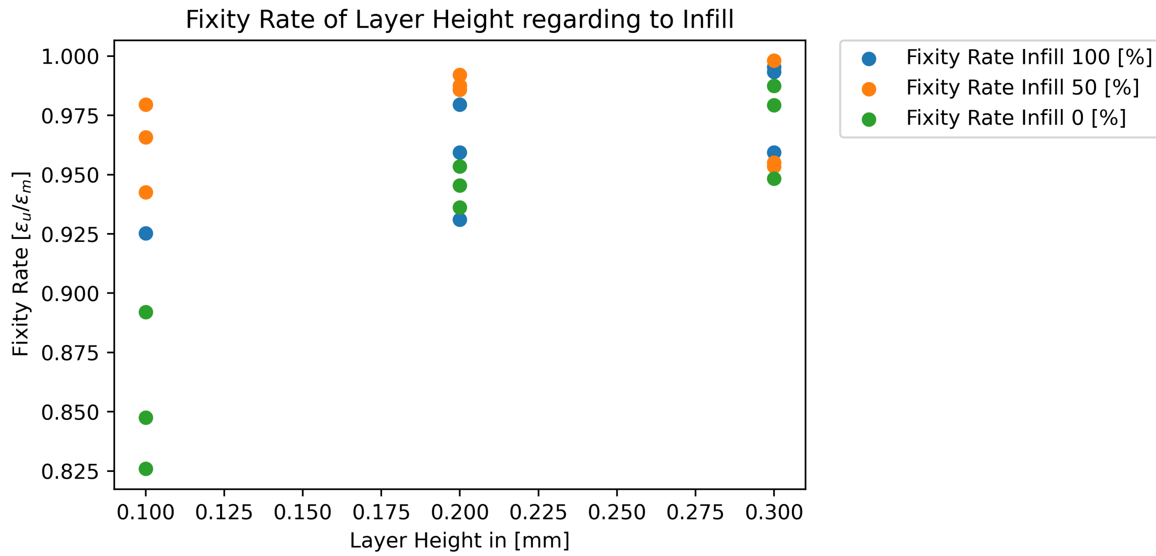

Figure 3.5: Fixity Rate of Layer Height regarding to Infill. 0% infill samples score on average lower on fixity rate than the 50% and 100% samples

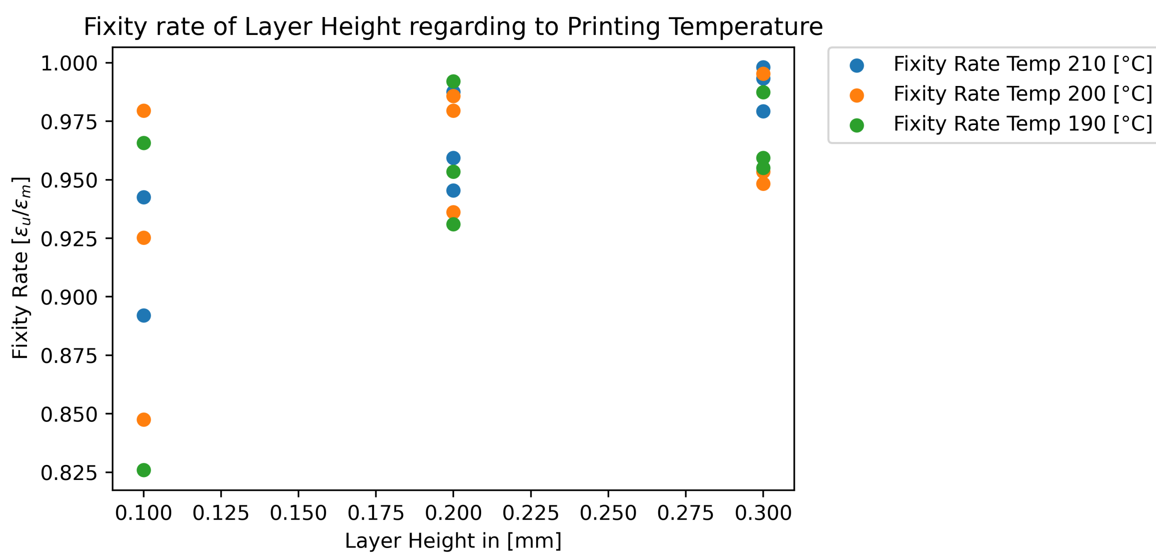

Figure 3.6: Fixity Rate of Layer Height regarding to Printing Temperature. No relation seems present from the data set

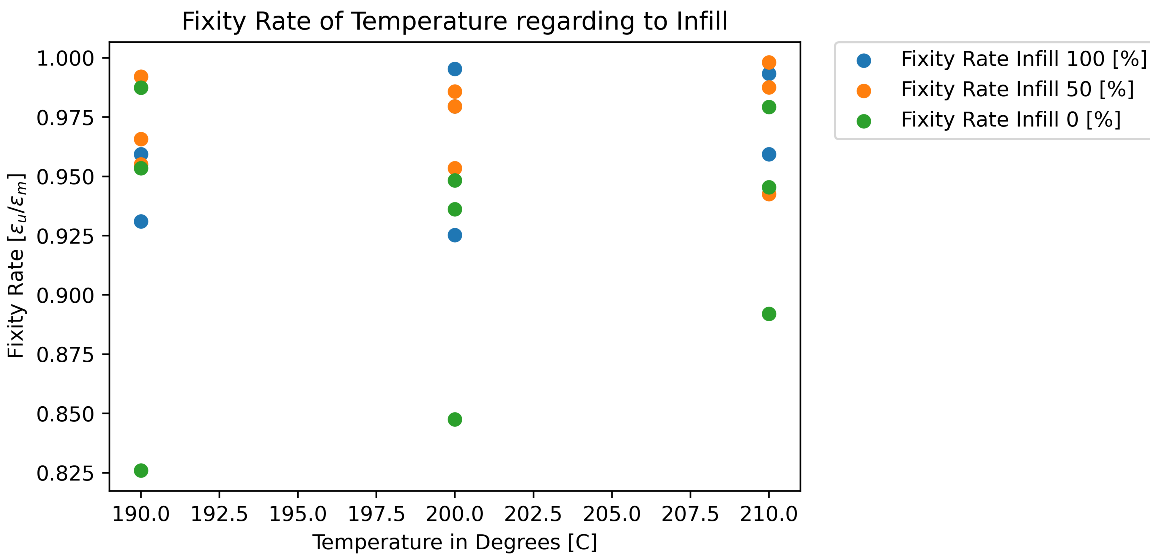

Figure 3.7: Fixity Rate of Printing Temperature regarding to Infill. No relation seems present from the data set

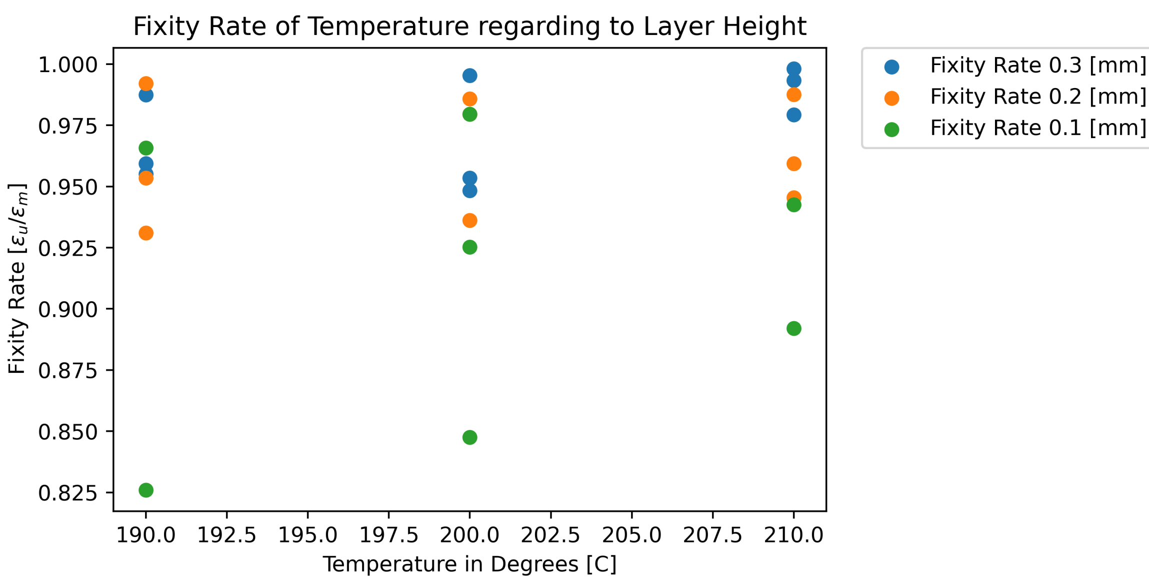

Figure 3.8: Fixity Rate of Printing Temperature regarding to Layer Height. No relation seems present from the data set

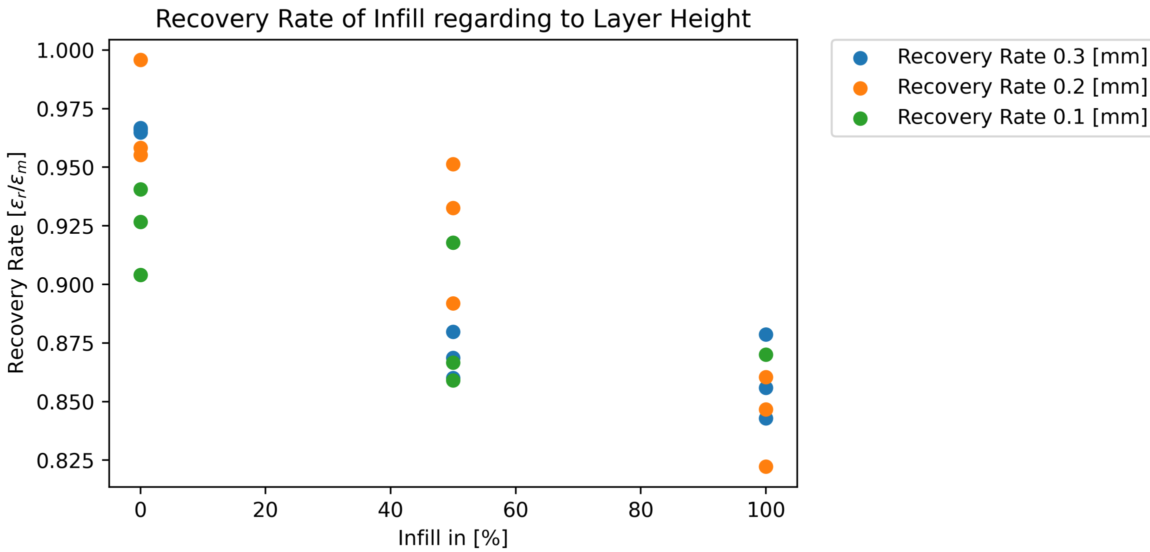

Figure 3.9: Recovery Rate of Infill regarding to Layer Height. A strong linear relation seems apparent from the displayed data set

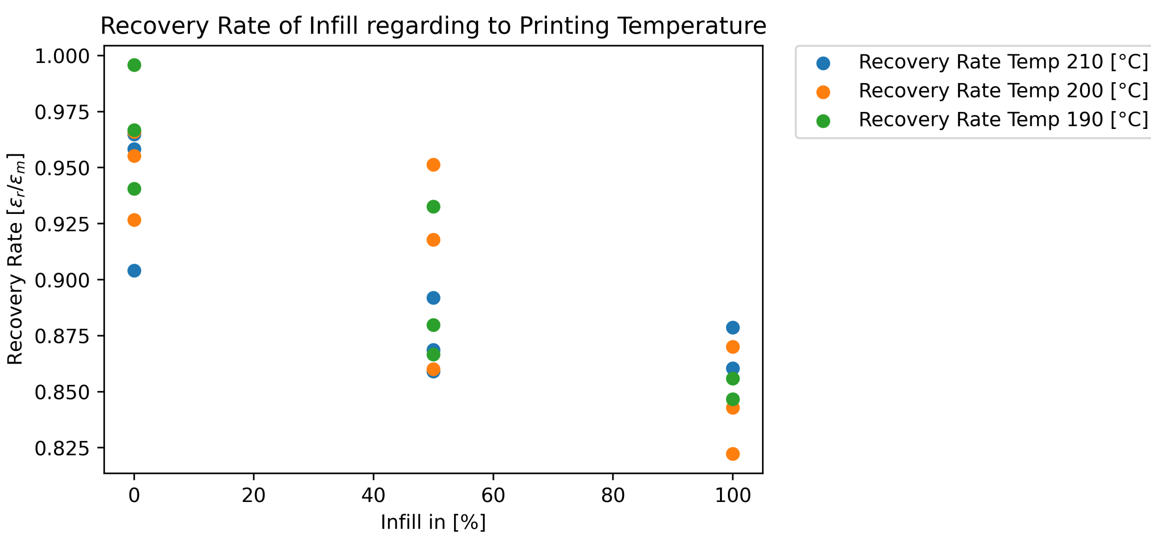

Figure 3.10: Recovery Rate of Infill regarding to Printing Temperature. A strong linear relation seems apparent from the displayed data set

