

Development of a Software Solution for the Use of Slicers for Polymer Materials for Additive Manufacturing with Concretes

David Scheidt

Thesis for the attainment of the academic degree

Master of Science (M.Sc.)

at the Department of Civil and Environmental Engineering of the Technical University of Munich.

Examiner:

Prof. Dr. Oliver Fischer

Supervisor:

M.Sc. Daniel Auer

Submitted:

Munich, 12.06.2025

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated.
I have only used the resources given in the list of references.

Munich, 12.06.2025

David Scheidt

Danksagung

Ich möchte mich an dieser Stelle ganz herzlich bei Herrn Prof. Dr. Oliver Fischer und meinem Betreuer Daniel Auer für die Chance bedanken mich mit einem meiner Hobbies auch auf wissenschaftlicher Ebene auseinandersetzen zu dürfen. Trotz der Vorerfahrung in Bezug auf 3D-Druck waren Teile dieser Arbeit nicht immer einfach und wären ohne die vielseitige Hilfsbereitschaft, das Vertrauen in meine Arbeit, aber auch die Geduld an mancher Stelle nicht möglich gewesen.

Generell möchte ich mich hier bei meiner gesamten Freundesgruppe bedanken die mich auf diesem Weg begleitet hat und die ich neben unseren gemeinsamen Mittagessen in der Mensa in Zukunft sicher sehr vermissen werde.

Ganz besonderer Dank gilt allerdings einigen Freunden, die ich im Laufe meines Studiums kennengelernt durfte und ohne deren Hilfe ich an mancher Stelle sicher einen anderen Weg eingeschlagen hätte. Dir Leo für die gemeinsamen Abende im Zeichensaal vor den Mathe-Tutorien und deine Unterstützung während meines gesamten Studiums auf die ich mich immer verlassen konnte. Markus für deinen ansteckenden Enthusiasmus, der mich neben deinen Witzen in der Vorlesung durch den kompletten Master getragen hat. Euch Hanna, Sarah und Sofie für die tollen Urlaube und Freizeitaktivitäten die mich immer wieder auf andere Gedanken und neue Ideen gebracht haben.

Abschließend möchte ich mich natürlich auch bei meinen Eltern bedanken, deren Rückhalt auf vielen Ebenen mir dieses Studium erst ermöglicht hat und die mich bei jeder meiner Entscheidungen mit Rat und Zuspruch ermutigt haben weiterzugehen.

Abstract

This thesis investigates the potential of improving the digital planning and fabrication process in robot-assisted concrete-3D-printing at the Technical University of Munich through the automated conversion of slicer outputs into robot-readable KUKA Robot Language (KRL) code. Following an introduction to the fundamentals of additive construction with concrete, the current state of the art is presented, including both the hardware in use and existing software solutions for path generation, control, and visualization.

To address the research question, the developed Python-based program is then introduced. It automatically imports and interprets G-code data from a slicer and translates it into KRL commands for the available robot. At the same time, the system generates a structured 3D-file in Rhinoceros3D that includes layer assignments, line codes, and positional data. This enables transparent control and preparation of the printing process. Parameters such as printing speed, extrusion commands, tool positions, and control signals (e.g., pump control) are processed and consistently transferred into the target language.

The developed program is subsequently tested using a representative application case and evaluated with regard to its functionality, flexibility, and expandability. The results demonstrate that the automated pipeline significantly simplifies the workflow, reduces manual effort, and identifies potential sources of error at an early stage. The thesis concludes with a critical reflection on the system's limitations and an outlook on possible extensions, including the integration of multiple slicers and the digital mapping of the entire process.

Contents

Abstract	iv
1 Introduction	1
2 State of the Art	4
2.1 Historical Development of Additive Manufacturing Techniques	4
2.2 Normative Classification of Printing Processes	5
2.3 Overview of Additive Manufacturing Methods for Concrete	6
2.3.1 Selective Binder Application	6
2.3.2 Material Depositing Methods	10
2.3.3 Slip Formwork	12
2.4 Classification of the Printing Process Used	13
2.5 Robotic Systems in Additive Manufacturing for Construction	14
2.5.1 Stationary Systems	14
2.5.2 Mobile Systems	17
2.6 Specification of the Hardware Used	18
2.6.1 Robot	18
2.6.2 Pump	21
2.6.3 Mixer	22
2.6.4 Print Bed	23
2.6.5 Overview	24
3 From Model to Printed Object	25
3.1 Overview	25
3.2 Workflow for 3D-Printed Objects	26
3.2.1 Design Process	26
3.2.2 Slicing Process	26
3.2.3 Comparison of Different Slicer Outputs	33
3.2.4 G-Code Analysis	34
3.2.5 Print	38
3.2.6 Summary	39
4 Software Development: Implementation of the Process Chain	40
4.1 Introduction	40
4.2 Program Overview	41
4.3 Input Parameters [folder: <i>user_input</i>]	43
4.3.1 Cura Setup [file: <i>printer.def.json & extruder.def.json</i>]	43
4.3.2 General Setup [file: <i>setup.json</i>]	44
4.3.3 Validation of User Input [module: <i>validate_value</i>]	47
4.4 Parsing and Interpretation of G-Code [package: <i>gcode</i>]	48
4.5 Robot Integration and Kinematic Modeling [package: <i>robot</i>]	50
4.5.1 Coordinate Frames	50
4.5.2 Robot Implementation Using <i>setup.json</i> [folder: <i>user_input</i>]	53
4.5.3 Mathematical Background [module: <i>mathematical_operators</i>]	57
4.5.4 Kinematic Modeling [module: <i>kinematics</i>]	61

4.6	Analysis of Line Geometry and Extrusion Values [package: <i>pump</i>]	68
4.6.1	Calculation of Line Width [module: <i>calculate_linewidth</i>]	68
4.6.2	Calculation for Pump Commands [module: <i>calculate_flow, calculate_rpm</i>]	69
4.7	Translation to KUKA Robot Language [package: <i>krl</i>]	71
4.7.1	List of Collected Information	71
4.7.2	Initialization & Reset of Robot	72
4.7.3	Main Section	73
4.8	Print Report [package: <i>report</i>]	75
4.9	Output in Rhino [package: <i>rhino</i>]	77
4.9.1	Setup of Rhinoceros3D File [sub-package: <i>pre_process</i>]	77
4.9.2	Write Information to Rhino File [sub-package: <i>process</i>]	78
4.10	Implementation of Safety Mechanisms	82
4.10.1	Program Abort Errors [<i>main-function</i>]	82
4.10.2	Check Object Size [package: <i>g_code</i> ; module: <i>fits_printbed</i>]	82
4.10.3	Deactivate Outputs [<i>main-function</i>]	84
5	Validation and Limitations	89
5.1	Validation [package: <i>tests</i>]	89
5.1.1	Validation of Rhino Output	89
5.1.2	Validation of Robot Kinematics	90
5.2	Limitations	91
6	Outlook	93
6.1	Simple Extensions of the Program	93
6.2	Advanced Extensions of the Program	93
7	Conclusion	95
A	G-Code Listings	96
A.1	Prusa Slicer 2.9.2 – G-Code Listing	96
A.2	OrcaSlicer 2.2.0 – G-Code Listing	117
A.3	Ultimaker Cura 5.8.0 – G-Code Listing	141
B	Fixed Arguments for Cura	156
C	List of Usable and Validated Commands from <i>printer.def.json</i> With Comments	158
D	List of Usable and Validated Commands from <i>extruder.def.json</i> With Comments	194
Bibliography		197

List of Tables

3.1	Color Legend and HEX-Code for the Different Line Types Used in Figure 3.7	29
3.2	Selection of Different G-Code Commands Used with Marlin [47]	32
3.3	Custom Slicing Parameters Used for Test-Object Generation	33
3.4	Line Type Labels Used by Different Slicers	35
3.5	Example of G-Code Inserted Before and After Layer Change in PrusaSlicer	36
4.1	Visualization and Notation Styles Used for Input and Output Formats	41
4.2	Overview of Data Types Used in <i>setup.json</i>	47
4.3	Explanation of Regular Expressions Used for Parsing G-Code	48
4.4	Definition of Variables Used in Equation 4.44	68
4.5	Typical G-Code Patterns for FDM-Style Slicing and Printing Using <i>Ultimaker Cura 5.8.0</i> .	70
4.6	Overview of Parameters Used in Figure 4.25	71
4.7	Extended Symbol Overview for KRL Line Definition	73
5.1	Summary of Validation Tests for the Kinematics Module	90

List of Figures

1.1 Comparison of Productivity Rate for Different Sectors in the US Since 1947	1
1.2 Automated Concrete Wall Construction Using Large-Scale 3D Printing On-Site [18]	2
2.1 The Three Basic Methods of 3D Printing Adapted from [30]	4
2.2 Timeline of Significant 3D-Printing Related Projects in Construction Summarized from [54]	5
2.3 Process Principles of Additive Manufacturing According to DIN EN ISO 52900 [60]	6
2.4 Overview on State of the Art Concrete-3D-Printing Methods Adapted from [100] and Adjusted	6
2.5 Process for Selective Binder Application Taken from [51]	7
2.6 Schematic of Selective Binder Activation Adapted from [45]	7
2.7 Schematic of Selective Paste Intrusion Adapted from [45]	8
2.8 Schematic of Binder Jetting Adapted from [45]	8
2.9 Schematic Figure of D-Shape Printer Taken from [46]	9
2.10 Process for Material Deposition Printing Taken from [51]	10
2.11 Schematic of Shortcrete 3D-Printing Adapted from [40]	11
2.12 Schematic for Slip Formwork 3D-Printing	12
2.13 Classification of the 3D-Printing Process Discussed in the Existing Processes Adapted from [100]	13
2.14 Classification of Robotic Systems Used for Concrete-3D-Printing	14
2.15 Example of a Gantry System Robot with Defined Axis [108]	15
2.16 Example of a Delta System Robot [8]	15
2.17 Example of Serial System Robots Working in Parallel [43]	16
2.18 Schematic of "INSTATIQ" During Operation [38]	17
2.19 Technical Data for KUKA KR 340 R3300	18
2.20 Picture of Tool Frame in Use	19
2.21 Basic Movement Types for KUKA KR 340 R3300 [43]	19
2.22 Side View and Front View of Pump with Measurements [mm] [66]	21
2.23 Mixer with Measurements Given in [mm] [65]	22
2.24 Print Bed Consisting of PERI MAXIMO Elements (Curvature Due to Lens Distortion)	23
2.25 Rendering of Robot Cell with Print Related Hardware	24
3.1 Technical Drawing of Test-Object with Measurements in [mm] Made Using Fusion 360 [11]	25
3.2 Design Process of Test-Object Using Fusion 360 [11]	26
3.3 Object turned from Solid Body to Boundary Representation as .stl	27
3.4 Cutting-Plane Intersecting Different Facets of Polygon Mesh [5]	27
3.5 Intersection Between Cutting-Plane and Facet Creating Start- and Endpoint of Line Segment [5]	28
3.6 Selection of Different Infill Patterns Used in Polymer 3D-Printing [14]	29
3.7 Overview on Different Linetypes Used After Slicing of the Test-Object	29
3.8 Different Bed Adhesion Features Taken from [6]	30
3.9 Slicer Output for Prusa Slicer Version 2.9.2	33
3.10 Slicer Output for Orca Slicer Version 2.2.0	34
3.11 Slicer Output for Cura Version 5.10.0	34
3.12 Difference in Line Types Regarding <i>Bridge</i> Declaration	36
3.13 Overview on Printer and Printed Part	38

3.14 Summary of 3D-Printing Process	39
4.1 Progress Bar Referring to Packages in Program	41
4.2 Overview of Program with All Implemented Modules and Information on Different Sub-Operations	42
4.3 Progress Bar for Chapter 4.3	43
4.4 Overview of the Different Sections Included in <i>setup.json</i> With Descriptions of Each Input Parameter	46
4.5 Progress Bar for Chapter 4.4	48
4.6 Progress Bar for Chapter 4.5	50
4.7 Comparison of Spherical and Non-Spherical Wrists for Serial Robots [27]	50
4.8 Coordinate Systems for Robot Joints of Kuka KR 340 R3300, Adapted from [16].	51
4.9 Overview of Relevant Coordinate Systems for Robot, Object and Bed Modeled in Fusion 360 [11]	52
4.10 Overview of "base_coordinates" Measurements for <i>setup.json</i>	53
4.11 Essential Measurements for Robotic Arm, Adapted from [16].	54
4.12 Overview of "tool_offset" Measurements for <i>setup.json</i>	54
4.13 Orientation of Coordinate Systems for Robotic Arm, according to [16].	55
4.14 Components of Spacial transformation	57
4.15 Individual Rotation Matrices	57
4.16 Gimbal and Gimbal-Locked System, Taken from [110]	59
4.17 Overview of Class <i>robotOPW</i> from Module <i>kinematics</i> inside Package <i>robot</i>	61
4.18 Necessary Measurements Needed to Determine Position of C_{wrist} wrt. \$ROBOTROOT .	62
4.19 Flowchart for <i>forward_kinematics</i> -Function Inside <i>robotOPW</i> -Class	64
4.20 Coordinate Frames and Available Transformation Matrices	65
4.21 Flowchart for <i>inverse_kinematics</i> -Function Inside <i>robotOPW</i> -Class	67
4.22 Progress Bar for Chapter 4.6	68
4.23 Geometry for Filament and Line Segment	68
4.24 Progress Bar for Chapter 4.7	71
4.25 Dictionary for Arbitrary G-Code Line Used Inside the Program to Build .src-file	71
4.26 Example of KRL-Command for Linear Movement (RPM Used for Pump Control)	73
4.27 Progress Bar for Chapter 4.8	75
4.28 Progress Bar for Chapter 4.9	77
4.29 Schematic of Layer and Sublayer Structure Inside .3dm-file	77
4.30 Overview of Rhino-File After Initialization	78
4.31 Visualization of test object in Rhinoceros3D	80
4.32 Attributes Used for Labeling in Rhinoceros3D	80
4.33 Screenshot of Rhinoceros3D Interface with Test Object Sliced and Visualized (beginning & end [yellow], start [green], stop [red])	81
4.34 Test Object (Scaled) on Print Bed in Top View (Taken from Report)	82
4.35 Flowchart of <i>main</i> -Function (Part 1)	85
4.36 Flowchart of <i>main</i> -Function (Part 2)	86
4.37 Flowchart of <i>main</i> -Function (Part 3)	87
4.38 Flowchart of <i>main</i> -Function (Part 4)	88
5.1 Comparison of Measurements for Sliced Test-Object (Scaled 8000 %)	89

Listings

4.2.1 Example of a Python-Function	41
4.2.2 Example of a <i>.json</i> -File	41
4.3.1 Section from <i>extruder.def.json</i>	43
4.3.2 Example Input for "cura_arguments" from <i>setup.json</i>	44
4.3.3 Console Output from Setup Validation with Error Message	45
4.3.4 Console Output from Setup Validation with Error Message	47
4.4.1 RegEx Pattern Used in <i>simplify_gcode</i>	48
4.4.2 Dictionary Generated for Each New Information Given on Coordinates, Line Type or Extrusion	49
4.5.1 Example Input to Geometrically Define KUKA KR 340 R3300 Taken from <i>setup.json</i>	56
4.6.1 Example Input "characteristic_curve" of Pump in <i>setup.json</i>	69
4.7.1 Console Output from <i>setup replace_strings</i> with Error Message	72
4.7.2 Example for Movement Command and Line Type Annotation Used in <i>.src</i> -Output	74
4.7.3 Example for Layer Timer and Minimum Layer Time Implemented for <i>.src</i> -Output	74
4.10. Example of CLI Error Code for Badly Located Print (See Figure 4.34b)	83
4.10. Example of CLI Error Code for Objects Larger than Printbed (See Figure 4.34c)	83
A.1.1G-Code Generated Using Prusa Slicer [68] (Initialization)	96
A.1.2G-Code Generated Using Prusa Slicer [68] (Main)	96
A.1.3G-Code generated using Prusa Slicer [68] (Reset)	112
A.2.1G-Code Generated Using Orca Slicer [63] (Initialization)	117
A.2.2G-Code Generated Using Orca Slicer [63] (Main)	117
A.2.3G-Code Generated Using Orca Slicer [63] (Reset)	134
A.3.1G-Code Generated Using Ultimaker Cura [97] (Initialization)	141
A.3.2G-Code Generated Using Ultimaker Cura [97] (Main)	141
A.3.3G-Code Generated Using Ultimaker Cura [97] (Reset)	155
B.1 Fixed arguments specified in <i>preset_arguments.py</i>	156

Abbreviations

A	Activator
A_i	Rotational Axes (i) of Robot
ACDC	Admixture Controlled Digital Casting
AG	Aggregat
AM	Additive Manufacturing
B	Binder
CAD	Computer-Aided Design
CDPR	Cable-Driven Parallel Robot
CLI	Command Line Interface
Cura	Ultimaker Cura (Version 5.8.0)
DOF	Degrees of Freedom
DSC	Digital Casting System
FDM	Fused Deposition Modeling
G-code	Geometric Code
GUI	Graphical User Interface
KRL	KUKA Robot Language
Orca	OrcaSlicer (Version 2.2.0)
OPW	Ortho-Parallel-Wrist
P_{AUX}	Auxiliary Point
Prusa	Prusa Slicer (Version 2.9.2)
Rhino	Rhinoceros 3D (CAD software)
rpm	Revolutions per Minute
SC3DP	Shotcrete 3D-Printing
SDC	Smart Dynamic Casting
SLA	Stereolithography
SLS	Selective Laser Sintering
TCP	Tool Center Point
TUM	Technical University of Munich

1 Introduction

The German Economic Institute identified an annual demand of nearly 373 000 housing units in Germany by the end of 2024. However, since 2021, only around 295 000 units have been completed per year. In Cologne, for instance, newly constructed housing covers only about 40 % of the demand. Especially in major cities such as Berlin, Munich, or Stuttgart, the supply of newly created living space falls significantly short of expectations needed to meet existing demand. [107]

One structural factor contributing to this imbalance is the persistently low productivity growth within the construction industry. According to the German Construction Industry Federation [32], productivity in the German construction sector has exceeded its 1991 reunification benchmark only once (in 1992) and, as of mid-2024, remains approximately 23 % below that level.

While long-term productivity data for Germany as a whole remains limited, international analyses indicate that this is not a nationally isolated phenomenon. According to a study by the consulting firm McKinsey, the U.S. construction sector has recorded an average annual productivity increase of just 0.1 % since 1947, placing it at the bottom of all industrial sectors in terms of long-term productivity growth (see Figure 1.1) [50].

In the United States, labor productivity in construction has declined since 1968, in contrast to rising productivity in other sectors

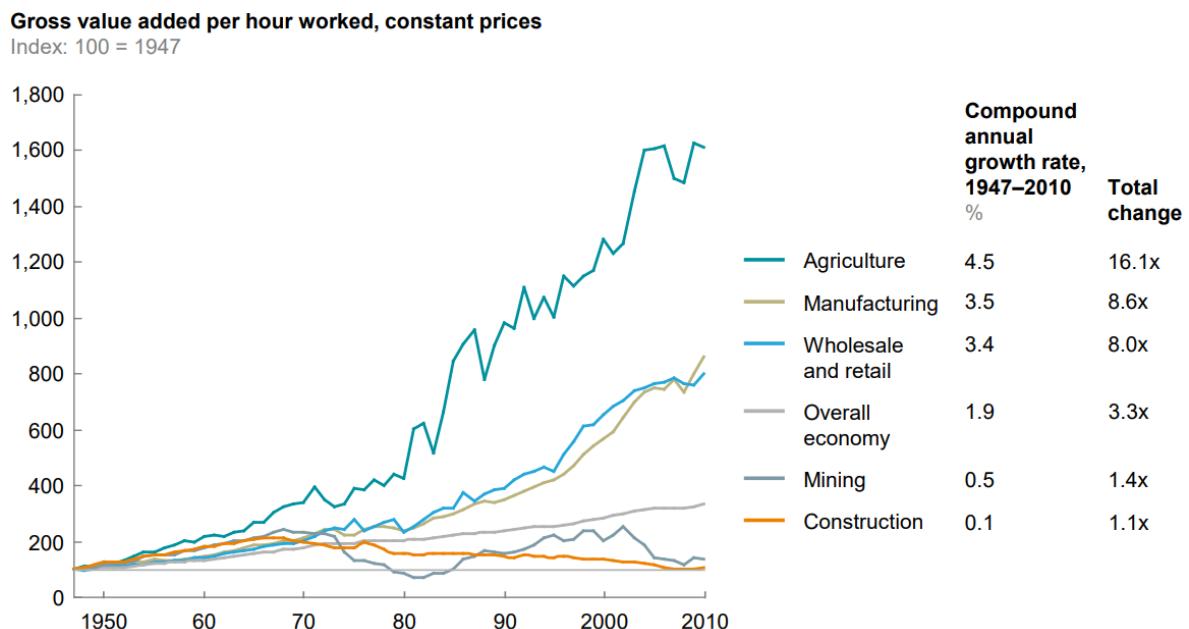


Figure 1.1 Comparison of Productivity Rate for Different Sectors in the US Since 1947

This stagnation not only affects the housing market, as previously discussed, but also contributes to the growing backlog of renovation and maintenance in Germany's infrastructure projects, such as the refurbishment of bridges [17, 37] or the expansion and upkeep of the railway network [22]. In both cases, the average condition of the total stock of infrastructure is now rated only as "satisfactory". Continued operation in these sectors will thus require substantial additional investments and labor for maintenance and renewal in the coming years [13], which must be provided by the construction industry. How, then, can the supply gap in construction services be reduced, or even closed in the future?

When comparing the share of automated labor in the construction sector to other industries, it quickly becomes apparent that manual work still dominates in construction. A 2021 survey conducted by the robotics manufacturer ABB revealed that only about 55 % of the roughly 1900 construction companies surveyed in China, the USA, and Europe integrate robots into their process chains. In contrast, the automotive sector utilizes robotic systems in the value chain in 84 % of the surveyed companies, revealing a gap of approximately 30 %. [3]

Closing this gap could not only help the construction industry overcome the prevailing shortage of skilled labor [3], but also increase productivity rates. However, how can automation technology be effectively integrated into the complex environment of the construction industry, where most projects are unique in both planning and execution?

A promising approach lies in the combination of *additive manufacturing* (AM), also known as *3D-printing*, and concrete construction. In this context, users or companies can incorporate their individual ideas and solutions into an automated manufacturing process. The robot responsible for producing the project can be reprogrammed depending on the requirements, but it is designed to perform the same task, such as the production of waffle slabs or entire building stories in an automated manner (see Figure 1.2).



Figure 1.2 Automated Concrete Wall Construction Using Large-Scale 3D Printing On-Site [18]

By combining various hardware systems and integrating them within a homogeneous software solution, housing units [89] or bridge components [12] could be constructed quickly and efficiently, as demonstrated by the examples mentioned. While 3D-printing has long been part of the repertoire in industries such as aerospace and mechanical engineering, the technology has not yet fully arrived in the construction sector. According to [86], the current market capitalization for 3D-printing within the construction industry is estimated at around 1.9 billion US\$ compared to a total additive manufacturing market capitalization of approximately 20 billion US\$. However, it should be noted that figures for the construction industry vary significantly depending on the source [74, 101]. Nevertheless, all forecasts clearly indicate that the market is expected to grow strongly, with projected annual growth rates of up to 54 % [86]. As this projection shows, the industry places well-founded hopes in this technological branch.

Therefor also research institutes and universities like the Technical University of Munich (TUM) take part in this development. Through its involvement in the Collaborative Research *Additive Manufacturing in Construction* (AMC TRR 277) of the German Research Foundation [19], TUM is part of a network of institutes aiming to drive innovation in this subfield.

As part of this thesis, an integrative software platform is developed that systematically consolidates the majority of required process steps from the digital 3D model to automated concrete printing. The objective is to represent the workflow in a transparent, comprehensible, and largely automated manner for the user. The tasks range from importing and positioning the geometry, to generating printing paths, and finally to outputting machine-readable control data for a 6-axis industrial robot.

The thesis starts with a brief introduction to the historical, technological, and regulatory foundations of 3D-concrete-printing. Building on this, the current state of the art is presented, followed by a delimitation to relevant system components available at TUM.

Subsequently, the complete process from geometric modeling to slicing and machine execution is analyzed in detail. Existing slicer programs are compared and their functionalities examined more closely. Particular attention is given to adapting an existing open-source slicer originally designed for polymer-based 3D-printing to suit the requirements of concrete printing. The slicer will then be implemented into a whole new program such that it can automatically generate printing paths, interpret process parameters, and produce complete robot programs based on parameterized user input.

In addition, the generated motion data are visually structured in a Rhino file, enabling intuitive digital pre-checks and validation of all boundary conditions. This not only establishes a complete digital process chain, but also contributes to improved accessibility, reproducibility, and automation in robot-assisted construction at TUM.

Finally, a critical reflection on the capabilities and limitations of the developed software is conducted. Potential extensions, such as runtime optimization, user-friendliness, or technological enhancements are discussed, and approaches for future developments are outlined.

2 State of the Art

2.1 Historical Development of Additive Manufacturing Techniques

The origins of additive manufacturing can be traced back to the early 1940s. In 1941, the American entrepreneur and inventor William E. Urschel from Valparaiso, Indiana, filed a patent for his so-called “Urschel Wall Building Machine.” This device was capable of depositing concrete using mechanical guides as “[...] solidifiable material into the form of a strip [...]” and then using it “as a course or layer in the formation of a wall” [106]. One of the three prototypes built is still on display in Valparaiso, Indiana, USA [105].

After these early developments in concrete, the focus of technological advancements shifted for more than three decades toward polymer-based methods. A significant milestone was reached in 1971, when Wyn Kelly Swainson filed a patent for the curing of thermosetting materials using “radiation beams” [109]. In 1980, Hideo Kodama followed with another patent that laid the groundwork for what is known today as *Stereolithography* (SLA) [33]. Figure 2.1a schematically illustrates how a UV light source is used in SLA technology to harden UV-sensitive resin layer by layer at the intended locations. This process was significantly refined in the following years by Charles Hull, who filed a patent in 1984 for an “apparatus for production of three-dimensional objects by stereolithography,” which was granted in 1986. Just one year later, Hull launched the first commercial 3D printer, the “SLA-1,” through his newly founded company “3D Systems” [2] [10].

In the same year, Carl R. Deckard filed a patent for a powder bed fusion process, which became the foundation for what is now known as *Selective Laser Sintering* (SLS). In this method, layers of plastic or metal powder are applied and selectively fused by a laser (see Figure 2.1b). With the invention of *Fused Deposition Modeling* (FDM) by the couple Scott and Lisa Crump in 1989, the third of today’s three main 3D printing processes was patented [85]. Unlike previous technologies, the object is not formed within a material reservoir; instead, the geometry is extruded layer by layer from a nozzle (see Figure 2.1c). The company Stratasys [87], that emerged from this idea remains, like 3D Systems [2], is still a major player in the industry to this day [94, 54].

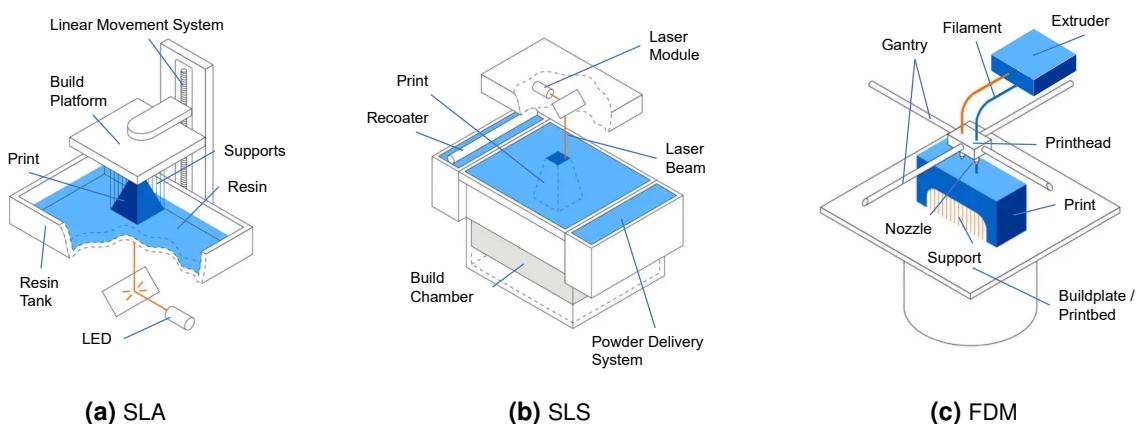


Figure 2.1 The Three Basic Methods of 3D Printing Adapted from [30]

With the expiration of the patent held by Scott and Lisa Crump in 2009, the previously tightly regulated technology of FDM became accessible to a broader public. For the first time, numerous companies, research institutions, and private developers were able to design their own FDM-based printing systems and bring them to market without licensing fees or patent-related restrictions. This development played a significant role in making additive manufacturing accessible to a wide user base and in substantially expanding its use beyond industrial applications.

The construction industry, whose productivity growth has been considered below average compared to other sectors since the 1950s as mentioned in the Introduction, also began to recognize the potential of additive manufacturing. 3D-printing not only promises to automate labor-intensive construction processes but also offers new design freedom and the possibility of significantly reducing material usage and construction time.

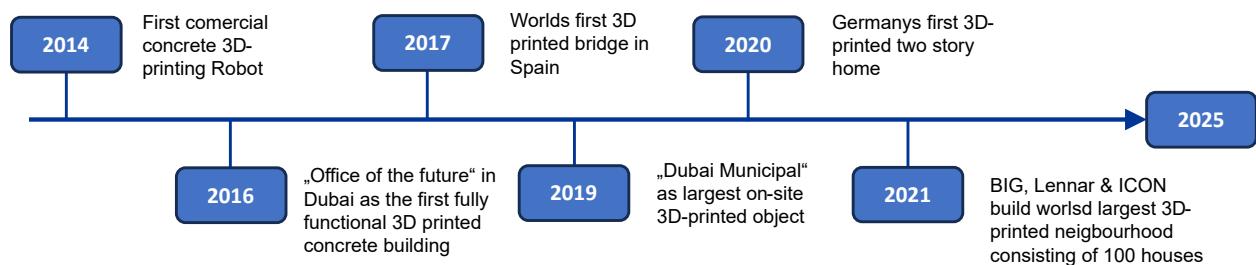


Figure 2.2 Timeline of Significant 3D-Printing Related Projects in Construction Summarized from [54]

From the mid-2010s onward, a continuous increase in 3D-printing-related projects in the construction sector can be observed. Figure 2.2 presents a selection of relevant developments from 2014 onward in chronological order. These include both individual projects and systematic research initiatives. Research groups such as the Collaborative Research Centre *Additive Manufacturing in Construction* (AMC TRR 277), which acts as a joint initiative between the Technical University Braunschweig and the Technical University of Munich funded by the German Research Foundation [19], as well as research teams at ETH Zurich [28] and the 3DCP-consortium at TU Eindhoven [26] have been driving innovation in this field continuously ever since. The goal is to further develop additive manufacturing methods for use in the construction sector and to make their application economically and sustainably attractive compared to conventional fabrication techniques.

2.2 Normative Classification of Printing Processes

With the increasing use of additive manufacturing processes in the construction industry, the need for a normative framework for the associated terminology and process descriptions also grew. Since 2015, the DIN EN ISO 52900 standard [60] has served as the basis for defining printing-related terminology. In addition, the standard classifies the currently available additive manufacturing processes.

Fundamentally, the standard distinguishes between “single-stage” and “multi-stage” manufacturing processes (see Figure 2.3). Single-stage processes are characterized by the fact that components are produced in a “[...] single operation” both in their “[...] fundamental geometric form” and with their “[...] fundamental material properties.” Actions such as removing formwork or cleaning the surface are considered part of the single-stage process.

In contrast, multi-stage manufacturing processes involve components being produced in “[...] two or more operations.” In this case, the geometric form is first created, and the material properties are subsequently stabilized, for example, through oven sintering or bonding [60].

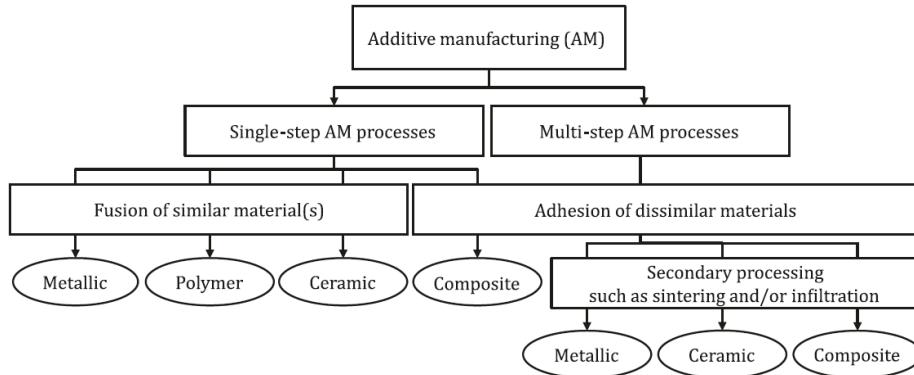


Figure 2.3 Process Principles of Additive Manufacturing According to DIN EN ISO 52900 [60]

2.3 Overview of Additive Manufacturing Methods for Concrete

Since this thesis focuses exclusively on the 3D-printing of concrete, this chapter provides an overview of the current state of the art in this specific field. Other materials addressed by DIN EN ISO52900 are not part of this thesis. Figure 2.4 presents an overview of the methods that will be explained in more detail below.

All methods listed therein can be classified as single-stage printing processes according to DIN EN ISO52900 (see Section 2.2).