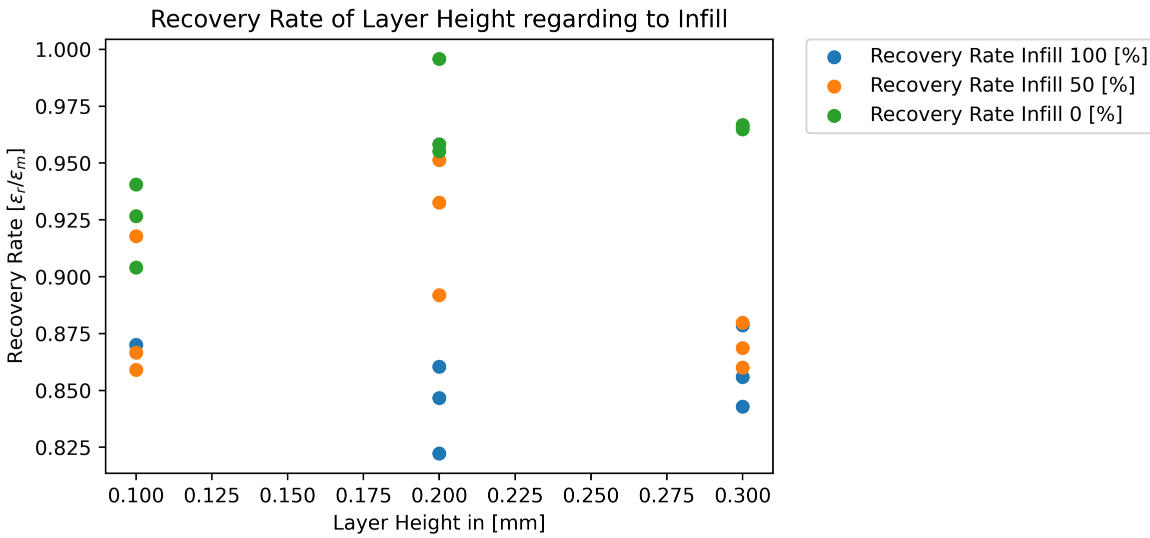

Figure 3.11: Recovery Rate of Layer Height regarding to Infill. On average the 0% infill samples have a better recovery rate than the 50% and 100% infill samples

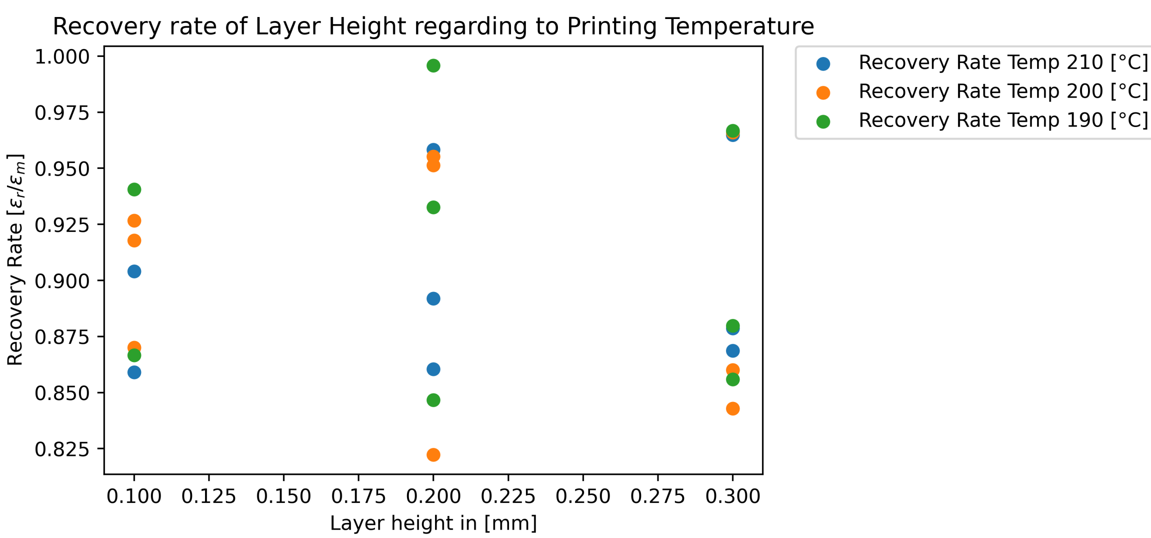

Figure 3.12: Recovery Rate of Layer Height regarding to Printing Temperature. On average the 0% infill samples have a better recovery rate than the 50% and 100% infill samples

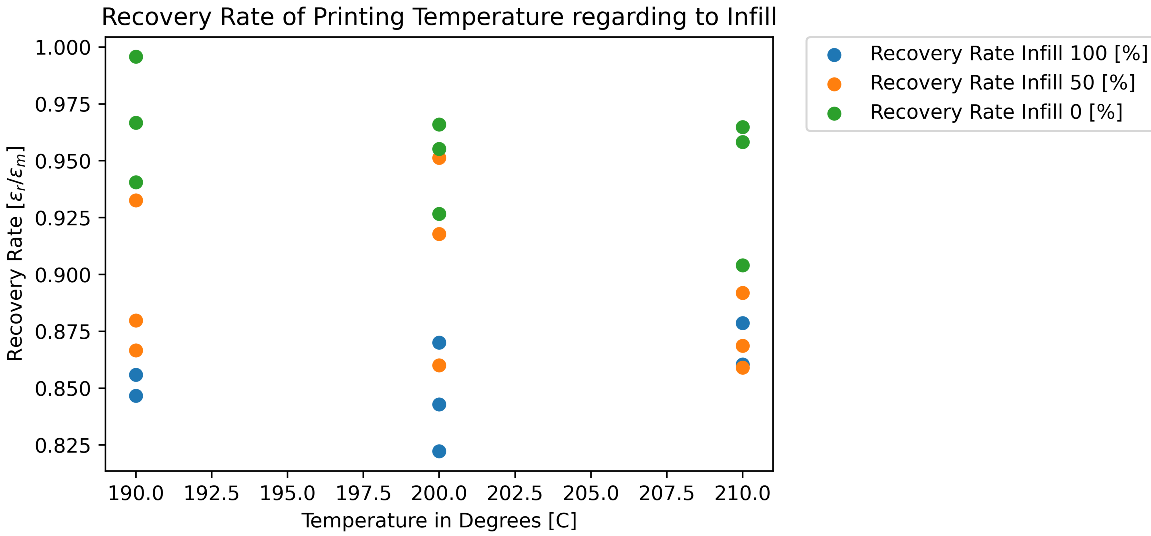

Figure 3.13: Recovery Rate of Printing Temperature regarding to Infill. No relation seems present in this data set

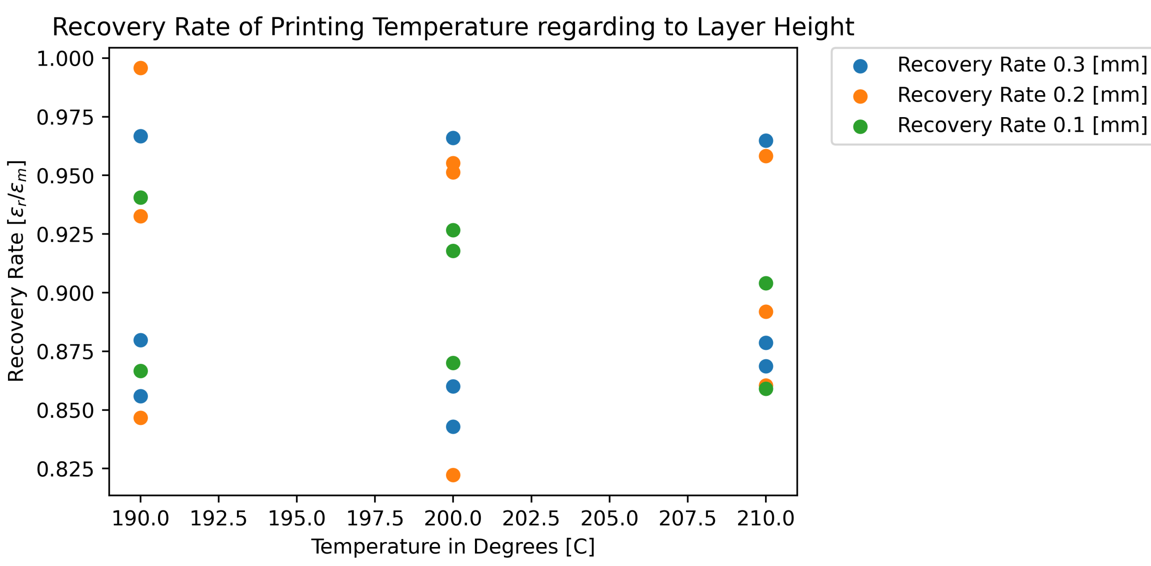

Figure 3.14: Recovery Rate of Printing Temperature regarding to Layer Height. No relation seems present in this data set

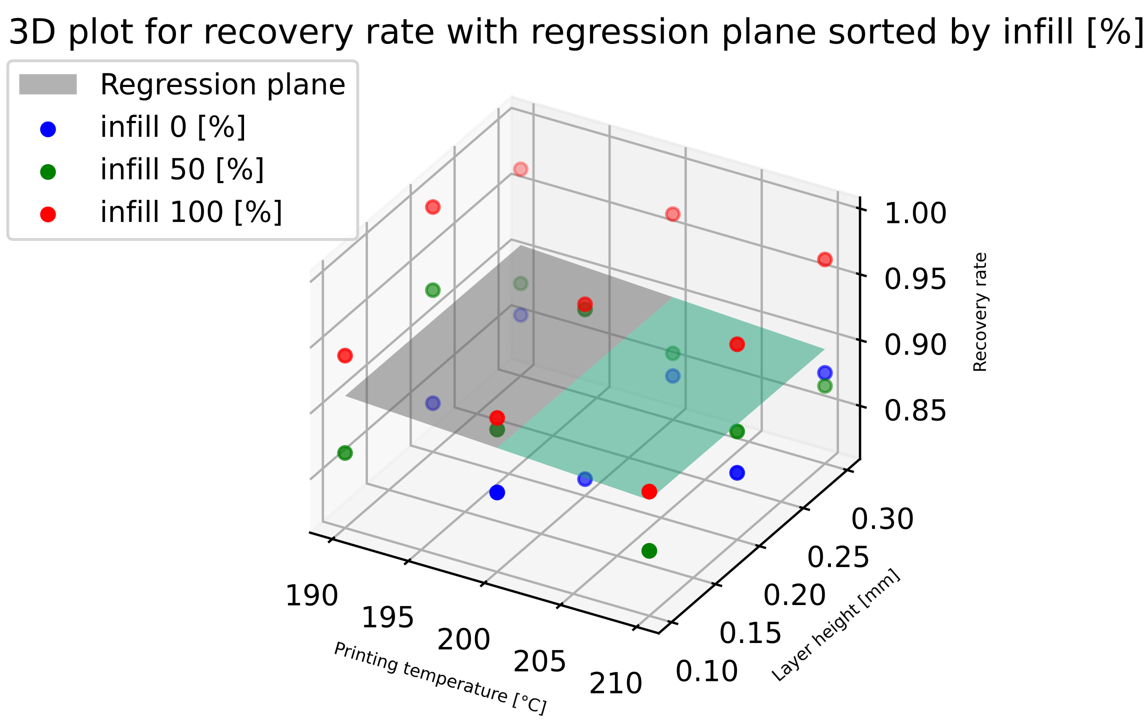

Figure 3.15: 3D plot for Recovery rate with a regression plane sorted by infill[%]. For recovery rate no apparent relation can be seen, with minimal deviations within the plane

3.2 Discussion

From the results we can state that the printing temperature does not affect either the fixity rate or recovery rate to such a degree that is apparent from the tests we have run. The correlation between the fixity rate and the infill can be explained by the line structure that forms between every filled line. For a rectilinear infill pattern, every line in a 100% infill structure is surrounded by 4 other lines(except for the outer lines) which helps to keep each other in check and distribute stress evenly. This structure takes more effort to deform since there is more material that can resist this change to form, but when cooled down can keep its form in a more stable state.

Lengthwise shrinkage has occurred in multiple samples, profoundly so in the 0% infill samples. Shrinkage of 0% up to 30% have been found during testing. This can be linked to the printing speed of the sample, which refers to the speed in mm/s with which the print-head moves during filament deposition. This effect has been studied by Rajkumar et al.[41] and Kauffman et al.[42] previously. They observed that higher printing speeds promote poor print quality with regards to shrinkage rates in multiple materials. For lower printing speeds lower shrinkage strain has been found as seen in fig 3.23

This phenomenon can be attributed to the sparse deposition of material, which leads to an increase in voids within the structure and a reduction in adhesion properties between adjacent lines. Consequently, this diminishes the mechanical performance of the printed object [43][44]. The voids which can form can be seen in fig. 3.24 where a PLA sample has been cut through perpendicular to the print direction. Nonetheless, it was observed that higher printing speeds facilitated an enhanced shape transformation response, despite the concurrent decrease in mechanical properties. One of the solutions to this problem could be to print with a different nozzle which has a larger nozzle diameter. This change in nozzle size leads to an increase in extrusion width. Less lines have to be printed side

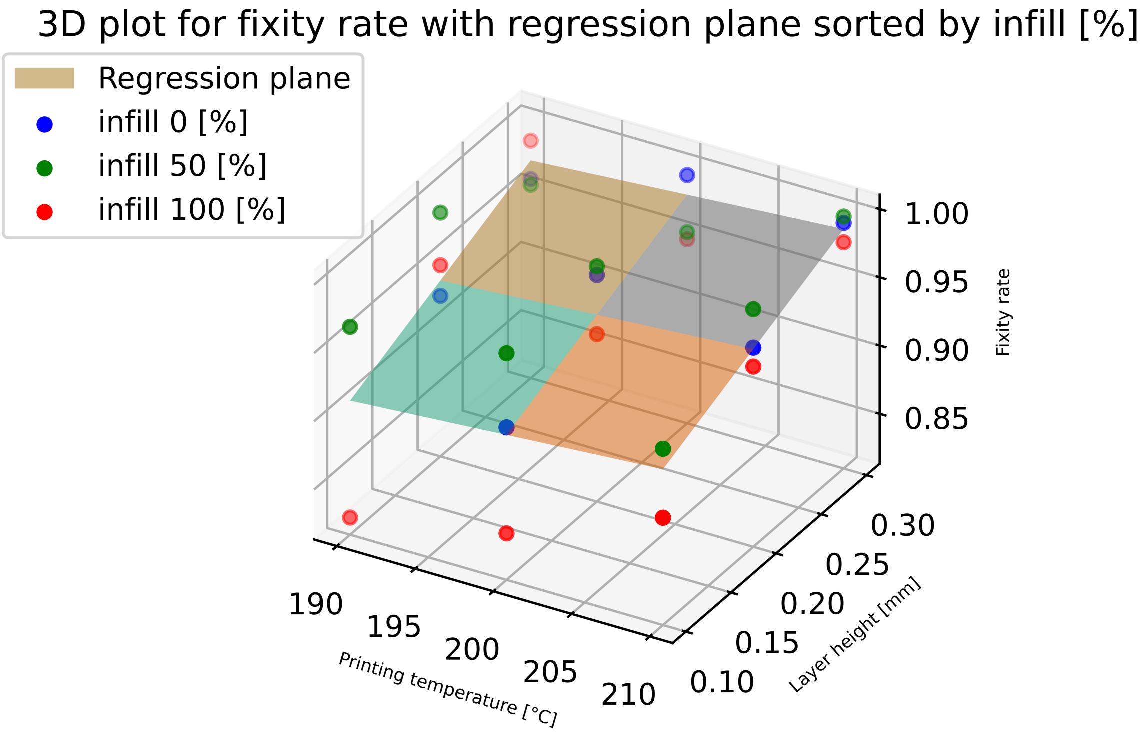

Figure 3.16: 3D plot for Fixity rate with a regression plane sorted by infill[%]. A linear relation between the fixity rate and the layer height seems apparent. Temperature might be loosely related.