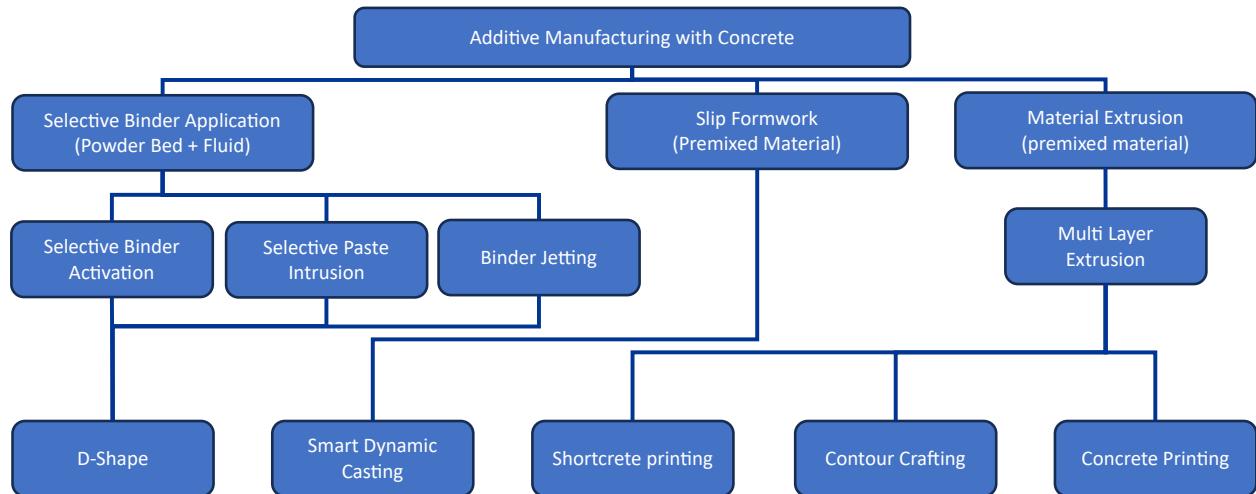


Figure 2.4 Overview on State of the Art Concrete-3D-Printing Methods Adapted from [100] and Adjusted

2.3.1 Selective Binder Application

The fundamental principle behind the method of *Selective Binder Application* is the layer-by-layer bonding of a particle bed through the targeted combination of different materials. In the context of 3D-concrete-printing, the following three components are typically used:

- a) an aggregate [AG] (e.g., gravel or sand),
- b) a binder [B] (e.g., cement), and
- c) an activator [A] (e.g., water).

The manufacturing process is based on the repeated execution of two process steps. Similar to the SLS method presented in Section 2.1, the component is built up layer by layer within a particle bed:

1. application of a uniform layer of unbound particles using a distribution device,
2. selective deposition of a liquid binder via a nozzle onto the regions intended for solidification.

Once the printing process is complete, the printed object can be exposed by removing the unbound particles from the bed [45]. Figure 2.5 shows a schematic representation of the individual process steps.

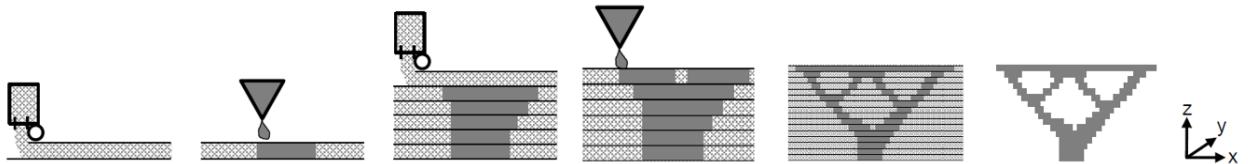


Figure 2.5 Process for Selective Binder Application Taken from [51]

The manufacturing process using *Selective Binder Application* can be further divided into three sub-variants:

Selective Binder Activation

In the method of *Selective Binder Activation*, the printing process is based on a pre-mixed particle bed composed of a fine-grained aggregate (typically with particle diameters < 1 mm, e.g. sand) and a hydraulic binder (e.g. cement). These two components (aggregate [AG] and binder [B]) are homogeneously mixed before the actual printing process and applied as a loose layer at the beginning of each layer.

Targeted activation is then carried out by a local or area-wise application of a liquid, which typically consists of water or a reactive activator solution. This liquid is deposited onto the regions of the particle bed intended for solidification using a nozzle or spray system. The local addition of the activator causes a chemical reaction between the binder and the activator in the treated areas. In the case of cement and water, hydration leads to the formation of a solid structure that binds the particles together in the affected zone. By repeating these steps, the desired geometry is created within the build volume. Unwetted areas remain unbound and can be removed after the printing process, provided they are not completely enclosed by hardened material. This allows the finished component to be exposed afterwards. The geometric resolution of the process depends primarily on the particle size of the bed and the precision of the activator application [45].

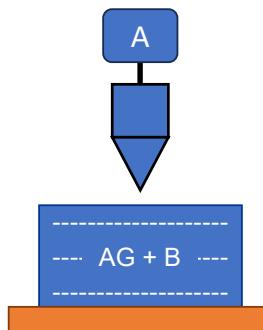


Figure 2.6 Schematic of Selective Binder Activation Adapted from [45]

Selective Paste Intrusion

In contrast to *Selective Binder Activation*, the method of *Selective Paste Intrusion* does not involve mixing the binder into the particle bed in advance. Instead, the binder is combined with the activator, typically as cement paste, and applied directly and selectively through a nozzle. The particle bed usually consists of coarser material such as gravel or coarse sand, with particle sizes of up to 5 mm [45].

The cement paste is specifically applied to the regions designated for solidification and penetrates the pore spaces of the particle bed. Its purpose is to fill the voids between the individual grains completely, thereby creating continuous bonding within the structure. In comparison to *Selective Binder Activation*, this method allows for greater layer thicknesses. This increases the printing speed and makes the method particularly attractive for large-scale applications in the construction sector.

One of the most critical aspects of this process is ensuring that the paste fully infiltrates the particle bed. Only with complete penetration can a homogeneous internal structure and consistent material properties throughout the component be achieved [104].

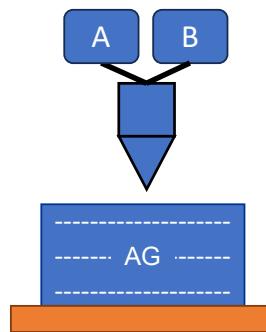


Figure 2.7 Schematic of Selective Paste Intrusion Adapted from [45]

Binder Jetting

Binder Jetting is considered the highest-resolution process currently available within the category of selective binding in powder beds. Unlike the previously described methods, it typically employs a resin–hardener-based binder system [45]. Due to the materials used, this process is not directly associated with concrete-3D-printing. However, it is included here for the sake of completeness.

As with *Selective Binder Activation*, the powder bed in binder jetting consists of a pre-mixed combination of materials. In contrast to that method, the powder bed does not contain the binder itself but rather the activator. The actual binder is applied selectively through a nozzle. The chemical reaction takes place locally at the defined contact points within the powder bed, resulting in the selective bonding of particles, similar to the other previously discussed methods.

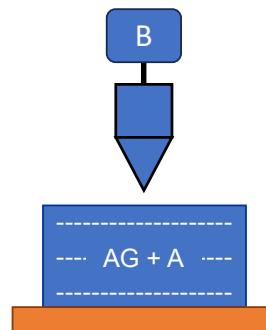


Figure 2.8 Schematic of Binder Jetting Adapted from [45]

Due to the high level of detail achievable, this technology is primarily used in the field of formwork construction, especially for foundry applications. Depending on the geometry of the component, it can offer economic advantages over conventional methods, such as *computerized numerical control milling* (CNC), in terms of both production time and manufacturing costs [104]. For large-scale applications, as commonly found in the construction industry, the technology has so far been used only selectively. This is mainly due to the limited build volume, which currently reaches a maximum of approximately $4 \times 2 \times 1$ meters [81] [79].

D-Shape

An example of integrating the previously described methods into a large-scale application is the so called *D-Shape* process, developed by Enrico Dini in 2008. This system combines the principles of selective binder application with a large build volume of approximately 5×5 meters. This setup enables the processing of large-volume objects with layer thicknesses of up to 5 mm [1] and a maximum particle size of up to 20 mm [104].

The printing system is based on a gantry-like frame mounted on four vertically aligned supports. Within this frame, the devices for granulate deposition and selective binder application move parallel to the build surface (see Figure 2.9). This configuration enables the layer-wise fabrication of large-format structures with comparatively high geometric resolution.

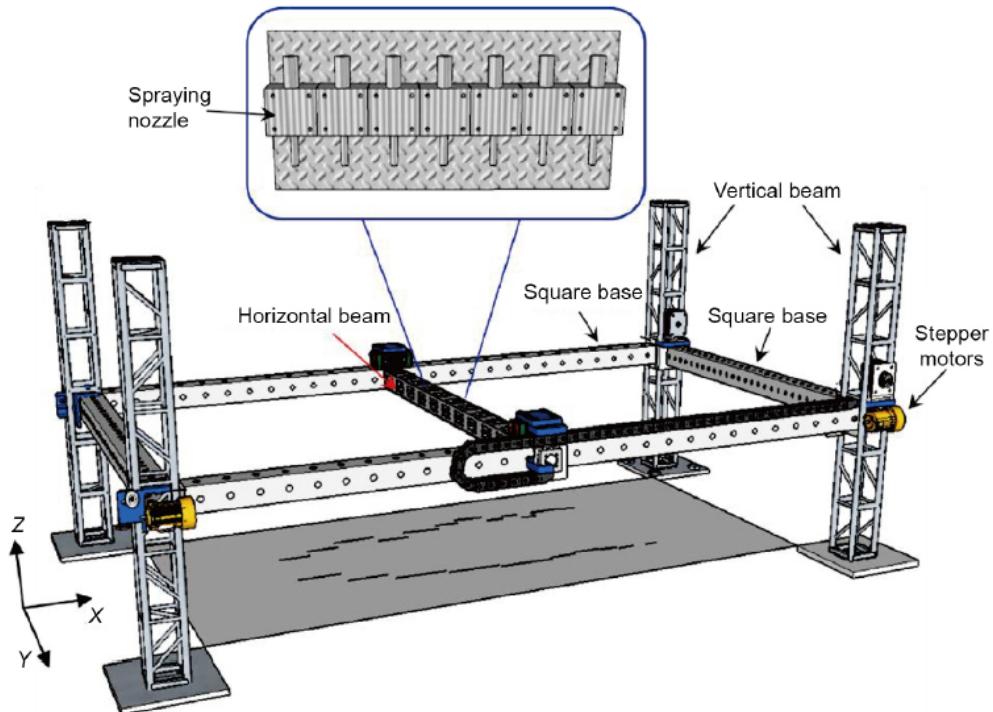


Figure 2.9 Schematic Figure of D-Shape Printer Taken from [46]

2.3.2 Material Depositing Methods

In general, both deposition-based and selective printing processes can be divided into a sequence of individual process steps that are repeated multiple times until the component is fully completed. These recurring steps are schematically illustrated for the deposition-based methods discussed below in Figure 2.10 and can be categorized as follows [51]:

1. Transportation of the build material from the storage system to the printhead,
2. Processing at the printhead and / or extrusion of the material through the nozzle,
3. Deposition of the material, accompanied by its deformation during placement,
4. Deposition of subsequent layers, including the self-weight-induced loading of previously deposited upper layers.

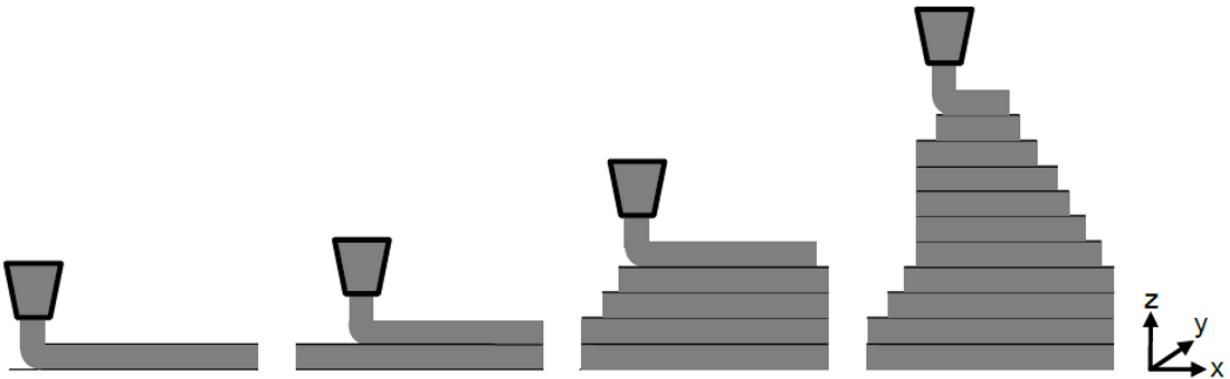


Figure 2.10 Process for Material Deposition Printing Taken from [51]

The so-called deposition-based methods are characterized by a higher material throughput compared to *Selective Binder Application*, which results in significantly shorter printing times for many applications. These properties make them particularly attractive for the fabrication of large-format components with low geometric detail. As a result, these methods have increasingly become a focus of both application and research of 3D-printing in construction [104].

This subfield of concrete-3D-printing can also be further subdivided. The following section presents the three most significant methods currently in use.

Multi Layer Extrusion

One form of deposition-based methods is known as *Multi Layer Extrusion*. This method follows the process principle described in Section 2.3.2 and is fundamentally based on the Fused Deposition Modeling technique developed by Scott and Lisa Crump (see Chapter 2.1).

Multi Layer Extrusion transfers this concept to the processing of concrete and similar materials. The building material is extruded layer by layer through a nozzle and deposited in a stable form on top of the previous layers. Numerous technical variants of this method now exist, differing in terms of material feed, printhead or nozzle configuration, and movement mechanics. The following section provides a closer look at three of the most relevant implementations currently in use.

A) Contour Crafting

In the mid-2010s, Behrokh Khoshnevis from the University of Southern California, in collaboration with NASA, developed a system intended to enable the automated construction of wall structures on the lunar surface. In this method, fresh concrete is conveyed to the nozzle by means of a pump and then deposited layer by layer as a continuous bead. Simultaneously, the outer surface of the printed object is smoothed using an integrated trowel in order to produce a uniform exterior.

After constructing the outer structure, the internal cavity is filled either by injection or by pouring another material [39]. The method is known as *Contour Crafting* and is considered one of the early large-scale applications of multi-layer extrusion.

As a further development of *Contour Crafting*, the so-called *CONPrint3D* method was developed at the Dresden University of Technology. The goal of this technology is to achieve a comparatively high printing speed of up to 10 m/min by increasing the material throughput, allowing for cross-sections of up to $150 \times 50 \text{ mm}^2$ and the use of aggregates with particle sizes of up to 16 mm [100]. A specially designed nozzle enables the direct fabrication of load-bearing wall structures, including integrated reinforcement, within the framework of a continuous on-site printing process.

To realize the required build volume, the printhead is mounted at the end of a boom-like extension on a mobile platform. This configuration makes it possible to construct entire building sections directly on-site, section by section [53].

B) Concrete Printing

The *Concrete Printing* system can be positioned in terms of its application between *Contour Crafting* and the *CONPrint3D* process. With layer heights between 5 and 25 mm and a lower printing speed compared to *CONPrint3D* [46], the method aims to achieve higher resolution and improved geometric precision.

The maximum size of the printed object is limited by the available build volume of the stationary printing system. As a result, the method is particularly suited for serial off-site prefabrication and is primarily used in controlled production environments [100].

C) Shotcrete 3D-Printing

The *Shotcrete 3D-Printing* (SC3DP) method, developed at the Technical University Braunschweig and illustrated schematically in Figure 2.11, represents another robot-assisted technique of additive manufacturing in the field of material extrusion. The method is based on conventional shotcrete technology and combines automated component fabrication with pneumatic material application [100].

The fully mixed fresh concrete is conveyed to the nozzle using a progressive cavity pump. At the nozzle, the material is accelerated by compressed air, which enables the concrete to be applied at high velocity to the target position [40]. This principle allows for rapid construction progress while maintaining high flexibility with regard to both the geometry of the component and the direction of material application. In addition, the process permits the integration of reinforcement elements directly during printing. To improve surface quality, a targeted smoothing step can be performed after the printing process. This step reduces irregularities and prepares the surface for subsequent construction work [104].

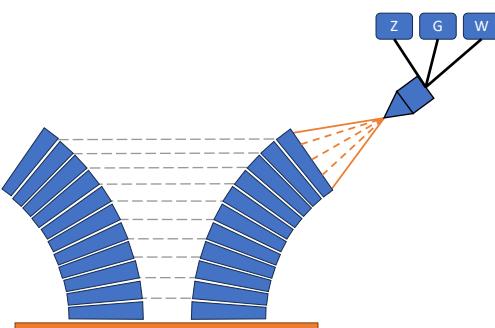


Figure 2.11 Schematic of Shotcrete 3D-Printing Adapted from [40]

2.3.3 Slip Formwork

The field of concrete printing using *Slip Formwork*-techniques, similar to *Multi Layer Extrusion*, can be divided into several distinct technical developments. One of the earliest and most well-known implementations of this principle is the *Smart Dynamic Casting* (SDC) process, developed at ETH Zurich in 2012 [100]. The basic principle of the Slip Formwork process can be described as follows: One or more formwork elements, which are small in dimension compared to the printed structure, are moved, rotated, or locally deformed either continuously or stepwise along the desired component geometry. At the same time, fresh concrete is introduced into the formwork.

Once the previously deposited concrete has sufficiently hardened to support the self-weight of the next section, the formwork is repositioned and the process is repeated [52]. This method enables the fabrication of continuous, geometrically variable concrete structures with high precision, without the need for conventional large-scale formwork systems.

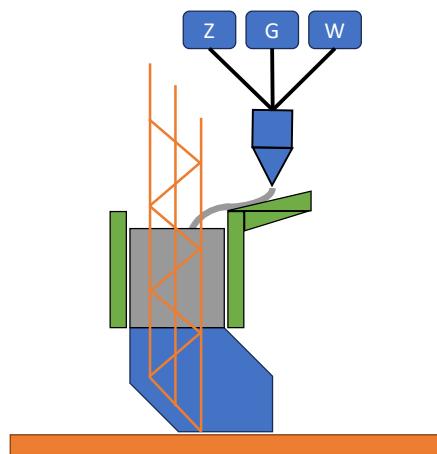


Figure 2.12 Schematic for Slip Formwork 3D-Printing

To reduce the pressure exerted by viscous fresh concrete on the formwork during the construction process, the *Smart Dynamic Casting* system developed at ETH Zurich was further advanced into what is now called the *Digital Casting System* (DCS). By precisely controlling the concrete hydration rate, it becomes possible to adjust the fresh concrete strength to match the construction progress. This enables a significant reduction in the thickness of the formwork elements without compromising the structural stability of the component [44].

Based on this principle, ETH Zurich also developed the process known as *Eggshell with thin printed formwork*. The method uses a thin-walled polymer formwork produced through additive manufacturing. The shell can either be fabricated in advance or printed simultaneously with the concrete casting process, and is then filled with fresh concrete. The concrete mix used, which is referred to as *Set-on-Demand Concrete* hardens rapidly, allowing the system to be self-stabilizing. Because the special concrete places less stress on the formwork, the walls can be made extremely thin (in the referenced example, only 1.5 mm in thickness). The integration of reinforcement elements is also easily achievable with this method.

Another development is the so-called *Admixture Controlled Digital Casting* (ACDC). In this approach, a prefabricated flexible formwork made of PVC or PTFE-glass fiber membrane is used, which is held in shape by a minimal external support structure. The concrete is then filled, depending on the cross section at a rate of up to 2.5 m/h [44].

In contrast to conventional slipforming methods, where the formwork is moved step by step along the printed object, both the *Eggshell* and *Admixture Controlled Digital Casting* methods rely on fully prefabricated formwork. The actual casting process takes place afterward, or in stages, within this fixed form. These approaches open up new design possibilities and allow for high geometric precision while reducing the amount of formwork required.

2.4 Classification of the Printing Process Used

Since numerous ongoing research projects exist in the field of concrete-3D-printing, the methods presented in Chapter 2.3 can only represent a limited portion of the current technological landscape. A more comprehensive and in-depth systematization of the various approaches can be found in the sources cited above [100, 46, 52, 104].

The results presented in this thesis focus exclusively on the method of material extrusion based on the *Concrete Printing* principle. The printing material is assumed to be a pre-mixed mineral-based compound, and the printing process is not designed for multi-material systems as referred to in Chapter 2.3.3 (see Figure 2.13).

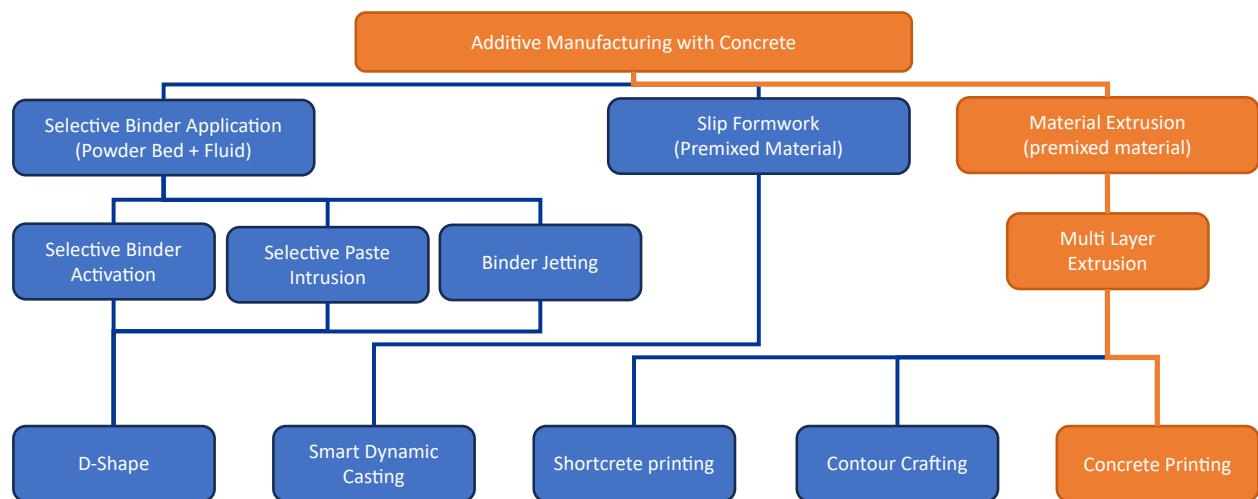


Figure 2.13 Classification of the 3D-Printing Process Discussed in the Existing Processes Adapted from [100]

2.5 Robotic Systems in Additive Manufacturing for Construction

Following the previous chapter's overview of various 3D-concrete-printing methods, this chapter focuses on the robotic hardware used to execute the printing processes. In this thesis, currently employed robotic systems for 3D-printing are classified into two main categories (see Figure 2.14):

Stationary systems remain fixed in position relative to the printed object throughout the entire printing process. As a result, their workspace is limited to the area reachable from their base position. These systems are typically used when the component to be printed can be fully positioned within this workspace.

Mobile systems, by contrast, are capable of moving relative to the printed object either manually or autonomously during the printing process. This movement involves not only the tool unit (end effector) of the robot, but also the relative motion of the entire robotic unit and its components with respect to the object. Consequently, significantly larger components can be fabricated, as the effective workspace of the system is extended by its mobility.

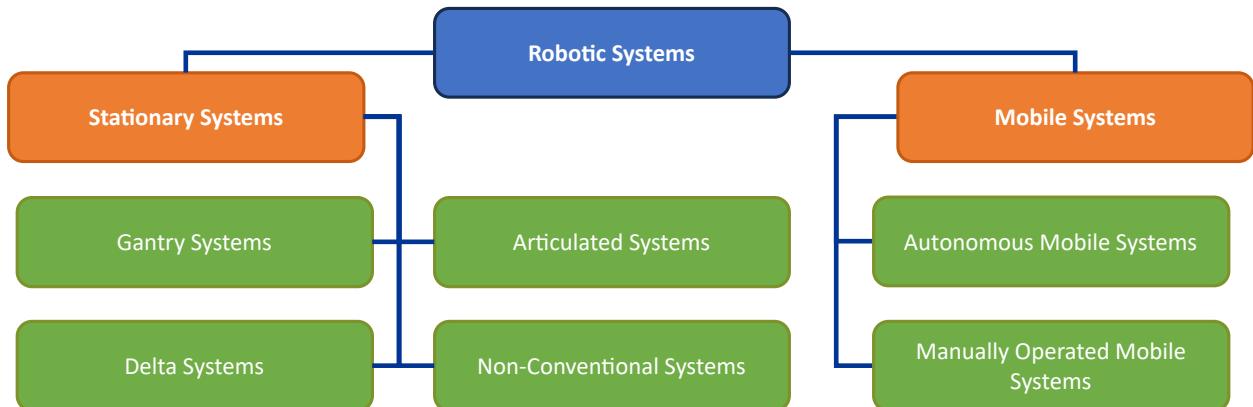


Figure 2.14 Classification of Robotic Systems Used for Concrete-3D-Printing

2.5.1 Stationary Systems

Gantry Systems

So-called *gantry systems* or *portal systems* are based on three linearly movable axes, which are mounted orthogonally to one another. Each axis is responsible for movement along one of the three Cartesian spatial directions (X , Y , Z). As a result, the system is capable of positioning the tool head precisely and independently along each of these directions.

Due to their simple mechanical structure, gantry systems can be adapted to various build volumes. At the same time, they offer high positioning accuracy, making them particularly suitable for large-scale and repeatable applications in additive manufacturing [72]. Because of their simplicity and straightforward kinematics, such systems are already widely in use. Depending on the project requirements, the complexity of the kinematic configuration can be increased, for example by adding a rotational joint around the Z -axis [18, 36].

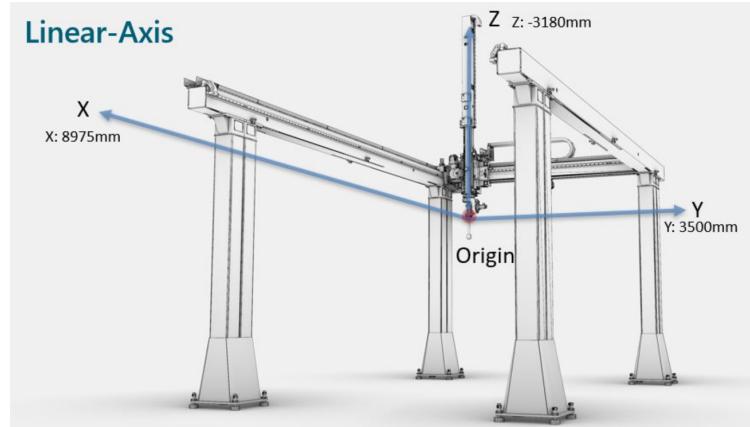


Figure 2.15 Example of a Gantry System Robot with Defined Axis [108]

Delta Systems

Delta systems can be structurally distinguished from conventional gantry systems with relative ease. They consist of two platforms connected by several movable arms. One of these platforms forms the stationary base, while the other acts as the end effector and moves through space relative to the base (see Figure 2.16). Each arm is controlled individually to enable precise relative motion between the platforms.

Although the mechanical structure is significantly more complex than that of Cartesian systems, the delta principle offers the advantage of higher dynamic performance. Since a single motion is distributed across multiple components, each individual component requires less travel. As a result, considerably higher speeds can be achieved. Due to these advantages over gantry systems, manufacturers such as WASP increasingly rely on delta configurations as a basis for concrete-3D-printing [103]. A key disadvantage, however, is the comparatively limited workspace and the greater overall footprint of the system compared to portal systems with an equivalent build volume [8].

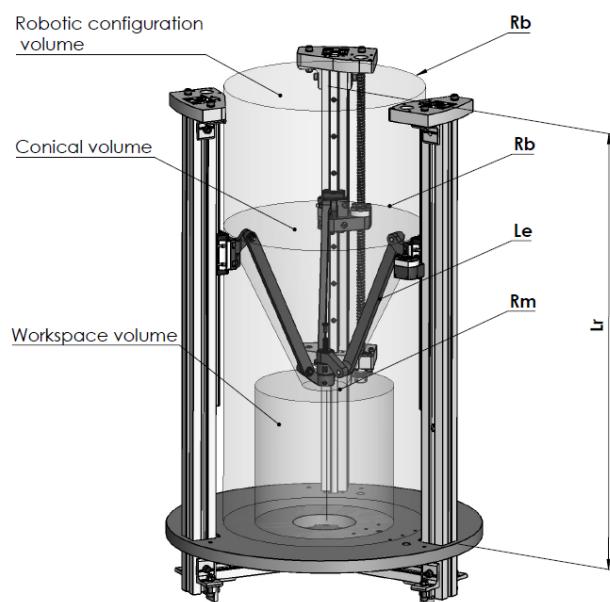


Figure 2.16 Example of a Delta System Robot [8]

Serial Systems

The most versatile form of robotic systems is represented by so-called *serial robots*, also referred to as *articulated robots*. They consist of a serial arrangement of segments connected by joints with varying *Degrees of Freedom* (DOF) [56]. Typically, the foundation of the robotic system, which is also referred to as the *base* is fixed to the reference frame (e.g., the ground) and forms the rotational axis A1, around which the robot can rotate. The other end of the robot arm, known as the *flange*, carries the attached tool, the tip of which forms the robot's *end effector* [99]. The end effector can be oriented and positioned in space depending on the joint configuration and joint limits.

Compared to cartesian systems, the mechanical setup is considerably more complex. However, it also enables significantly greater freedom of movement, particularly with respect to the orientation of the tool relative to the robot base. This capability makes serial robots especially suitable for tasks that require complex, non-linear motion sequences and varying tool orientations. Typical applications include automated painting, welding, or assembly processes involving entire components, such as the joining of car body parts in the automotive industry [64].



Figure 2.17 Example of Serial System Robots Working in Parallel [43]

Non-Conventional Systems

In addition to the commonly used stationary robotic systems described above, a wide range of other, partly highly specialized configurations exists, many of which are already well established in research and development. These include, among others, *continuum robots* [61], *hexapods* or *Stewart platforms* [4], and *Cable-Driven Parallel Robots (CDPRs)* [92].

Such systems are often characterized by unique kinematic properties, a high degree of flexibility, or suitability for specific application areas. However, due to their relatively limited use in large-scale additive manufacturing for construction and the associated practical constraints, they are not discussed in detail in this thesis.

2.5.2 Mobile Systems

In contrast to stationary robotic systems, mobile robots offer significantly greater flexibility and a substantially expanded workspace. Because the entire robot base can be repositioned relative to the reference object, large-scale or segmented printing tasks can be executed step by step using a single system.

However, increased mobility also comes with certain disadvantages. Additional degrees of freedom typically result in reduced positioning accuracy and require greater effort in setting up and calibrating the system. In particular, the precise localization and fixation of the robot during the printing process pose additional technical challenges. Furthermore, all connected systems, such as the pump and mixer, must also be designed for mobility in order to accommodate changes in the position of the printing unit. The payload capacity of the robot is also lower compared to a stationary unit due to limited anchoring possibilities.

The following section presents selected mobile robotic systems that are currently used in the field of concrete-3D-printing.

Manually Operated Mobile Systems

These robots represent a hybrid configuration consisting of a numerically controlled printhead mounted on a semi-automated or manually movable platform.

The system must be positioned at the beginning of each operation. Once one section is completed, it is moved by an operator to a new target position, where it can resume work on the next section. The system is not capable of completing an entire project autonomously once the printing process has begun. Systems such as the "P1" (see Figure 2.18) developed by Putzmeister and further advanced by INSTATIQ [69, 38], or the robotic arm on a movable platform developed by APIS COR [77], fall into this category.

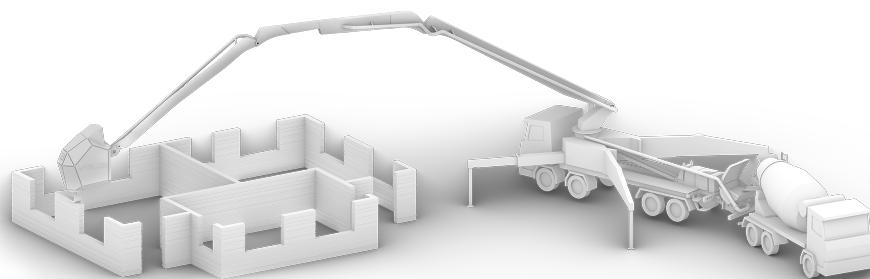


Figure 2.18 Schematic of "INSTATIQ" During Operation [38]

Autonomous Mobile Systems

Fully autonomous robotic systems are capable of independently adjusting their absolute position relative to the component during the printing process without external intervention or input. They navigate the construction site autonomously and are, in principle, capable of completing the entire printing process without continuous human supervision. This would make it possible to fabricate even large-scale components entirely using a single robotic unit.

However, such a level of automation has not yet been fully achieved in the construction industry with the current state of technology. At present, mostly semi-autonomous systems are available in practice. These require manual remeasurement, position adjustment, or external control, as described in the previous section. Nevertheless, research results such as those presented in [88] and [25], along with market-ready robotic systems such as the *Jaibot* by Hilti AG [34] or the *BauBot* by Fischer Holding GmbH & Co. KG [29] demonstrate that this technology offers more than just theoretical potential.

2.6 Specification of the Hardware Used

The software developed in this thesis is specifically tailored to the hardware available at the Technical University of Munich. The following section outlines the relevant specifications and the interaction of the different components during a 3D-printing operation.

2.6.1 Robot

The robotic system used is a *KUKA KR 340 R3300*. This serial industrial robot is equipped with six rotational axes (A1–A6), which are depicted in Figure 2.19b and is designed for applications with high load requirements and an extended working range. The robot itself weighs approximately 2.4 tonnes. Its maximum reach, measured from the robot base (axis A1) to the wrist (axis A5), is 3326 mm. The nominal payload capacity is 340 kg, but it can be increased to up to 418 kg depending on the position of the center of mass of the payload [42].

The repeatability of the robot's positioning is specified according to DIN EN ISO 9283 as ± 0.08 mm. Figure 2.19a shows the workspace that can be reached through the rotational motion of joints A2 to A5. This workspace is further extended by the rotation around axis A1 and is approximately spherical around the robot base.