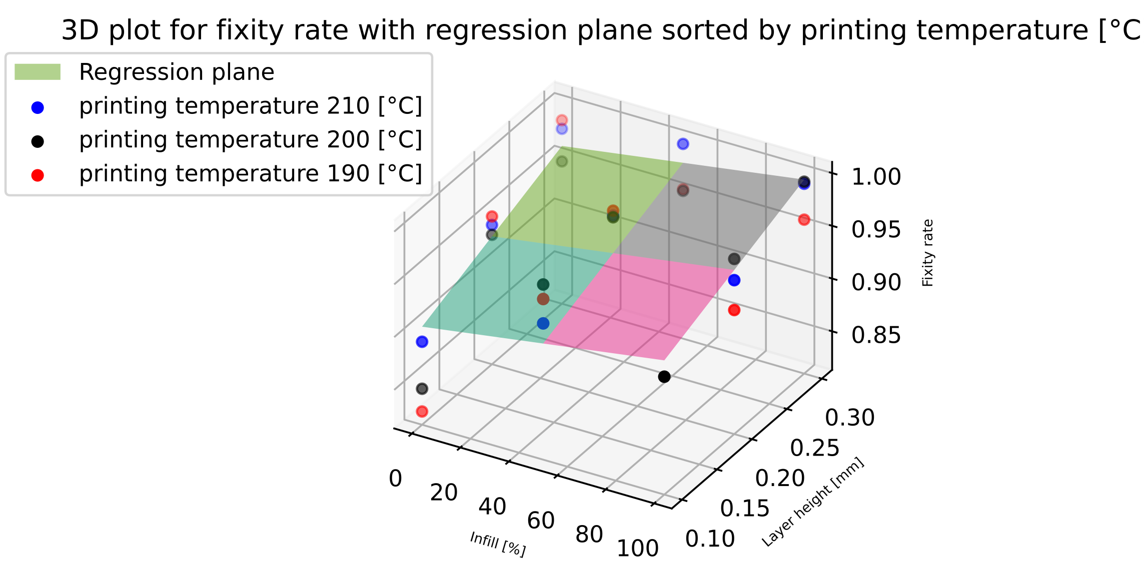

Figure 3.17: 3D plot for Fixity rate with a regression plane sorted by temperature[°C]. A relation between the fixity rate and layer height is apparent. There might also be a relation between fixity rate and the infill

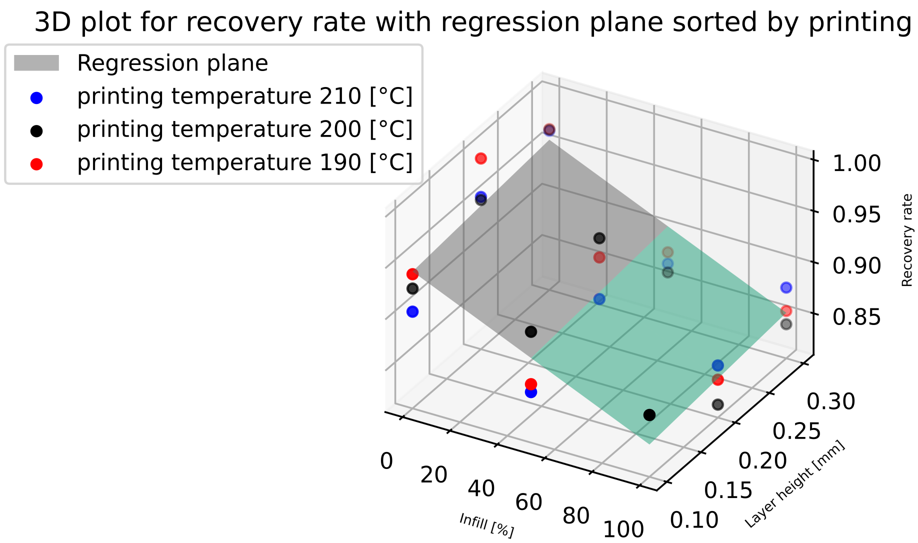

Figure 3.18: 3D plot for recovery rate with a regression plane sorted by temperature[°C]. A clear relation between the recovery rate and infill is displayed. The layer height does not seem to influence the recovery rate

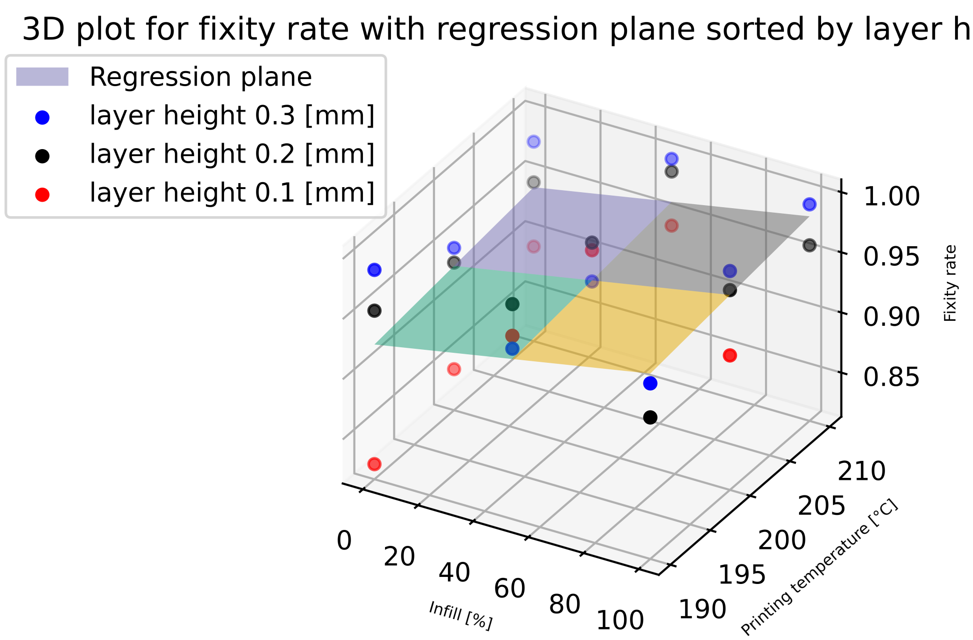

Figure 3.19: 3D plot for fixity rate with a regression plane sorted by layer height [mm]. A minor relation seems visible in the data set. between the infill and fixity rate. Temperature does not affect the fixity rate

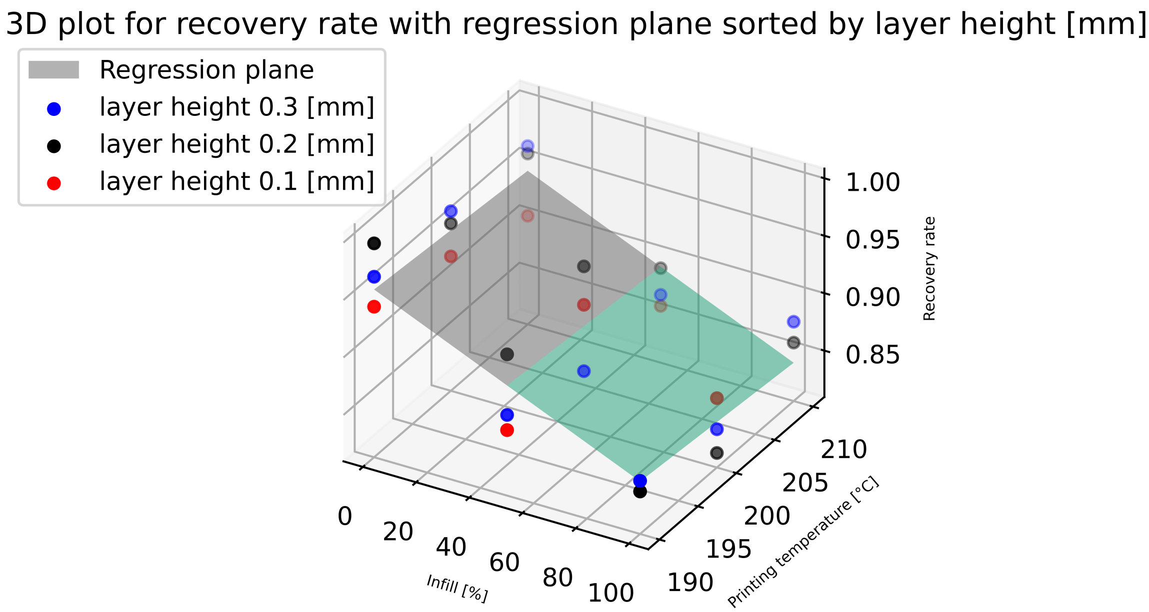

Figure 3.20: 3D plot for recovery rate with regression plane sorted by layer height [mm]. A relation between the recovery rate and infill is visible within the data set. Temperature does not affect the recovery rate

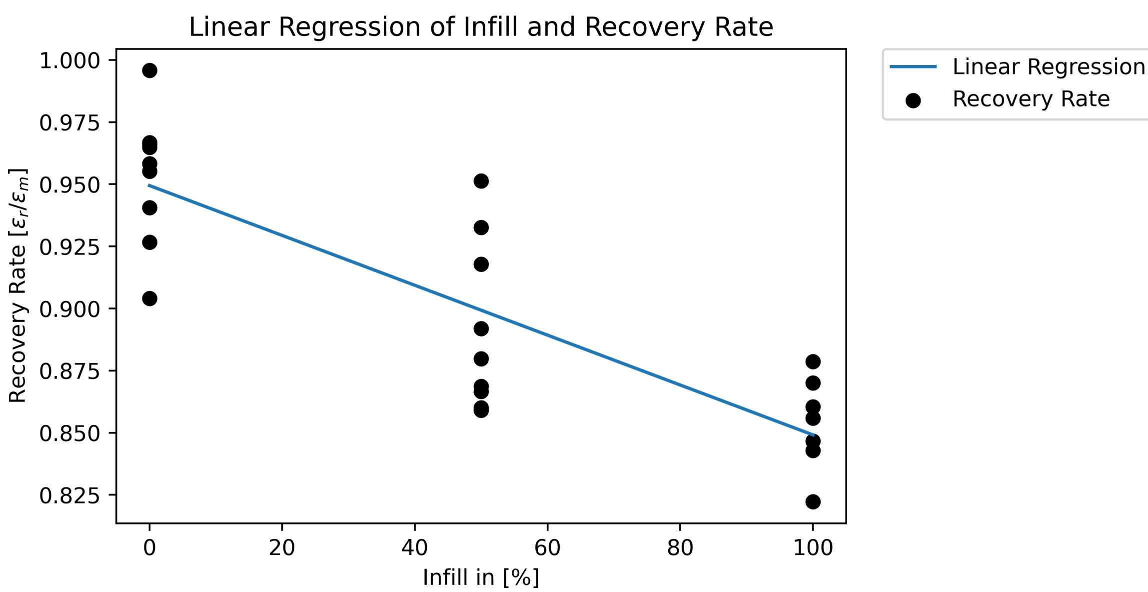

Figure 3.21: Linear regression of the Infill related to the Recovery Rate. A linear relation is apparent within this data set as displayed

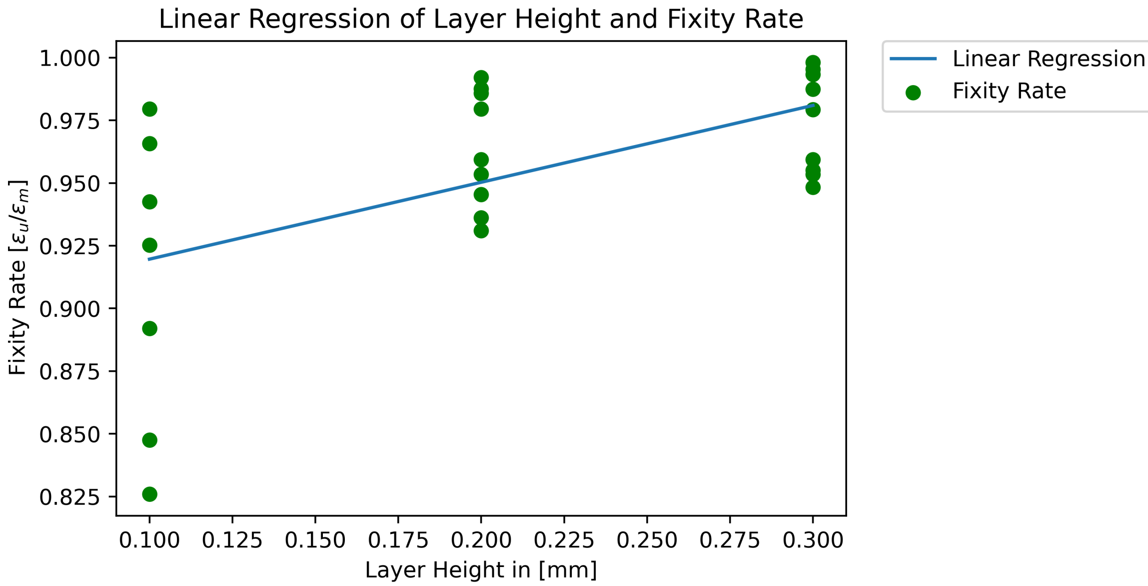

Figure 3.22: Linear regression of the Layer Height related to the Fixity Rate. A relation is apparent within this data set as displayed. It's form seems likely linear. Some of the lowest recorded fixity rates are a result of shrinkage of material which could be further researched how to make a correction factor related to the shrinkage in order to make a more accurate model

by side in order to realize the same geometry which leads to a lower probability of included voids as shown in fig. 3.25

Rajkumar et al. [41] also concluded based on other work that self-bending action is dominated by the print direction of the top layers. When infill density is decreased the top layers can shrink more because they have less material surrounding them which entails fewer constraints to deformation. This effect can be seen in fig. 3.26 and fig. 3.27 of my experiment. Multiple 0% infill samples printed at 0.1mm layer height exhibited this self-bending behaviour. At 0.2mm and 0.3mm this effect was significantly less noticeable.

When increasing the layer height from 0.1mm to 0.3mm using a sample of the same dimensions will results in a increase in total number of deposited lines. PLA filaments will increase their length and decrease their diameter from 1.75mm to 0.4mm during extrusion[45], however part of its previous form is retained by the SME. This causes the effect that for samples printed at 0.3mm a noticeable increase in their height was found during testing ranging from 0.5 to 2 mm. The layers try to expand to their previous spooled form which entails an increase in diameter of each individual layer line. This theory is further supported two more observations made during testing: the tips of the samples printed at a layer height of 0.3mm bend up and outwards and when printed at 0% infill the print begins to resemble a tube like shape. Because the corners of the samples are weaker bound in their plane they have the tendency to move out of plane due to the swelling/SME that occurs. The tube like shape with the walls starting to bulge outwards, such as in fig. 3.29, probably occurs due to the SME, the layer lines start to expand, shrinking their length and increasing their thickness. Because there is no central structure binding the walls together the walls start to expand outwards to relieve their own stress. The 0% degree also displayed a concave surface that probably occurs due to the swelling and expanding of the sides. When these start to deform the top and bottom side also want to deform, but the perimeters of the top and bottom side are forced inward due to the expanding walls. This causes the top and bottom side to expand inwards. This swelling also caused the samples to grow in size beyond what was able to fit under the bridge part of the testing structure and it was necessary to

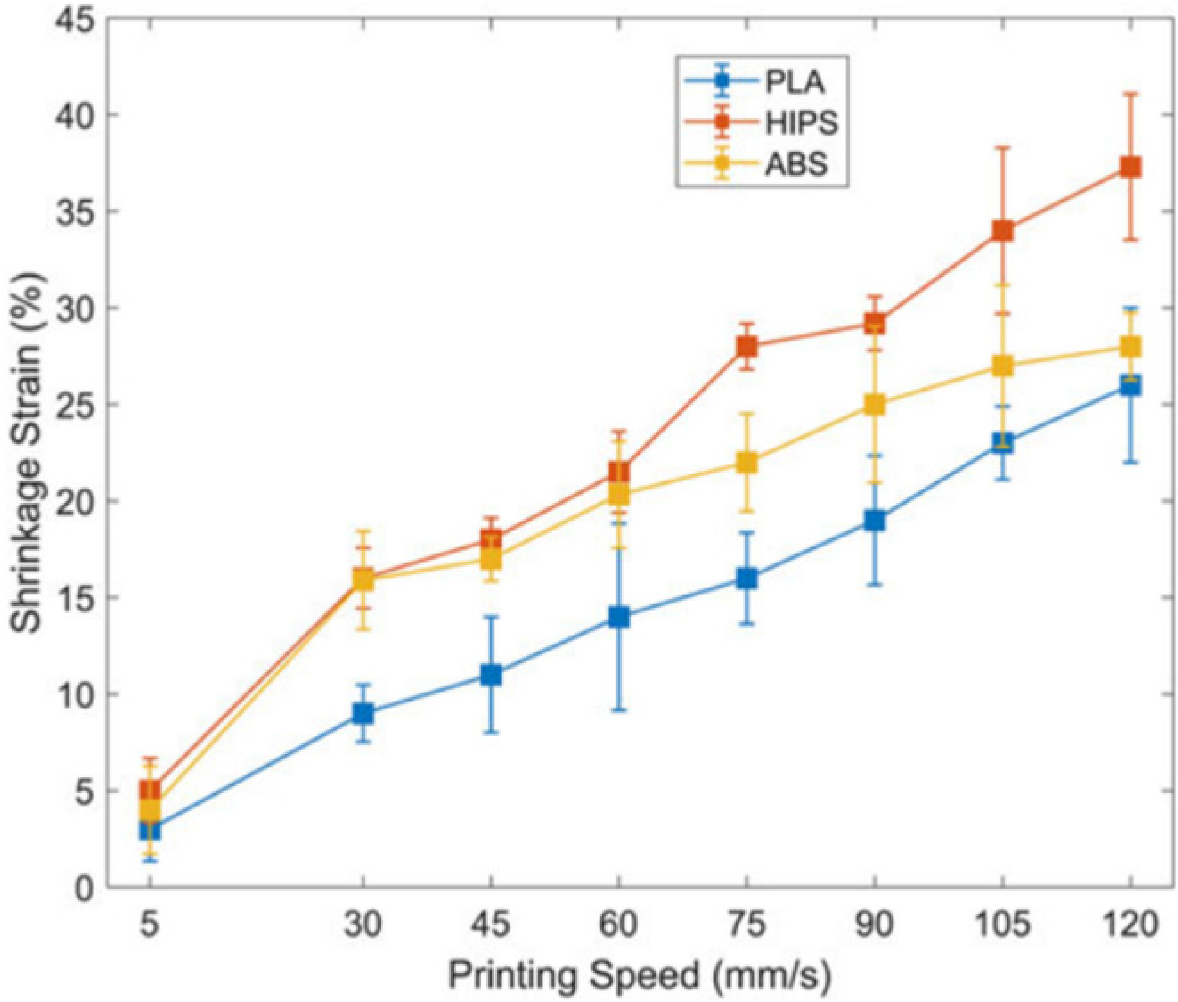

Figure 3.23: Effect of printing speed on the shrinkage strain [41]. Material from: Rajkumar, A.R.; Shanmugam, K, Additive manufacturing-enabled shape transformations via FFF 4D printing, published (2018)

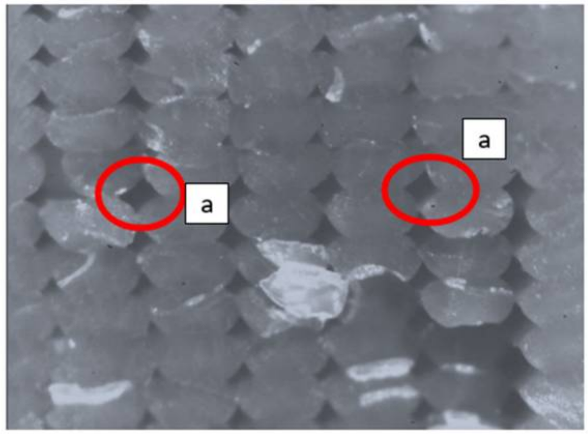

Figure 3.24: Defect appearance on a PLA specimen: (a) material voids between adjacent lines. Reused from [43]

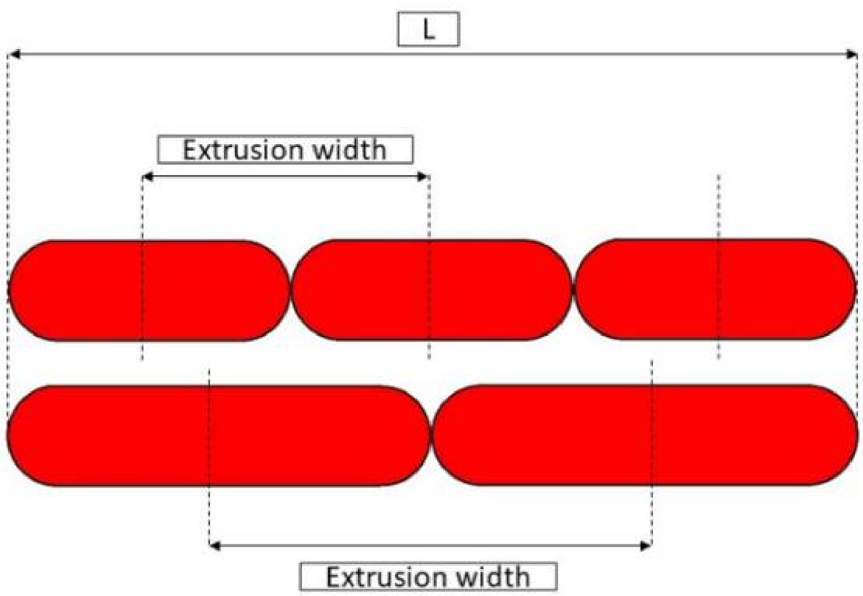

Figure 3.25: Lower extrusion width (on top) and higher extrusion width (on bottom) compared for a given dimension L. Reused from [43]

Figure 3.26: 0% infill sample printed at a layer height of 0.1mm after the programming step. Notice the concave top layers of the sample and the bulging sided

Figure 3.27: 0% infill sample printed at a layer height of 0.1mm after the programming step. This sample also has concave top layers and out of plane curving ends

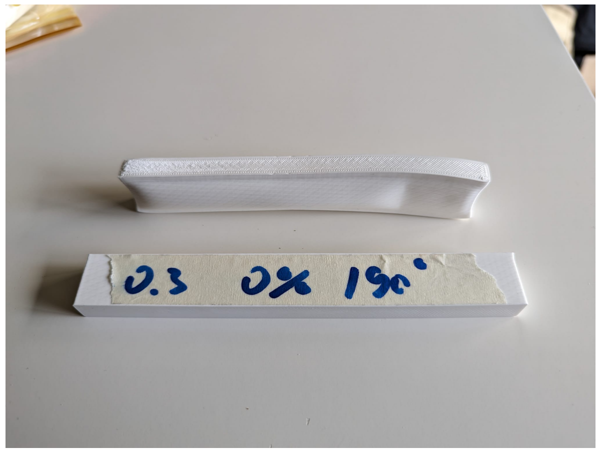

Figure 3.28: 0% infill sample printed at 0.3mm layer height after the recovery step. The sample is exhibiting swelling and bulging of the sides. the short end have become concave

Figure 3.29: 0% infill sample printed at 0.3mm layer height after the recovery step as seem from above. exhibiting swelling and bulging of the sides

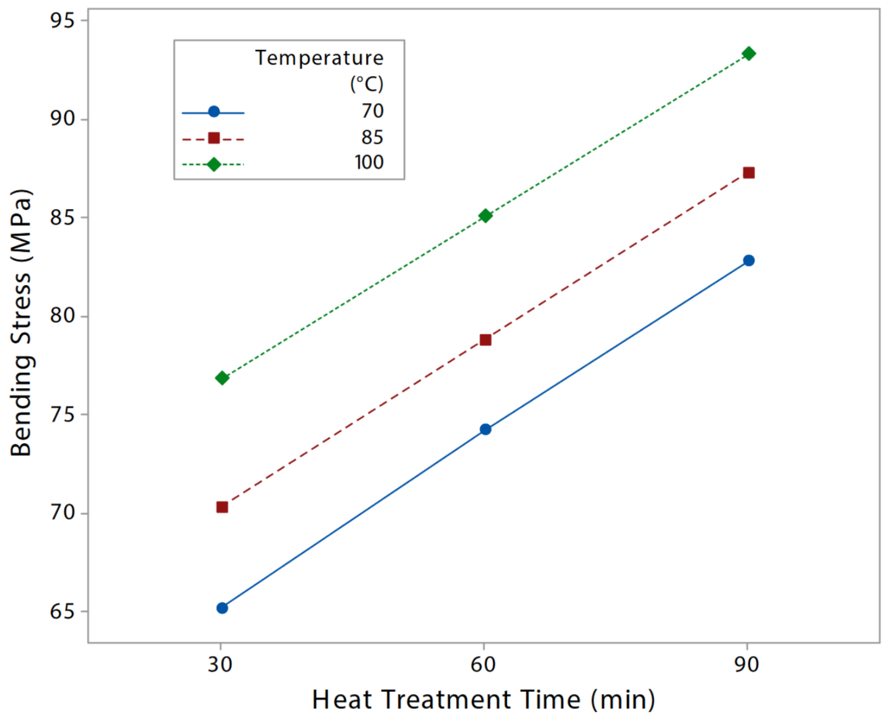

Figure 3.30: Effect of heat treatment on bending stress. Reused from [46]

remove this part of the testing structure as a result.

Two 100% infill samples became extremely hard and tough when submitted to the programming cycle. After two minutes in the water, which was the same as the other 100% infill samples, they became hard to the touch and extremely hard to bend. So much so that a part of the testing structure shattered because of the required force to bend the sample. The fixity rate was so low that the values have not been used in the rest of the experiment. Recovery was not possible after cooling and heating the two samples.