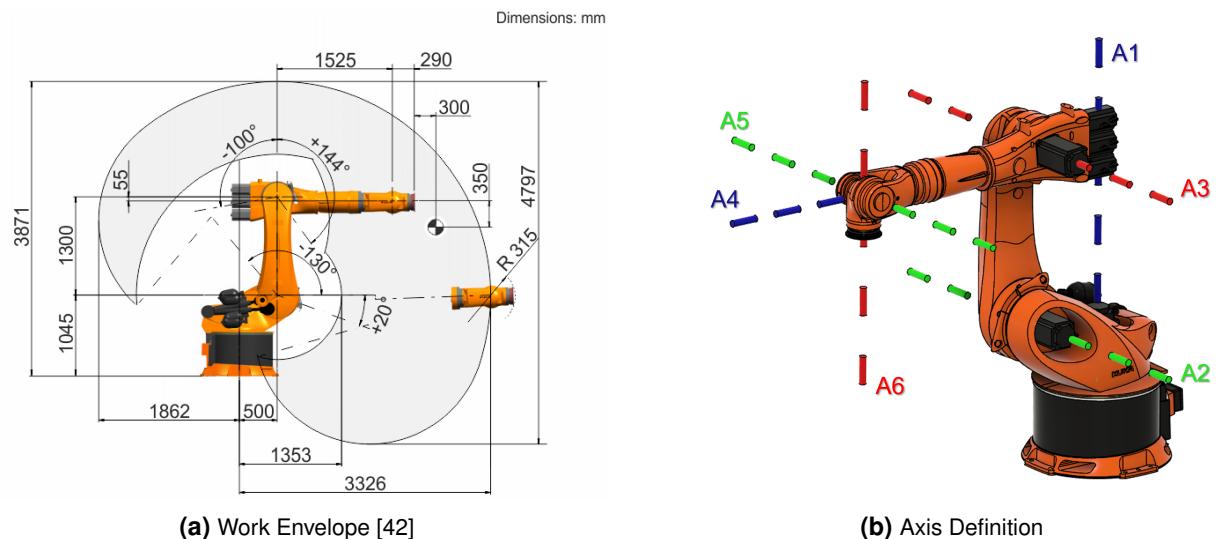
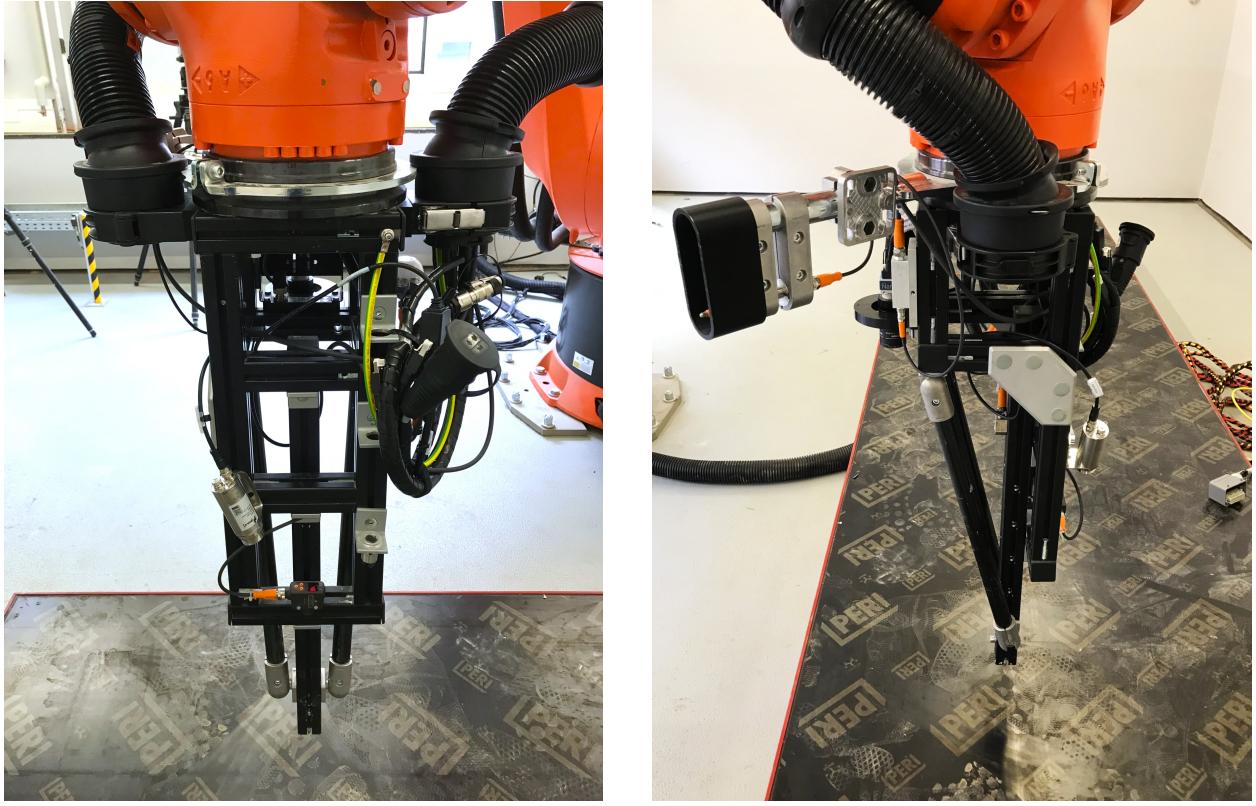


Figure 2.19 Technical Data for KUKA KR 340 R3300

At the flange, which forms the end of the KUKA robotic system and includes axis A6, an additional frame with dimensions of approximately $100 \times 30 \times 30$ cm is mounted. In addition to the nozzle of the printhead, this frame supports various measurement and control devices. These include, among others, a camera and a thermal imaging camera, a ring-shaped lighting system for uniform illumination of the printed object, and a contact sensor for the precise detection of the print bed position. The nozzle itself is connected to the material hose via a standardized Whitworth G1/2" thread, allowing for quick replacement and flexible selection of different nozzle geometries. A photograph of the complete tool assembly as used is shown in Figure 2.20.

The nozzle of the printhead defines the so-called *Tool Center Point* (TCP) of the robot, which is the point in space where the trajectory of movement is executed. The printing material is conveyed to the nozzle via the previously mentioned material hose with a nominal diameter of DN25. This hose is routed within the designated empty conduits along the robot arm.



(a) Front View of Tool-Frame Attached to Robot Flange

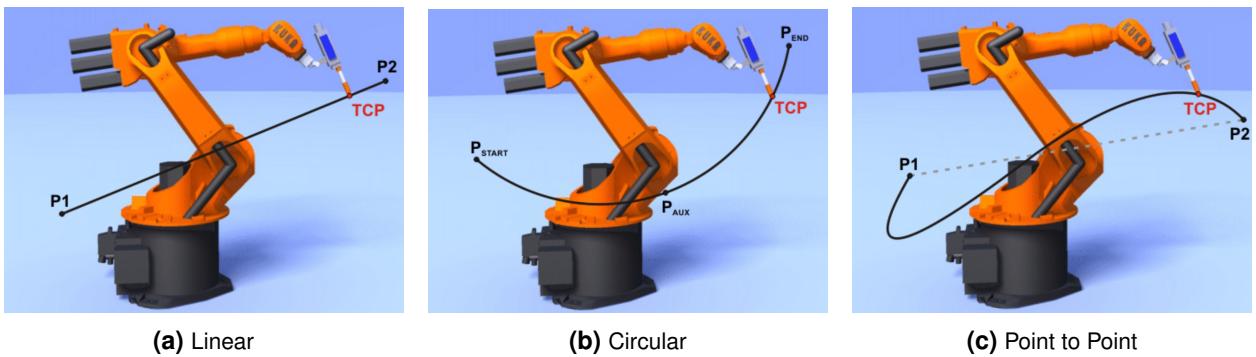
(b) Side View of Tool-Frame Attached to Robot Flange

Figure 2.20 Picture of Tool Frame in Use

The robot can be operated in two ways. First, manual control is possible using the *smartPAD* provided by KUKA, which features a graphical user interface (GUI). This allows the operator to intuitively and step-by-step move the robot into the desired position.

Second, the robot can be controlled by executing predefined motion sequences through code-based commands. For this purpose, a program written in the manufacturer-specific machine language *KUKA Robot Language* (KRL) is transferred to the robot. A complete KRL program typically consists of two file types, although in some cases all relevant information can be contained within one single file: The first file is the *.src* file, which contains the actual motion commands and the process logic. The second is the associated *.dat* file, where robot-specific parameters such as tool data, coordinate sets, and internal variables are defined.

To execute motion, the robot is capable of performing a variety of movement types. The three fundamental types of motion are illustrated in Figure 2.21.

**Figure 2.21** Basic Movement Types for KUKA KR 340 R3300 [43]

The visually simplest of these is the so-called *linear motion*, which connects the current tool coordinates to the target point along a straight path. The resulting trajectory represents the shortest distance between the two points and follows this line exactly.

Circular motions require, in addition to the start and end points, a third point, that is known as the *auxiliary point* (P_{AUX}). The robot moves along an arc that passes through all three specified points: the start point, end point, and auxiliary point.

In *point-to-point* motions, as in linear motion, only the start and end points are defined. However, unlike linear motion, the robot calculates a trajectory that reaches the target as quickly as possible. This computation takes into account factors such as the maximum angular velocities of the individual joints, which can be defined in the corresponding *.dat* file. As a result, the actual path is not predictable and may deviate significantly from a straight line between the start and end positions.

In addition to the predefined trajectory, velocity-related parameters can also be transmitted to the robot. These include the translational velocity (in m/s), the angular velocities of the individual axes (in rad/s), the acceleration (i.e., the first time derivative of velocity), and the so-called *jerk* (the second time derivative of velocity), which represents the abrupt change in acceleration (measured in m/s³) and significantly affects the precision of the robot. These parameters can be defined with high precision for each motion sequence, allowing for targeted control of the robot's dynamic behavior.

2.6.2 Pump

In order to ensure continuous material supply at the nozzle throughout the printing process, fresh concrete is constantly conveyed toward the printhead. For this purpose, a delivery pump of the type *PFT Swing L 400V FU*, manufactured by Knauf PFT GmbH Co. KG, is used. Depending on the configuration of the feed screw and the type of material being conveyed, the pump reaches a maximum delivery rate of 90 liters per minute and is equipped with a storage tank holding 70 liters. It is powered by a 6.05 kW electric motor operating at a variable speed ranging from 146 to 458 revolutions per minute (rpm). The maximum operating pressure is 30 bar, enabling conveying distances of up to 30 meters depending on the material consistency [66]. The pump dimensions are illustrated schematically in Figure 2.22.

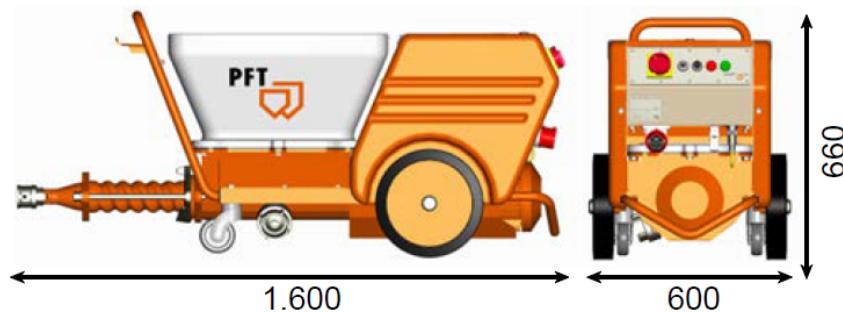


Figure 2.22 Side View and Front View of Pump with Measurements [mm] [66]

The underlying delivery mechanism is based on the principle of a progressive cavity pump. In this system, a metallic rotor with eccentric geometry rotates along a circular path inside an elastic stator. This motion continuously generates sealed conveying chambers that transport the material uniformly from the storage tank to the outlet. The flow rate can be controlled via the motor speed (rpm), which in turn is adjusted by a control signal in the range of 0 to 10 V. This allows for continuous speed variation within the specified range.

This pumping technology enables a low-pulsation material flow with consistent discharge performance, which is particularly beneficial for maintaining uniform print quality in a layer-based 3D-printing processes.

2.6.3 Mixer

Since the capacity of the pump's storage tank is insufficient to ensure continuous material supply, even during low-volume printing processes, additional material feeding during the printing operation is required. For this purpose, a continuous mixer is positioned in close proximity to the pump, supplying fresh material directly into the pump's feed hopper.

The mixer used is a *PFT HM 24* from Knauf PFT GmbH & Co. KG. The unit is equipped with a 3.3 kW electric motor and achieves a mixing output of up to 50 liters per minute at a defined speed of 280 rpm. The dry mortar (cement-based), supplied in bagged form, is continuously mixed with water in the mixer. Therefor water is introduced with a predefined dispense rate via a DN12 connection rated for pressures up to 2.5 bar. Additionally, aggregates with a maximum grain size of up to 6 mm can be processed [65]. This combination of continuous mixing and direct feeding ensures a reliable and uniform material supply, which is a critical factor for maintaining quality and process stability in concrete-3D-printing. A schematic representation of the mixer including rough dimensions is shown in Figure 2.23.



Figure 2.23 Mixer with Measurements Given in [mm] [65]

2.6.4 Print Bed

The print bed consists of two rectangular bolted formwork panels from the MAXIMO series by PERI [31]. The panels have dimensions of 3000×1200 mm and 1500×1200 mm, respectively, and are rigidly connected on the underside using M24 bolts. The entire print bed is supported by four pairs of height-adjustable feet, which allow for precise leveling of the surface. Each foot is additionally equipped with a load cell, enabling accurate measurement of both the total weight of the print and the weight distribution across the bed.

Any deviations in the bed's position or its parallelism with respect to the robot's nozzle prior to printing can be precisely detected using a contact sensor mounted on the robot's tool frame. These deviations are then compensated internally by the robot system in software. Throughout this thesis, a perfectly positioned print bed is assumed. Figure 2.24 shows the print bed in its assembled state.

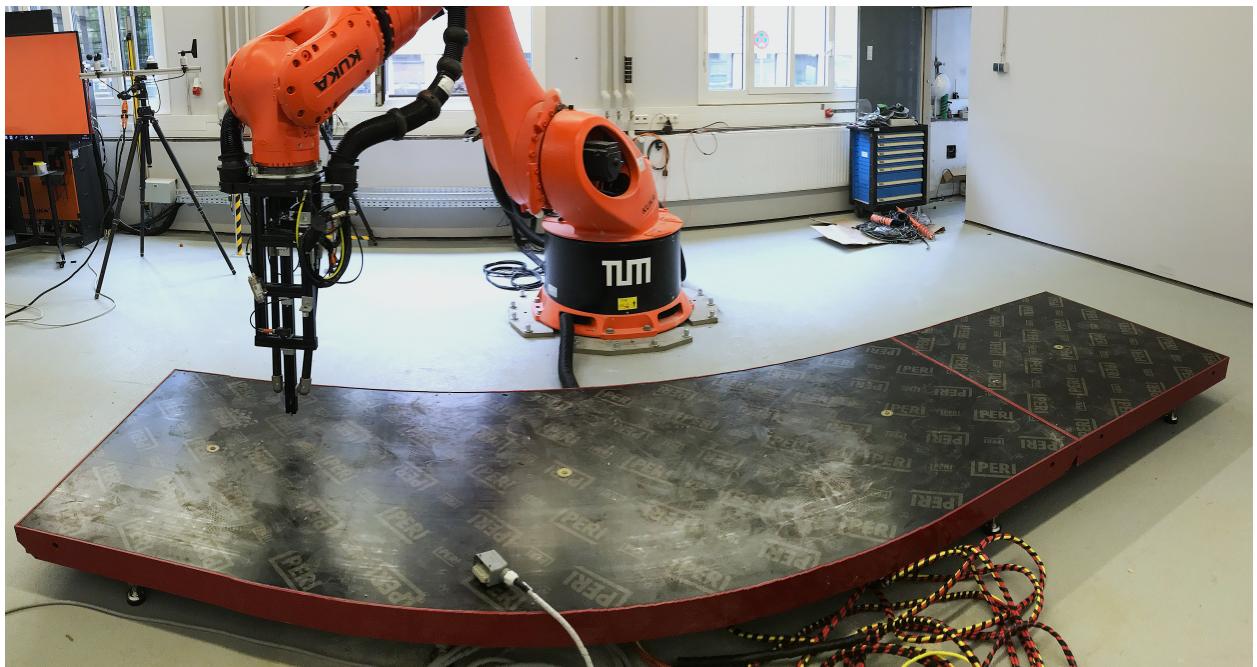


Figure 2.24 Print Bed Consisting of PERI MAXIMO Elements (Curvature Due to Lens Distortion)

2.6.5 Overview

Figure 2.25 shows a rendering of the hardware setup used during the printing process. Visible components include the *KUKA KR 340 R3300* with mounted tool frame, the print bed made of formwork panels, the delivery pump, and the continuous mixer used for material supply. This configuration forms the technical basis for the concrete-3D-printing processes investigated in this thesis.

During printing, mineral-based material is first mixed with water in the mixer. Depending on the amount of water added, both the water-to-cement (w/c) ratio or the consistency of the material can be precisely adjusted. The material-specific properties and the exact composition of the concrete mixtures are not considered in detail within the scope of this work.

Once the components are mixed, they are transferred from the mixer to the pump via the discharge pipe. From there, the material is conveyed through a DN25 hose to the nozzle. Through computer-controlled nozzle movement with partially constant speed, consistent beads can be deposited onto the print bed. Layer by layer, the desired object is created.

The objective of this work is to enable a better interaction between the various hardware systems and to provide a software platform that facilitates future printing operations.

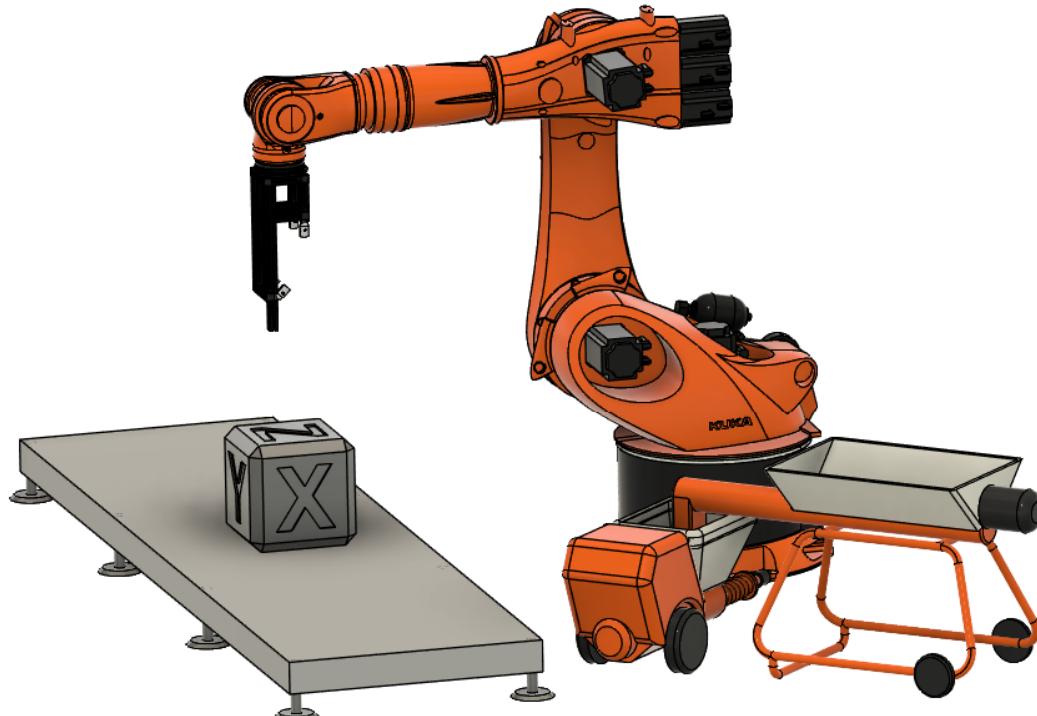


Figure 2.25 Rendering of Robot Cell with Print Related Hardware

3 From Model to Printed Object

3.1 Overview

This chapter traces the complete workflow from digital design to the physical printing of an object, using an exemplary component. The aim is to provide a practical explanation of the essential steps within the additive manufacturing process, ranging from the creation of a 3D model in a CAD environment to the handover to the printer.

A specially designed test object serves as the demonstrator. It combines typical geometric elements such as rectangular contours, circular segments, and a cylindrical recess (see Figure 3.1). Test geometries like this are widely used in 3D-printing, with the complexity of the structure depending on the printing method employed [55]. The illustrated object was created using *Computer Aided Design* (CAD) software [11].

The example shown initially illustrates the process exclusively for polymer-based materials using the Fused Deposition Modeling technique, without yet considering concrete-3D-printing. This method was chosen because, among the three common printing processes (SLA, SLS, and FDM), it is most comparable to concrete printing. A Cartesian 3D printer, such as the gantry printer described in Section 2.5.1, serves as the printing system. The considered objects are processed using planar slicing [57] and printed layer by layer. The terminology and process steps follow the slicing workflow described in [95]. The transfer of these procedures to systems that differ in material and process technology, such as concrete-3D-printing, is addressed in the following chapters.

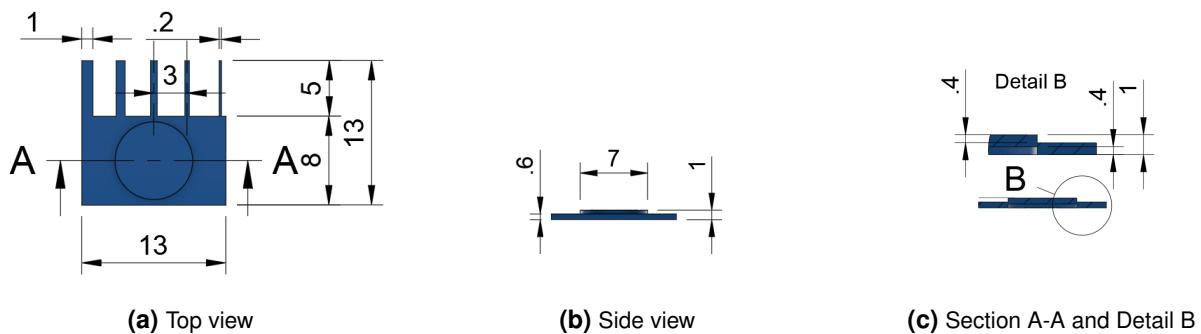


Figure 3.1 Technical Drawing of Test-Object with Measurements in [mm] Made Using Fusion 360 [11]

3.2 Workflow for 3D-Printed Objects

3.2.1 Design Process

The first step on the way to a printed component is the design of a suitable 3D model. Figure 3.2 illustrates the creation process of the test object used in this work.

Starting from a two-dimensional sketch that contains all essential features of the geometry as a projection onto the X - Y -plane, the rectangular base is first extruded. In the next step, the cylindrical extrusion is added, positioned centrally within the rectangular area. Finally, the circular cutout within the base is created using a *cut* operation.

This structure forms the basis for further processing in the slicer creating the required tool path and the subsequent transfer to the printer.

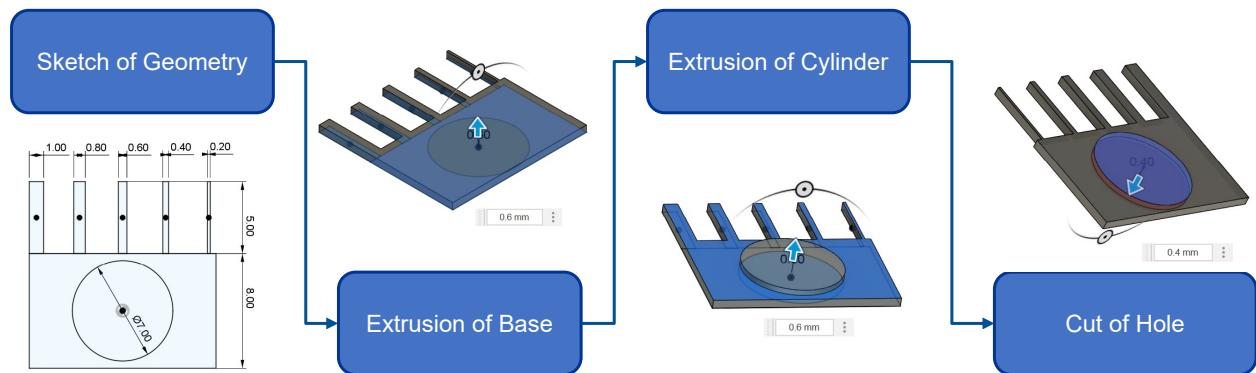


Figure 3.2 Design Process of Test-Object Using Fusion 360 [11]

3.2.2 Slicing Process

Once the digital 3D model has been created, it can be exported in a suitable file format and transferred to a slicer software. Common exchange formats include *.stl*, as well as *.step*, *.obj*, *.dae*, or *.3mf*.

The main task of the slicer is to divide the three-dimensional volume model into a sequence of individual layers. Within each layer, the object is described by line segments that define the motion paths of the printhead. This decomposition into discrete layers and path patterns results in a structured representation that a cartesian 3D printer can interpret and execute layer by layer. As a result, the object is no longer treated as a solid volume but as a superposition of thin, extruded material strands.

The following section provides a detailed examination of this process.

From Solid to Poly-Surface

In order to divide the volume model into individual layers and lines, it is first converted within the slicer into a file format known as the stereolithography file, or *.stl* for short. This format, developed by the aforementioned company 3D Systems [2], describes the object as a hollow geometry based on a triangulated surface mesh. During this conversion, the original solid model is approximated by a network of flat triangular facets. Each of these facets is defined by just three vertices, making triangles the simplest possible planar element for representing flat and angled surfaces. As a result, the volumetric properties of the model are discarded, and its geometry is described exclusively by its outer surfaces.

Figures 3.3a and 3.3b schematically illustrate how a solid body is transformed into such a polygon mesh. In the example shown, a simple cube is approximated using twelve individual triangles. Figure 3.3c shows the actual test object opened as an *.stl* file. It clearly demonstrates that the level of detail in the model is directly related to the number of polygons used: the more complex the geometry, the finer the resulting triangulation.

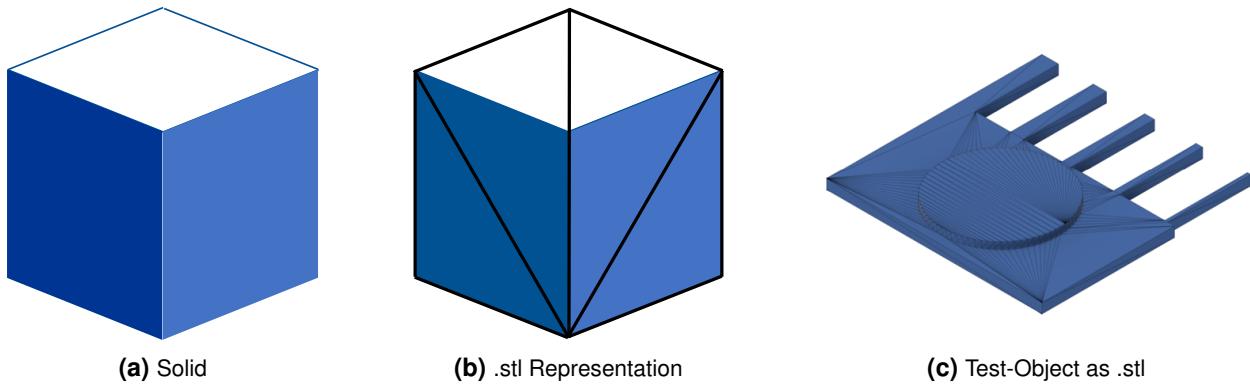


Figure 3.3 Object turned from Solid Body to Boundary Representation as .stl

Additionally, it becomes evident that curved structures, such as the cylindrical section of the test object shown above are not represented as exact circular segments, but are instead approximated by a series of straight line segments.

This simplification facilitates the subsequent slicing of the model geometry into planar cross-sections, as each surface is already defined by linear elements.

From Poly-Surface to Print-Path

Since the specific procedures used in slicing can vary significantly depending on the software, settings, and printing strategy, the following section provides only an abstracted representation of the process. Based on the works of [57], [5], and [96], a simplified outline is given of how a 3D model can be converted into a printable path. The actual implementation and sequence may differ between slicers and also depend on parameters such as material type, layer height, or motion strategy.

The first step in converting the mesh geometry into printable tool paths involves assigning mesh facets (polygons) to individual layers. For this purpose, the model is intersected along the Z -axis using a series of cutting planes spaced according to the selected *layer height*. Each of these planes is intersected with the triangular faces of the *.stl* mesh. The goal is to generate a valid line segment from the intersection between each facet and its corresponding plane. This segment will later form part of the closed contour of that layer.

Figure 3.4, adapted from [5], illustrates the geometric configurations that can occur during this intersection. Only configurations 1, 2, and 3 produce valid and usable results, since in these cases the slicing plane intersects exactly two edges of a triangle, thus defining a unique line segment.

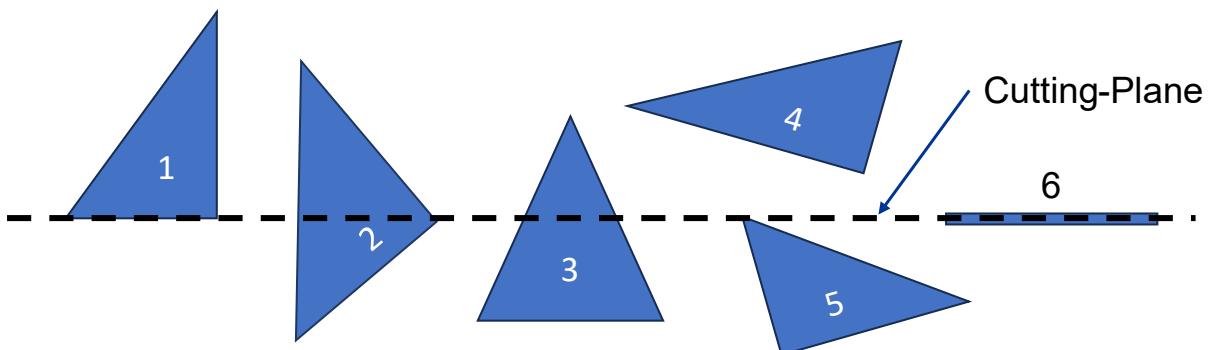


Figure 3.4 Cutting-Plane Intersecting Different Facets of Polygon Mesh [5]

- **Case 4:** The triangle lies entirely outside the current cutting plane and cannot be assigned to this layer.
- **Case 5:** The plane touches only a single vertex of the triangle. As this yields a point coordinate rather than a line segment, the case must be ignored.
- **Case 6:** All three vertices lie in the cutting plane. Although this geometrically defines a face, no unique line segment is created, and thus the case is also discarded.

If a valid intersection is detected, two points can be extracted from the triangle that define the start and end of a line segment. As the entire mesh is processed, multiple partially connected segments are generated. In a subsequent step, these segments are joined to form a continuous path that represents the outer contour of the object at the respective cutting plane height.

Figure 3.5 illustrates the resulting intersection points for Case 3. This procedure is then repeated for all required layers. The resulting tool path defines the outer *outer wall* of the object.

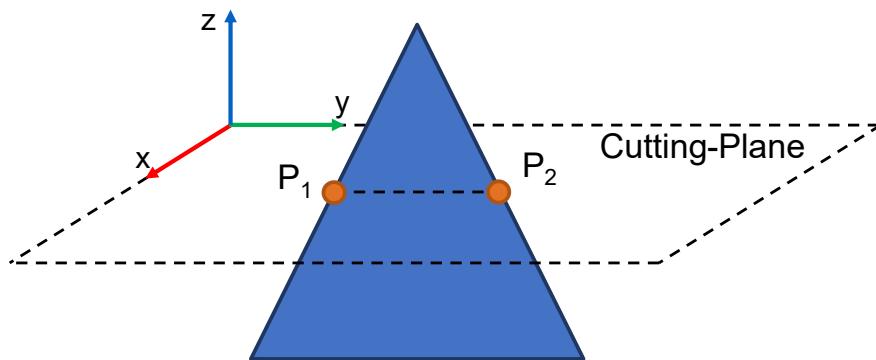


Figure 3.5 Intersection Between Cutting-Plane and Facet Creating Start- and Endpoint of Line Segment [5]

To allow the printhead to move between individual layers, additional connecting paths are required. Along these paths, however, no material is deposited, i.e. no extrusion takes place. Such non-extruding movements are generally referred to as *travel* moves. To ensure an efficient printing process, the travel speed is significantly increased compared to extrusion moves.

When transitioning to a travel move, it is common practice to retract the material away from the nozzle. This is especially true for materials prone to oozing from the nozzle without extrusion commands. This backward motion of the material in the opposite direction of the extrusion is known as *retract* or *retraction*, and it is essential to ensure print quality.

Since the printed object is not intended to consist solely of a hollow outer shell, additional steps are required in the generation of the print path. First, the path representing the *outer wall* can be offset inward by one extrusion width. The resulting additional path is referred to as the *inner wall*.

Because facets that lie exactly in the cutting plane are not considered during slicing, the object must also be closed at the top and bottom. The planes generated in this step are referred to as *surface*, and a further distinction is made depending on their position: the *bottom surface* closes the object in the direction of the print bed, while the *top surface* closes it in the direction of the printhead.

To optimize material usage, the number of wall lines and surface layers can usually be specified explicitly by the user within the slicer. The remaining volume is then filled with the so-called *infill*. This consists of repeated geometric patterns, and its density can be set by the user. Figure 3.6 shows a selection of different infill patterns.