Kartal et al.[46] have studied the effect of annealing of PLA under different circumstances. They annealed samples for 30, 60 or 90 minutes at a temperature of 70, 85 and 100 degrees celcius. Furthermore they stated that:"According to the results, it has been shown that with increasing heat treatment temperature and duration, there are significant improvements in the mechanical properties of PLA plastics, such as tensile strength, elastic modulus, Shore D hardness value, and bending stress. In particular, applying heat treatment at 85°C and for 90 minutes enabled PLA plastics to achieve the highest mechanical properties"[46]. The 100 degree Celcius annealed samples reach a Shore D hardness value of 75-76, as opposed to the average of 69 [47]. The Bending stress also increased to 77 93 MPa.Particularly, after applying a thermal treatment of 85 °C and 90 minutes duration, the highest flexural stress values (93 MPa) were reached for PLA plastics, as shown in Fig. 3.30. This annealing process is probably what happened on a small scale to the 100% infill prints when comparing the findings of the study done by Kartal et al. and my observations.

Chapter 4

Conclusion

The research questions that were stated in the beginning of this paper were as follows

- How to control the SME of PLA through printing parameters?

This will be answered through the following sub questions:

- What are the mechanical and chemical properties of PLA and at what temperature does the SME occur?
- What are the printing parameters of an FDM printer and how will they be modified?
- How do we qualitatively describe the SME parameters?
- What test setup will be used to deform the test structure?
- What relations can be found between parameters?

The following answers and conclusions have been found during this study. The SME of PLA is possible to influence and control when applying different values of the infill and layer height used for the testing structure. The printing temperature does not seems to have a significant influence on either the fixity or recovery rate. This means that the printing temperature can be adjusted independently of the required effect of the SME. The infill mainly affects the recovery rate of the samples and the layer height mainly affects the fixity rate of the samples. Depending on the application a focus can be made on either property. When the structure has to fit through a narrow opening the fixity rate can be more important than the recovery rate. Vice versa when the final state of the structure is critical to the application, such as a stent placed in a body, a focus should be on maximizing the recovery rate. PLA is quite a resilient material, while being cheap and widely available. The SME of PLA occurs around 65 °C. The fixity rates and recovery rates have been described in eq. 2.2 and 2.1 as ratios of desired and resulting angles in order to quantitatively describe the effects the printing parameters have on the SME. The testing structure has been displayed in Fig.2.5. The found relations between fixity rate and layer height, and recovery rate and infill both seem linear in nature. These relation can be used for constructing a model with which expected behaviour of 4D printed parts and structures can be predicted. Which in turn enables better options for designing parts.

4.1 Future prospects and reflection

The results of this thesis offer new insights on the interplay between the dependent (fixity and recovery rate) and independent (printing temperature, layer height and infill) variables which can be leveraged for future research in 4D printing. The printing temperature has no apparent affect on either the fixity rate or recovery rate and as such does not influence the quantified aspects of the shape memory affect of PLA. This leaves room for the user to adjust the extrusion temperature based on mechanical properties required for the application, since characteristics such as tensile strength can be influences by this parameter as shown by Ansari et al.[48]. Correlation between the fixity rate and layer height, as well as recovery rate and infill have been observed, analyzed and discussed. A linear regression has been found for these correlations showcased in equation 3.1 and 3.2. These regressions could be used to construct a comprehensive model to simulate expected behaviour of 4D printed parts and structures.

The lengthwise shrinkage which was prominent in the 0% infill samples are attributed to the printing speed, especially of the top layers as shown in earlier studies done by Rajkumar et al.[41]

and Kauffman et al.[41]. This shrinkage is attributed to voids inside the structure due to the speed, however while it compromises mechanical properties the structure does have a stronger shape memory effect. One solution to increase or decrease the amount of voids inside of a structure is to extrude material through a differently sized nozzle. This can increase or decrease the amount of self-bending action the 4D printed part experiences. Further research could be done with applying different printing speeds to different parts of a structure to pre-program a structure with shape memory effects.

When the layer height increased the printed samples expanded more due to the shape memory effect the PLA has in its filament form, a round 1.75mm diameter spooled wire. This expansion leads to observable changes in the samples geometry, including bending, bulging and shrinking. The type of geometry change is linked to both the infill and layer height, since an internal structure restricts movement of the geometry. Further research could focus on the geometry changes and at what critical levels of infill drastic changes in behaviour can be found, in particular at the 0% to 15% infill levels.

When heating the 0.1mm, 100% infill samples above the glass transition temperature a chance to accidentally anneal the samples can occur. The root cause for this is once again the voids which can be introduced during printing. At a small layer height with a 100% structure the entire structure has a low number of voids in its system. When heated the layers will start to merge together and start to form a solid part and with that change its accompanying mechanical properties[46]. While this is great for structural integrity, it also reduces the shape memory effect. Further research can investigate how this annealing can be applied for 4D structures that have to bear loads when placed.

When regarding how the findings of this paper can be applied in practice we look at three industries: The aerospace, medical and automotive industry. In the aerospace industry lightweight and durable components are critical, The ability to control shape memory effect of 4D printed structures could change and optimize design and manufacturing processes. For example a 4D printed wing structure could be designed with properties tailored to specific flight conditions, such as temperature in order to optimize performance and fuel efficiency[49]. In the medical field the exciting possibility of personalized healthcare through 4D printed implants could be possible. Medical practitioners can fine tune printing parameters in order to achieve an optimal balance between fixity rate, recovery rate and mechanical properties tailored to the patient[50]. For instance, orthopedic implants can be designed to expand or contract to fit the contours of a patient´s bone, enhancing stability and promoting faster healing. In the automotive industry manufacturers can apply the found knowledge to produce components with shape memory properties to improve crash-worthiness and energy absorption during incidents[51]. For example 4D printed bumpers or similar crash structures could be designed to deform during an accident and return to their original shape when heated.

Another purpose for the 4D printed parts could be in the bearing industry. Plastic bearings are widely used, but can be prone to certain issues, such as degradation and extreme loads. When applying 4D printing a structure can be made with inclusions for self-lubricating properties and self healing capabilities for an extended lifetime of the bearing.

The found relations can also be useful for applications regarding space travel. One of the issues with space travel is weight, because the heavier the system is that will be launched, more fuel and resources have to be spent in order to achieve flight to outer space. When a part has to adjust to different surroundings, such as temperature or pressure, a 4D printed part which is able to control its shape and function based on these stimuli can help with lowering the overall weight of a space faring vessel.

Large scale applications such as self assembling furniture could be realised. When a material with a suitable glass transition point is chosen, it could be possible to have an entire closed arrive within a flat package which has to be opened, left inside a room of the appropriate temperature, which will start the SME and the closet will self assemble without further help. The found relations could be used to find a combination of fixity rate, recovery rate and required mechanical properties to enable such a design.

The fashion industry could experience a major upheaval with the possibility to have clothing or accessories which are able to transform based on the current situation. Think of earrings shaped like a flower which are able to open and close based on the surrounding temperature. Shoes could be designed in order to deliver more comfort and start to open up when the temperature starts to rise to allow more air in and preventing sweaty feet. A chain mail structure like shirt could be printed which can shrink and expand based on temperature and humidity in order to enhance the wearing experience of the user. Keeping its structure closed in cold temperatures and creating gaps within the meta structure to enhance breath-ability of the clothing.

The practical implications of this research extend across diverse industries, offering opportunities for innovation and advancement. By leveraging the power of 4D printing, researchers and engineers can unlock new possibilities for flexible designs, performance optimization, and personalized solutions, ultimately shaping the future of additive manufacturing and 4D printing technology.

Bibliography

[1] I. Gibson, D. Rosen, B. Stucker, and M. Khorasani, *Additive Manufacturing Technologies.* 2021.

[2] C. Hull and C. W. *U.S. Pat. 4575330, 1986.*

[3] S. A. Tofail, E. P. Koumoulos, A. Bandyopadhyay, S. Bose, L. O'Donoghue, and C. Charitidis, "Additive manufacturing: scientific and technological challenges, market uptake and opportunities," *Materials Today*, vol. 21, no. 1, pp. 22–37, 2018.

[4] C. M. González-Henríquez, M. A. Sarabia-Vallejos, and J. Rodriguez-Hernandez, "Polymers for additive manufacturing and 4D-printing: Materials, methodologies, and biomedical applications," *Progress in Polymer Science*, vol. 94, pp. 57–116, 2019.

[5] J. K. Choi, A. Zhu, B. G. Jenkins, S. Hattori, K. E. Kil, T. Takagi, S. Ishii, T. Miyakawa, and A. L. Brownell, "Combined behavioral studies and in vivo imaging of inflammatory response and expression of mGlu5 receptors in schnurri-2 knockout mice," *Neuroscience Letters*, vol. 609, pp. 159–164, 2015.

[6] W. Gao, Y. Zhang, D. Ramanujan, K. Ramani, Y. Chen, C. B. Williams, C. C. Wang, Y. C. Shin, S. Zhang, and P. D. Zavattieri, "The status, challenges, and future of additive manufacturing in engineering," *CAD Computer Aided Design*, vol. 69, pp. 65–89, 2015.

[7] Y. Y. C. Choong, S. Maleksaeedi, H. Eng, S. Yu, J. Wei, and P. C. Su, "High speed 4D printing of shape memory polymers with nanosilica," *Applied Materials Today*, vol. 18, p. 100515, 2020.

[8] G. X. Gu and M. J. Buehler, "Tunable mechanical properties through texture control of polycrystalline additively manufactured materials using adjoint-based gradient optimization," *Acta Mechanica*, vol. 229, no. 10, pp. 4033–4044, 2018.

[9] A. Nishiguchi, H. Zhang, S. Schweizerhof, M. F. Schulte, A. Mourran, and M. Möller, "4D Printing of a Light-Driven Soft Actuator with Programmed Printing Density," *ACS Applied Materials and Interfaces*, vol. 12, no. 10, pp. 12176–12185, 2020.

[10] F. Libonati, G. X. Gu, Z. Qin, L. Vergani, and M. J. Buehler, "Bone-Inspired Materials by Design: Toughness Amplification Observed Using 3D Printing and Testing," *Advanced Engineering Materials*, vol. 18, no. 8, pp. 1354–1363, 2016.

[11] F. P. Melchels, M. A. Domingos, T. J. Klein, J. Malda, P. J. Bartolo, and D. W. Hutmacher, "Additive manufacturing of tissues and organs," *Progress in Polymer Science*, vol. 37, no. 8, pp. 1079–1104, 2012.

[12] X. Kuang, *Introduction to 4D printing: Methodologies and materials.* Elsevier Inc., 2021.

[13] F. Kabirian, P. Mela, and R. Heying, "4D Printing Applications in the Development of Smart Cardiovascular Implants," *Frontiers in Bioengineering and Biotechnology*, vol. 10, no. May, pp. 1–11, 2022.

[14] M. Javaid and A. Haleem, "4D printing applications in medical field: A brief review," *Clinical Epidemiology and Global Health*, vol. 7, no. 3, pp. 317–321, 2019.

[15] L. Sun, T. X. Wang, H. M. Chen, A. V. Salvekar, B. S. Naveen, Q. Xu, Y. Weng, X. Guo, Y. Chen, and W. M. Huang, "A brief review of the shape memory phenomena in polymers and their typical sensor applications," 6 2019.

[16] M. Vert, J. Chen, K.-H. Hellwich, P. Hodge, T. Nakano, C. Scholz, S. Slomkowski, and J. Vohlidal, "Nomenclature and terminology for linear lactic acid-based polymers (iupac recommendations 2019)," *Pure and Applied Chemistry*, vol. 92, no. 1, pp. 193–211, 2020.

[17] O. Martin and L. Avérous, "Poly(lactic acid): plasticization and properties of biodegradable multiphase systems," *Polymer*, vol. 42, no. 14, pp. 6209–6219, 2001.

[18] Ceresana., "Bioplastics - study: Market, analysis, trends," *https://ceresana.com/en/produkt/market-study-bioplastics.*

[19] W. M. Huang, Z. Ding, C. C. Wang, J. Wei, Y. Zhao, and H. Purnawali, "Shape memory materials," *Materials Today*, vol. 13, no. 7-8, pp. 54–61, 2010.

[20] X. Wu, W. M. Huang, Y. Zhao, Z. Ding, C. Tang, and J. Zhang, "Mechanisms of the shape memory effect in polymeric materials," *Polymers*, vol. 5, no. 4, pp. 1169–1202, 2013.

[21] C. C. Hornat, M. Nijemeisland, M. Senardi, Y. Yang, C. Pattyn, S. van der Zwaag, and M. W. Urban, "Quantitative predictions of maximum strain storage in shape memory polymers (smp)," *Polymer*, vol. 186, p. 122006, 2020.

[22] B. B. Samal, A. Jena, S. K. Varshney, and C. S. Kumar, *4D printing: An experimental case study on processing of shape memory polymer by FDM/FFF for nature inspired structures.* INC, 2023.

[23] Matweb, "MatWeb, Your Source for Materials Information," *MatWeb*, pp. 1–2, 2015.

[24] S. Jacobsen and H. Fritz, "Plasticizing polylactide - the effect of different plasticizers on the mechanical properties," *Polymer Engineering and Science*, vol. 39, no. 7, p. 1303 – 1310, 1999. Cited by: 468.

[25] P. Tdrrndii, "Biodegradable Self-Reinforced Composite Materials," vol. 10, pp. 29–34, 1992.

[26] L. Fambri, A. Pegoretti, R. Fenner, S. D. Incardona, and C. Migliaresi, "Biodegradable fibres of poly (L-lactic acid) produced by melt spinning," vol. 38, no. 1, pp. 79–85, 1997.

[27] D. W. Grijpma, A. J. Nijenhuis, P. G. T. V. Wijk, and A. J. Pennings, "High impact strength as-polymerized PLA," vol. 578, pp. 571–578, 1992.

[28] A. Södergård and M. Stolt, "Properties of lactic acid based polymers and their correlation with composition," *Progress in Polymer Science*, vol. 27, no. 6, pp. 1123–1163, 2002.

[29] J. Suder, Z. Bobovsky, Z. Zeman, J. Mlotek, and M. Vocetka, "The influence of annealing temperature on tensile strength of polylactic acid," *MM Science Journal*, vol. 2020, no. November, pp. 4132–4137, 2020.

[30] P. Latko-Duralek, K. Dydek, and A. Boczkowska, "Thermal, Rheological and Mechanical Properties of PETG/rPETG Blends," *Journal of Polymers and the Environment*, vol. 27, no. 11, pp. 2600–2606, 2019.

[31] K. Yu, Q. Ge, and H. J. Qi, "Reduced time as a unified parameter determining fixity and free recovery of shape memory polymers," *Nature Communications*, vol. 5, pp. 1–9, 2014.

[32] A. Lendlein and R. Langer, "Biodegradable, elastic shape-memory polymers for potential biomedical applications," *Science*, vol. 296, no. 5573, pp. 1673–1676, 2002.

[33] B. K. Kim, S. Y. Lee, and M. Xu, "Polyurethanes having shape memory effects," *Polymer*, vol. 37, no. 26, pp. 5781–5793, 1996.

[34] P. Schober, C. Boer, and L. A. Schwarte, "Correlation Coefficients: Appropriate Use and Interpretation," *Anesthesia Analgesia*, vol. 126, no. 5, 2018.

[35] I. S. Best, "Analysis of a slotted column silo system," 2022.

[36] S. Venkateshan, *Mechanical Measurements*. 2015.

[37] K. T. Kyun, "T test as a parametric statistic," *kja*, vol. 68, no. 6, pp. 540–546, 2015.

[38] N. J. Nagelkerke, "A note on a general definition of the coefficient of determination," *Biometrika*, vol. 78, no. 3, pp. 691–692, 1991.

[39] K. Kobayashi and M. U. Salam, "Comparing simulated and measured values using mean squared deviation and its components," *Agronomy Journal*, vol. 92, no. 2, pp. 345–352, 2000.

[40] G. K. Uyanık and N. Güler, "A study on multiple linear regression analysis," *Procedia - Social and Behavioral Sciences*, vol. 106, pp. 234–240, 2013. 4th International Conference on New Horizons in Education.

[41] A. R. Rajkumar and K. Shanmugam, "Additive manufacturing-enabled shape transformations via FFF 4D printing," *Journal of Materials Research*, vol. 33, no. 24, pp. 4362–4376, 2018.

[42] J. Kauffman and T. Dolma, "Analysis of shape memory properties in 3d printed pla," 2018.

[43] P. Ferretti, C. Leon-Cardenas, G. M. Santi, M. Sali, E. Ciotti, L. Frizziero, G. Donnici, and A. Liverani, "Relationship between fdm 3d printing parameters study: Parameter optimization for lower defects," *Polymers*, vol. 13, 7 2021.

[44] Y. Tao, F. Kong, Z. Li, J. Zhang, X. Zhao, Q. Yin, D. Xing, and P. Li, "A review on voids of 3d printed parts by fused filament fabrication," 11 2021.

[45] M. Mehrpouya, H. Vahabi, S. Janbaz, A. Darafsheh, T. R. Mazur, and S. Ramakrishna, "4d printing of shape memory polylactic acid (pla)," *Polymer*, vol. 230, p. 124080, 2021.

[46] F. KARTAL and A. KAPTAN, "Effects of annealing temperature and duration on mechanical properties of pla plastics produced by 3d printing," *European Mechanical Science*, vol. 7, pp. 152–159, 9 2023.

[47] https://www.matweb.com/search/DataSheet.aspx?MatGUID=ab96a4c0655c4018a8785ac4031b9278, "Overview of materials for polylactic acid (pla) biopolymer."

[48] A. A. Ansari and M. Kamil, "Effect of print speed and extrusion temperature on properties of 3d printed pla using fused deposition modeling process," *Materials Today: Proceedings*, vol. 45, pp. 5462–5468, 2021. Second International Conference on Aspects of Materials Science and Engineering (ICAMSE 2021).

[49] K. Ntouanoglou, P. Stavropoulos, and D. Mourtzis, "4d printing prospects for the aerospace industry: a critical review," *Procedia Manufacturing*, vol. 18, pp. 120–129, 2018. 18th Machining Innovations Conference for Aerospace Industry, MIC 2018, 28-29 December 2018, Garbsen Germany.

[50] M. Javaid and A. Haleem, "4d printing applications in medical field: A brief review," *Clinical Epidemiology and Global Health*, vol. 7, no. 3, pp. 317–321, 2019.

[51] A. Raina, M. I. U. Haq, M. Javaid, S. Rab, and A. Haleem, "4d printing for automotive industry applications," *Journal of The Institution of Engineers (India): Series D*, vol. 102, pp. 521–529, 2021.

Appendix A

Scientific Paper

Appendix B

Python code

B.1 Graphs and statistics sorted by layer height

```
# -*- coding: utf-8 -*-
"""
Created on Fri Feb  9 17:07:53 2024

@author: besov
"""

# -*- coding: utf-8 -*-
"""
Created on Tue Apr 26 11:25:11 2022

@author: besov
"""

import pandas as pd
import glob
import matplotlib.pyplot as plt
import numpy  as np
from scipy import stats




# Loading all files from folder into workable python lists

li = []
dfs = pd.read_csv(r'C:\Users\besov\OneDrive\School\Thesis\python\data lh.csv');
data= dfs.values
li.append(dfs)
list1 = li[0]
infill=[None] * len(list1)
temp=[None] * len(list1)
lh=[None] * len(list1)
fru=[None] * len(list1)
frl=[None] * len(list1)
rru=[None] * len(list1)
rrl=[None] * len(list1)

for num in range(0 , len(data)):
    temp[num] = data[num, 3]
    infill[num] = data[num, 2]
    lh[num] = data[num, 1]
    fru[num] = data[num, 4]
```

```
    frl[num] = data[num, 5]
    rru[num] = 180 - data[num, 6]
    rrl[num] = 180 - data[num, 7]



#taking average of measure angles from the top and bottom

def mean(numbers):
    return float(sum(numbers)) / max(len(numbers), 1)
rr = [mean(i) for i in zip(rru,rrl)]
def mean(numbers):
    return float(sum(numbers)) / max(len(numbers), 1)
fr = [mean(i) for i in zip(fru,frl)]

fr= [x/38.5 for x in fr]

#rr = list(set(rr) - 180)



rr= [x/180 for x in rr]

'''
#plot of data

fig=plt.figure(dpi=500)
plt.scatter(temp[0:9], fr[0:9], label='Fixity Rate 0.3 [mm]')
plt.scatter(temp[9:18], fr[9:18], label='Fixity Rate 0.2 [mm]')
plt.scatter(temp[18:26], fr[18:26], label='Fixity Rate 0.1 [mm]')
plt.ylabel('Fixity Rate [$_u/_m$]')
plt.xlabel('Temperature in Degrees [C]')
plt.title('Fixity Rate of Temperature regarding to Layer Height')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()


fig=plt.figure(dpi=500)
plt.scatter(temp[0:9], rr[0:9], label='Recovery Rate 0.3 [mm]')
plt.scatter(temp[9:18], rr[9:18], label='Recovery Rate 0.2 [mm]')
plt.scatter(temp[18:26], rr[18:26], label='Recovery Rate 0.1 [mm]')
plt.ylabel('Recovery Rate [$_r/_m$]')
plt.xlabel('Temperature in Degrees [C]')
plt.title('Recovery Rate of Printing Temperature regarding to Layer Height')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()

fig=plt.figure(dpi=500)
plt.scatter(infill[0:9], fr[0:9], label='Fixity Rate 0.3 [mm]')
plt.scatter(infill[9:18], fr[9:18], label='Fixity Rate 0.2 [mm]')
plt.scatter(infill[18:26], fr[18:26], label='Fixity Rate 0.1 [mm]')
plt.ylabel('Fixity Rate [$_u/_m$]')
```

```
plt.xlabel('Infill in [%]')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.title('Fixity Rate of Infill regarding to Layer Height')
plt.show()



fig=plt.figure(dpi=500)
plt.scatter(infill[0:9], rr[0:9], label='Recovery Rate 0.3 [mm]')
plt.scatter(infill[9:18], rr[9:18], label='Recovery Rate 0.2 [mm]')
plt.scatter(infill[18:26], rr[18:26], label='Recovery Rate 0.1 [mm]')
plt.ylabel('Recovery Rate [$_r/_m$]')
plt.xlabel('Infill in [%]')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.title('Recovery Rate of Infill regarding to Layer Height')
plt.show()

'''



from sklearn import linear_model



x = list (zip(infill, temp))


X = pd.DataFrame(x, columns = ['infill', 'temp'])
y = rr

regr = linear_model.LinearRegression()
regr.fit(X, y)
print(regr.coef_)

asdf = regr.predict(X)


coeff_df = pd.DataFrame(regr.coef_, X.columns, columns=['Coefficient'])
e=coeff_df.to_numpy()[1:2]
f=coeff_df.to_numpy()[0:1]
d=regr.intercept_

x = [0, 50, 100]
y = [190, 200, 210]

Xx,Yy = np.meshgrid(x,y)
Zz = (d + f*Xx + e*Yy)

zdata = rr
ydata = temp
xdata = infill
```

```
fig = plt.figure(dpi=1200)
ax = plt.axes(projection='3d', proj_type = 'ortho')

surf = ax.plot_surface(Xx, Yy, Zz,cmap='Dark2', label='Regression plane', alpha=0.5);

ax.scatter3D(xdata[0:9], ydata[0:9], zdata[0:9], color='blue', depthshade=True, label='layer height 0
ax.scatter3D(xdata[9:18], ydata[9:18], zdata[9:18], color='black', depthshade=True, label='layer heig
ax.scatter3D(xdata[18:26], ydata[18:26], zdata[18:26], color='red', depthshade=True, label='layer hei

#ax.view_init(45, 45)


#ax.invert_xaxis()
plt.legend(bbox_to_anchor=(0.15, 1), loc='upper right', borderaxespad=0)
ax.set_xlabel('Infill [%]', fontsize=6)
ax.set_ylabel('Printing temperature [°C]', fontsize=6)
ax.set_zlabel('Recovery rate', fontsize=6)
plt.title('3D plot for recovery rate with regression plane sorted by layer height [mm]')
plt.show()
```

```
from sklearn import linear_model



x = list (zip(infill, temp))


X = pd.DataFrame(x, columns = ['infill', 'temp'])
y = fr

regr = linear_model.LinearRegression()
regr.fit(X, y)
print(regr.coef_)

asdf = regr.predict(X)


coeff_df = pd.DataFrame(regr.coef_, X.columns, columns=['Coefficient'])
e=coeff_df.to_numpy()[1:2]
f=coeff_df.to_numpy()[0:1]
d=regr.intercept_

x = [0, 50, 100]
y = [190, 200, 210]
```

```
Xx,Yy = np.meshgrid(x,y)
Zz = (d + f*Xx + e*Yy)

zdata = fr
ydata = temp
xdata = infill

fig = plt.figure(dpi=1200)
ax = plt.axes(projection='3d', proj_type = 'ortho')

surf = ax.plot_surface(Xx, Yy, Zz,cmap='Dark2', label='Regression plane', alpha=0.5);

ax.scatter3D(xdata[0:9], ydata[0:9], zdata[0:9], color='blue', depthshade=True, label='layer height 0
ax.scatter3D(xdata[9:18], ydata[9:18], zdata[9:18], color='black', depthshade=True, label='layer heig
ax.scatter3D(xdata[18:26], ydata[18:26], zdata[18:26], color='red', depthshade=True, label='layer hei

#ax.view_init(45, 45)


#ax.invert_xaxis()
plt.legend(bbox_to_anchor=(0.15, 1), loc='upper right', borderaxespad=0)
ax.set_xlabel('Infill [%]', fontsize=6)
ax.set_ylabel('Printing temperature [°C]', fontsize=6)
ax.set_zlabel('Fixity rate', fontsize=6)
plt.title('3D plot for fixity rate with regression plane sorted by layer height [mm]')
plt.show()
```

```
'''
#fig=plt.figure()

#plt.scatter(lh[0:9], fr[0:9], label='Fixity Rate 0.3mm')
#plt.scatter(lh[0:9], rr[0:9], label='Recovery Rate 0.3mm')

#plt.scatter(lh[9:18], fr[9:18], label='Fixity Rate 0.2mm')
#plt.scatter(lh[9:18], rr[9:18], label='Recovery Rate 0.2mm')

#plt.scatter(lh[18:26], fr[18:26], label='Fixity Rate 0.1mm')
#plt.scatter(lh[18:26], rr[18:26], label='Recovery Rate 0.1mm')

#plt.ylabel('Angle in degrees')
#plt.xlabel('Layer Height in mm')
#plt.title('Experiment with layer height of 0.3mm and ')
#plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
#plt.show()



print(np.corrcoef(lh[0:18], fr[0:18]))
print(np.corrcoef(lh, rr))

print(np.corrcoef(temp, fr))
print(np.corrcoef(temp, rr))

print(np.corrcoef(infill, fr))
print(np.corrcoef(infill, rr))

from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score



st=lh
st=np.array(st)
ste = st.reshape(len(st), 1)
regr = linear_model.LinearRegression()
regr.fit(ste, fr)
stress = regr.predict(ste)
fig = plt.figure(dpi=500)
plt.plot(ste, stress, label='Linear Regression')
plt.scatter(lh, fr, color='green', label='Fixity Rate')
plt.ylabel('Fixity Rate [$_u/_m$]')
```

```
plt.xlabel('Layer Height in [mm]')
plt.title('Linear Regression of Layer Height and Fixity Rate')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)


# The mean squared error
print('Infill stats')
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(fr, stress)))
# The coefficient of determination: 1 is perfect prediction
print("Coefficient of determination (R2): %.2f" % r2_score(fr, stress))

corravg = np.corrcoef(fr, st)
print('Pearson\'s correlation avg :', corravg[:,1])
print('')
print('Linear regression in the form of y=a+bx')
print('a=', regr.intercept_)
print('b=', regr.coef_)
#print('mean =', np.mean(rr))
#print('standard deviation =', np.std(rr))
print('')


t_value,p_value=stats.ttest_ind(st,stress)
print('t-value for two tailed test is %f'%t_value)
print('p-value for two tailed test is %f'%p_value)
#print('mean =', np.mean(stress))
#print('standard deviation =', np.std(stress))
'''
```