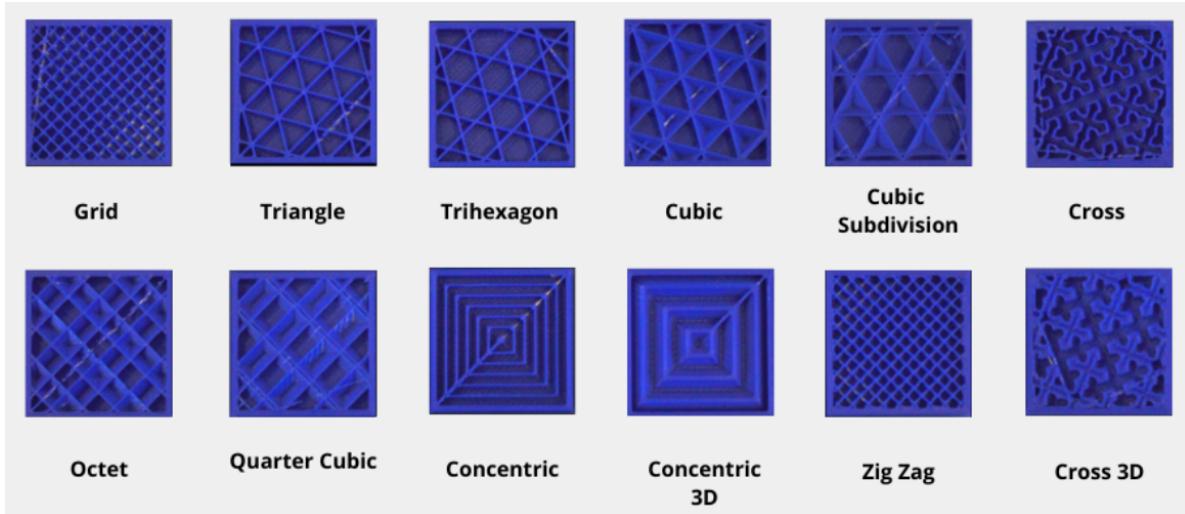
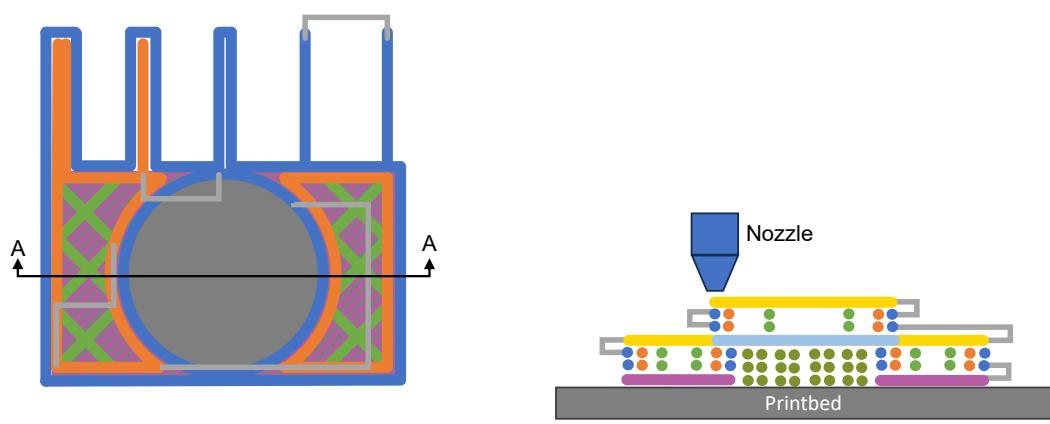


Figure 3.6 Selection of Different Infill Patterns Used in Polymer 3D-Printing [14]

If two distant points within the model need to be connected without any underlying material from previous layers, the resulting line segment is commonly referred to as a *bridge*. When using fast-setting materials, such bridges can span distances that exceed the nozzle diameter many times—sometimes by several hundred times.

However, if the geometry is such that the bridge can no longer support its own weight across the bridges span or would collapse under the weight of subsequent layers, the use of a auxiliary structure becomes necessary. This is known as a *support*. A support structure may be anchored either directly on the print bed or on the object itself and is built up layer by layer until it reaches the critical geometry. After the printing process is complete, the support must be removed in order to obtain the printed object in its intended shape according to the CAD model. Figure 3.7 provides a schematic overview of the various line types using the test object. A corresponding assignment of these line types is given in Table 3.1.



(a) Top View at $Z = 0.4$ mm (without Support)

(b) Section A-A form Figure 3.7a (with Support)

Figure 3.7 Overview on Different Linetypes Used After Slicing of the Test-Object

Table 3.1 Color Legend and HEX-Code for the Different Line Types Used in Figure 3.7

Wall Outer	#0065BD	Infill	#A2AD00	Wall Inner	#E37222
Bridge	#5E94D4	Top Surface	#FEDE34	Support	#7D922A
Bottom Surface	#B55CA5	Travel	#999999		

To ensure positional accuracy and geometric fidelity of model and printed part, the adhesion between the printed object and the print bed is of critical importance. For materials with poor bed adhesion, it may be necessary to implement additional measures to prevent detachment during printing. One basic method is to increase the extrusion volume in the first layer, which without changing the nozzle-to-bed distance presses the material more firmly onto the bed surface.

Over time, various alternative techniques have been developed to improve first-layer reliability. Among these, the so-called *skirt*, which represents a closed perimeter around the printed object (see Figure 3.8a) and serves mainly to prime the nozzle by extruding material before the actual print begins. This helps avoid under-extrusion (insufficient material flow) in the initial layer.

The *brim* (see Figure 3.8b), by contrast, is explicitly designed to improve adhesion. It enlarges the contact area of the first layer (*Layer 0*) by adding an outwardly extending edge around the object. After printing, this extra rim is mechanically removed, and the part is usually deburred to eliminate remaining artifacts.

A more robust method is the use of a *raft* (see Figure 3.8c), which slightly lifts the object above the print bed. The gap beneath is filled with a multi-layer structure, forming essentially a platform, that increases the effective contact surface and acts as a bonding interface between bed and object. This helps mitigate issues such as *warping*, which can occur due to uneven material shrinkage during cooling and lead to detachment of lower layers or corner lifting. The raft absorbs these effects, reducing their impact on the final geometry.

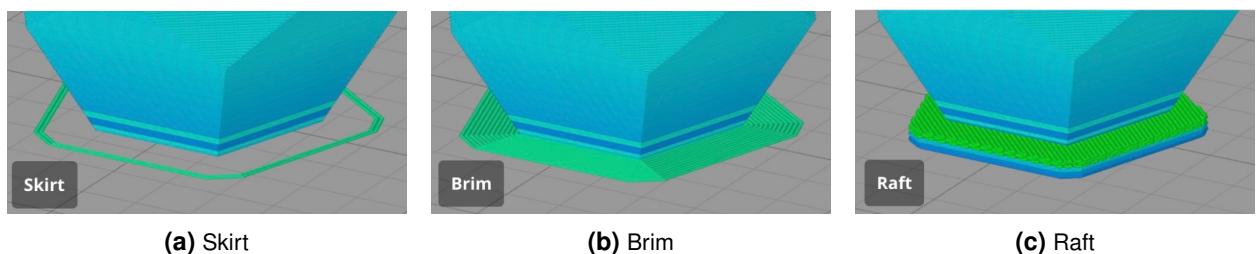


Figure 3.8 Different Bed Adhesion Features Taken from [6]

The line types described above and their respective functions provide a summary of the most commonly used path types in the slicing process. However, depending on the slicer software used, both the terminology and the specific function of individual line types may vary. A more detailed analysis and comparison of different slicer outputs is presented in Chapter 3.2.3.

Machine Code Generated by the Slicer

To make the linear movements defined during the slicing process interpretable for the printer, the coordinate information must be translated into a machine-readable language understood by the printer's firmware interpreter. This is typically achieved by generating commands in the so-called geometric code language (G-code), as specified in the DIN 66025 standard [59].

A typical G-code command includes target coordinates, which are specified using the prefixes X, Y, and Z, each followed by a numerical value. Additionally, movement parameters and command patterns are included. For instance, the F command defines the maximum feed rate in millimeters per minute. The extrusion of material is controlled via the prefix E, whose value specifies the volume of material extruded during the movement, usually given in mm³.

In addition to these commands, which are referred to as *General Commands* and build the essential base for the printing process the printer must also process instructions for managing hardware components and settings. These are known as *Miscellaneous Commands* and are identified by the prefix M. They are often accompanied by an S parameter, which defines the value to be set.

In Table 3.2, several key G-code commands are presented as examples, along with explanations of how they are interpreted by the firmware *Marlin* [47] used in this thesis.

The listed G-code commands represent a large portion of the input parameters used to control the printer. A comprehensive overview of all commands supported by the *Marlin* firmware can be found in [48].

However, since both the selection and the sequence of these commands are largely determined by the slicing software used, the following section will outline the general structure of a typical G-code output. In addition, key parameters generated by various slicers are compared and discussed.

Table 3.2 Selection of Different G-Code Commands Used with Marlin [47]

Command	Example	Explanation
General Commands (G-Commands)		
G0 – Linear Move	G0 X<pos> Y<pos> Z<pos> F1500	Moves the printhead linearly to the given coordinates at a feed rate of 1500 mm/min. No material is extruded.
G1 – Linear Move + Material Extrusion	G1 X<pos> Y<pos> Z<pos> E3.5 F<val>	Moves linearly to the target while extruding 3.5 mm ³ of material.
G28 – Auto Home	G28 X Y Z	Moves all (or specified [X Y Z]) axes to their endstops to establish the printer's reference (home) position.
G92 – Set Position	G92 E0	Resets the current extruder value to 0 mm ³ without movement.
Miscellaneous Commands (M-Commands)		
M82 – Set Extruder to Absolute Mode	M82 G1 X<pos> Y<pos> Z<pos> E10	Sets the extruder to absolute mode. E10 extrudes to 10 mm ³ absolute.
M83 – Set Extruder to Relative Mode	M83 G1 X<pos> Y<pos> Z<pos> E5	Sets the extruder to relative mode. E5 adds 5 mm ³ to the current position.
M104 – Set Extruder Temperature	M104 S200	Sets target temperature to 200 °C, proceeds immediately without waiting.
M105 – Get Temperature	M105	Requests current nozzle and bed temperatures. Returns values like T:200 /200 B:60 /60.
M106 – Fan On	M106 S128	Turns on the fan at 50 % speed to cool freshly extruded material.
M107 – Fan Off	M107	Turns off the currently active fan.
M109 – Set Temperature and Wait	M109 S200	Sets the temperature to 200 °C and pauses execution until it is reached.
M201 – Set Max Acceleration	M201 X1000 Y1000 Z500 E<val>	Defines the maximum acceleration [mm/s ²] for each axis during printing moves.
M203 – Set Max Feedrate	M203 X500 Y500 Z100 E<val>	Sets the maximum speed [mm/s] each axis can move.
M204 – Set Acceleration	M204 S400 P<val> T<val> R<val>	Specifies acceleration values for different move types: default (S), printing (P), travel (T), and retract (R).
M205 – Advanced Settings	M205 S<val> B<val> X8.0 Y8.0 Z0.5	Configures advanced motion settings like minimum segment time (B), jerk limits (X/Y/Z), and buffer behavior.

3.2.3 Comparison of Different Slicer Outputs

As an example, the outputs of three widely used slicing programs are compared: *PrusaSlicer* version 2.9.2 [68], and *OrcaSlicer* version 2.2.0 [63], *Ultimaker Cura* version 5.8.0 [97].

The corresponding example files, fully documented in the appendix (see Appendix A), are based on the previously introduced test object. To ensure a consistent basis for comparison, the default print profile for the *Ender 3* printer by Creality [80] was used in all three programs. In each case, the profile is configured for a layer height of 0.2 mm. As a result, the entire test object consists of five layers (Layer 0 to Layer 4). In addition, the following user-defined parameters were adjusted uniformly from each given preset:

Table 3.3 Custom Slicing Parameters Used for Test-Object Generation

Parameter	Command	Description	Unit/Type
Line Width	0.4	Width of extruded line	mm
Infill Density	20	Percentage of internal part volume to be filled	%
Infill Pattern	Grid	Internal fill structure using a grid layout	Pattern
Top Layers	1	Number of solid layers at the top	Integer
Bottom Layers	1	Number of solid layers at the bottom	Integer
Wall Line Count	2	Number of perimeters around the part	Integer
Thin Wall Printing	true	Enables printing of features smaller than the nozzle diameter (e.g., 0.2 mm)	Boolean

PrusaSlicer 2.9.2

For the settings used in *PrusaSlicer* (referred to as Prusa hereafter), the estimated print time is two minutes, with approximately 40 % of the movements classified as *travel* (see Figure 3.9c). With a selected filament diameter of 1.75 mm, the total printed volume amounts to 104.16 mm³, corresponding to a filament length of 0.06 meters. The results generated with Prusa are shown in Figure 3.9.

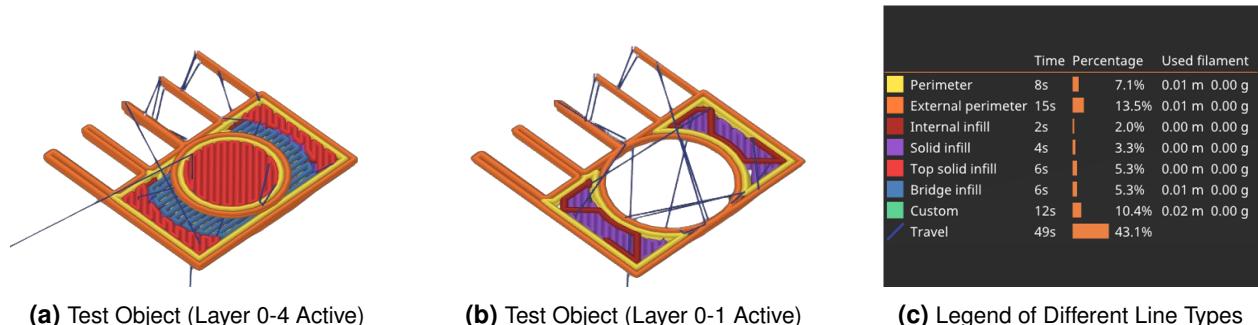


Figure 3.9 Slicer Output for Prusa Slicer Version 2.9.2

OrcaSlicer 2.2.0

OrcaSlicer (referred to as Orca hereafter) provides a more detailed breakdown of results. For example, in addition to *retract* movements, it also explicitly lists inverse retractions, during which material is fed back toward the nozzle and refers them as *unretract*. The material consumption is estimated at 0.05 m, while the total print time, including printer preparation, is also estimated at two minutes. Figure 3.10 presents a summary of the Orca graphical user interface.

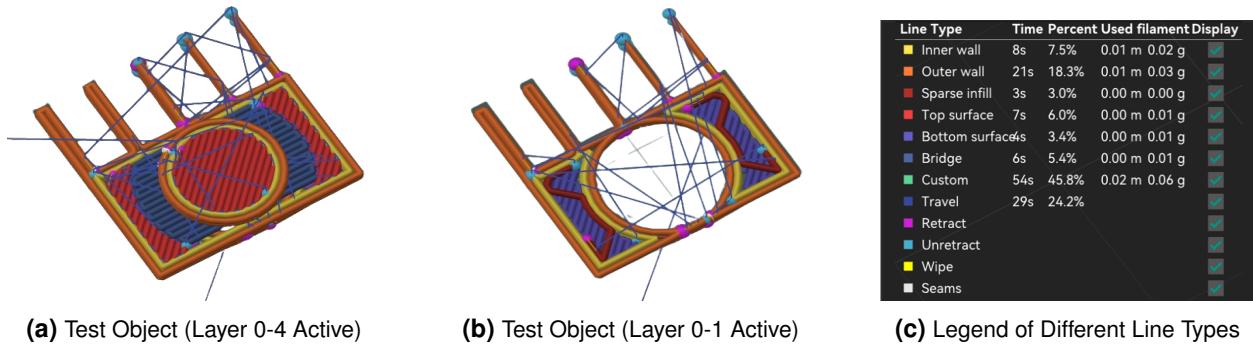


Figure 3.10 Slicer Output for Orca Slicer Version 2.2.0

Cura 5.8.0

Ultimaker Cura (referred to as Cura hereafter) estimates the expected print time at one minute, with only about 9 % of that time attributed to *travel* movements. Unlike Prusa, Cura also tracks the time spent on *retractions*, which account for approximately 22% of the total print time. With the same filament diameter of 1.75 mm, Cura estimates a significantly lower material usage of just 0.03 m. All visually accessible results are shown in Figure 3.11.

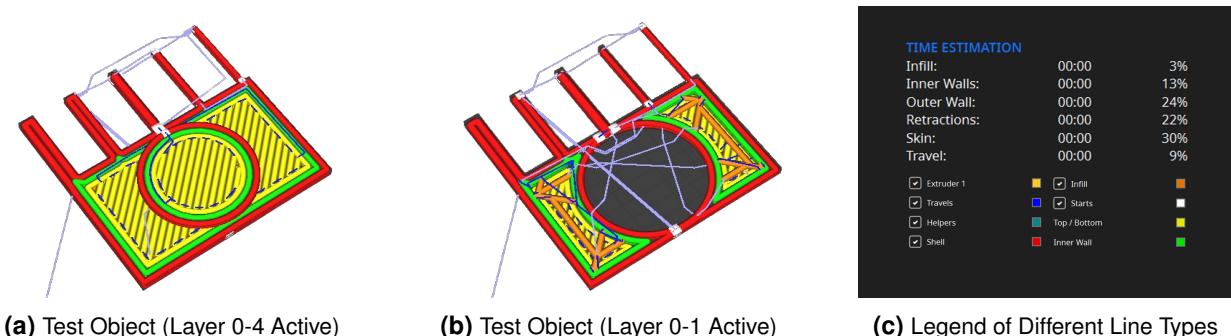


Figure 3.11 Slicer Output for Cura Version 5.10.0

3.2.4 G-Code Analysis

Initialization Commands

Based on the G-code files presented in Appendix A, which form the basis of the data discussed earlier, it becomes clear that the generated code is structured into three sections, each highlighted with a distinct background color in the listings. The first part in Appendix A consists of an initialization section, which begins with comments (introduced by a semicolon ;) related to the respective print object. In the case of Prusa, this section includes information such as line width (see Listing A.1.1.1, lines 1–11), while Orca also adds data on the number of layers and the object dimensions (see Listing A.2.1.1, lines 1–15). Cura, by contrast, limits itself to object-specific details such as the minimum and maximum dimensions, the selected layer height, and material information including filament diameter and length (see Listing A.3.1.1, lines 1–12).

Following the initial block of general information, Prusa and Orca include a configuration section in which printing parameters are defined using M200 commands (see Table 3.2). These parameters include, among others, speed limits, accelerations, and so-called jerk values (cf. Chapter 2.6.1). Cura omits explicit parameter constraints at this stage and instead provides a warning system within its graphical user interface . If an incompatible value is entered there, a warning message appears. This happens, provided that the correct printer profile is selected.

In contrast, to both Prusa and Orca, Cura uses this stage to set the target temperatures for the nozzle and heated bed, thereby preparing the printer for the upcoming print process.

Speaking of which, the configuration block is followed in both Prusa and Orca by the command G90, which sets the printer's coordinate system to absolute positioning relative to the home position (cf. Table 3.2, command G28). This command marks the beginning of the so-called *start G-code*, which can be customized by the user within the slicer. While Cura does not explicitly include the G90 command, it also defaults to absolute positioning based on the home position. In Cura, the start G-code begins at line 19.

The purpose of the start G-code is to optimally prepare the printer for the actual printing process. This includes, in particular, ensuring that the first layer adheres reliably to the print bed and avoiding print artifacts, which will be discussed in Chapter 3.2.5 caused by excessive or residual material around the nozzle during the heating phase.

Print Commands

As observed, both Prusa and Orca use only the G1 command for all movements, regardless of whether material is extruded (indicated by an E<val> entry within the G-code line). Cura, by contrast, adheres more closely to the formal G-code specification, using G0 commands for non-extrusion moves and G1 commands for lines that include material extrusion via E<val>. This makes the code much more readable and aids visual understanding.

The printing of the object itself proceeds similarly across all three slicers (see Listings A.1.1.2, A.2.1.2, and A.3.1.2), with only minor differences in the handling of start and end points of travel movements, and in the structure of the infill pattern. The latter shows particularly clear deviations between Cura and the other two slicers (see Figures 3.9b, 3.10b, and 3.11b).

Notable differences are also found in the naming of individual line types, which are indicated prior to each extrusion via the comment prefix ;TYPE: in the G-code (highlighted in the Appendix A). Each slicer uses its own terminology, offering insight into the internal structuring and segmentation of the generated code.

Table 3.4 provides an overview of all line type designations used in the appendix.

Table 3.4 Line Type Labels Used by Different Slicers

Prusa	Orca	Cura
External perimeter	Outer wall	WALL - OUTER
Perimeter	Inner wall	WALL - INNER
Solid infill	Bottom surface	SKIN
Internal infill	Sparse infill	FILL
Bridge infill	Bridge	
Top solid infill	Top surface	
Custom	Custom	

As shown in Table 3.4, Cura distinguishes between line types in a significantly less differentiated manner than the other two programs. For instance, bridge structures are not explicitly labeled as a distinct line type in Cura but are instead grouped under the general label SKIN (see Listing A.3.2.2, line 695 ff.). This umbrella term also includes lines that are identified as *Bottom Surface* or *Top Surface* in Orca and as *Solid infill* or *Top solid infill* in Prusa.

The absence of a *Custom* type in Cura can be attributed to the fact that user-defined G-code sections during initialization are not tagged with a specific line type. Although these custom commands are included in the generated G-code, they are not semantically annotated as they are in Prusa or Orca.

Figure 3.12 provides a graphical illustration of these categorization differences, based on the user interfaces of the three programs.

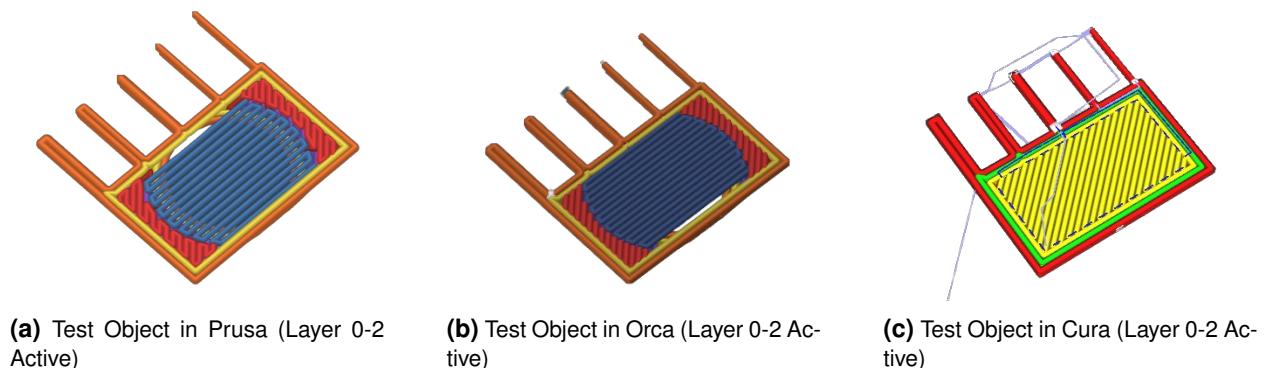


Figure 3.12 Difference in Line Types Regarding *Bridge* Declaration

During the transition between individual layers, all slicers insert a specific G-code block. In the case of Prusa and Orca, this block can only be partially customized via the printer-specific settings within the slicer. Which could be an issue when adapting the G-code for concrete-3D-printing.

As an example, Table 3.5 presents the layer change G-code as defined in Prusa under the fields *Before layer change G-code* and *After layer change G-code*. The table displays both the user-defined entries within the slicer (left column) and the actual output G-code block (right column), taken from Listing A.1.2.2, lines 433–454.

Table 3.5 Example of G-Code Inserted Before and After Layer Change in PrusaSlicer

Prusa Input	G-Code Output
Before layer change G-Code ;BEFORE_LAYER_CHANGE G92 E0 ;layer_z After layer change G-Code ;AFTER_LAYER_CHANGE ;layer_z	; LAYER_CHANGE ; Z :0.4 ; HEIGHT :0.2 ; BEFORE_LAYER_CHANGE G92 E0 ; 0.4 G1 E-3.5 F3600 ; WIPE_START G1 F7200 G1 X110.416 Y116.028 E-0.40842 G1 X110.959 Y116.028 E-0.25793 G1 X109.977 Y115.046 E-0.65966 G1 X109.977 Y114.680 E-0.17399 ; WIPE_END G1 Z0.4 F9000 ; AFTER_LAYER_CHANGE ; 0.4 G1 Z .4 G1 X115 .779 Y116 .743 G1 E5 F2400

The command `layer_z` represents a slicer variable that returns the current Z -height within the G-code. The *Before layer change* sequence is then extended by an automatically generated retract command (`G1 E-3.5 F3600`). This initiates an automatic so-called *wipe* sequence, which is intended to remove excess material that may unintentionally ooze from the nozzle.

For this purpose, a series of short retracts and movements in the X - Y plane is executed, during which the adhering material is wiped off inside the part geometry.

Because the retracted material amount (extrusion value E) in this sequence is defined relative to the previous position (cf. `G92 E0` in Table 3.5), it must be compensated at the beginning of the new layer. This is achieved by a corresponding extrusion move in the last line of the G-code block shown in Table 3.5, which resets the E value relative to its value before the layer change to zero.

Similar sequences can also be observed in Orca (see for example Listing A.2.2.2, lines 478–494). Cura, by contrast, performs the layer change without an automatic in-model nozzle wipe beforehand (see for example Listing A.3.2.2, lines 766–774). This sequence, which consists of the printing of different line types followed by a layer change is repeated cyclically until the object is completed.

Reset Commands

Once printing is complete, all three slicers initiate an end sequence (see Listings A.1.3.3, A.2.3.3, and A.3.3.3). Similar to the start sequence, this section includes commands that can be defined by the user directly within the slicer (for Prusa, see Listing A.1.3.3, lines 1308–1315). During this sequence, the printhead is moved away from the object, and the build plate is positioned in a way that allows for convenient removal of the finished part. In addition, M-commands such as `M104` and `M107` are used to disable the nozzle heater and deactivate any running fans.

A large portion of the following code block in Prusa and Orca contains metadata about the parameters used during the slicing process (for Prusa, see Listing A.1.3.3, lines 1316–1691, for Orca see Listing A.2.3.3, lines 1453–1932). Cura, once again, is more restrained in this regard and provides only limited information about the input parameters used.

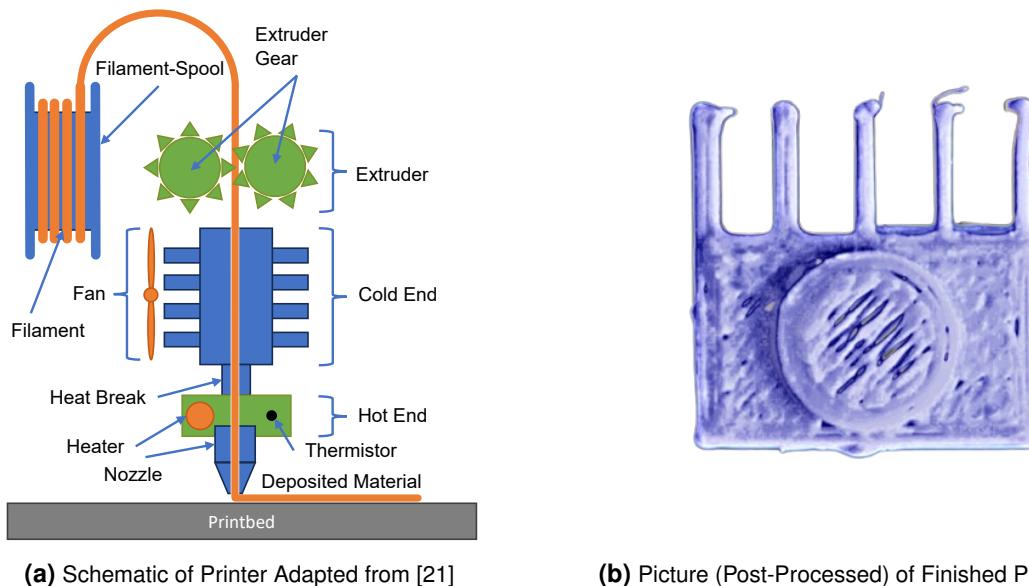
After the slicer has generated the complete G-code file, the actual fabrication process begins with its interpretation and execution by the printer. The next section therefore focuses on the practical realization of the print path and describes how the object is built up layer by layer.

3.2.5 Print

As briefly outlined in Chapter 2.1, the object is now constructed layer by layer on the print bed based on the generated G-code. Figure 3.13a provides a detailed schematic overview of the components required for this process in an FDM printer.

The raw material is typically supplied as filament on a spool and is continuously fed into the system by the extruder with its subsequent gear assembly. From there, it enters the so-called *hotend assembly*, which consists of two individual components. The task of the latter *hot end* is to heat the filament to the required melting temperature for printing. During this process the temperature is precisely monitored and controlled by an integrated thermistor.

To prevent the melt zone from extending from the hot end toward the *cold end*, cooling fins and an associated fan ensure the targeted dissipation of excess heat. If this heat is not sufficiently removed, a phenomenon known as *heat creep* [73] may occur. In this case, the filament begins to soften prematurely inside the cold end. As a result, the necessary feeding pressure from the extruder gears can no longer be reliably maintained, potentially leading to a nozzle blockage caused by backflow or excess material.



(a) Schematic of Printer Adapted from [21]

(b) Picture (Post-Processed) of Finished Printed Part

Figure 3.13 Overview on Printer and Printed Part

In addition to the mechanical and thermal influences described in this chapter, a variety of other defects may occur during the printing process. These include insufficient bed adhesion or lifting of the print from the bed (warping), inconsistent layer deposition, under- or over-extrusion, as well as temperature-related issues such as stringing (the formation of fine polymer threads between extrusion stop and start). Figure 3.13b shows the completed print of the test object after removal from the print bed. Visible are, among other things, artifacts caused by extrusion stops on the “fingers” of the object, as well as under-extrusion in the topmost layer of the cylinder. These problems are primarily influenced by the interaction between the hardware used, filament quality, and printing parameters. A comprehensive overview of common FDM-related defects and their remedies can be found, for example, in [67].

A detailed investigation of these aspects is intentionally omitted in this thesis, as the material used may behave totally different from the described thermoplastics. It is henceforth assumed that the printing environment is properly calibrated and that the process is carried out under favorable conditions.

Once printing is successfully completed, the finished object must be detached from the print bed. If support structures have been used, they must be removed in a separate post-processing step. These actions mark the end of the printing process. The printed part is now ready for further post-processing or direct use.

3.2.6 Summary

The processes described in the preceding chapters form the basis for transferring polymer-based FDM printing to mineral materials. Figure 3.14 provides a summary of the entire workflow. Feedback loops may be required between or within individual process steps to ensure an optimal print result.

As shown, the general structure of the slicers is quite similar. This is particularly evident in the case of Prusa and Orca, which can be attributed to the fact that both are based on the same open-source slicing engine: Slic3r [7].

The insights gained in this chapter will now be transferred to the use of concrete as a printing material and applied to the available hardware (cf. Chapter 2.6.1 ff.).

The slicer used as a foundation for this purpose is *Ultimaker Cura*. Despite the slight disadvantages discussed earlier compared to Prusa and Orca, Cura was selected due to its nearly unrestricted configuration capabilities. In particular, parameters such as nozzle diameter and print bed dimensions can be freely defined without constraint. Additionally, the user retains full control over the generation of G-code, such as that shown in Table 3.5. Owing to the increased control over output and the author's prior experience with this slicer, Cura serves as the foundation for the program described in the following chapters.

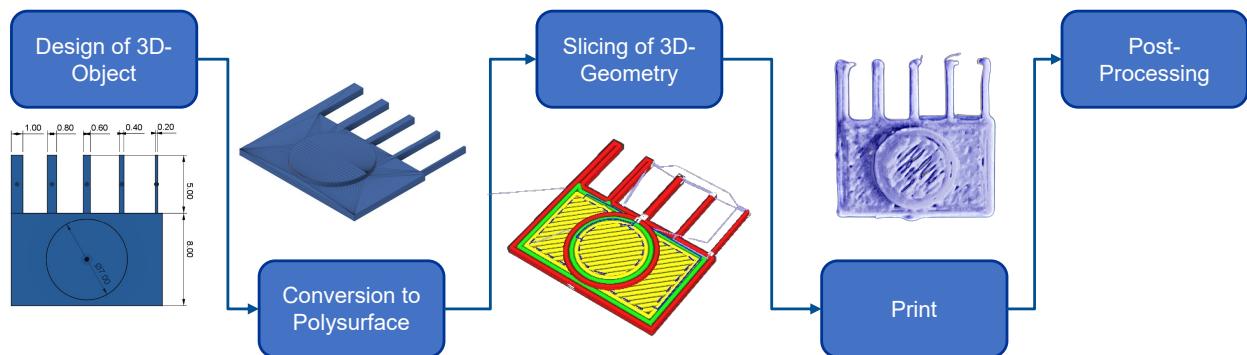


Figure 3.14 Summary of 3D-Printing Process

4 Software Development: Implementation of the Process Chain

4.1 Introduction

This chapter focuses exclusively on the software developed within the scope of this thesis and its individual components. It begins with an overview to provide a basic understanding of the program's architecture and the structure of its subfunctions. Subsequently, the functionality and implementation of the central modules are discussed in detail in the respective sections.

The program itself, along with its full documentation, is freely available via the following [GitHub project](#). The implementation was carried out using *Python* [71] in version 3.12. The project is structured into several packages, each divided into dedicated modules. Within these modules, arguments are processed using functions and classes. A complete list of additional external libraries used for the program is documented in the *requirements.txt* file included in the GitHub repository. To improve code quality and structure, AI-based tools were selectively employed [62]. These tools were used exclusively to improve structure, readability and robustness but were not used for the automated generation of functional program components.

The goal of the software is to integrate various existing software tools and thereby make the relevant information from the FDM slicing process usable for the available KUKA robot, while simultaneously simplifying or automating the process chain up to printing. To achieve this, a user-provided *.stl* file is automatically converted into G-code using established slicer software. Subsequently, the G-code is parsed to extract data relevant to concrete-3D-printing and adapted to the specific hardware. A core feature of the program is the translation of G-code into the machine language used by KUKA (KRL), as well as the code-based control of the extrusion pump. The most important process parameters are then compiled for the user in a summary report in *.docx* format. Finally, a *.3dm* file is generated from the collected data, providing a structured visual representation of all relevant information within Rhinoceros 3D (Rhino) [75].

The slicing process is based on the previously introduced slicer *Ultimaker Cura* [97], version 5.8.0. As discussed in Chapter 3.2.3, Cura produces a less detailed output compared to Prusa and Orca. However, it offers virtually unrestricted configuration options, including print bed dimensions and nozzle diameters, and supports direct execution of the *CuraEngine.exe* via a command line interface (CLI), which handles the slicing process. Further information on this is provided in [98]. This functionality is of central importance for the automation and flexibility of the overall workflow.