B.2 Graphs and statistics sorted by printing temperature

```
# -*- coding: utf-8 -*-
"""
Created on Fri Feb  9 17:07:53 2024

@author: besov
"""

# -*- coding: utf-8 -*-
"""
Created on Tue Apr 26 11:25:11 2022

@author: besov
"""

import pandas as pd
import glob
import matplotlib.pyplot as plt
import numpy  as np
from scipy import stats
```

```
# Loading all files from folder into workable python lists

li = []
dfs = pd.read_csv(r'C:\Users\besov\OneDrive\School\Thesis\python\data temp.csv');
data= dfs.values
li.append(dfs)
list1 = li[0]
infill=[None] * len(list1)
temp=[None] * len(list1)
lh=[None] * len(list1)
fru=[None] * len(list1)
frl=[None] * len(list1)
rru=[None] * len(list1)
rrl=[None] * len(list1)

for num in range(0 , len(data)):
    temp[num] = data[num, 3]
    infill[num] = data[num, 2]
    lh[num] = data[num, 1]
    fru[num] = data[num, 4]
    frl[num] = data[num, 5]
    rru[num] = 180 - data[num, 6]
    rrl[num] = 180 - data[num, 7]



#taking average of measure angles from the top and bottom

def mean(numbers):
    return float(sum(numbers)) / max(len(numbers), 1)
rr = [mean(i) for i in zip(rru,rrl)]
def mean(numbers):
    return float(sum(numbers)) / max(len(numbers), 1)
fr = [mean(i) for i in zip(fru,frl)]

fr= [x/38.5 for x in fr]

#rr = list(set(rr) - 180)



rr= [x/180 for x in rr]





'''
```

```
fig=plt.figure(dpi=500)
plt.scatter(infill[0:8], fr[0:8], label='Fixity Rate Temp 210 [°C]')
plt.scatter(infill[8:17], fr[8:17], label='Fixity Rate Temp 200 [°C]')
plt.scatter(infill[17:26], fr[17:26], label='Fixity Rate Temp 190 [°C]')
plt.ylabel('Fixity Rate [$_u/_m$]')
plt.xlabel('Infill in [%]')
plt.title('Fixity Rate of Infill regarding to Printing Temperature')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()


fig=plt.figure(dpi=500)
plt.scatter(lh[0:8], fr[0:8], label='Fixity Rate Temp 210 [°C]')
plt.scatter(lh[8:17], fr[8:17], label='Fixity Rate Temp 200 [°C]')
plt.scatter(lh[17:26], fr[17:26], label='Fixity Rate Temp 190 [°C]')
plt.ylabel('Fixity Rate [$_u/_m$]')
plt.xlabel('Layer Height in [mm]')
plt.title('Fixity rate of Layer Height regarding to Printing Temperature')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()


fig=plt.figure(dpi=500)
plt.scatter(infill[0:8], rr[0:8], label='Recovery Rate Temp 210 [°C]')
plt.scatter(infill[8:17], rr[8:17], label='Recovery Rate Temp 200 [°C]')
plt.scatter(infill[17:26], rr[17:26], label='Recovery Rate Temp 190 [°C]')
plt.ylabel('Recovery Rate [$_r/_m$]')
plt.xlabel('Infill in [%]')
plt.title('Recovery Rate of Infill regarding to Printing Temperature')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()


fig=plt.figure(dpi=500)
plt.scatter(lh[0:8], rr[0:8], label='Recovery Rate Temp 210 [°C]')
plt.scatter(lh[8:17], rr[8:17], label='Recovery Rate Temp 200 [°C]')
plt.scatter(lh[17:26], rr[17:26], label='Recovery Rate Temp 190 [°C]')
plt.ylabel('Recovery Rate [$_r/_m$]')
plt.xlabel('Layer height in [mm]')
plt.title('Recovery rate of Layer Height regarding to Printing Temperature')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()

'''
from sklearn import linear_model



x = list (zip(infill, lh))


X = pd.DataFrame(x, columns = ['infill', 'lh'])
```

```
y = fr

regr = linear_model.LinearRegression()
regr.fit(X, y)
print(regr.coef_)

asdf = regr.predict(X)


coeff_df = pd.DataFrame(regr.coef_, X.columns, columns=['Coefficient'])
e=coeff_df.to_numpy()[1:2]
f=coeff_df.to_numpy()[0:1]
d=regr.intercept_

x = [0, 50, 100]
y = [0.1, 0.2, 0.3]

Xx,Yy = np.meshgrid(x,y)
Zz = (d + f*Xx + e*Yy)

zdata = fr
ydata = lh
xdata = infill

fig = plt.figure(dpi=1200)
ax = plt.axes(projection='3d', proj_type = 'ortho')

surf = ax.plot_surface(Xx, Yy, Zz,cmap='Dark2', label='Regression plane', alpha=0.5);

ax.scatter3D(xdata[0:8], ydata[0:8], zdata[0:8], color='blue', depthshade=True, label='printing tempe
ax.scatter3D(xdata[8:17], ydata[8:17], zdata[8:17], color='black', depthshade=True, label='printing t
ax.scatter3D(xdata[17:26], ydata[17:26], zdata[17:26], color='red', depthshade=True, label='printing

#ax.view_init(45, 45)


#ax.invert_xaxis()
plt.legend(bbox_to_anchor=(0.15, 1), loc='upper right', borderaxespad=0)
ax.set_xlabel('Infill [%]', fontsize=6)
ax.set_ylabel('Layer height [mm]', fontsize=6)
ax.set_zlabel('Fixity rate', fontsize=6)
plt.title('3D plot for fixity rate with regression plane sorted by printing temperature [°C]')
plt.show()
```

```
x = list (zip(infill, lh))


X = pd.DataFrame(x, columns = ['temp', 'lh'])
y = rr

regr = linear_model.LinearRegression()
regr.fit(X, y)
print(regr.coef_)

asdf = regr.predict(X)


coeff_df = pd.DataFrame(regr.coef_, X.columns, columns=['Coefficient'])
e=coeff_df.to_numpy()[1:2]
f=coeff_df.to_numpy()[0:1]
d=regr.intercept_

x = [0, 50, 100]
y = [0.1, 0.2, 0.3]

Xx,Yy = np.meshgrid(x,y)
Zz = (d + f*Xx + e*Yy)

zdata = rr
ydata = lh
xdata = infill

fig = plt.figure(dpi=1200)
ax = plt.axes(projection='3d', proj_type = 'ortho')

surf = ax.plot_surface(Xx, Yy, Zz,cmap='Dark2', label='Regression plane', alpha=0.5);

ax.scatter3D(xdata[0:8], ydata[0:8], zdata[0:8], color='blue', depthshade=True, label='printing tempe
ax.scatter3D(xdata[8:17], ydata[8:17], zdata[8:17], color='black', depthshade=True, label='printing t
ax.scatter3D(xdata[17:26], ydata[17:26], zdata[17:26], color='red', depthshade=True, label='printing

#ax.view_init(45, 45)


#ax.invert_xaxis()
plt.legend(bbox_to_anchor=(0.15, 1), loc='upper right', borderaxespad=0)
```

```
ax.set_xlabel('Infill [%]', fontsize=6)
ax.set_ylabel('Layer height [mm]', fontsize=6)
ax.set_zlabel('Recovery rate', fontsize=6)
plt.title('3D plot for recovery rate with regression plane sorted by printing temperature [°C]')
plt.show()
```

```
''''
#plot of data

fig=plt.figure()

plt.scatter(temp[0:9], fr[0:9], label='Fixity Rate 0.3mm')
plt.scatter(temp[0:9], rr[0:9], label='Recovery Rate 0.3mm')

plt.scatter(temp[9:18], fr[9:18], label='Fixity Rate 0.2mm')
plt.scatter(temp[9:18], rr[9:18], label='Recovery Rate 0.2mm')

plt.scatter(temp[18:26], fr[18:26], label='Fixity Rate 0.1mm')
plt.scatter(temp[18:26], rr[18:26], label='Recovery Rate 0.1mm')

plt.ylabel('Angle in degrees')
plt.xlabel('Temperature in Degrees [C]')
plt.title('Experiment with layer height of 0.3mm and ')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()



fig=plt.figure()

plt.scatter(infill[0:9], fr[0:9], label='Fixity Rate 0.3mm')
plt.scatter(infill[0:9], rr[0:9], label='Recovery Rate 0.3mm')

plt.scatter(infill[9:18], fr[9:18], label='Fixity Rate 0.2mm')
plt.scatter(infill[9:18], rr[9:18], label='Recovery Rate 0.2mm')

plt.scatter(infill[18:26], fr[18:26], label='Fixity Rate 0.1mm')
plt.scatter(infill[18:26], rr[18:26], label='Recovery Rate 0.1mm')


plt.ylabel('Angle in degrees')
plt.xlabel('Infill in [%]')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.title('Experiment with layer height of 0.3mm')
plt.show()


fig=plt.figure()

plt.scatter(lh[0:9], fr[0:9], label='Fixity Rate 0.3mm')
plt.scatter(lh[0:9], rr[0:9], label='Recovery Rate 0.3mm')

plt.scatter(lh[9:18], fr[9:18], label='Fixity Rate 0.2mm')
plt.scatter(lh[9:18], rr[9:18], label='Recovery Rate 0.2mm')

plt.scatter(lh[18:26], fr[18:26], label='Fixity Rate 0.1mm')
```

```
plt.scatter(lh[18:26], rr[18:26], label='Recovery Rate 0.1mm')

plt.ylabel('Angle in degrees')
plt.xlabel('Layer Height in mm')
plt.title('Experiment with layer height of 0.3mm and ')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()



#print(np.corrcoef(lh, fr))
#print(np.corrcoef(lh, rr))

#print(np.corrcoef(temp, fr))
#print(np.corrcoef(temp, rr))

#print(np.corrcoef(infill, fr))
#print(np.corrcoef(infill, rr))

from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score



st=temp
st=np.array(st)
ste = st.reshape(len(st), 1)
regr = linear_model.LinearRegression()
regr.fit(ste, rr)
stress = regr.predict(ste)
#fig = plt.figure(dpi=1000)
plt.plot(ste, stress, label='linear regression average stress')
plt.scatter(temp, rr, color='black', label='average stress time corrected')

# The mean squared error
print('Infill stats')
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(rr, stress)))
# The coefficient of determination: 1 is perfect prediction
print("Coefficient of determination (R2): %.2f" % r2_score(rr, stress))

corravg = np.corrcoef(rr, st)
print('Pearson\'s correlation avg :', corravg[:,1])
print('')
print('Linear regression in the form of y=a+bx')
print('a=', regr.intercept_)
print('b=', regr.coef_)
#print('mean =', np.mean(rr))
#print('standard deviation =', np.std(rr))
print('')


t_value,p_value=stats.ttest_ind(st,stress)
```

```
print('t-value for two tailed test is %f'%t_value)
print('p-value for two tailed test is %f'%p_value)
#print('mean =', np.mean(stress))
#print('standard deviation =', np.std(stress))
'''
```