4.2 Program Overview

The program is designed primarily as a serial pipeline to convert the G-code generated using Cura into various output files. Figure 4.2 provides an overview of the individual packages and their corresponding modules. The core of the program is the `c3dp_main` function (Concrete-3D-Printing), which coordinates the functionality of all subordinate modules and is responsible for information exchange, data storage, and function calls.

For slicing the current implementation of the program only contains a single package under `slicer`, which handles interaction with the `CuraEngine`. However, due to its modular design, the program is intended to be easily extendable to include additional slicers or functionalities. While this potential is referenced repeatedly throughout the thesis, it is not explored in depth.

The following sections examine the individual components of the packages and folders in greater detail. Each chapter heading indicates the corresponding folder or package to which the described function belongs inside the program. To improve the readability of inputs and code references, the formatting conventions shown in Table 4.1 will be used throughout this thesis to visually distinguish relevant program components and arguments.

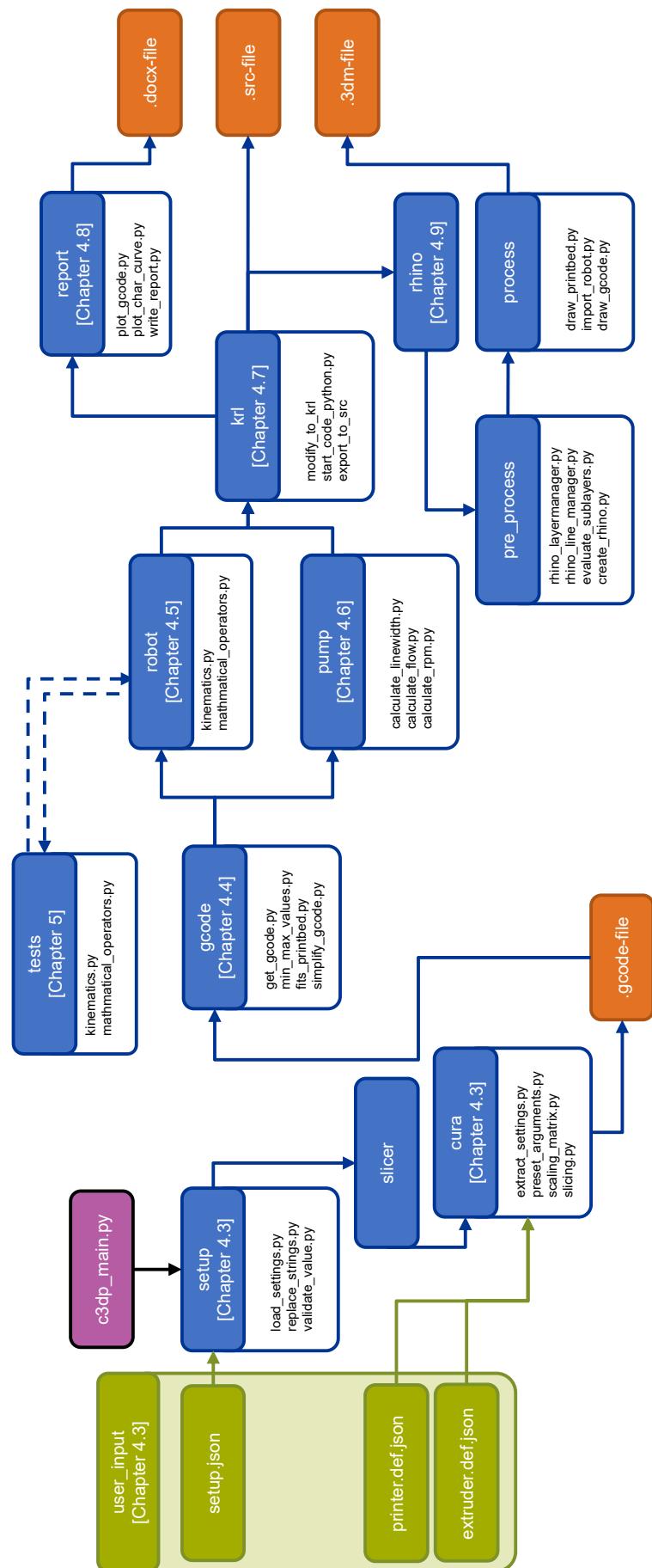
Table 4.1 Visualization and Notation Styles Used for Input and Output Formats

Python Code	
<pre>1 def example(alpha: int, beta: float) -> None: 2 gamma = alpha + beta 3 print(f"Sum of {alpha} and {beta} is {gamma}")</pre>	
.json-File	
<pre>"machine_nozzle_size": { "default_value": 25, "type": "float", "unit": "mm" }</pre>	
CLI Output	
<pre>[INFO] This is an example for a CLI Output line</pre>	
Module \ Folder \ Function	user_input
File	<code>printer.def.json</code>
Argument \ Variable	"default_value"
Section in <code>setup.json</code>	"Directory"

At the beginning of each chapter, a progress bar is included to indicate where the content of the upcoming section fits within the overall sequence of the `main` function. Therefor the color scheme used in all associated flow diagrams follows a consistent logic throughout this thesis. Cells highlighted in green represent input values, while orange cells generally indicate either output values or conditional operations such as `if`-statements. In contrast, cells shaded in blue refer either to packages and folders that contain functional components or to individual functions themselves. Figure 4.1 illustrates a representative progress bar, as used at the beginning of each section in this chapter to indicate which parts of the overall workflow have already been executed and which will be covered next.



Figure 4.1 Progress Bar Referring to Packages in Program

**Figure 4.2** Overview of Program with All Implemented Modules and Information on Different Sub-Operations

4.3 Input Parameters [folder: *user_input*]

The user input required for executing the program is organized within the dedicated *user_input* module. This module consists of three *.json* files, which are divided into a general setup file (*setup.json*) and two configuration files required by Cura (*printer.def.json* and *extruder.def.json*).

The following section first focuses on the files relevant for interfacing with Cura.



Figure 4.3 Progress Bar for Chapter 4.3

4.3.1 Cura Setup [file: *printer.def.json* & *extruder.def.json*]

The configuration files required for the slicing process were taken directly from the installation directory of Ultimaker Cura 5.8.0. They contain, in a nested structure, all input parameters necessary for passing information about the printer and extruder to the slicer. The original versions of the duplicates included in the *user_input* folder can be found in the Cura installation directory at the following paths:

1. **Printer:** C:\Program Files\UltiMaker Cura 5.8.0\share\cura\resources\definitions
2. **Extruder:** C:\Program Files\UltiMaker Cura 5.8.0\share\cura\resources\extruders

The described files consist of nested dictionaries that assign a value to each process parameter. The value specified in the "default_value" field is used as the input during the slicing process when invoked via the CLI. Listing 4.3.1 provides an example of how an input parameter is defined in the *extruder.def.json* file.

Listing 4.3.1 Section from *extruder.def.json*

```
"machine_nozzle_size":  
{  
    "default_value": 25,  
    "description": "The inner diameter of the nozzle. Chan  
        ge this setting when using a non-standard nozzle si  
        ze.",  
    "label": "Nozzle Diameter",  
    "maximum_value_warning": "100",  
    "minimum_value": "0.001",  
    "settable_per_extruder": true,  
    "settable_per_mesh": false,  
    "type": "float",  
    "unit": "mm"  
}
```

The *CuraEngine.exe*, which is used for the slicing process, retrieves only the most deeply nested parameters from the two *.def.json* files. Appendix C and D provide tabular overviews of all supported parameters along with their corresponding input types.

Unfortunately, due to the lack of output data in the generated G-code, it was not possible to retrieve all input parameters from the G-code file in post-processing. As a result, certain inputs are fundamentally excluded from use. These are explicitly marked in Appendix C and D. In addition, some inputs were deliberately fixed to avoid requiring duplicate user entries. Listing B.1 summarizes all such fixed parameters. Most of these variables are defined through user input in *setup.json* and must therefore be sourced from that file. For other fixed values, such as "ironing_enabled", the necessary information could not be retrieved from the G-code and was therefore defined apriori to ensure a smooth process and avoid errors.

When creating new printer profiles or editing existing parameters for the printer or extruder, it is particularly important to ensure that the syntax throughout the entire *def.json* file is correct and that the

value specified under "default_value" matches the data type defined in the "type" field, as these are used in the slicing process as input values. A full validation of all arguments has been deliberately omitted due to the complex structure of the files and the large number of potential error sources. As a result, incorrect, incomplete, or inconsistent entries, as well as missing separators within the Cura specific input files, may lead to faulty execution of the program.

Replacing the existing files in the `user_input` directory with user-specific variants is entirely feasible. In addition to replacement, multiple printer profiles may be stored either in the `user_input` folder or in any other user-defined directory. The only requirement is that the corresponding path to the `printer.def.json` file is correctly specified in `setup.json` under the entry "cura_printer_config_file_path" (see Figure 4.4). Furthermore, the associated extruder file `extruder.def.json` must reside in the same folder and must be correctly referenced in `printer.def.json` under "machine_extruder_trains". An extension to additional slicers would also be possible at this point by adding new packages to the `slicer` directory.

4.3.2 General Setup [file: `setup.json`]

All remaining inputs that are not directly required for slicer operation are defined in `setup.json`. Figure 4.4 provides an overview of the various input parameters located under the "settings" section, along with their respective parent sections.

The structure of the `setup.json` file closely mirrors that of `extruder.def.json` and `printer.def.json`. The "Directory" section contains entries for both input and output files and folders. It is important to ensure that the value assigned to "input_name" refers exclusively to an .stl file, as this is required to prevent errors in the output. The existence of the specified input file is automatically checked within the main function. Output folders and file paths defined in `setup.json` are created automatically by the program if they do not already exist in the referenced directory.

The largest portion of user input in `setup.json` is found under the "Robot" section. These parameters, along with the entries under "Pump", are discussed in more detail in Chapter 4.5.2 and Chapter 4.6.

The slicer used is defined via the "slicer_name" field within the "Slicer" section. This ensures the correct mapping of line types listed in Chapter 3.2.4. If not developed any further in the future, only the entry "Cura" produces a complete output.

The "Cura" section contains slicer-specific input. Among other things, it specifies the path to the `CuraEngine.exe` file included with Cura 5.8.0. Additionally, the "cura_arguments" field allows arguments from `printer.def.json` and `extruder.def.json` that are normally read by `CuraEngine.exe` to be overwritten. Listing 4.3.2 shows an example input for this parameter.

Listing 4.3.2 Example Input for "cura_arguments" from `setup.json`

```

    "cura_arguments": {
        "description": "Additional arguments to quickly change the
                      settings for CURA slicing. These overwrite the values
                      in the printer and extruder config data",
        "value": {
            "layer_height_0": "15",
            "layer_height": "15",
            "mesh_position_x": "0",
            "min_wall_line_width": "25",
            "retraction_enable": "true"
        },
        "type": "dict[str,str]"
    }
}

```

If one of the input parameters predefined in Listing B.1 is nevertheless specified by the user in `setup.json`, it will be automatically overwritten by the corresponding default value during program execution. In such cases, the program issues a warning message but continues the execution process without interruption. Listing 4.3.3 provides an example of such a warning for the parameter "retraction_enable".

Listing 4.3.3 Console Output from Setup Validation with Error Message

```
[INFO] Validating and updating user arguments...
[INFO] retraction_enable = False from setup.json is valid
[WARNING] Preset value applied for 'retraction_enable': True
[INFO] Starting to slice Test Object.stl
```

The field "cura_line_types_dict" is used to group the line type identifiers defined by Cura into a set of uniform categories. This categorization is intended to ensure that adapting the program for use with other slicers does not require fundamental changes to the code structure. A total of eight line type categories are distinguished, of which types 1 through 7 can be explicitly assigned by the user:

1. **surface**: Lines that are coplanar with the object, typically lying within the $X-Y$ plane.
2. **wall_outer**: The outermost, visible perimeter of the object.
3. **wall_inner**: All additional wall lines located within the object interior that are not visible in on the finished part.
4. **infill**: Material used to fill the space between inner and outer walls.
5. **bridge**: Lines that span unsupported regions and would hang freely in the air if support structure is not generated.
6. **curb**: A collective category for all bed-adhesion features, as shown in Figure 3.8.
7. **support**: Support material that is not part of the actual print object.
8. **unknown**: This internal category is not visible to the user. It serves as a fallback for any line types that cannot be matched to a predefined entry in "cura_line_types_dict" and is used as the default value during program execution.

The final section of input parameters is labeled "Rhino". All entries in this section directly affect the rendering and output of the .stl file as visualized tool paths in the generated .3dm file. Details on these inputs are provided in Chapter 4.9.

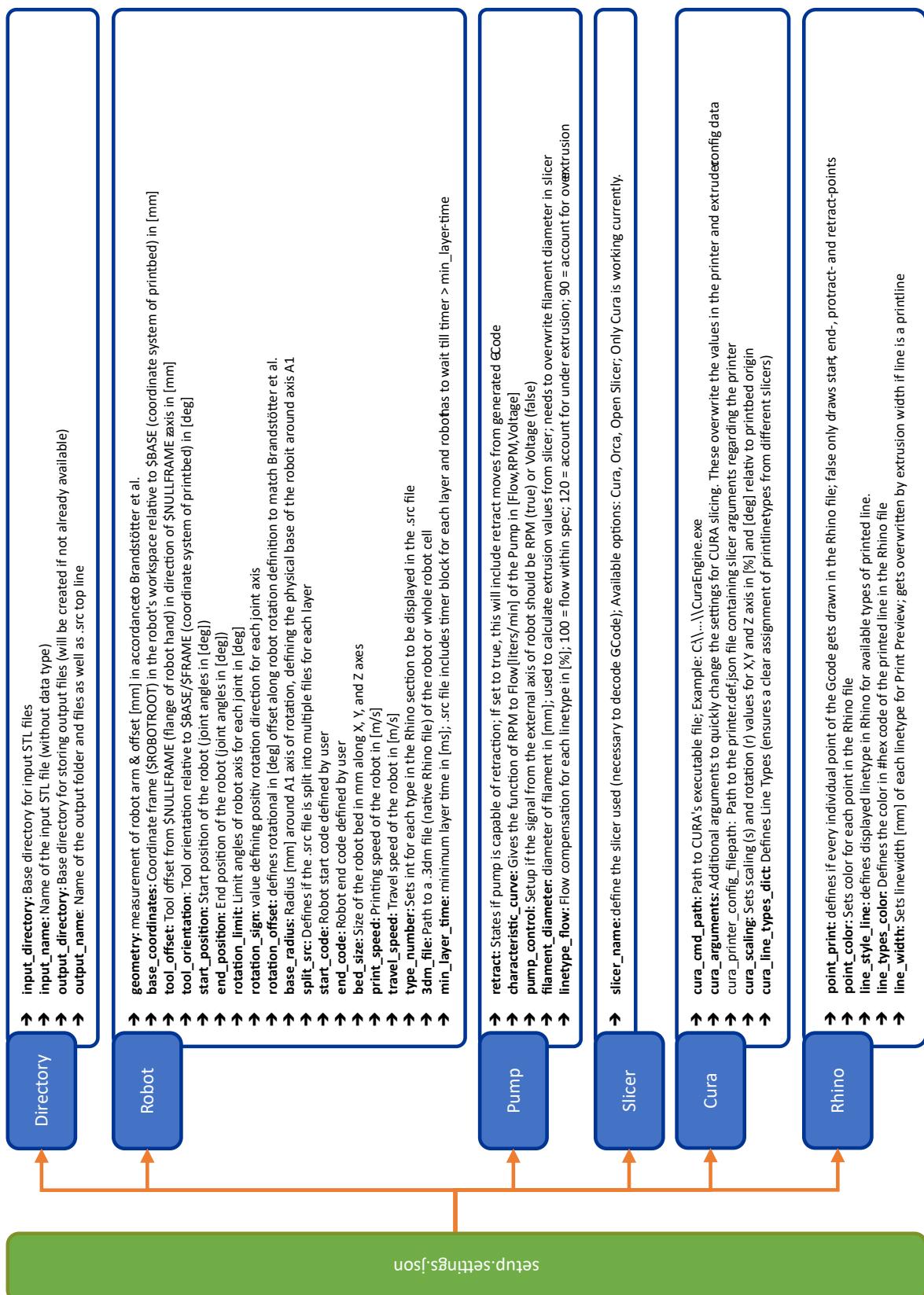


Figure 4.4 Overview of the Different Sections Included in `setup.json` With Descriptions of Each Input Parameter

4.3.3 Validation of User Input [module: `validate_value`]

To prevent potential errors in the `setup.json` file caused by incorrect user input, additional validation functions have been implemented within the `setup` package. As already established for `printer.def.json` and `extruder.def.json`, the data type for each user-defined parameter is explicitly specified within its respective entry. Table 4.2 provides an overview of all supported data types along with illustrative examples.

Table 4.2 Overview of Data Types Used in `setup.json`

Data Type	Description	Example Value
str	String	KUKA KR340 R 3300
int	Integer	10000
float	Floating Point Number	43.99
bool	Boolean	false
dict[str,str]	Dictionary [key = str, value = str]	{'layer_height_0': '15', 'layer_height': '15', 'infill_li...'} {'A1': -1, 'A2': 1, 'A3': 1, 'A4': -1, 'A5': 1, 'A6': -1}
list[str]	List [value = str]	[';Printer: ?id?', ';Declarations for RSI']
dict[str,list[int]]	Dictionary [key = str, value = list of int]	{'A1': [-185, 185], 'A2': [-130, 20], 'A3': [-100, 144], ...}
dict[str,list[str]]	Dictionary [key = str, value = list of str]	{'surface': ['SKIN'], 'wall_outer': ['WALL-OUTER'], ...}

If input data types inside `setup.json` are nested, simple constructs such as `dict[str, float]` are automatically recognized and recursively passed through the validation function until all input values have been fully checked to match the data type specified. For more complex, deeply nested data types, specific `if`-conditions have been implemented within the `validate_value` module to handle these cases in a targeted manner. When adding new input parameters in the future, this module must be extended accordingly.

If a user entry does not match the expected data type, the program issues a detailed error message pointing directly to the erroneous input location and terminates execution without producing any output. Once the error regarding data type has been corrected, the `main` function can be re-executed. An example of such an error message is shown in Listing 4.3.4.

Listing 4.3.4 Console Output from Setup Validation with Error Message

```
[ERROR] Invalid input in setup.json
[ERROR] Configuration error found
[WARNING] settings.Slicer.slicer_name: Invalid value '1' for type
          'str'
```

4.4 Parsing and Interpretation of G-Code [package: *gcode*]

`user_input` `.gcode_file` `gcode` `robot` `pump` `krl` `.src-file` `report` `.docx-file` `rhino` `.3dm-file`

Figure 4.5 Progress Bar for Chapter 4.4

If the G-code has been generated as described in Chapter 3 and stored in the directory specified by "output_directory", it must subsequently be imported into Python. For this purpose, it is first necessary to determine which parts of the code are of essential relevance. To avoid unintended movements caused by the initialization and reset commands shown in Chapter 3.2.3, the program deliberately omits the code blocks at the beginning and end of the G-code that describe the actual print object. The corresponding user inputs in Cura are part of the fixed user parameters shown in Listing B.1, ensuring that accidental activation is excluded. Furthermore, the relevant fields in the `.def.json` files (see "machine_start_gcode", "machine_end_gcode", "machine_extruder_start_gcode" as well as "machine_extruder_end_gcode") can be safely defined as empty strings.

The G-code itself is subsequently imported into Python as a list of strings via the module `get_gcode` and then parsed by the module `min_max_values` to determine the minimum and maximum coordinates within the list. As the full set of information on position is not always given from the start of the G-code, these serve as initial fallback values for the following module to ensure that complete coordinate information (*X*, *Y*, and *Z*) is available starting from line 1 of the G-code.

Using *regular expressions* (RegEx) and the corresponding Python library [102], the G-code is then analyzed. Listing 4.4.1 shows the relevant lines of code from the `simplify_gcode` module. Table 4.3 provides a detailed explanation of the associated search patterns.

Listing 4.4.1 RegEx Pattern Used in `simplify_gcode`

```

57     # Step 1: Define regex patterns
58     type_pattern = r"(?i)[;]?TYPE[:\s]*(.+)"
59     gcode_pattern = r"(G[01])?(?:\s*F[\d\.\.]+)?(?:\s*X\s*([-?\d\.\.]+))?
60        (?:\s*Y\s*([-?\d\.\.]+))?(?:\s*Z\s*([-?\d\.\.]+))?(?:\s*E\s*([-?\d\.\.]+))?"
61     feedrate_pattern = r"(G[01])\s*F[\d\.\.]+$"
62     g92_pattern = r"G92\s+E0" # Matches "G92 E0"
63
64     # Step 2: Determine extrusion mode (M82/M83)
65     absolute_mode = not any("M83" in line for line in gcode[:300])

```

Table 4.3 Explanation of Regular Expressions Used for Parsing G-Code

Regex Pattern	Explanation of Search Criteria
<code>type_pattern</code>	Finds a line containing the word TYPE, optionally preceded by \$ or ;, using [;\$]?. It is case-insensitive due to (?i) and allows colons or whitespace between TYPE and its value using [:\s]*. The value is captured in (.+).
<code>gcode_pattern</code>	Parses G-code motion lines using (G[01])? for optional G0/G1 command, followed by optional feedrate (?:\s*F[\d\.\.]+)?, and optional coordinates X, Y, Z, and extrusion E, each using non-capturing optional groups like (?:\s*X\s*([-?\d\.\.]+)).
<code>feedrate_pattern</code>	Matches G0/G1 lines that end in a feedrate using (G[01])\s*F[\d\.\.]+\$. The line must end (\$) after the feedrate expression, ensuring exclusivity of the match.
<code>g92_pattern</code>	Identifies an extrusion reset command G92 E0 using G92\s+E0. Allows one or more spaces between G92 and E0 via \s+.

As shown in the listing, filtering is based on four distinct character sequences. The first step involves identifying the line type as explained in Chapter 3.2.3. Since all analyzed slicers indicate the type with the prefix `TYPE:`, this string is used as the primary search criterion. The string that follows is stored temporarily under the variable `"type_"` as the current line type.

Next, motion commands are filtered using the pattern `"gcode_pattern"`. The relevant lines start with either G0 or G1 and represent the general commands that define the desired print path. The type of movement as discussed in Table 3.2 is automatically determined within the function and stored for further use under the variable `"move"`. The associated coordinate and extrusion values are grouped and captured using parentheses `("` and `)` in the regular expression and continuously updated in the variables `"current_x"`, `"current_y"`, and `"current_z"`. The value specified under E, which indicates the amount of extruded material, is also extracted and stored in the variable `"e_relative"`. When interpreting the extrusion value, it is important to consider whether the G-code uses absolute or relative extrusion. This information is derived from the commands M82 and M83, as explained in Table 3.2, and is always specified at the beginning of the G-code file (see Appendix A). The corresponding line is identified using the variable `"absolute_mode"` and is expected to be found within the first 300 lines of G-code (see Listing 4.4.1, Step 2). Based on this, the relative amount of extrusion per segment (`"E_Rel"`) can later be calculated precisely (see Chapter 4.6).

As a third criterion, lines that contain only a feedrate value without any accompanying position or extrusion data are identified. Although these lines are not yet processed further, they are already considered in anticipation of a future extension that may incorporate speed information from the slicer.

Finally, the code filters for G92 commands, which reset the extrusion value to zero. This information enables continuous calculation of the actual amount of material extruded during each linear movement. All other commands found in the G-code, such as temperature settings or fan controls (cf. Table 3.2) are ignored by the program, as they are not required for the subsequent steps. This is either due to the characteristics of the mineral-based material, which does not require as many commands as polymer material, or because the corresponding hardware components (like a part-cooling-fan) are not available or used in the given setup. This selective filtering streamlines data processing and improves the overall runtime of the program.

To ensure that the program remains largely independent of slicer-provided metadata in the G-code, the current layer height (`"Layer_Height"`) and the layer number itself (`"Layer"`) are determined automatically.

Listing 4.4.2 illustrates how the extracted information is stored as a *dictionary* for each individual line segment. Each new entry related to coordinates or extrusion in the G-Code results in the creation of a new dictionary. All dictionaries are then appended to a list for further processing. This list will be updated and extended during the execution of the program. Chapter 4.7 will discuss what the full set of information looks like before the file that numerically controls the robot is finally generated and exported. Finally, the maximum layer value is retrieved from the last dictionary in the list and stored in a dedicated variable within the `main` function.

Listing 4.4.2 Dictionary Generated for Each New Information Given on Coordinates, Line Type or Extrusion

```

147     gcode_entry = {
148         "Move": move,
149         "X": current_x,
150         "Y": current_y,
151         "Z": current_z,
152         "E_Rel": e_relative if move == "G1" else 0,
153         "Layer": current_layer,
154         "Type": type_,
155         "Layer_Height": layer_height,
156     }

```

4.5 Robot Integration and Kinematic Modeling [package: *robot*]

This chapter addresses the integration of the robot into the digital process chain, as well as the underlying kinematic modeling for a specific class of 6-DOF serial robots. The objective is to implement a class within the program that enables verification of whether all positions contained in the G-code can be reached by the robot defined in *setup.json*.



Figure 4.6 Progress Bar for Chapter 4.5

For this purpose, the underlying coordinate systems are described first, followed by the mathematical foundations of the kinematic modeling. Subsequently, the implementation of the kinematic calculations itself is explained, which forms the basis for the physical execution of the robot's movements.

4.5.1 Coordinate Frames

The robot model used at the TUM is a serial manipulator with a spherical wrist. Since the axes A4 to A6 intersect at a single point, the combined kinematics of these three joints resemble that of a ball joint and are comparable to the human wrist. Figure 4.7 illustrates how the arrangement of axes in this wrist configuration differs from other robotic arm designs. The point of intersection of these three axes plays a crucial role in later simulation steps and is referred to as C_{wrist} in the following.

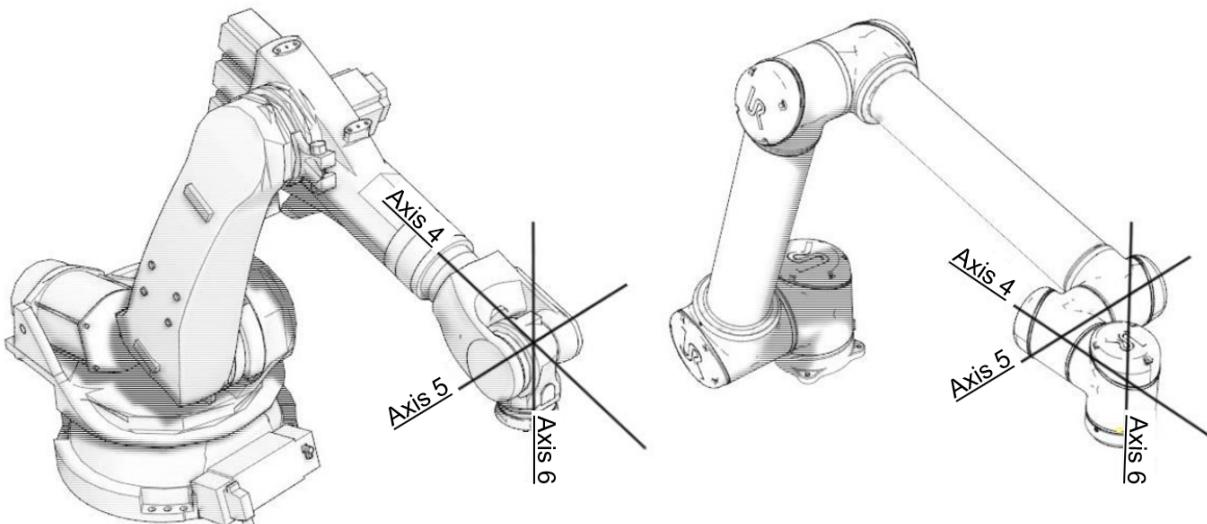


Figure 4.7 Comparison of Spherical and Non-Spherical Wrists for Serial Robots [27]

The coordinate frames of the remaining degrees of freedom (axes A1 to A3) are arranged orthogonally to one another. As a result, they can be transformed into adjacent coordinate systems by rotations of $\pi/2$ about one or more axes. Figure 4.8 illustrates the coordinate systems and axis orientations along the kinematic chain, as positively defined by the manufacturer for the KUKA KR 340 R3300. The Z -axis of each joint always represents the rotational axis (A_i) for the corresponding movement, with the rotation angle θ_i defined as mathematically positive.

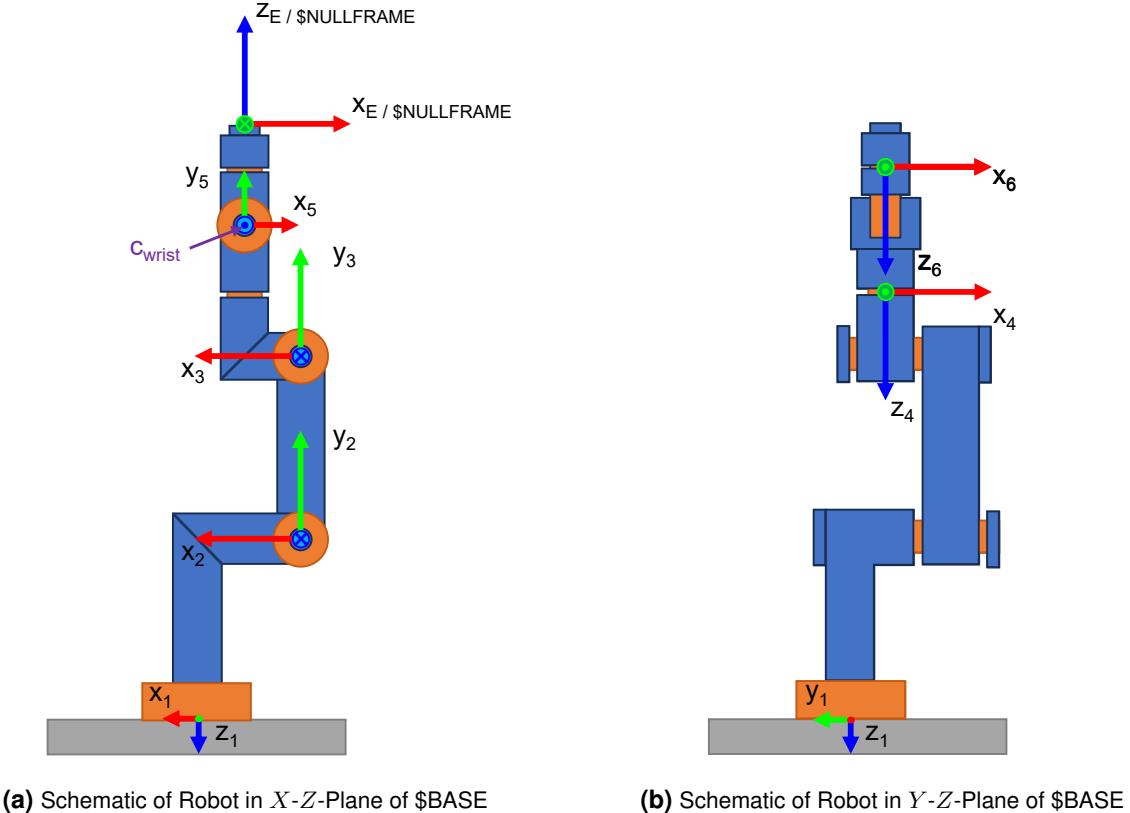


Figure 4.8 Coordinate Systems for Robot Joints of Kuka KR 340 R3300, Adapted from [16].

In addition to the robot-internal coordinate systems shown in Figure 4.8, external objects must also be assigned reference frames to be identified correctly in the following sections. This ensures a clear localization of both the robot relative to the print bed and the printed object relative to the robot. Following the naming conventions used in [41], the following coordinate frames are defined:

- **\$BASE:** Origin of the print bed. Both the coordinates of the printed object and the robot's base position and orientation are referenced to this coordinate system.
- **\$ROBOTROOT:** Coordinate system coinciding with the origin of axis A1 (cf. Figure 4.8a, 4.8b, axes x_1 , y_1 , and z_1). This coordinate system defines the robot's reference point relative to the print bed in the X - Y plane. Unlike the orientation shown in Figure 4.8, its Z -axis is defined to point upward.
- **\$NULLFRAME:** Coordinate system located at the origin of axis A6; describes the orientation of the flange on the robot arm.
- **\$TOOL:** Local coordinate system of the tool mounted on the flange.
- **\$FRAME:** Temporary coordinate system of a point on the print path. Its orientation corresponds to that of \$BASE, with the respective Z -axes aligned in parallel. When the robot moves to a coordinate, the origins of \$NULLFRAME and \$TOOL coincide. Moreover, the Z -axis of \$TOOL is collinear with that of the currently addressed point, thereby excluding non-planar tool motions relative to the print bed.

Figure 4.9 illustrates the various coordinate systems in their overall context. The coordinate system of \$ROBOTROOT is embedded within the robot base and thus not visible. Its orientation is aligned with that of the \$BASE coordinate system. For clarity sake, the \$TOOL reference frame has been omitted from the visualization; depending on the values specified under "tool_offset", it could, for instance, be located in the extension of \$NULLFRAME. Due to the chosen implementation, the \$TOOL coordinate system must have the same orientation as \$NULLFRAME.

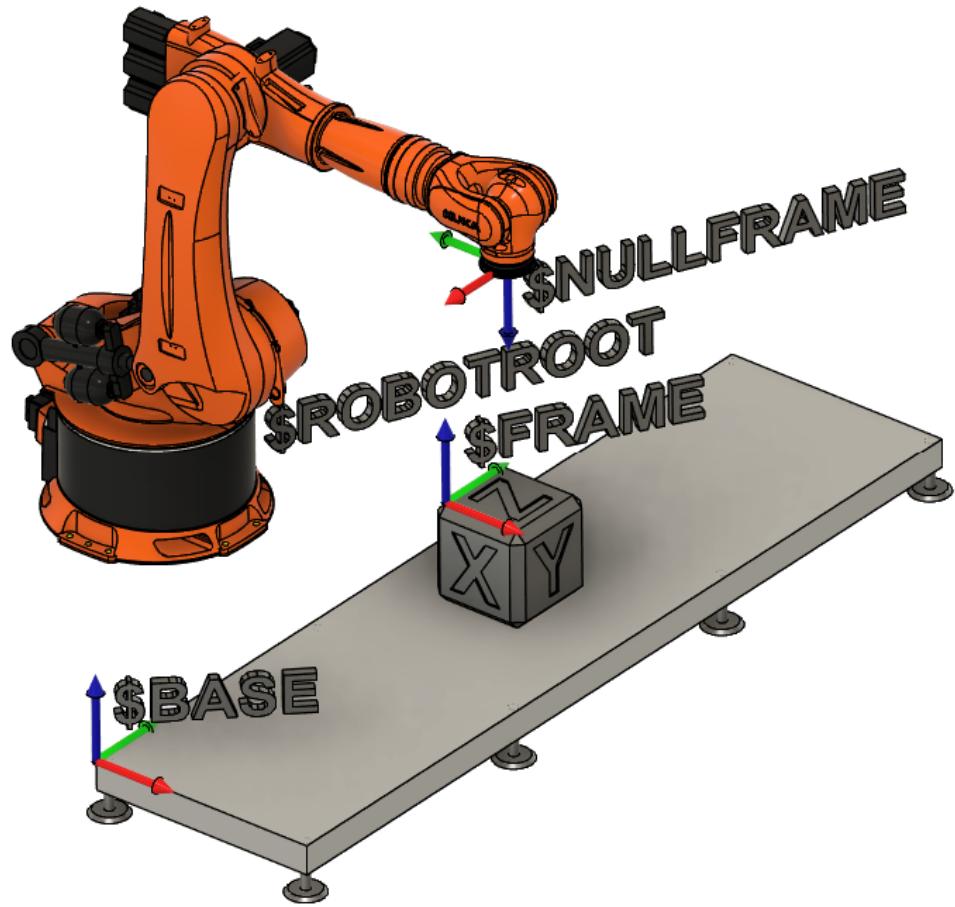


Figure 4.9 Overview of Relevant Coordinate Systems for Robot, Object and Bed Modeled in Fusion 360 [11]

All relevant local coordinate systems as well as the overarching reference frames have thus been described and clearly defined. The following chapter focuses on the required user inputs in *setup.json* that enable the integration of any robot with an identical kinematic structure into the program.

4.5.2 Robot Implementation Using `setup.json` [folder: `user_input`]

The definition of the robot used in the program has been implemented in a fully parametric manner, allowing the integration of various robot types with an identical geometric base structure. This ensures a modular program structure and allows the system to be used flexibly for future configurations or entirely different setups. The section "Robot" in `setup.json` is largely dedicated to this input.

To begin with, the user must specify the position of the robot relative to `$BASE`. For this purpose, the offset from `$BASE` to `$ROBOTROOT` is defined under "base_coordinates" in `setup.json`, as well as the rotations around the X , Y , and Z -axis using "C", "B", and "A", respectively, in that order with reference to `$BASE`. Figure 4.10 schematically illustrates the corresponding dimensions.

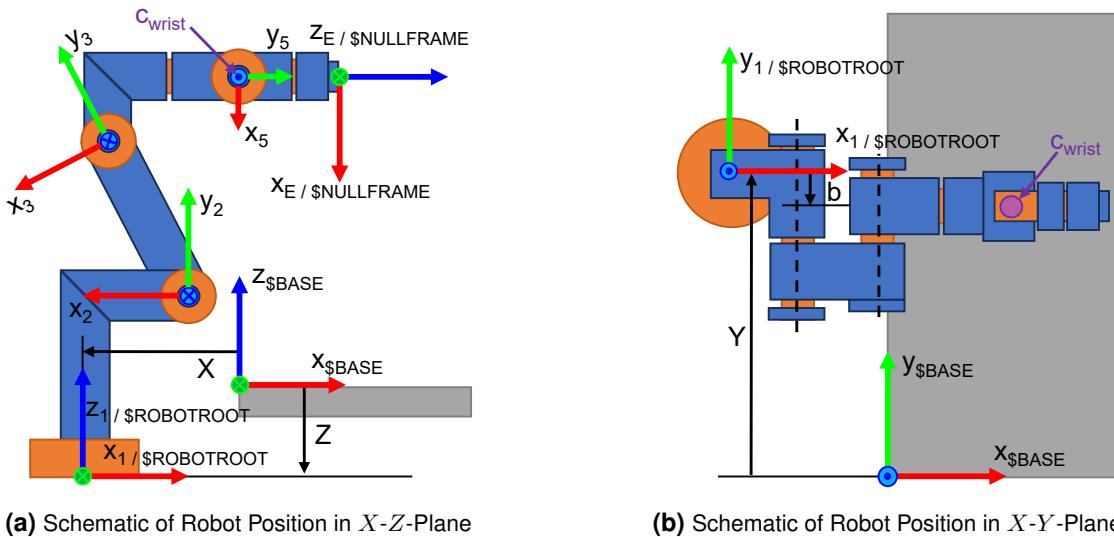
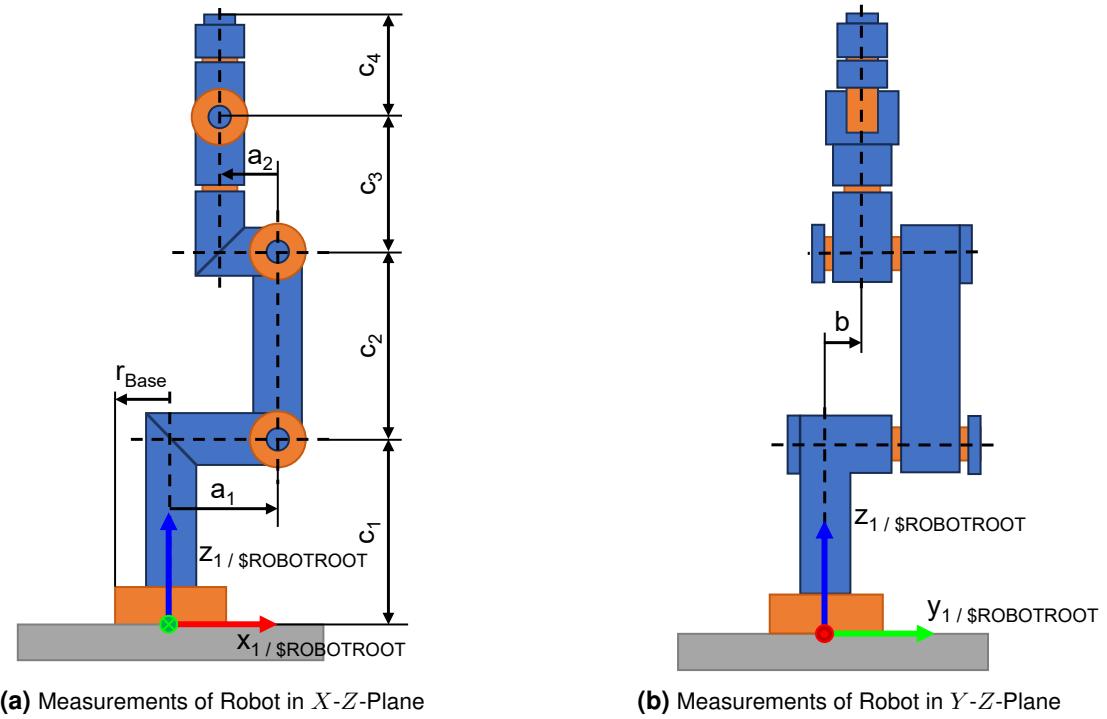


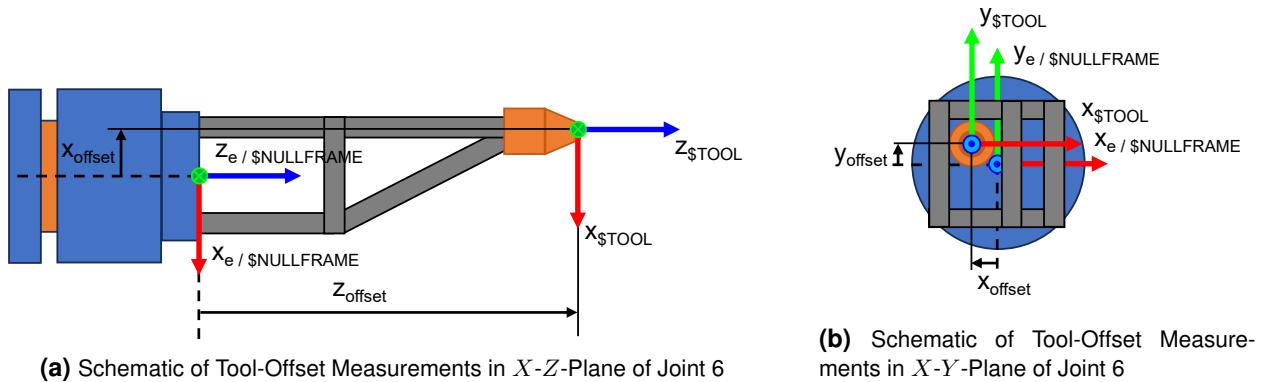
Figure 4.10 Overview of "base_coordinates" Measurements for `setup.json`

Furthermore, the user must configure the robot's geometry according to the specifications provided by the manufacturer. The corresponding parameters are stored in the "geometry" dictionary within `setup.json`. Figure 4.11 illustrates all relevant dimensions in a schematic context.

In addition, the parameter "base_radius" defines the radius of the robot base around the Z -axis of `$ROBOTROOT`. This value is used in later processing steps to detect potential collisions between the robot arm and its own base. Unlike the illustration, the computational assumption takes a conservative approach by extending the base radius along the entire length of " c_1 ".

**Figure 4.11** Essential Measurements for Robotic Arm, Adapted from [16].

Besides the geometric properties of the robot arm, the dimensions of the tool head mounted on the robot's flange must also be specified by the user. For this purpose, the offset of the tool relative to the $\$TOOL$ coordinate system is defined in millimeters and can be customized using "tool_offset" inside `setup.json`. Figure 4.12 illustrates the relevant measurements in both side and top view.

**Figure 4.12** Overview of "tool_offset" Measurements for `setup.json`

Since the orientation of the tool relative to the print bed or the printed object must also be specified in addition to its dimensions, this is defined using the parameter "tool_orientation". As previously explained, the orientation is given relative to $\$BASE$ as the reference coordinate system and remains constant throughout the printing process. The three parameters "A", "B", and "C" represent successive rotations about the Z -, Y -, and X -axes of $\$BASE$, in the specified order.

The rotation is defined as *intrinsic*, meaning that the underlying coordinate system rotates along with each applied transformation. In contrast, an *extrinsic* rotation assumes a fixed coordinate system, where each rotation is applied with respect to the original global axes. Chapter 4.5.3 provides a more detailed explanation of the corresponding mathematical background.

Since the orientation of the coordinate systems of individual robot axes can vary depending on the manufacturers definition, the program must also allow for flexible input in this regard. To still ensure consistent internal processing, the direction of the Z -axis for each joint must be specified using the parameter "rotation_sign". This parameter refers to the reference coordinate systems of each joint as illustrated in Figure 4.13.

The input is provided as a dictionary, in which each axis is assigned a *boolean*-value. A value of "true" indicates that the direction of the Z -axis for each joint as defined by the manufacturer corresponds to the reference system shown (cf. Figure 4.13). If the value is "false", the Z -axis is inverted, while the *right-handed* structure of the coordinate system is preserved. Regarding the configuration defined by KUKA for the KR 340 R3300 (see Figure 4.8), this results in the dictionary for "rotation_sign" shown in Listing 4.5.1.

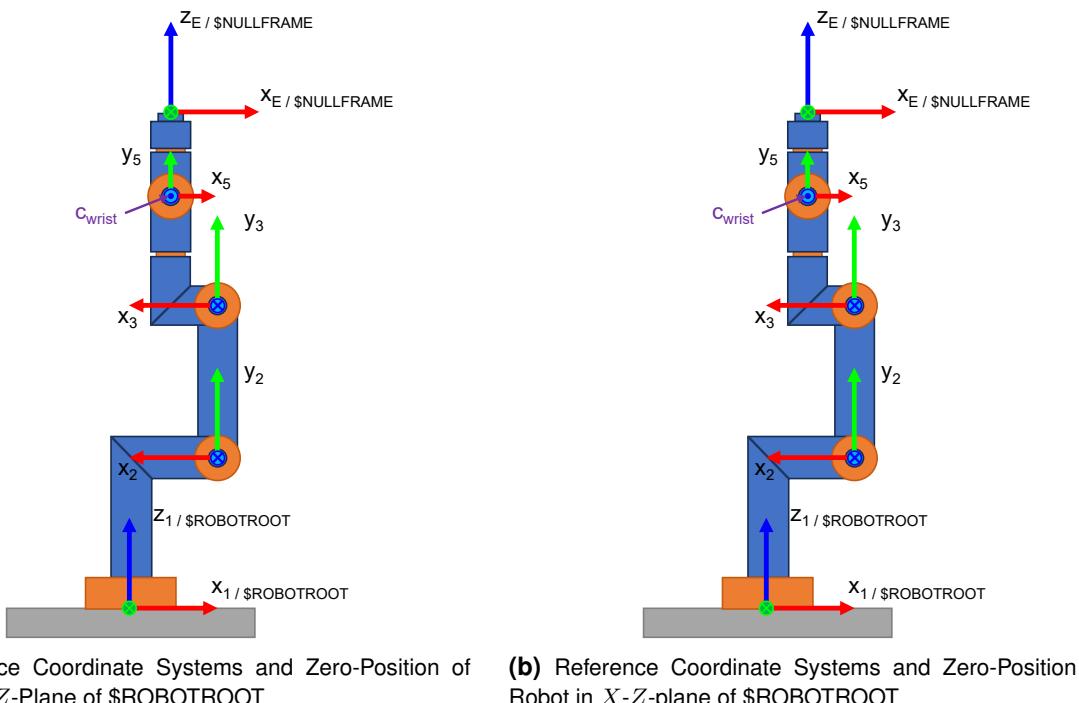


Figure 4.13 Orientation of Coordinate Systems for Robotic Arm, according to [16].

To ensure that the robot does not collide with itself during motion execution, manufacturers typically specify limits for the rotational angles of each joint in both the positive and negative directions relative to the joint's zero position (i.e., joint rotation = 0°). Since the definition of this zero position can vary between manufacturers, it is necessary to again establish a unified convention within the program. The orientation shown in Figure 4.13 serves as the reference configuration referred to as the zero position, which is consistently used throughout the implementation.

To configure a specific robot, the deviation between the manufacturer's defined zero position and the referenced orientation must be specified. This is done with regard to the coordinate frames given in Figure 2.19b using the parameter "rotation_offset" in the *setup.json* file. Following that, the allowable joint angle ranges relative to the robot-specific coordinate systems can be defined using the parameter "rotation_limit". Listing 4.5.1 provides an example of how these parameters are set for the KUKA KR 340 R3300 robot used in this study.

Listing 4.5.1 Example Input to Geometrically Define KUKA KR 340 R3300 Taken from *setup.json*

```

    "id": {
        "description": "name of robot",
        "value": "KUKA KR340 R3330",
        "type": "str"
    },
    "geometry": {
        "description": "measurement of robot arm & offset [mm] in accordance to Brandstatter et. al [https://de.mathworks.com/matlabcentral/fileexchange/48468-inverse-kinematics-of-serial-robots-with-wrist?searchTerm=prof_contrib_link]",
        "value": {"a1": 500, "a2": 55, "b": 0, "c1": 1045, "c2": 1300, "c3": 1525, "c4": 290},
        "type": "dict[str,float]"
    },
    "base_radius": {
        "description": "Radius [mm] around A1 axis of rotation; ensures the robot head never intersects with its own base (around joint axis A1)",
        "value": 400,
        "type": "float"
    },
    "tool_offset": {
        "description": "Tool offset from $NULLFRAME (flange of robot hand) in direction of $NULLFRAME in [mm]; This leads to T($NULLFRAME,$TCP) as no rotation is involved",
        "value": {"X": -10.99, "Y": -0.86, "Z": 917.61},
        "type": "dict[str,float]"
    },
    "tool_orientation": {
        "description": "Tool orientation relative to $BASE/$FRAME (coordinate system of printbed) in [deg]; This leads to R($BASE/$FRAME,$TCP); A(angle related to Z), B(angle related to Y), C(angle related to X)",
        "value": {"A": 0, "B": 0, "C": 180},
        "type": "dict[str,float]"
    },
    "rotation_sign": {
        "description": "value defining positive rotation direction of axis; true == along z-axis definition, false == against z-axis definition",
        "value": {"A1": false, "A2": true, "A3": true, "A4": false, "A5": true, "A6": false},
        "type": "dict[str,bool]"
    },
    "rotation_offset": {
        "description": "defines rotational in [deg] offset along robot rotation definition to match 'Brandstatter et al.'",
        "value": {"A1": 0, "A2": -90, "A3": 0, "A4": 0, "A5": 0, "A6": 0},
        "type": "dict[str,float]"
    },
    "rotation_limit": {
        "description": "Limit angles of robot axis for each joint in [deg]; valid rotation is between those angles",
        "value": {"A1": [-185, 185], "A2": [-130, 20], "A3": [-100, 144], "A4": [-350, 350], "A5": [-120, 120], "A6": [-350, 350]},
        "type": "dict[str,list[int]]"
    }
}

```

4.5.3 Mathematical Background [module: *mathematical_operators*]

The following section outlines the mathematical principles on which the computational methods for robot kinematics derived by [16] are based upon. Central to this is the definition of a *homogeneous transformation matrix*, which serves as the foundation for describing positions and orientations in general robotic configurations.

The purpose of a homogeneous transformation matrix is to transform a vector defined in a local coordinate system into another, typically global, coordinate system. After the transformation is applied, the vector is no longer expressed relative to its original base frame, but relative to the target frame. This allows for the consistent representation of kinematic chains across multiple coordinate systems.

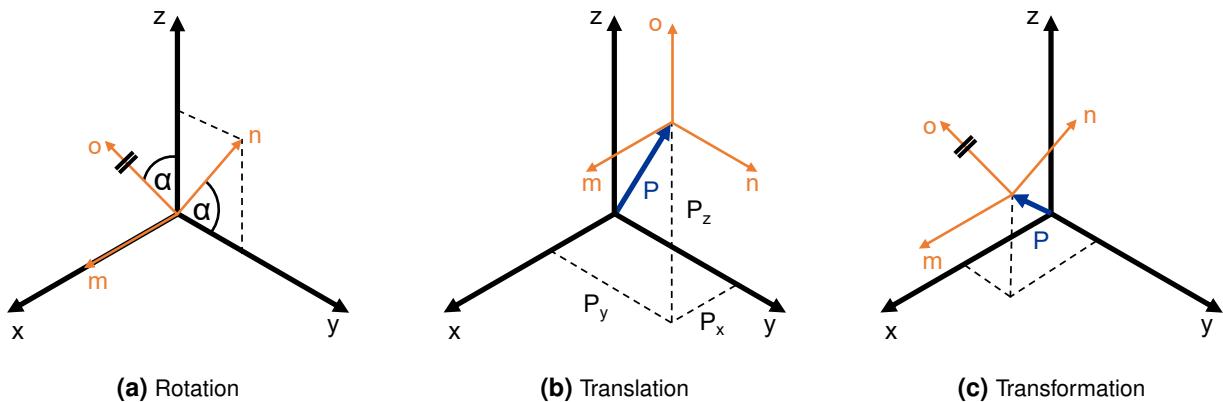


Figure 4.14 Components of Spacial transformation

A homogeneous transformation matrix consists of two independently defined components (cf. Figure 4.14): a rotation and a translation. The translation describes the displacement of the coordinate origin (cf. Figure 4.14b), while the rotation (cf. Figure 4.14a) alters the orientation of the coordinate axes. In this process, for both rotation as well as translation the orthogonality of the coordinate system is preserved. Each component will be discussed in more detail in the following sections.

Rotation

The rotation consists of three successive partial rotations, each performed about one of the three principal axes of the coordinate system (X , Y , Z). As explained in Chapter 4.5.2, the rotation is carried out *intrinsically* in the present case. Figure 4.15 shows the corresponding rotation matrices for the individual spatial directions. A derivation of these matrices can be found in [35].

$$R_X(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \quad R_Y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad R_Z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(a) Rotation About X -Axis

(b) Rotation About Y -Axis

(c) Rotation About Z -Axis

Figure 4.15 Individual Rotation Matrices

Orientation is often described using the so-called *Euler*-angles, which are typically composed of a sequence of three elemental rotations. These rotations are taken from the principal axes X (*roll*), Y (*pitch*), and Z (*yaw*), and follow specific axis combinations of two of the three axes, such as XYX , XZX , or ZYZ , depending on the chosen convention. The subsequent use of all three rotational axes in this context is known as the *Bryan–Tait* definition [90, 24]. This definition is characterized by

the sequential application of the individual rotations in a specified order, mathematically expressed through the successive multiplication of the corresponding rotation matrices. This convention is used for movement calculation by both KUKA as well as the program itself.

In the case of an intrinsic rotation, the multiplication is carried out from right to left, where the leftmost matrix represents the last rotation to be applied, and the rightmost matrix corresponds to the first. For an XYZ rotation sequence, this means:

$$\mathbf{R}_{(\alpha,\beta,\gamma)}^{\text{intr.}} = R_Z(\gamma) \cdot R_Y(\beta) \cdot R_X(\alpha) \quad (4.1)$$

Rotation about the X -axis is therefore applied first, followed by rotation about the Y -axis and finally about the Z -axis. Each subsequent rotation is performed relative to the already transformed coordinate system.

A key characteristic of this rotational representation is that the order of rotations has a significant effect on the result. Since rotations in three-dimensional space are generally *non-commutative*, the following relation holds, for example:

$$R_X(\alpha) \cdot R_Y(\beta) \neq R_Y(\beta) \cdot R_X(\alpha) \quad (4.2)$$

The orientation of a body after a rotational operation depends significantly on the order in which the individual rotations about the principal axes are executed. Since rotations in three-dimensional space are non-commutative, the same set of angles can lead to different final orientations depending on the sequence in which they are applied. Therefore, the rotation order — for example, in the form of a ZYX or XYZ convention — must always be explicitly specified. Equation 4.3 shows the resulting overall rotation matrix for an intrinsic rotation in XYZ order:

$$\mathbf{R}_{(\alpha,\beta,\gamma)}^{\text{intr.}} = \begin{bmatrix} \cos \beta \cos \gamma & \cos \gamma \sin \alpha \sin \beta - \cos \alpha \sin \gamma & \cos \alpha \cos \gamma \sin \beta + \sin \alpha \sin \gamma \\ \cos \beta \sin \gamma & \cos \alpha \cos \gamma + \sin \alpha \sin \beta \sin \gamma & -\cos \gamma \sin \alpha + \cos \alpha \sin \beta \sin \gamma \\ -\sin \beta & \cos \beta \sin \alpha & \cos \alpha \cos \beta \end{bmatrix} \quad (4.3)$$

In the following, this notation for intrinsic rotations is used in a compact form. Here, A denotes the reference coordinate system in which the angles are defined, and B represents the target coordinate system:

$$\mathbf{R}_{A \rightarrow B} = \mathbf{R}_{(\alpha,\beta,\gamma)}^{\text{intr.}} = R_Z(\gamma) \cdot R_Y(\beta) \cdot R_X(\alpha) = \mathbf{R}_{XYZ} \left(\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \right) \quad (4.4)$$

To apply a rotation to a given vector, it must be multiplied point-wise (i.e., matrix-based) from the left by the previously computed rotation matrix. Equation 4.5 illustrates the structure of this operation using the system shown in Figure 4.14a as an example:

$$\begin{bmatrix} m \\ n \\ o \end{bmatrix} = \underbrace{\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}}_{\text{Rotationmatrix } R} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (4.5)$$

In addition to the forward computation of the rotation matrix from given Euler-angles, it is also possible to compute the angle values from an existing rotation matrix. For this purpose, special variants of inverse trigonometric functions are used. Therefor most notably *atan2* is used, which determines both the angle magnitude and the correct quadrant unambiguously.

The following representation shows the corresponding formulas for calculating the two possible sets of solutions for the angles γ , β , and α , based on an XYZ rotation sequence. A detailed derivation can be found in [83].

$$\gamma_1 = -\arcsin(R_{13}) \quad (4.6)$$

$$\gamma_2 = \pi - \gamma_1 \quad (4.7)$$

$$\beta_1 = \text{atan2}\left(\frac{R_{12}}{\cos \gamma_1}, \frac{R_{11}}{\cos \gamma_1}\right) \quad (4.8)$$

$$\beta_2 = \text{atan2}\left(\frac{R_{12}}{\cos \gamma_2}, \frac{R_{11}}{\cos \gamma_2}\right) \quad (4.9)$$

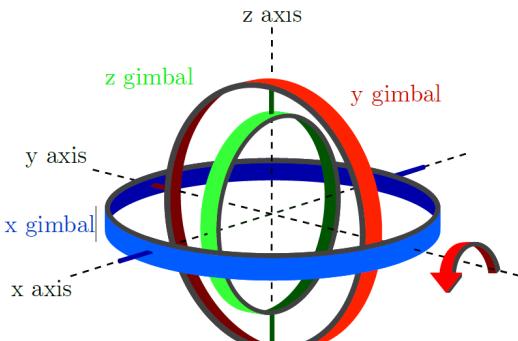
$$\alpha_1 = \text{atan2}\left(\frac{R_{23}}{\cos \gamma_1}, \frac{R_{33}}{\cos \gamma_1}\right) \quad (4.10)$$

$$\alpha_2 = \text{atan2}\left(\frac{R_{23}}{\cos \gamma_2}, \frac{R_{33}}{\cos \gamma_2}\right) \quad (4.11)$$

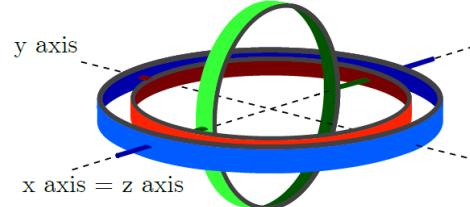
Since the sine function \arcsin is not injective in the interval $[0; \pi]$, which means that multiple solutions can exist for a given value, more than one angle may correspond to the same function value. Consequently, each of the three Euler-angles has two possible solutions that differ by π (or 180°), as illustrated in Equation (4.6) and Equation (4.7).

In the implementation of the `mathematical_operators`-module, only the first solution $(\alpha_1, \beta_1, \gamma_1)$ is computed. The second solution, however, is already incorporated in the kinematic calculations (see [16]) and can therefore be disregarded at this point.

A critical limitation of the Euler-angle and Bryan-Tait representation is the potential occurrence of a singularity for certain combinations of angle parameters. Specifically, when the pitch angle reaches $\pm 90^\circ$ (i.e., $\beta = \pm \frac{\pi}{2}$), two of the three rotation axes align, resulting in the loss of one degree of freedom. This phenomenon is known as *Gimbal Lock* and prevents certain orientations from being uniquely defined. In such cases, rotations about the affected axes can cancel each other out, meaning that not all three degrees of freedom can be independently controlled. Figure 4.16 illustrates the loss of a degree of freedom due to overlapping rotation axes in a gimbal system.



(a) Gimbal System in Base Configuration



(b) Gimbal System with Gimbal-Lock between *X* and *Z*

Figure 4.16 Gimbal and Gimbal-Locked System, Taken from [110]

To counteract this phenomenon during computation of individual angles from a given rotation, it is necessary to explicitly handle this special case. An unambiguous solution is typically achieved by fixing one of the angles to a defined value (e.g., $\alpha = 0$), allowing the remaining two angles to be uniquely determined. The following equations show the solution for special case depicted in Figure 4.16 with $\beta = +\frac{\pi}{2}$.

$$\alpha = 0 \quad (4.12)$$

$$\beta = \frac{\pi}{2} \quad (4.13)$$

$$\gamma = \text{atan2}(R_{12}, R_{13}) \quad (4.14)$$

Alternatively, the issue of *Gimbal Lock* could be completely avoided by using the so-called *quaternion* representation. A detailed explanation of this approach can be found in [35]. However, since this issue occurs only rarely within the robots kinematics and the solution presented in Equations 4.12 – 4.14 is sufficient for the most relevant cases, the use of the more complex quaternion method is deliberately omitted in this thesis.

Translation

Translation, as illustrated in Figure 4.14b, refers to a pure shift of the origins position for the coordinate system in space. The input quantity required for this operation is the displacement vector \mathbf{P} , which specifies both the direction and the magnitude of the translation. It consists of the three displacement components in the X , Y , and Z directions and can be represented as shown in Equation 4.15:

$$\mathbf{P} = \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} \quad (4.15)$$

To apply a translation to an existing vector \mathbf{L} , the corresponding displacement vector \mathbf{P} must be added to it. Equation 4.16 schematically illustrates this operation:

$$\mathbf{L}_{\text{new}} = \mathbf{L} + \mathbf{P} = \begin{bmatrix} L_x \\ L_y \\ L_z \end{bmatrix} + \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} L_{x,\text{new}} \\ L_{y,\text{new}} \\ L_{z,\text{new}} \end{bmatrix} \quad (4.16)$$

Transformation

From the previously explained independent components of rotation (3×3 matrix) and translation (3×1 vector) between two different coordinate systems A and B a combined transformation matrix can be formulated based on the structure of a matrix-vector product, where A represents the base configuration and B the transformed system. However, as shown in Equation 4.17, a direct multiplication with a 3×1 vector is not defined:

$$\underbrace{\begin{bmatrix} R_{3 \times 3} & t_{3 \times 1} \end{bmatrix}}_{3 \times 4-\text{Matrix}} \cdot \underbrace{\begin{bmatrix} \mathbf{v}_{3 \times 1} \end{bmatrix}}_{3 \times 1-\text{Vector}} = \text{not defined} \quad (4.17)$$

To ensure the compatibility of the multiplication, both the transformation matrix $T_{B \rightarrow A}$ and the vector are converted into the so-called *homogeneous coordinate space*. For this purpose, the 3×4 matrix is extended by a fourth row $[0 \ 0 \ 0 \ 1]$, and the 3×1 vector is extended by an additional entry with the value 1, resulting in a 4×1 vector. This ensures that the translation component is preserved correctly during the multiplication, while the rotational part remains unchanged. Equation 4.18 shows the structure of the complete multiplication:

$$\underbrace{\begin{bmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}_{B \rightarrow A}} \cdot \underbrace{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}_{\mathbf{L}_A} = \underbrace{\begin{bmatrix} x_{\text{new}} \\ y_{\text{new}} \\ z_{\text{new}} \\ 1 \end{bmatrix}}_{\mathbf{L}_B} \quad (4.18)$$

To transform from system B back to system A , the transformation matrix must be inverted [58]. Since all coordinate systems used are orthonormal, the rotation matrix is orthogonal. In this case, the inverse equals the transpose [58]:

$$\mathbf{T}^{-1} = \mathbf{T}^T \quad (4.19)$$

Transformations across multiple systems are also possible. In this case, the individual matrices can be multiplied. Each matrix links two adjacent coordinate systems. The following relation applies:

$$\mathbf{T}_{B \rightarrow A} \cdot \mathbf{T}_{A \rightarrow C} = \mathbf{T}_{B \rightarrow C} \quad (4.20)$$

This means that a transformation from system C through A into system B can be described by chaining the two individual transformations. The multiplication is always performed from right to left, in accordance with the order of reference systems. This property underlies hierarchical kinematic chains and is applied in the following sections.

4.5.4 Kinematic Modeling [module: *kinematics*]

As described in Chapter 2.6.1, the robot used in this thesis features a kinematic structure with an orthogonal-parallel base (axes A1 – A3) and a spherical wrist (axes A4 – A6). The *kinematics* module is therefore tailored specifically to this geometry. This so-called OPW architecture (**O**rtho-**P**arallel with **s**pherical **W**rist) is characteristic of many industrial 6-DOF serial manipulators.

The basis for the calculations is an analytical solution to the kinematic problem for this robot configuration, as introduced by Brandstötter et al. [16]. Unlike the classical *Denavit–Hartenberg* parameterization [23], which is applicable to arbitrary joint configurations, this approach relies on a compact set of parameters that can be derived directly from the robot's mechanical dimensions. This allows for a more straightforward and robust implementation in the case of these specific robot types.

In general, the kinematic analysis of robotic systems distinguishes between two approaches depending on the given parameters.

1. **Forward Kinematics:** All joint angles (θ_1 – θ_6) of the robot are known. The task is to determine the resulting position and orientation of the flange (i.e., \$NULLFRAME) or the tool center point (TCP) with respect to the reference frame \$ROBOTROOT. For each complete set of joint angles within the joint limits, there exists at most one valid solution.
2. **Inverse Kinematics:** The position and orientation of the TCP in space are given. The goal is to determine the joint angles required to reach this position. Depending on the pose and due to the multiple degrees of freedom of the robot, there may be one, multiple, or no valid solutions for the specified TCP or end-effector configuration.

For both problem types, the analytical method presented in [16] provides closed-form solutions, which are discussed in more detail in the following sections. An overview of the structure of the implemented module is shown in Figure 4.17. The Figure illustrates the contents of the class used to model the robot and outlines the required input and output values

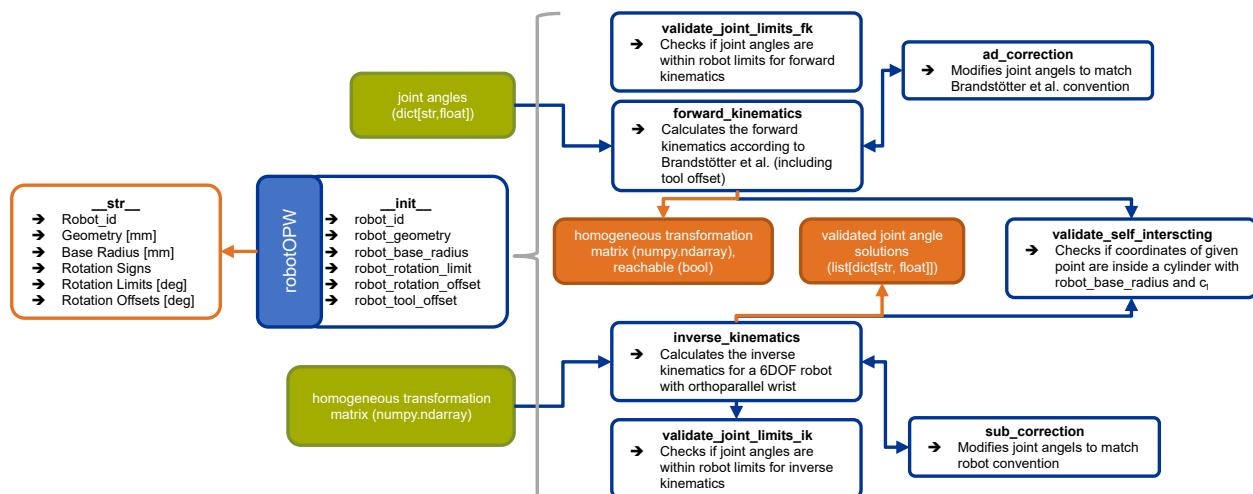


Figure 4.17 Overview of Class *robotOPW* from Module *kinematics* inside Package *robot*

Forward Kinematics

To compute the forward kinematics, the joint angles θ_1 to θ_6 are provided as input parameters. The function first verifies whether these values lie within the joint angle limits defined under "rotation_limit". If this condition is not met, the computation is terminated at this point, and an empty array is returned. Otherwise, the angles are transformed to the universal convention as depicted in Figure 4.13 and a complete homogeneous transformation matrix is calculated based on the equations presented in [16], describing the position and orientation of the robot flange relative to the base coordinate system \$ROBOTROOT (or coordinate frame 1). A flowchart illustrating the main components of this function is shown in Figure 4.19.