B.3 Graphs and statistics sorted by infill

```
# -*- coding: utf-8 -*-
"""
Created on Fri Mar 22 14:00:05 2024

@author: besov
"""

# -*- coding: utf-8 -*-
"""
Created on Fri Feb  9 17:07:53 2024

@author: besov
"""

# -*- coding: utf-8 -*-
"""
Created on Tue Apr 26 11:25:11 2022

@author: besov
"""

import pandas as pd
import glob
import matplotlib.pyplot as plt
import numpy  as np
from scipy import stats




# Loading all files from folder into workable python lists

li = []
dfs = pd.read_csv(r'C:\Users\besov\OneDrive\School\Thesis\python\data infill.csv');
data= dfs.values
li.append(dfs)
list1 = li[0]
infill=[None] * len(list1)
temp=[None] * len(list1)
lh=[None] * len(list1)
fru=[None] * len(list1)
frl=[None] * len(list1)
rru=[None] * len(list1)
```

```
rrl=[None] * len(list1)

for num in range(0 , len(data)):
    temp[num] = data[num, 3]
    infill[num] = data[num, 2]
    lh[num] = data[num, 1]
    fru[num] = data[num, 4]
    frl[num] = data[num, 5]
    rru[num] = 180 - data[num, 6]
    rrl[num] = 180 - data[num, 7]



#taking average of measure angles from the top and bottom

def mean(numbers):
    return float(sum(numbers)) / max(len(numbers), 1)
rr = [mean(i) for i in zip(rru,rrl)]
def mean(numbers):
    return float(sum(numbers)) / max(len(numbers), 1)
fr = [mean(i) for i in zip(fru,frl)]

fr= [x/38.5 for x in fr]

#rr = list(set(rr) - 180)



rr= [x/180 for x in rr]

'''
#plot of data

fig=plt.figure(dpi=500)
plt.scatter(temp[0:7], fr[0:7], label='Fixity Rate Infill 100 [%]')
plt.scatter(temp[7:16], fr[7:16], label='Fixity Rate Infill 50 [%]')
plt.scatter(temp[16:26], fr[16:26], label='Fixity Rate Infill 0 [%]')
plt.ylabel('Fixity Rate [$_u/_m$]')
plt.xlabel('Temperature in Degrees [C]')
plt.title('Fixity Rate of Temperature regarding to Infill')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()


fig=plt.figure(dpi=500)
plt.scatter(lh[0:7], fr[0:7], label='Fixity Rate Infill 100 [%]')
plt.scatter(lh[7:16], fr[7:16], label='Fixity Rate Infill 50 [%]')
plt.scatter(lh[16:26], fr[16:26], label='Fixity Rate Infill 0 [%]')
plt.ylabel('Fixity Rate [$_u/_m$]')
plt.xlabel('Layer Height in [mm]')
plt.title('Fixity Rate of Layer Height regarding to Infill')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
```

```
plt.show()


fig=plt.figure(dpi=500)
plt.scatter(temp[0:7], rr[0:7], label='Recovery Rate Infill 100 [%]')
plt.scatter(temp[7:16], rr[7:16], label='Recovery Rate Infill 50 [%]')
plt.scatter(temp[16:26], rr[16:26], label='Recovery Rate Infill 0 [%]')
plt.ylabel('Recovery Rate [$_r/_m$]')
plt.xlabel('Temperature in Degrees [C]')
plt.title('Recovery Rate of Printing Temperature regarding to Infill')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()


fig=plt.figure(dpi=500)
plt.scatter(lh[0:7], rr[0:7], label='Recovery Rate Infill 100 [%]')
plt.scatter(lh[7:16], rr[7:16], label='Recovery Rate Infill 50 [%]')
plt.scatter(lh[16:26], rr[16:26], label='Recovery Rate Infill 0 [%]')
plt.ylabel('Recovery Rate [$_r/_m$]')
plt.xlabel('Layer Height in [mm]')
plt.title('Recovery Rate of Layer Height regarding to Infill')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()

'''



'''


from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score


fig=plt.figure(dpi=500)
st=infill
st=np.array(st)
ste = st.reshape(len(st), 1)
regr = linear_model.LinearRegression()
regr.fit(ste, rr)
stress = regr.predict(ste)
#fig = plt.figure(dpi=1000)
plt.plot(ste, stress, label='Linear Regression')
plt.scatter(infill, rr, color='black', label='Recovery Rate')
plt.ylabel('Recovery Rate [$_r/_m$]')
plt.xlabel('Infill in [%]')
plt.title('Linear Regression of Infill and Recovery Rate')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)

# The mean squared error
print('Infill stats')
```

```
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(rr, stress)))
# The coefficient of determination: 1 is perfect prediction
print("Coefficient of determination (R2): %.2f" % r2_score(rr, stress))

corravg = np.corrcoef(rr, st)
print('Pearson\'s correlation avg :', corravg[:,1])
print('')
print('Linear regression in the form of y=a+bx')
print('a=', regr.intercept_)
print('b=', regr.coef_)
#print('mean =', np.mean(rr))
#print('standard deviation =', np.std(rr))
print('')


t_value,p_value=stats.ttest_ind(st,stress)
print('t-value for two tailed test is %f'%t_value)
print('p-value for two tailed test is %f'%p_value)
#print('mean =', np.mean(stress))
#print('standard deviation =', np.std(stress))

'''
'''
x = list (zip(temp, lh))


X = pd.DataFrame(x, columns = ['temp', 'lh'])
y = fr


from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=7, random_state=0)


from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
a=regressor.fit(X_train, y_train)


stress = regressor.predict(X)

regressor = LinearRegression()
a=regressor.fit(X_train, y_train)



coeff_df = pd.DataFrame(regressor.coef_, X.columns, columns=['Coefficient'])
b=coeff_df.to_numpy()[1:2]
a=coeff_df.to_numpy()[0:1]
d=regressor.intercept_
#x = np.linspace(0,0.4,10)
#y = np.linspace(0.05,0.240,10)
```

```
x = [190, 200, 210]
y = [0.1, 0.2, 0.3]

Xx,Yy = np.meshgrid(x,y)
Zz = (d + a*Xx + b*Yy)

zdata = fr
ydata = lh
xdata = temp

fig = plt.figure(dpi=1200)
ax = plt.axes(projection='3d', proj_type = 'ortho')

surf = ax.plot_surface(Xx, Yy, Zz,cmap='Dark2', label='Regression plane', alpha=0.5);
#surf._facecolors2d = surf._facecolor3d
#surf._edgecolors2d = surf._edgecolor3d
#fng._facecolors3d= 'black'
#fng._edgecolors3d= 'black'
#surf._facecolors3d= 'black'
#fng._edgecolors2d= 'black'
ax.scatter3D(xdata[0:7], ydata[0:7], zdata[0:7], color='blue', depthshade=True, label='0 [%]');
ax.scatter3D(xdata[7:16], ydata[7:16], zdata[7:16], color='green', depthshade=True, label='50 [%]');
ax.scatter3D(xdata[16:26], ydata[16:26], zdata[16:26], color='red', depthshade=True, label='100 [%]')

plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
ax.set_xlabel('Temperature [°C]')
ax.set_ylabel('Layer height [mm]')
ax.set_zlabel('Fixity rate')
plt.title('3D plot for fixity rate with regression plane')
plt.show()

'''



from sklearn import linear_model

x = list (zip(temp, lh))


X = pd.DataFrame(x, columns = ['temp', 'lh'])
y = fr

regr = linear_model.LinearRegression()
regr.fit(X, y)
print(regr.coef_)

asdf = regr.predict(X)


coeff_df = pd.DataFrame(regr.coef_, X.columns, columns=['Coefficient'])
e=coeff_df.to_numpy()[1:2]
```

```
f=coeff_df.to_numpy()[0:1]
d=regr.intercept_

x = [190, 200, 210]
y = [0.1, 0.2, 0.3]

Xx,Yy = np.meshgrid(x,y)
Zz = (d + f*Xx + e*Yy)

zdata = fr
ydata = lh
xdata = temp

fig = plt.figure(dpi=1200)
ax = plt.axes(projection='3d', proj_type = 'ortho')

surf = ax.plot_surface(Xx, Yy, Zz,cmap='Dark2', label='Regression plane', alpha=0.5);

ax.scatter3D(xdata[0:7], ydata[0:7], zdata[0:7], color='blue', depthshade=True, label='infill 0 [%]')
ax.scatter3D(xdata[7:16], ydata[7:16], zdata[7:16], color='green', depthshade=True, label='infill 50
ax.scatter3D(xdata[16:26], ydata[16:26], zdata[16:26], color='red', depthshade=True, label='infill 10

plt.legend(bbox_to_anchor=(0.15, 1), loc='upper right', borderaxespad=0)
ax.set_xlabel('Printing temperature [°C]', fontsize=6)
ax.set_ylabel('Layer height [mm]', fontsize=6)
ax.set_zlabel('Fixity rate', fontsize=6)
plt.title('3D plot for fixity rate with regression plane sorted by infill [%]')
plt.show()
```

```
x = list (zip(temp, lh))


X = pd.DataFrame(x, columns = ['temp', 'lh'])
y = rr

regr = linear_model.LinearRegression()
regr.fit(X, y)
print(regr.coef_)

asdf = regr.predict(X)


coeff_df = pd.DataFrame(regr.coef_, X.columns, columns=['Coefficient'])
e=coeff_df.to_numpy()[1:2]
f=coeff_df.to_numpy()[0:1]
d=regr.intercept_
```

```
x = [190, 200, 210]
y = [0.1, 0.2, 0.3]

Xx,Yy = np.meshgrid(x,y)
Zz = (d + f*Xx + e*Yy)

zdata = rr
ydata = lh
xdata = temp

fig = plt.figure(dpi=1200)
ax = plt.axes(projection='3d', proj_type = 'ortho')

surf = ax.plot_surface(Xx, Yy, Zz,cmap='Dark2', label='Regression plane', alpha=0.5);

ax.scatter3D(xdata[0:7], ydata[0:7], zdata[0:7], color='blue', depthshade=True, label='infill 0 [%]')
ax.scatter3D(xdata[7:16], ydata[7:16], zdata[7:16], color='green', depthshade=True, label='infill 50
ax.scatter3D(xdata[16:26], ydata[16:26], zdata[16:26], color='red', depthshade=True, label='infill 10

plt.legend(bbox_to_anchor=(0.15, 1), loc='upper right', borderaxespad=0)
ax.set_xlabel('Printing temperature [°C]', fontsize=6)
ax.set_ylabel('Layer height [mm]', fontsize=6)
ax.set_zlabel('Recovery rate', fontsize=6)
plt.title('3D plot for recovery rate with regression plane sorted by infill [%]')
plt.show()

#print(np.corrcoef(rr, temp))

'''



fig = plt.figure(dpi=500)
ax = plt.axes(projection='3d', proj_type = 'ortho')




ax.scatter3D(xdata[0:7], ydata[0:7], zdata[0:7], color='blue', depthshade=True, label='0 [%]');
ax.scatter3D(xdata[7:16], ydata[7:16], zdata[7:16], color='green', depthshade=True, label='50 [%]');
ax.scatter3D(xdata[16:26], ydata[16:26], zdata[16:26], color='red', depthshade=True, label='100 [%]')
#ax.view_init(45, 45)
surf = ax.plot_trisurf(xdata[0:7], ydata[0:7], zdata[0:7],cmap='Blues', label='Regression plane', alp
surf = ax.plot_trisurf(xdata[7:16], ydata[7:16], zdata[7:16],cmap='Greens', label='Regression plane',
surf = ax.plot_trisurf(xdata[16:26], ydata[16:26], zdata[16:26],cmap='Reds', label='Regression plane'

#ax.invert_xaxis()
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
ax.set_xlabel('Temperature [°C]')
ax.set_ylabel('Layer height [mm]')
ax.set_zlabel('Recovery rate')
```

```
plt.title('3D plot for recovery rate')
plt.show()
```

```
fig = plt.figure(dpi=500)
ax = plt.axes(projection='3d', proj_type = 'ortho')
```

```
zdata = fr
ydata = lh
xdata = temp

ax.scatter3D(xdata[0:7], ydata[0:7], zdata[0:7], color='blue', depthshade=True, label='0 [%]');
ax.scatter3D(xdata[7:16], ydata[7:16], zdata[7:16], color='green', depthshade=True, label='50 [%]');
ax.scatter3D(xdata[16:26], ydata[16:26], zdata[16:26], color='red', depthshade=True, label='100 [%]')

#ax.view_init(45, 45)
surf = ax.plot_trisurf(xdata[0:7], ydata[0:7], zdata[0:7],cmap='Blues', label='Regression plane', alp
surf = ax.plot_trisurf(xdata[7:16], ydata[7:16], zdata[7:16],cmap='Greens', label='Regression plane',
surf = ax.plot_trisurf(xdata[16:26], ydata[16:26], zdata[16:26],cmap='Reds', label='Regression plane'

#ax.invert_xaxis()
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
ax.set_xlabel('Temperature [°C]')
ax.set_ylabel('Layer height [mm]')
ax.set_zlabel('Fixity rate')
plt.title('3D plot for fixity rate')
plt.show()
```

```
'''
```

```
'''

fig=plt.figure()
plt.scatter(temp[0:7], rr[0:7], label='RR Infill 100%')
plt.scatter(temp[7:16], rr[7:16], label='RR Infill 50%')
plt.scatter(temp[16:26], rr[16:26], label='RR Infill 0%')
plt.ylabel('Recovery Rate')
plt.xlabel('Temperature in Degrees [C]')
plt.title('Experiment with layer height of 0.3mm and ')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()


fig=plt.figure()

plt.scatter(infill[0:9], fr[0:9], label='Fixity Rate 0.3mm')
plt.scatter(infill[0:9], rr[0:9], label='Recovery Rate 0.3mm')

plt.scatter(infill[9:18], fr[9:18], label='Fixity Rate 0.2mm')
plt.scatter(infill[9:18], rr[9:18], label='Recovery Rate 0.2mm')

plt.scatter(infill[18:26], fr[18:26], label='Fixity Rate 0.1mm')
plt.scatter(infill[18:26], rr[18:26], label='Recovery Rate 0.1mm')


plt.ylabel('Angle in degrees')
plt.xlabel('Infill in [%]')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.title('Experiment with layer height of 0.3mm')
plt.show()
```

```
fig=plt.figure()

plt.scatter(lh[0:9], fr[0:9], label='Fixity Rate 0.3mm')
plt.scatter(lh[0:9], rr[0:9], label='Recovery Rate 0.3mm')

plt.scatter(lh[9:18], fr[9:18], label='Fixity Rate 0.2mm')
plt.scatter(lh[9:18], rr[9:18], label='Recovery Rate 0.2mm')

plt.scatter(lh[18:26], fr[18:26], label='Fixity Rate 0.1mm')
plt.scatter(lh[18:26], rr[18:26], label='Recovery Rate 0.1mm')

plt.ylabel('Angle in degrees')
plt.xlabel('Layer Height in mm')
plt.title('Experiment with layer height of 0.3mm and ')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0)
plt.show()



#print(np.corrcoef(lh, fr))
#print(np.corrcoef(lh, rr))

#print(np.corrcoef(temp, fr))
#print(np.corrcoef(temp, rr))

#print(np.corrcoef(infill, fr))
#print(np.corrcoef(infill, rr))

from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score



st=infill
st=np.array(st)
ste = st.reshape(len(st), 1)
regr = linear_model.LinearRegression()
regr.fit(ste, rr)
stress = regr.predict(ste)
#fig = plt.figure(dpi=1000)
plt.plot(ste, stress, label='linear regression average stress')
plt.scatter(infill, rr, color='black', label='average stress time corrected')

# The mean squared error
print('Infill stats')
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(rr, stress)))
# The coefficient of determination: 1 is perfect prediction
print("Coefficient of determination (R2): %.2f" % r2_score(rr, stress))

corravg = np.corrcoef(rr, st)
```

```
print('Pearson\'s correlation avg :', corravg[:,1])
print('')
print('Linear regression in the form of y=a+bx')
print('a=', regr.intercept_)
print('b=', regr.coef_)
#print('mean =', np.mean(rr))
#print('standard deviation =', np.std(rr))
print('')


t_value,p_value=stats.ttest_ind(st,stress)
print('t-value for two tailed test is %f'%t_value)
print('p-value for two tailed test is %f'%p_value)
#print('mean =', np.mean(stress))
#print('standard deviation =', np.std(stress))
'''
```