The mathematical expressions required for the calculation are listed in Equations 4.21 to 4.34. These are derived in particular from the geometric relationships depicted in Figures 4.18a and 4.18b.

In the first step, the position-related solution is determined, which uniquely defines the location of the afore mentioned wrist center (C_{wrist}) in space. Once this has been established, the procedure of the program deviates from the method in [16] by introducing a self-collision check with the robot base. If the point C_{wrist} lies within a cylinder defined by the radius specified under "base_radius" and extending along the dimension " c_1 " of the robot, the computation is likewise terminated.

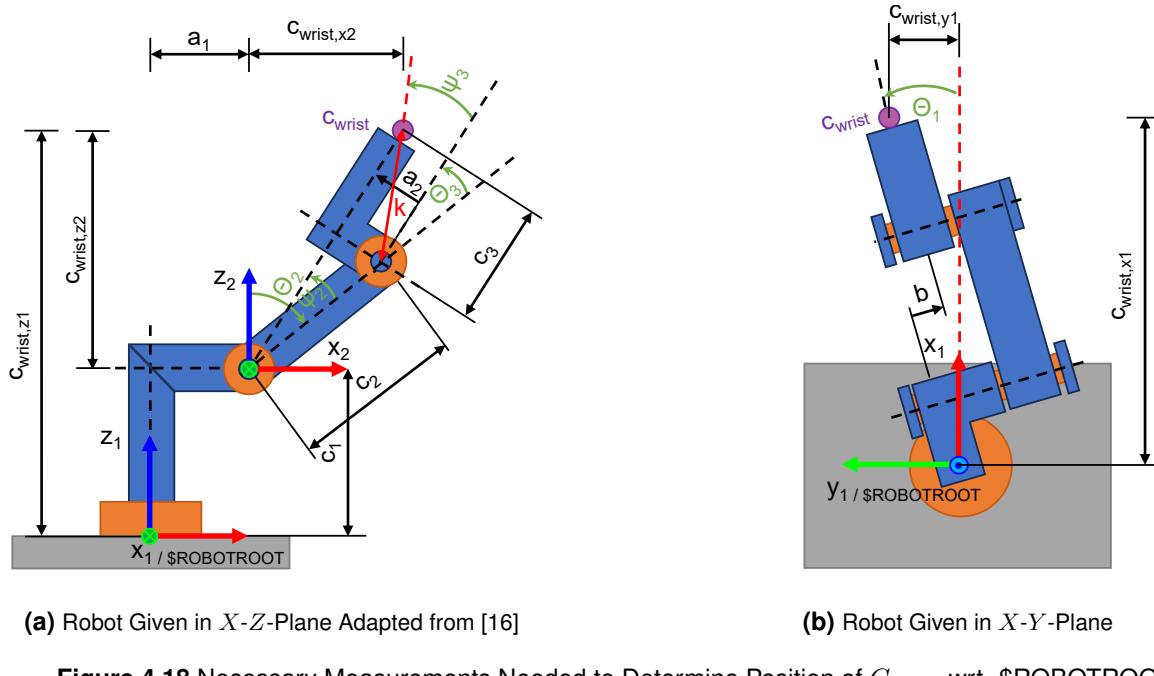


Figure 4.18 Necessary Measurements Needed to Determine Position of C_{wrist} wrt. \$ROBOTROOT

$$\psi_3 = \arctan\left(\frac{a_2}{c_3}\right) \quad (4.21)$$

$$k = \sqrt{a_2^2 + c_3^2} \quad (4.22)$$

$$c_{wrist,x2} = c_2 \cdot \sin(\theta_2) + k \cdot \sin(\theta_2 + \theta_3 + \psi_3) + a_1 \quad (4.23)$$

$$c_{wrist,y2} = b \quad (4.24)$$

$$c_{wrist,z2} = c_2 \cdot \cos(\theta_2) + k \cdot \cos(\theta_2 + \theta_3 + \psi_3) \quad (4.25)$$

$$c_{wrist,x1} = c_{wrist,x2} \cdot \cos(\theta_1) - c_{wrist,y2} \cdot \sin(\theta_1) \quad (4.26)$$

$$c_{wrist,y1} = c_{wrist,x2} \cdot \sin(\theta_1) + c_{wrist,y2} \cdot \cos(\theta_1) \quad (4.27)$$

$$c_{wrist,z1} = c_{wrist,z2} + c_1 \quad (4.28)$$

Subsequently, the position and orientation of the end-effector are determined based on the location of the wrist center C_{wrist} . To do this, a vector of length " c_4 " is added to the position of C_{wrist} in the direction of axis A6 (cf. Figure 4.11). The orientation results from the predefined alignment of axes A4 through A6.

The computation uses the rotation matrix $\mathbf{R}_{\$ROBOTROOT, \$NULLFRAME}$, which is calculated as shown in Equation 4.31 and describes the orientation of the end-effector relative to the robot base. In [16], this matrix is denoted as $\mathbf{R}_{0,E}$. By applying this matrix to the vector connecting C_{wrist} and the end-effector, its position in the $\$ROBOTROOT$ coordinate system can be determined.

The corresponding mathematical relationships are presented in Equations 4.29 through 4.32.

$$\mathbf{R}_{\$ROBOTROOT \rightarrow C_{wrist}} = \mathbf{R}_{A1, A2, A3} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} \quad (4.29)$$

$$\mathbf{R}_{C_{wrist} \rightarrow \$NULLFRAME} = \mathbf{R}_{A4, A5, A6} \begin{pmatrix} \theta_4 \\ \theta_5 \\ \theta_6 \end{pmatrix} \quad (4.30)$$

$$\mathbf{R}_{\$ROBOTROOT \rightarrow \$NULLFRAME} = \mathbf{R}_{\$ROBOTROOT \rightarrow C_{wrist}} \cdot \mathbf{R}_{C_{wrist} \rightarrow \$NULLFRAME} \quad (4.31)$$

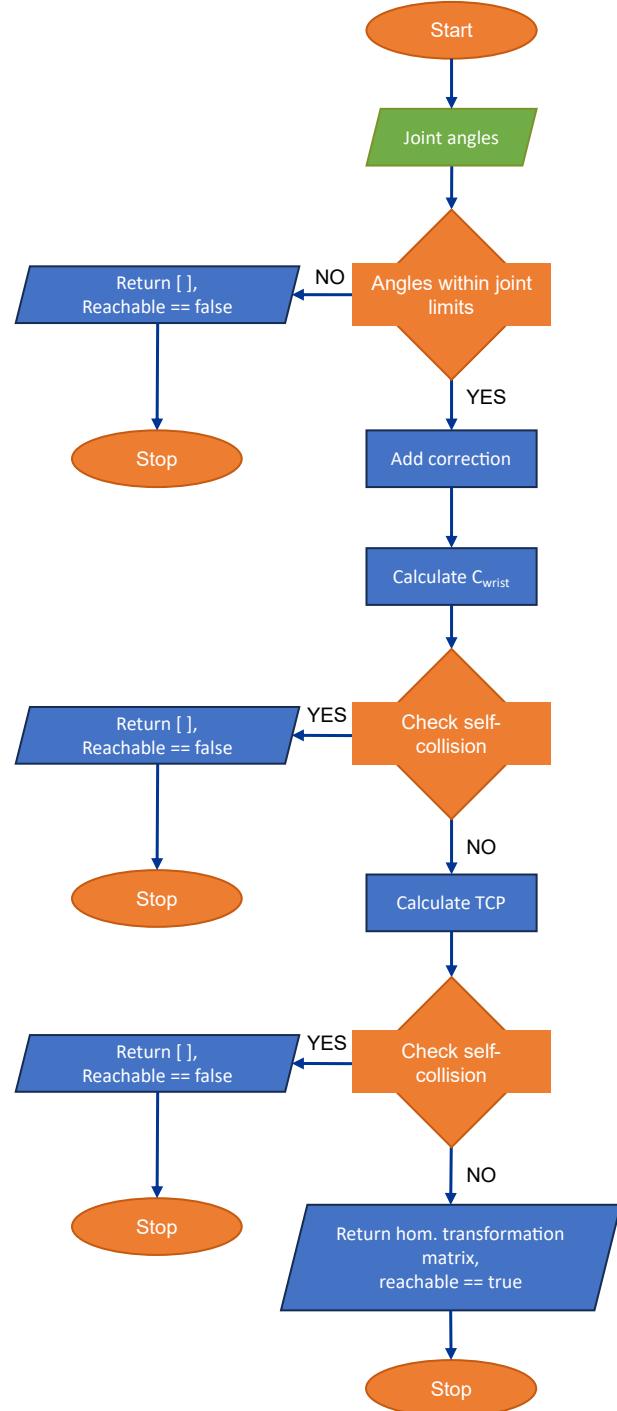
$$\mathbf{f} = \begin{bmatrix} c_{wrist,x1} \\ c_{wrist,y1} \\ c_{wrist,z1} \end{bmatrix} + c_4^{\$NULLFRAME} \cdot \mathbf{R}_{\$ROBOTROOT \rightarrow \$NULLFRAME} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.32)$$

The result from Equation 4.32 describes the position of the robot flange with respect to $\$ROBOTROOT$. For the origin of this coordinate system, again a self-collision check with the robot base is performed. Since the formulation in [16] does not account for additional tool attachments on the robot, the equation set was extended accordingly. It is assumed that the tool does not exhibit any rotation relative to the coordinate system of the robot flange ($\$NULLFRAME$), and that only a translational offset in the X , Y , or Z direction exists between $\$NULLFRAME$ and the TCP. The corresponding input values for this tool offset are extracted from the dictionary under the key "tool_offset" (cf. Chapter 4.5.2). As shown in Equation 4.33, this offset can also be computed by multiplying it with $\mathbf{R}_{\$ROBOTROOT \rightarrow \$NULLFRAME}$ (referred to as $\mathbf{R}_{0,E}$) and subsequently adding it to the position vector \mathbf{f} .

$$\mathbf{t} = \mathbf{f} + \mathbf{R}_{0 \rightarrow E} \begin{bmatrix} x_{\text{tool_offset}} \\ y_{\text{tool_offset}} \\ z_{\text{tool_offset}} \end{bmatrix} \quad (4.33)$$

If no self-collision is detected between the TCP and the virtual robot base, the homogeneous transformation matrix \mathbf{p} for the vector \mathbf{t} is constructed. It consists of the rotation matrix $\mathbf{R}_{0,E}$ and the vector \mathbf{t} as the translational component. The resulting matrix describes the position and orientation of the TCP, as determined by the provided joint angles θ_1 through θ_6 , relative to the coordinate system $\$ROBOTROOT$ (cf. Equation 4.34).

$$\mathbf{p}_{\text{homogeneous}, \$ROBOTROOT} = \begin{bmatrix} \mathbf{R}_{0 \rightarrow E} & \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (4.34)$$

**Figure 4.19** Flowchart for `forward_kinematics`-Function Inside `robotOPW`-Class

Inverse Kinematics

For the inverse kinematics, a homogeneous transformation matrix is provided that describes both the orientation and the position of the TCP relative to \$ROBOTROOT. Since all joint angles of the robot are apriori unknown, the implementation is considerably more complex compared to the forward kinematics and can yield up to eight individual solutions depending on the arm's configuration (cf. [16]).

To reduce computational effort and improve runtime, all standardizable pre-processing steps are consolidated before passing the homogeneous transformation matrix to *robotOPW*. This includes converting the coordinates defined by the G-code (given relative to \$BASE) into the \$ROBOTROOT coordinate system. This transformation is possible because all required input values are already known. Figure 4.20 illustrates all relevant coordinate systems given, as well as the transformation matrices that can be derived from the given inputs. The direction of each arrow corresponds to the direction defined by the respective input transformation matrix. As previously explained in Chapter 4.5.1, it is important to note that the origins of \$FRAME and the TCP are coincident, while the robot is commanded to move to the point specified by \$FRAME.

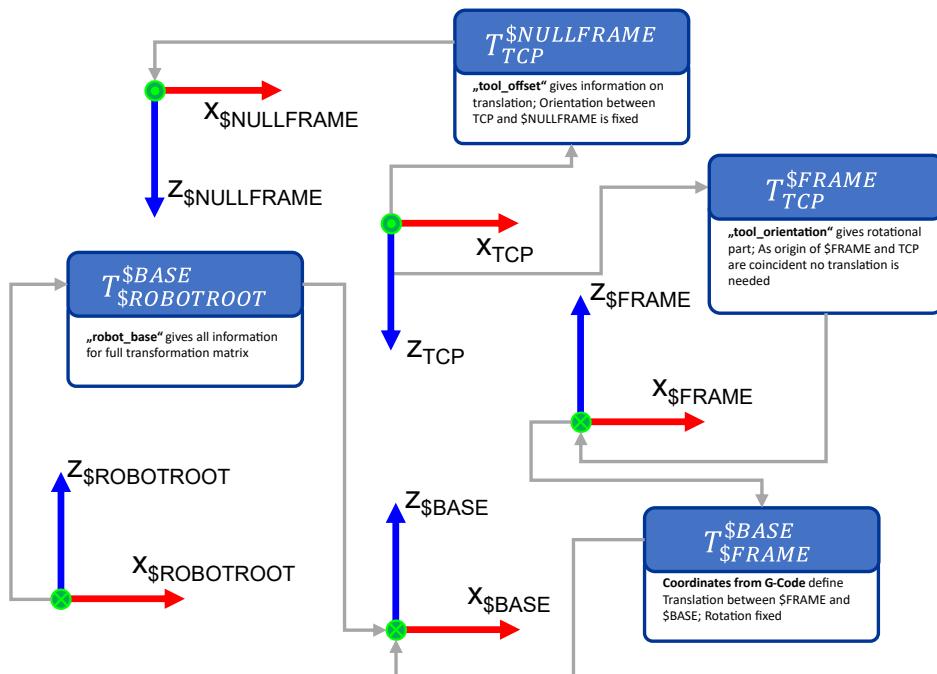


Figure 4.20 Coordinate Frames and Available Transformation Matrices

To construct the homogeneous transformation matrix, the rotational component is first derived. Equations 4.35 to 4.37 illustrate how this rotational part is composed.

$$\mathbf{R}_{\$ROBOTROOT \rightarrow \$BASE} = R_{ZYX} \begin{pmatrix} C_{base_coordinates} \\ B_{base_coordinates} \\ A_{base_coordinates} \end{pmatrix} \quad (4.35)$$

$$\mathbf{R}_{TCP \rightarrow \$BASE} = R_{ZYX} \begin{pmatrix} C_{tool_orientation} \\ B_{tool_orientation} \\ A_{tool_orientation} \end{pmatrix} \quad (4.36)$$

$$\mathbf{R}_{TCP \rightarrow \$ROBOTROOT} = \mathbf{R}_{\$BASE \rightarrow \$ROBOTROOT}^T \cdot \mathbf{R}_{TCP \rightarrow \$BASE} \quad (4.37)$$

The translational component, which expresses an arbitrary point (p) from \$BASE in \$ROBOTROOT, can be determined by combining and extending the previously introduced information into a homogeneous transformation matrix using the already introduced extension of $[0 \ 0 \ 0 \ 1]$ in the last row of the matrix:

$$\mathbf{T}_{\$ROBOTROOT \rightarrow \$BASE} = \begin{bmatrix} [\mathbf{R}_{\$ROBOTROOT \rightarrow \$BASE}] & \begin{bmatrix} x_{base_coordinates} \\ y_{base_coordinates} \\ z_{base_coordinates} \end{bmatrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.38)$$

$$\mathbf{p}^{\$ROBOTROOT} = \mathbf{T}_{\$BASE \rightarrow \$ROBOTROOT}^\top \cdot \mathbf{p}^{\$BASE} \quad (4.39)$$

The complete homogeneous transformation matrix for an arbitrary point from the G-code is shown in Equation 4.40. This matrix can now be passed to `robotOPW` to determine all possible joint configurations.

$$\mathbf{p}_{homogeneous}^{\$ROBOTROOT} = \begin{bmatrix} [\mathbf{R}_{TCP \rightarrow \$ROBOTROOT}] & \begin{bmatrix} \mathbf{p}_x, \$ROBOTROOT \\ \mathbf{p}_y, \$ROBOTROOT \\ \mathbf{p}_z, \$ROBOTROOT \end{bmatrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.40)$$

As described for the forward kinematics, the inverse kinematic was also extended to account for an offset between the `$TOOL` and `$NULLFRAME`. To achieve this, the coordinates defined under "tool_offset" are first transformed into the `\$ROBOTROOT` frame using $\mathbf{R}_{TCP \rightarrow \$ROBOTROOT}$ (cf. Equation 4.41 and 4.42). The position of C_{wrist} is then determined using Equation 4.43. A check for potential self-collisions of this point as well as compliance with the specified joint limits is also implemented in the program (see Figure 4.21).

$$\text{tool_offset}^{\$ROBOTROOT} = \mathbf{R}_{TCP \rightarrow \$ROBOTROOT} \cdot \begin{bmatrix} x_{\text{tool_offset}} \\ y_{\text{tool_offset}} \\ z_{\text{tool_offset}} \end{bmatrix} \quad (4.41)$$

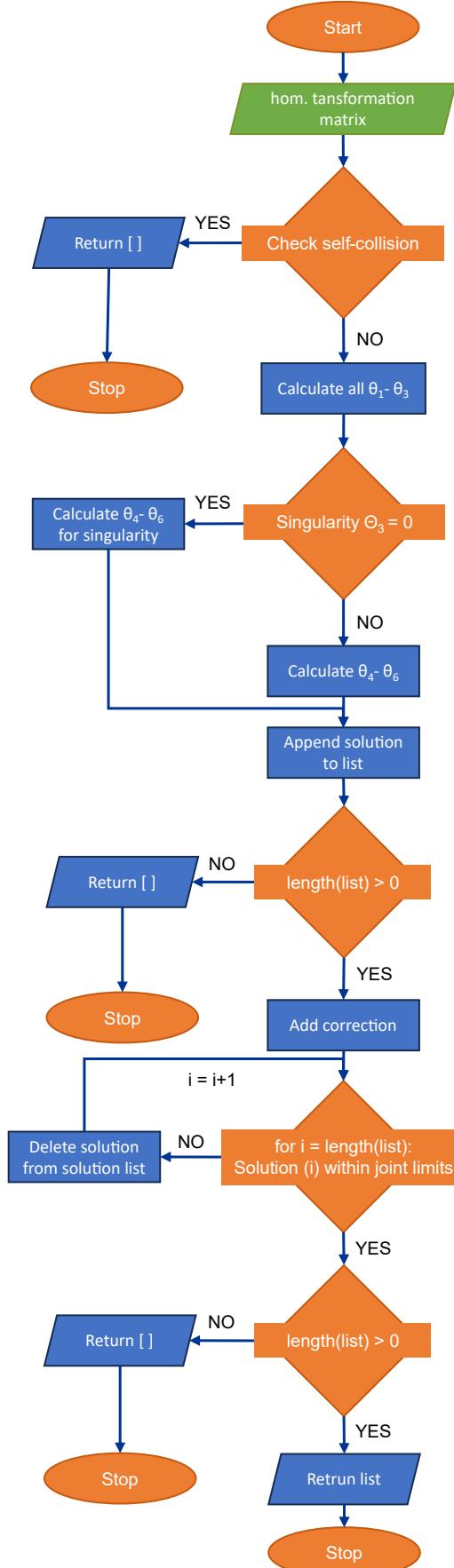
$$c_4^{\$ROBOTROOT} = c_4^{\$NULLFRAME} \cdot \mathbf{R}_{TCP \rightarrow \$ROBOTROOT} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.42)$$

$$\mathbf{C}_{wrist}^{\$ROBOTROOT} = \begin{bmatrix} p_x, \$ROBOTROOT \\ p_y, \$ROBOTROOT \\ p_z, \$ROBOTROOT \end{bmatrix} - \begin{bmatrix} x_{\text{tool_offset}} \\ y_{\text{tool_offset}} \\ z_{\text{tool_offset}} \end{bmatrix} + \mathbf{c}_4^{\$ROBOTROOT} \quad (4.43)$$

The remaining steps of the inverse kinematics follow the formulation presented in [16] and are therefore not discussed in detail. However, it should be noted that the formulation provided in [16] does not yield a unique solution in the case of a singularity at $\theta_5 = 0$. In this configuration, the rotation axes of θ_4 and θ_6 become collinear according to the coordinate definition in [16] (see Figure 4.8b). This situation corresponds to the *Gimbal Lock* case described in Chapter 4.5.3.

To still obtain a unique solution within the program, the rotation around θ_4 is set to zero, and only θ_6 is used to apply the required orientation. Reference [82] provides an overview of the typical singularity scenarios encountered in 6-DOF serial robots. Other singularities, apart from the so-called *wrist singularity*, in which axes A4 and A6 are collinear are not considered further, as they are not deemed relevant for the printing process with the given robot configuration.

As a result, it is now possible to verify the reachability of all given positions and joint angles. In addition a corresponding validation of the calculations has been conducted and is described in more detail in Chapter 5.1.2.

Figure 4.21 Flowchart for `inverse_kinematics`-Function Inside `robotOPW`-Class

4.6 Analysis of Line Geometry and Extrusion Values [package: *pump*]

[user_input](#) [.gcode_file](#) [gcode](#) [robot](#) [pump](#) [krl](#) [.src-file](#) [report](#) [.docx-file](#) [rhino](#) [.3dm-file](#)

Figure 4.22 Progress Bar for Chapter 4.6

4.6.1 Calculation of Line Width [module: *calculate_linewidth*]

The objective of this module is to determine the appropriate amount of material for each printing movement, thereby enabling integration of the pump into the process. To achieve this, the effective line width of each segment must first be calculated. This information is essential not only for controlling the extrusion precisely, but also for visualizing the print path in the Rhino model.

Since the line width of individual extrusion movements is not explicitly included in the G-code after slicing, and no geometric data from the associated *.stl* file can be extracted, the actual line width must be reconstructed from the extrusion values (E-values) contained in the G-code. Equation 4.44 shows the formula used by Cura to compute the E-value, from which the deposited line width can be derived in reverse [20]. As this is a secondary source and no official published formula could be identified, the equation used here was additionally validated through multiple control calculations and found to be plausible.

$$E_{\text{value}} = \frac{h \cdot SF \cdot d_n \cdot l}{\frac{\pi}{4} d_f^2} = \frac{4 \cdot h \cdot SF \cdot d_n \cdot l}{\pi \cdot d_f^2} \quad (4.44)$$

Table 4.4 Definition of Variables Used in Equation 4.44

Symbol	Name	Unit	Input
<i>h</i>	Layer height	mm	calculated
<i>SF</i>	Flow value	–	"lintype_flow" from <i>pump.setup.json</i>
<i>d_n</i>	Line width	mm	–
<i>l</i>	Segment length	mm	calculated
<i>d_f</i>	Filament diameter	mm	"filament_diameter" from <i>pump.setup.json</i>
<i>E_{value}</i>	Extrusion value	mm	G-Code input (relative value)

Cura assumes an idealized rectangular geometry for the extruded segment when calculating the material volume. The volume *V* of the deposited line segment is determined by the product of the layer height *h*, a scaling factor *SF*, the line width *d_n*, and the segment length *l*, i.e., *V* = *h* · *SF* · *d_n* · *l*.

To provide this volume during the printing process, Cura advances a filament strand with a known diameter *d_f* and circular cross-section. The required feed length of the filament is defined in the G-code using the E-value. Figure 4.23 illustrates the geometric relationship.

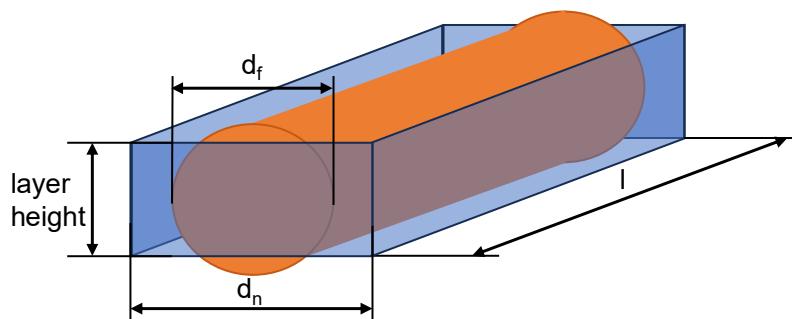


Figure 4.23 Geometry for Filament and Line Segment

The flow rate for each line segment cannot be retroactively extracted from the G-code, but it has already been assigned to a predefined line type via the `setup.json` file. This allows it to be uniquely traced for each line. Since all other parameters are known, the line width d_n remains the only unknown in the equation. The formula can therefore be rearranged to solve for d_n for each line segment.

$$d_n = \frac{E_{\text{value}} \cdot \pi \cdot d_f^2}{4 \cdot h \cdot SF \cdot l} \quad (4.45)$$

The following section explains how the line width calculated in this manner can be translated into commands for controlling the pump.

4.6.2 Calculation for Pump Commands [module: `calculate_flow, calculate_rpm`]

Flow [module: `calculate_flow`]

As described in Chapter 2.6.2, the pump serves a role analogous to the extruder of a conventional FDM printer by continuously supplying material to the nozzle throughout the printing process. For this, the volume of an entire line segment is extruded within a defined time interval.

The decisive factor for this is the nozzle's travel speed along the length of the segment (in mm/s), as it directly determines the required volumetric flow rate q_{flow} of the pump, which is specified in liters per minute. Ideally, the exact local speed profile along each line segment would be known to dynamically adapt the flow rate. However, since the actual travel speed cannot be predicted precisely as it is highly dependent on the orientation as well as prior or following commands, only the target speed specified by the user in the `setup.json` file under "print_speed" in m/s is available.

Equation 4.46 shows how this information based on the known parameters is used to calculate the required flow rate in mm³/s:

$$q_{\text{flow}}^{[\text{mm}^3/\text{s}]} = d_n^{[\text{mm}]} \cdot h^{[\text{mm}]} \cdot v_{\text{print}}^{[\text{m}/\text{s}]} \cdot 1000^{[\text{mm}/\text{m}]} \quad (4.46)$$

Conversion of Flow to RPM and Voltage [module: `calculate_rpm`]

Since the pump cannot be directly controlled via a volumetric flow rate, the calculated flow value must be converted into a usable control parameter. For this purpose, the pumps so-called *characteristic curve* is used, which describes the flow rate at the pump outlet as a function of the motor speed (abbreviated as rpm) or the motor voltage (V).

As no characteristic curve has been specified by Knauf PFT GmbH & Co. KG, a linear relationship is assumed for the example input: from 0 rpm at 0 L/min and 0 V up to 458 rpm at 90 L/min and 10 V, which corresponds to the maximum motor input voltage and flow rate of the pump [66].

Alternatively, the user can define a custom pump characteristic curve in the `setup.json` file under "characteristic_curve". The input is provided as a nested list, in which specific support points (defined by [Flow, RPM, Voltage]) along the curve are specified. Listing 4.6.1 shows an example configuration.

Listing 4.6.1 Example Input "characteristic_curve" of Pump in `setup.json`

```

"characteristic_curve": {
    "description": "Gives the function of RPM to Flow[liter/min] of the Pump in [Flow,RPM,Voltage]",
    "value": [[10, 146, 1], [0, 0, 0], [90, 456, 10]],
    "type": "list[list[float, float, float]]"
},

```

The discrete points are sorted in ascending order by *rpm* within the module, and the characteristic curve is calculated by mapping discrete points of the flow value from `calculate_flow` and interpolating linearly between two discrete values. The same is done for the given voltage numbers. The result

yields a continuous graph for rpm or voltage, which is then used to automatically control the pump during the printing process.

The user can choose whether the pump is controlled via voltage or rotational speed in the output *.src* file; this selection is also made in *setup.json* under the key "pump_control". The input specified there determines the value later used in the KRL output.

In addition, a velocity limit has been implemented to ensure that the nozzle does not move faster than the pump's maximum delivery rate allows. This is based on Equation 4.46, rearranged to solve for the printing speed v_{print} . The maximum flow rate q_{max} is used as the upper limit. The calculated speed thus corresponds to the highest permissible feed rate at which the desired line width can still be reliably achieved.

As described in Chapter 3.2.2, transitions between different line types can trigger retraction events, during which material is briefly pulled back to avoid printing artifacts at segment boundaries. However, due to the viscosity of the material used in this context, it is not practical to mechanically retract the material from the nozzle on a regular basis.

Therefore, it is proposed instead to integrate a fast-acting valve between the nozzle and the delivery hose. This valve is mounted on the robot's tool arm and prevents material from oozing during pure travel movements.

In Cura, specific movement patterns arise when switching between G1 and G0 commands. These are summarized in Table 4.5.

Table 4.5 Typical G-Code Patterns for FDM-Style Slicing and Printing Using *Ultimaker Cura 5.8.0*

Pattern 1	E-value	Pattern 2	E-value
G1 Extrusion	>0	G1 Extrusion	>0
G0 Travel	<0	G1 Retract	<0
G1 Extrusion	>0	G0 Travel	<0
		G1 Protract	= 0
		G1 Extrusion	>0
see Listing A.3.2, Line 266–268 (absolute Mode M82)		see Listing A.3.2, Line 72–80 (absolute Mode M82)	

The proposed valve can be controlled based on the extrusion values recorded in the *.src* file. If the E-value is less than zero, the valve is closed; if it is zero or greater, the valve is opened. To ensure consistent material flow and to avoid pressure buildup in the pump against a closed valve, both opening and closing should be performed with a slight delay. However, the control and implementation of the valve regulation are not part of this work due to the current lack of suitable hardware on the robot.

In the present setup, the printing process is carried out without retractions. Within Cura, material retractions can be completely disabled. The parameter "retract" specified in the *setup.json* file controls the corresponding input passed to Cura.

4.7 Translation to KUKA Robot Language [package: *krl*]

4.7.1 List of Collected Information



Figure 4.24 Progress Bar for Chapter 4.7

Based on the previously determined coordinates, line types, extrusion values, and speeds, a robot-interpretable *.src* file can now be generated. Figure 4.25 illustrates the structure of a single entry in the form of a dictionary of the before mentioned list that contains all information collected throughout the program. Each of these dictionary entries is subsequently converted into an individual KRL code line.

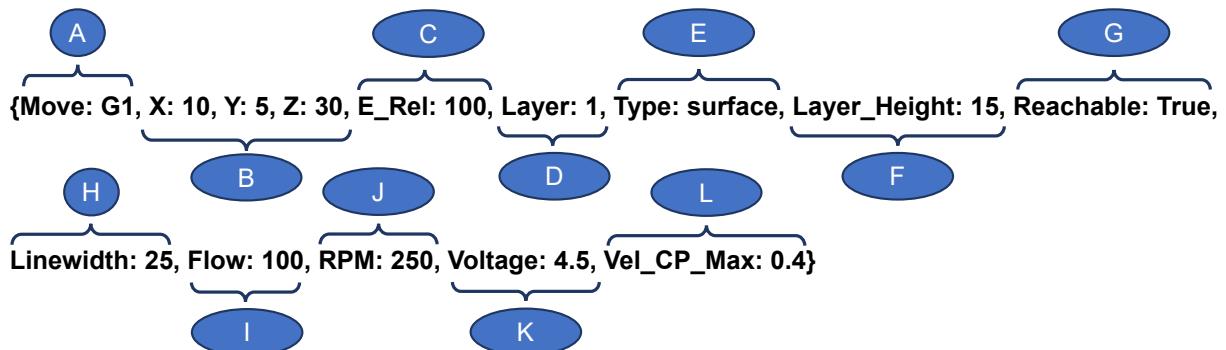


Figure 4.25 Dictionary for Arbitrary G-Code Line Used Inside the Program to Build *.src*-file

Table 4.6 Overview of Parameters Used in Figure 4.25

Symbol	Information	Source
A	Definition of move type (G1, G0)	G-Code [package: g_code]
B	Target coordinates of end point	G-Code [package: g_code]
C	Extrusion value relative to last extrusion	G-Code [package: g_code]
D	Layer number counting from 0	Calculated [package: g_code]
E	Line type of current section (surface, infill, retract, unknown, ...)	Assigned [package: g_code]
F	Current layer height	Calculated [package: g_code]
G	Coordinates reachable for specified robot configuration	Validated [package: robot]
H	Linewidth in mm of current line segment	Calculated [package: pump]
I	Flow related to current line type	Calculated [package: pump]
J	RPM of pump rotor needed to extrude specified line width	Calculated [package: pump]
K	Voltage of pump motor needed to extrude specified line width	Calculated [package: pump]
L	Maximum possible print velocity (depending on pump capability)	Calculated [package: pump]

The program code as a whole is structured analogously to the G-code and divided into three functional sections. The process begins with the initialization phase, during which local variables are declared and assigned, time and progress tracking is started, and return values are set. This is followed by the actual printing phase, in which the path data generated by Cura is translated line by line into KUKA Robot Language and output accordingly. After the printing process is complete, all previously initiated routines are terminated, and the system is returned to a defined initial state, as described in Chapter 3.2.3.

4.7.2 Initialization & Reset of Robot

The initialization and reset sections of the program can largely be configured by the user via the *setup.json* file. Custom code blocks in KRL can be defined under the keys "start_code" and "end_code", which are inserted into the resulting *.src* file without additional validation. To avoid redundant entries, for example when specifying print parameters multiple times inside *setup.json*, placeholders can be used. These placeholders are enclosed in question marks (e.g., ?id?) and reference corresponding keys in the *setup.json* file that contain the actual values (in this case id).

If a user-defined placeholder cannot be resolved due to a missing or incorrectly specified key in the *setup.json* file (e.g., ?ip? instead of ?id?), the system generates an error message, as shown in Listing 4.7.1, explicitly indicating the invalid entry. As a result, no *.src* file is created. To enable successful output, the error must first be corrected and the program restarted.

Listing 4.7.1 Console Output from *setup replace_strings* with Error Message

```
[ERROR] Key ?ip? inside robot start or end code not found
[WARNING] Key ?ip? not replaced
```

In addition to the manually defined code lines, the module *start_code_python* allows the input of a list of KRL-formatted strings into which dynamic variables from the Python program flow can be integrated. This enables the integration of additional information gathered during program execution into the generated *.src* file. It can be used for example for the automatic inclusion of the maximum number of layers, which was determined at the beginning of the process as described in Chapter 4.4. This information can in turn be used to continuously monitor print progress during execution. The processing of such dynamic content, however, is only possible within the corresponding Python module and can't be accessed from the *setup*-module.

A comparable module for the "end_code" section has not yet been implemented, as no practical use case has arisen so far. However, an extension analogous to *start_code_python* could be implemented at any time if needed.

4.7.3 Main Section

The main section of the `.src` file requires the collected information from the G-code (cf. Figure 4.25) to be converted into the format specified by KRL. Figure 4.26 illustrates the structure of a command line that contains all the information necessary for a linear movement.

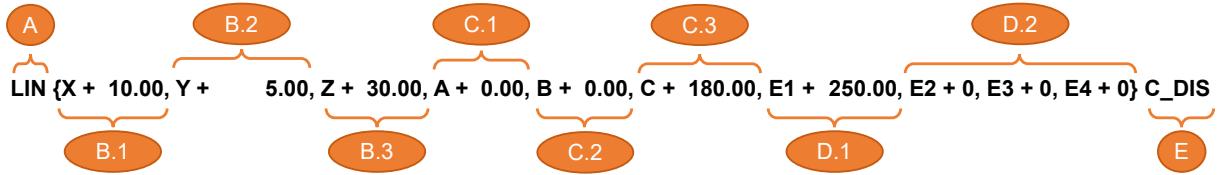


Figure 4.26 Example of KRL-Command for Linear Movement (RPM Used for Pump Control)

Table 4.7 Extended Symbol Overview for KRL Line Definition

Symbol	Information	Source
A	Definition of move type (LIN, CIRC, PTP)	LIN as only G1 or G0 commands occur in G-Code
B.1	Target coordinate of end point in x-direction	see B in Figure 4.25
B.2	Target coordinate of end point in y-direction	see B in Figure 4.25
B.3	Target coordinate of end point in z-direction	see B in Figure 4.25
C.1	Orientation of flange in relation to base coordinate system for z-axis of flange	set via "tool_orientation" ["A"]
C.2	Orientation of flange in relation to base coordinate system for y-axis of flange	set via "tool_orientation" ["B"]
C.3	Orientation of flange in relation to base coordinate system for x-axis of flange	set via "tool_orientation" ["C"]
D.1	Information for external axis of robot (here E1 gives information on extruded material along line and valve position)	see J or K in Figure 4.25
D.2	Unused external axis of robot (always set to 0)	Fixed value [package: pump]
E	Point is approximated to give a more continuous motion between points	mostly set to C_DIS

The commands in the `.src` file are formatted so that the decimal places of all numerical values are precisely aligned in vertical columns within the output file. To achieve this, the maximum character length for each numerical category is determined from the entire G-code in order to efficiently define the output field width. Subsequently, all values are rounded to a predefined number of decimal places, formatted with the appropriate sign, and right-aligned within their respective fields. This structure not only improves readability but also facilitates the targeted editing of individual values, for instance by column-wise selection and replacement.

In addition to the movement commands, the currently active line type is also output. The type designation stored under `E` in Figure 4.25 is converted into an integer value using the mapping defined in `setup.json` under `"type_number"`. This value is output at the beginning of each new line type and can be linked to a already implemented variable of the KUKA system, allowing the current line type to be displayed on the *smartPAD* user interface of the robot. In addition to the line type, the print speed is also specified at the start of each segment category. Although not yet implemented, defining individual speeds for different line types could be considered in the future. This would allow for optimized printing. As an example reducing speed for visually exposed segments like *surface* or *outer wall* could improve overall print quality. Since travel movements are not constrained by the pump's maximum flow rate, an independent speed setting is already implemented and can be specified via `"travel_speed"` in `setup.json`. Listing 4.7.2 illustrates an example output incorporating the afore mentioned features.

Listing 4.7.2 Example for Movement Command and Line Type Annotation Used in .src-Output

```

LIN {X + 586.97, Y + 2255.91, Z + 60.00, A + 0.00, B + 0.00,
     C + 180.00, E1 + 5.00, E2 + 0, E3 + 0, E4 + 0} C_DIS
LIN {X + 587.00, Y + 2257.18, Z + 60.00, A + 0.00, B + 0.00,
     C + 180.00, E1 + 5.00, E2 + 0, E3 + 0, E4 + 0} C_DIS

PATH_TYPE = 0
$VEL.CP=0.500
LIN {X + 587.00, Y + 2257.18, Z + 75.00, A + 0.00, B + 0.00,
     C + 180.00, E1 + 0.00, E2 + 0, E3 + 0, E4 + 0} C_DIS
LIN {X + 599.60, Y + 2257.18, Z + 75.00, A + 0.00, B + 0.00,
     C + 180.00, E1 + 0.00, E2 + 0, E3 + 0, E4 + 0} C_DIS

```

Furthermore, the generated .src file includes a timer that controls the minimum layer time. This timer is reset at the beginning of each new layer. If the robot reaches the end of the current layer before the time specified under "min_layer_time" in *setup.json* has elapsed, the system waits until the target time is reached. Since all *travel*-movements at the end of a layer are still assigned to that layer and every layer necessarily ends with such a movement, it is ensured that the valve mounted on the tool arm receives a command to close. This reliably prevents any unintended material discharge from the nozzle. Listing 4.7.3 illustrates the process using an example .src file.

Listing 4.7.3 Example for Layer Timer and Minimum Layer Time Implemented for .src-Output

```

LAYER = 0
$TIMER_STOP[4] = FALSE ; start layer timer

PATH_TYPE = 4 ; line type as int
$VEL.CP=0.133 ; velocity in m/s
[...]
WAIT FOR $TIMER[4] > 10000 ; time in ms

$TIMER_STOP[4] = TRUE ; stop layer timer
$TIMER[4] = 0 ; reset layer timer

LAYER = 1 ;layer change
$TIMER_STOP[4] = FALSE ; start layer timer

```

4.8 Print Report [package: *report*]

To provide a compact summary of the most relevant print parameters after successful execution of the program and in particular to allow the user to visually verify the position of the printed object on the print bed a report is automatically generated using the libraries *python-docx* [84], *scipy* [78], and *matplotlib* [49].



Figure 4.27 Progress Bar for Chapter 4.8

This report summarizes the key parameters of the print job and includes graphical evaluations of the characteristic pump curve and the printed object. The report is based on a predefined *.docx* template included in the program. Keys within the template are automatically replaced with the corresponding values, and the final file is saved in the output directory.

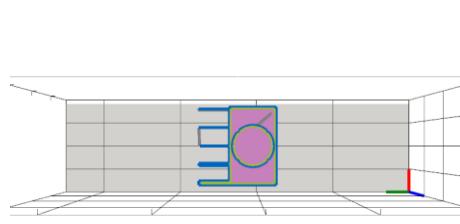
The report is structured into the following main sections:

- **Print section:** This section contains information about the processed geometry, including the file name, estimated volume, and the calculated weight of the print based on the extruded filament volume and a specific weight of 21 KN/m³. It also lists the overall object dimensions in all spatial directions as well as the minimum and maximum coordinates on the print bed, allowing for a quick plausibility check of the object placement.
- **Robot section:** This part lists the robot parameters used in the process. It includes the model designation, the base geometry in OPW format, and the tool offset and orientation. Additionally, the absolute position of the robot (\$ROBOTROOT) relative to the print bed (\$BASE) is given.
- **Printbed section:** This section provides the defined dimensions of the print bed along with the coordinates of the object's placement in the workspace.
- **Print parameter section:** This part lists the specified print and travel speeds, as well as the mapping of G-code line types. It provides a quick overview that helps with orientation inside the generated *.src* file.
- **Pump section:** This section specifies the pump control setup used during printing. It indicates whether retraction was enabled and whether the pump is controlled via rpm or voltage. The configured flow rates for each line type are listed as a percentage of the nominal flow rate.

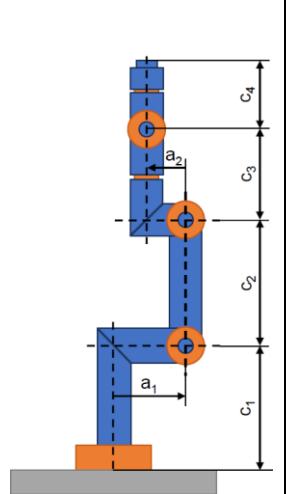
The clear structure of the report is intended to facilitate the review of all relevant parameters before starting the print and to ensure quick documentation of the selected settings. On the next page an example report generated for the test object using a scaling factor of 8000 percent is provided.

REPORT

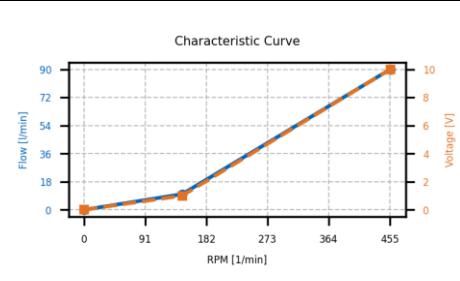
Print:

	Filename:	Slicer Test v3.stl
	Estimated weight [kg]:	72.636
	Estimated volume [liter]:	34.589
	Measurements [mm]:	[1015.0, 1019.566, 75.0]
	Coordinates [mm]:	
	Xmin: 92.5	Xmax: 1107.5
	Ymin: 1742.5	Ymax: 2762.066
	Zmin: 0	Zmax: 75.0

Robot:

	ID:	KUKA KR340 R3330									
	Geometry [mm]:	{'a1': 500, 'a2': 55, 'b': 0, 'c1': 1045, 'c2': 1300, 'c3': 1525, 'c4': 290}									
	Tool Offset [mm]:	{'X': -10.99, 'Y': -0.86, 'Z': 917.61}									
	Tool Orientation [deg]:	{'A': 0, 'B': 0, 'C': 180}									
	Printbed [mm]:	{'X': 1200, 'Y': 4500, 'Z': 2000}									
	Location [to Printbed in mm]:	{'X': -1460.9, 'Y': 2237.66, 'Z': -268.5, 'A': 0, 'B': 0.0, 'C': 0.0}									
	Set Print Speed [m/s]:	0.35	Travel Speed [m/s]:	0.5							
Line Type Dictionary:			Minimum Layer Time [sec]:	10.0							
Protract	-2	Travel	0	Wall outer	1	Surface	7	Bridge	5	Support	7
Retract	-1			Wall inner	2	Infill	4	Curb	6	Unknown	99

Pump:

	Retract	activated			
	Pump control:	RPM			
	Flow [%]:				
	Wall Outer:	100		Infill:	100
	Wall Inner:	100		Bridge:	100
	Surface:	100		Curb:	100
	Unknown:	100			

4.9 Output in Rhino [package: *rhino*]

In addition to the compact overview provided by the generated report, all collected information is also visualized in a Rhinoceros3D file (.3dm). The objective is to allow for a layer-by-layer inspection of the printing process, analogous to the graphical representation used in slicer programs. This includes the visualization of attributes such as print speed, line width and the reachability of individual points.



Figure 4.28 Progress Bar for Chapter 4.9

Therefor a Rhino file is first prepared with a suitable layer structure. Subsequently, the individual line segments are inserted into the corresponding layers and sub-layers. The following sections explain the complete process from raw data preparation to the finished Rhino model.

4.9.1 Setup of Rhinoceros3D File [sub-package: *pre_process*]

The first step consists of creating a template file in the specified output folder before any geometric data is added. The program systematically processes the following three steps:

1. **Creation of an empty .3dm file:** Using the *rhinoinside* library [9], an empty Rhino file is generated, accessing a local installation of Rhinoceros3D.
2. **Definition of custom line types:** To allow full control over the graphical representation of motion and print paths, specific line types are generated within the file. These enable the visual differentiation between different printing and travel operations.
3. **Construction of the layer and sublayer structure:** In addition to the three main layers (*toolpath*, *printbed*, *robot*), the structure within the *toolpath* layer is further subdivided. Sublayers are first created for each print layer, within which further sublayers for the individual line segments are added. This hierarchical structure allows the print process to be visualized step-by-step in Rhino by toggling the visibility of individual layers. Figure 4.29 schematically illustrates the layer structure.

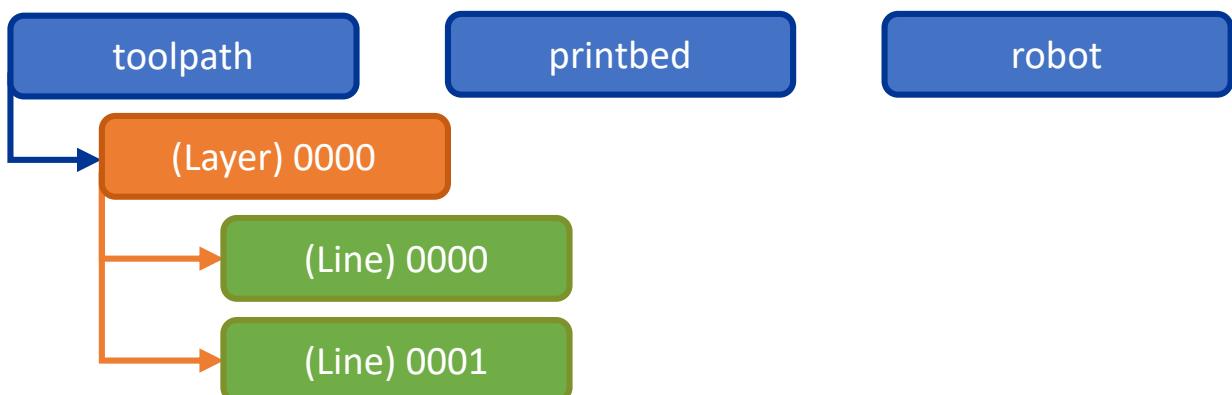


Figure 4.29 Schematic of Layer and Sublayer Structure Inside .3dm-file

4.9.2 Write Information to Rhino File [sub-package: *process*]

Preparation of the Rhino File [module: *import_robot, draw_printbed*]

Once the basic structure of the Rhino file has been created, the corresponding geometries can be added. To obtain a detailed digital representation of the entire hardware setup during the printing process, it is possible to import existing .3dm files of the robot cell. These files are referenced via the parameter "3dm_file" in the *setup.json* and are automatically imported when the Rhino file is created.

The imported geometry is then shifted by the displacement defined under "base_coordinates", relative to the origin (\$BASE) of the Rhino file. No rotation is applied. Therefore, the robot base (\$ROBOTROOT) must already lie at the origin of the imported geometry, and its coordinate system must be aligned parallel to the orientation of the print bed's coordinate system \$BASE.

Only the following geometry types are processed: points, curves, lines, B-reps, and meshes. Other entity types are ignored or may lead to errors.

In addition, the print bed is generated as a solid object based on the specified dimensions, with a fixed thickness of 15 cm in the negative Z-direction. It is placed at the origin of the global coordinate system, with its surface lying in the X-Y plane.

Figure 4.30 shows the resulting viewport in Rhino after completion of these initialization steps.

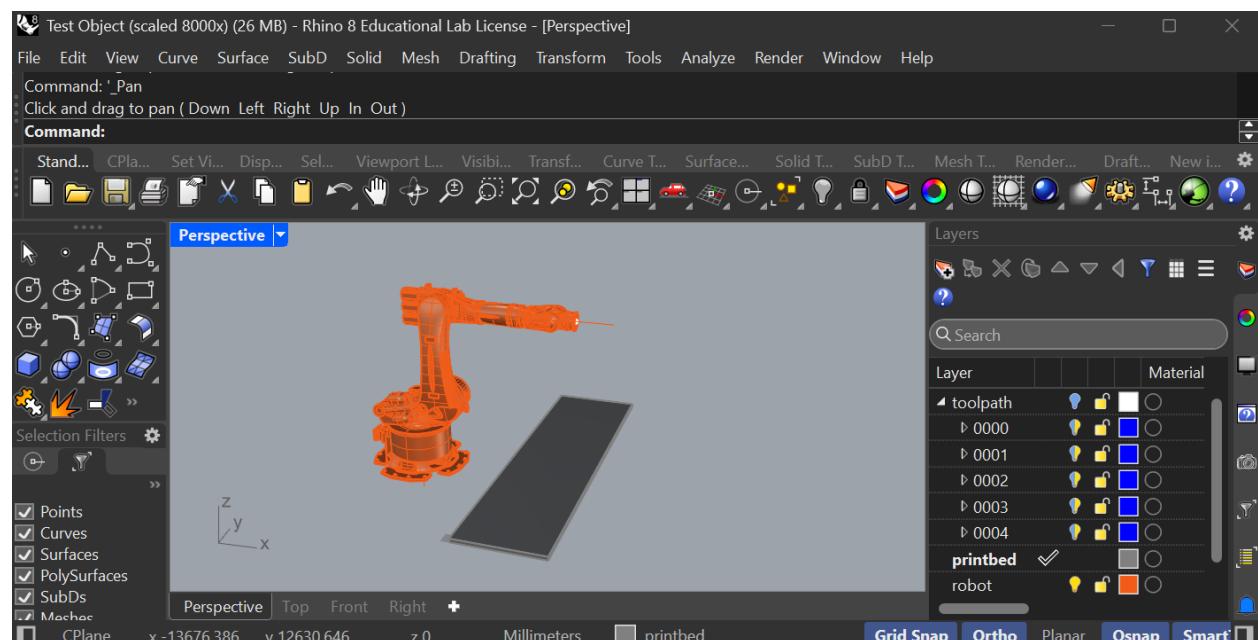


Figure 4.30 Overview of Rhino-File After Initialization

Visualisation of Tool Path [module: `extend_gcode, draw_gcode`]

After the basic structure of the Rhino file has been created, the generated print paths are transferred and visualised using two central modules:

1. **Numbering of lines and points:** Since a separate polyline is created for each line type (*travel*, *wall_outer*, *infill*, etc.), a new polyline must be started whenever the type changes within a print layer. The endpoint of the previous polyline is therefor duplicated and used as the start point of the new one. This task, along with the continuous numbering of lines and points within each layer, is handled by the `extend_gcode` module.
2. **Transfer of points and lines into the Rhino file:** The prepared and numbered points and line segments are transferred into the designated sublayers within the parent layer *toolpath*. This task is performed by the `draw_gcode` module.

In addition to importing the line geometries, the `draw_gcode` module highlights specific points using color to mark particular events during the printing process:

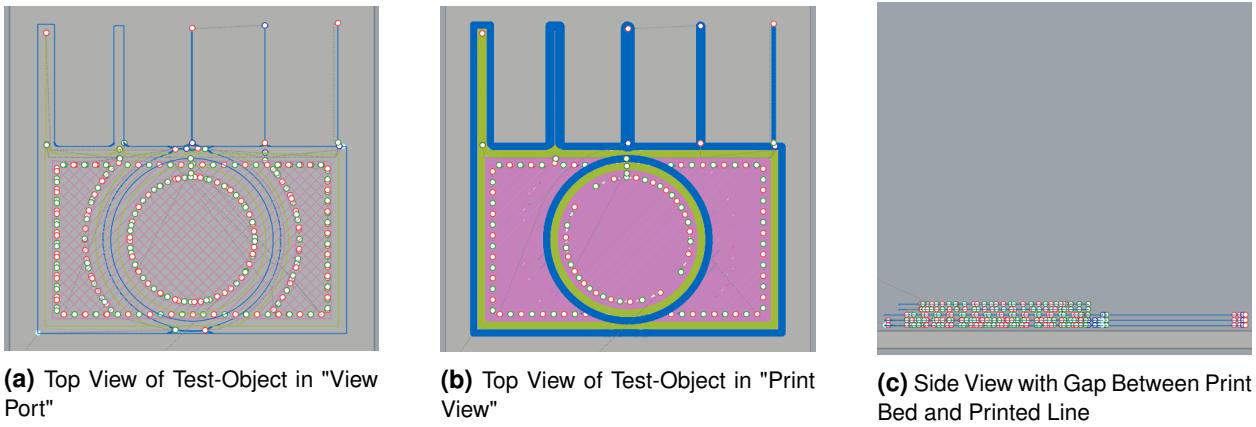
- **beginning:** Starting point of the entire print process (set via "start_position" in *setup.json*).
- **end:** Final point of the entire print process (defined by "end_position" in *setup.json*).
- **retract:** Point where material extrusion is paused for a travel move.
- **protract:** Point where material extrusion resumes for a new extrusion move.
- **start:** Starting point of a line segment.
- **stop:** Endpoint of a line segment.

Color assignments are defined via the "point_color" entry in the *setup.json* file. Points marked as "unreachable" (see Figure 4.25, section G) are always displayed in black, regardless of the configuration. The same applies to lines with unknown or erroneous types. This ensures that critical conditions remain clearly visible.

Four predefined line styles (*solid*, *dashed*, *dotted*, *dash_dotted*) are available for line visualization. The mapping to motion types is specified in "line_style_line" within the *setup.json* file.

To generate a three-dimensional representation of the object geometry, each line is assigned its calculated width (see Chapter 4.6.1). In Rhino, this is implemented using the so-called *Print Width*, which is displayed when *Print View* is activated [command: *PrintDisplay*: (State = On)]. Each line is rendered as a solid with a circular cross-section.

The actual geometry of the extruded lines (e.g., rectangular or elliptical, defined by line width and layer height) is not reproduced exactly as the line-segments represent the path of the nozzle tip and lie not within the center of each line segment. This leads to a slightly idealized representation and minor inaccuracies in the vertical dimensions of the segments. This design choice was made deliberately to reduce computational effort and memory usage. For the reason given, there is a vertical offset of half the layer height between layer 0 and the print bed. Figure 4.31 illustrates the described characteristics.

**Figure 4.31** Visualization of test object in Rhinoceros3D

In addition to their visual properties, all line and point objects are assigned name attributes that document their unique identification within the printing process. A continuous four-digit numbering system is used, starting at zero for each category:

- For lines: Layer / Line number / Segment number
(e.g., 0002/0007/0003 → Layer 3, 8th line within the layer, 4th segment).
- For points: Layer / Line number / Point number
(e.g., 0004/0016/0001 → Layer 5, 17th line within the layer, 1st point).

Note that if "point_print" is set to "false" in the *setup.json* file, only the previously listed point types are displayed visually to maintain clarity in the file. These are labeled as point 0000 and 0001. Intervening points are not counted. If this setting is changed to true, all contact points of individual line segments are additionally marked with colors.

Moreover, all line and point objects receive additional text attributes that store metadata such as local print speed, line width, or the extruded material volume. These attributes are intended to assist the user in conducting a detailed analysis in case a fault diagnosis on the physical print object becomes necessary. Figure 4.32 illustrates an example of the embedded information for line and point objects.

Key	Value
Point	0001
Layer	0004
Line	0016
Point_Info	stop

Key	Value
Velocity [m/s]	0.35
Voltage [V]	0.787
Layer	0002
Line	0007
Extrusion	1
RPM [1/min]	114.975
Flow [mm ³ /s]	131249.995
Linewidth [mm]	25.0
Type	wall_outer

(a) Example of Text Attributes Used for Points
(b) Reference for Specified Point and Line on Test-Object
(c) Example of Text Attributes Used for Lines

Figure 4.32 Attributes Used for Labeling in Rhinoceros3D

This completes the creation of the digital model. Figure 4.33 shows the full Rhino interface as it is presented to the user after successful program execution. In order to react to incorrect inputs during execution and to ensure the robustness of the process, additional safety mechanisms have been implemented. The explanation of these mechanisms is the subject of the following chapter.

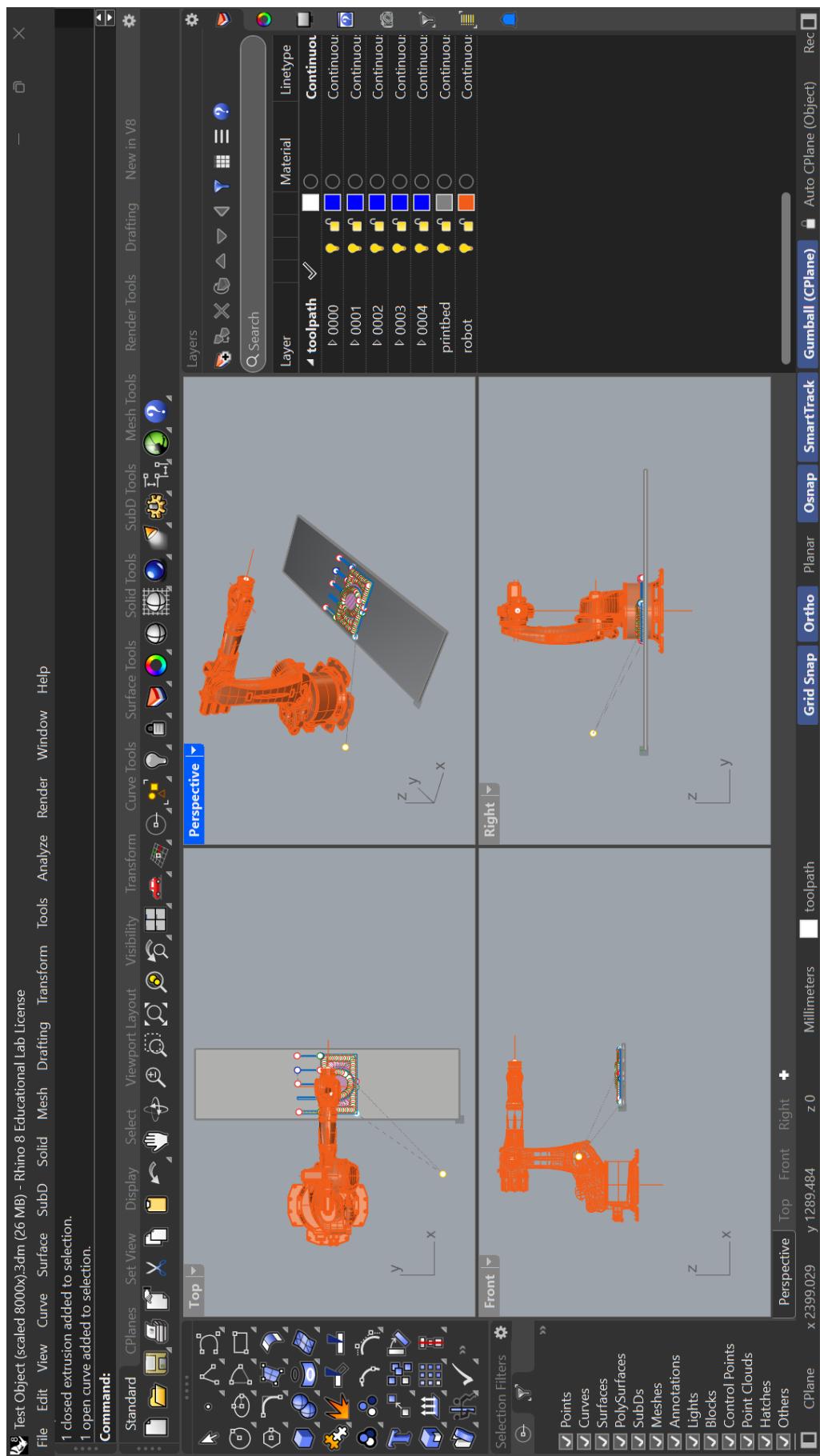


Figure 4.33 Screenshot of Rhinoceros3D Interface with Test Object Sliced and Visualized (beginning & end [yellow], start [green], stop [red])

4.10 Implementation of Safety Mechanisms

In addition to the safety features already described within the individual packages and modules, there are cross-system checks designed to ensure that no invalid files or outputs are generated. The following sections describe the most important functions in detail. The flowchart presented in Figures 4.35 to 4.37 provides an overview of the process flow within the *main*-function including all essential safety features.

4.10.1 Program Abort Errors [*main*-function]

In general, a complete termination of the program has been avoided wherever possible. However, there are four specific cases in which a controlled and defined abort of the program is executed.

- **Invalid input in *setup.json*:** To ensure that no erroneous KUKA code (.src file) is generated, the program terminates if a value in the *setup.json* file does not match the specified type defined in the corresponding "type" field.
- **File or directory not found:** If an error occurs while attempting to locate a specified file or folder the program is terminated, as no data would be available for further processing. This feature may be activated either due to a typographical error or a non-existent path.
- **Slicer or *.stl* file not found:** If either the slicer engine (for example, *CuraEngine.exe*) or the specified *.stl* file cannot be located at the given path, the program will abort. The process cannot proceed without these essential components.
- **No usable data in the *.gcode* file:** If the generated *.gcode* file does not contain any printable information after slicing, the program will be terminated. This may occur if the geometry of the object is too small relative to the defined print settings, such that no tool paths are generated.

4.10.2 Check Object Size [package: *g_code*; module: *fits_printbed*]

Since the *CuraEngine.exe* does not verify whether the object generated from the *.stl* file fits within the print bed defined in the *printer.def.json*, a separate validation is performed in the *fits_printbed* module. The goal is to ensure that the entire object lies within the specified build volume. Figure 4.34 illustrates the three possible scenarios that are evaluated in the following.

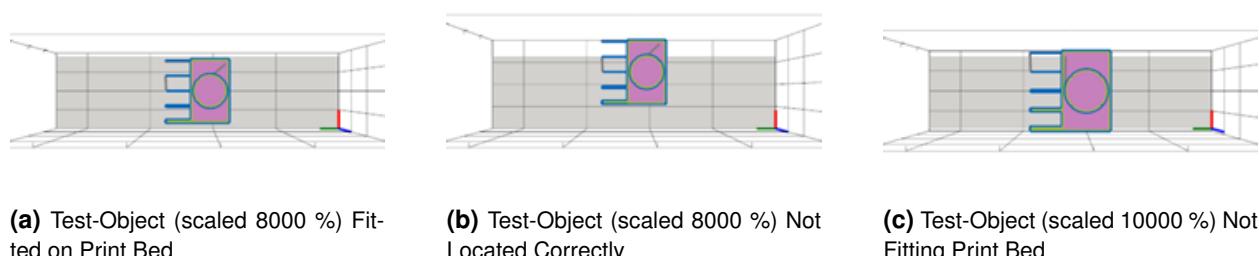


Figure 4.34 Test Object (Scaled) on Print Bed in Top View (Taken from Report)

The verification is based on the following input parameters:

- **bed_size:** (b_i) specifies the dimensions of the print bed in the *X*, *Y*, and *Z* directions.
- **min_values:** defines the minimum coordinates of the object.
- **max_values:** defines the maximum coordinates of the object.

First, the length of the object in each spatial direction X, Y, Z (hereinafter referred to as i) is calculated:

$$\Delta i = i_{\max} - i_{\min}, \quad (4.47)$$

The object only fits on the print bed if the following conditions are met in all spatial directions:

$$\Delta i \leq b_i \quad (4.48)$$

If any of these conditions are violated, the object is classified as too large.

Next, a position check is performed to verify whether the object lies within the limits of the print bed:

$$0 \leq i_{\min} \text{ and } i_{\max} \leq b_i \quad (4.49)$$

If these positioning conditions are not satisfied, a necessary shift is calculated. The required displacement Δp is determined for each spatial direction as:

$$\Delta p_i = \max(0, -i_{\min}) + \max(0, i_{\max} - b_i) \quad (4.50)$$

Full print approval is only granted if both the dimension check according to Equation (4.48) and the position check according to Equation (4.49) are successfully completed. If this is not the case, one of the two error codes shown in Listings 4.10.1 and 4.10.2 is issued.

Listing 4.10.1 Example of CLI Error Code for Badly Located Print (See Figure 4.34b)

```
[ERROR] Object not located fully on the printbed; shift at least
      {'dx': -207.5, 'dy': 0, 'dz': 0}
[WARNINg] .src file not generated due to invalid location of
          object      or object not fitting on printbed
```

Listing 4.10.2 Example of CLI Error Code for Objects Larger than Printbed (See Figure 4.34c)

```
[ERROR] Object doesn't fit printbed in given orientation
[WARNINg] .src file not generated due to invalid location of
          object      or object not fitting on printbed
```

4.10.3 Deactivate Outputs [*main-function*]

If the internal program switch "src" is set to "false" during execution (see Figure 4.35 to 4.38), the output of the .src file is suppressed. The following conditions can trigger this switch:

- **Invalid key in *setup.json*:** If a key specified within "start_code" or "end_code" does not exist in *setup.json*, the .src output will be skipped. Since the key could not be resolved, the .src file would be incomplete and corrupted.
- **Object not fully placed on the print bed:** If it is detected, as described in Chapter 4.10.2, that the object lies outside the defined boundaries of the print bed or is simply too large to fit the print bed, the generation of the .src file is also suppressed.
- **Kinematics check failed:** If it is determined that the joint angles for the defined start or end position are not within the limits specified under "joint_limits", or if the start position leads to a collision with the robot base, no .src file will be created. The same applies if no solution can be found for any of the positions extracted from the G-code using inverse kinematics.

This ensures that only robot control files are generated that have passed all relevant validation steps and meet all defined requirements.

In addition to the validation of input and output parameters, verifying the correctness of the program implementation itself is a crucial step. This topic is addressed in the following chapter.

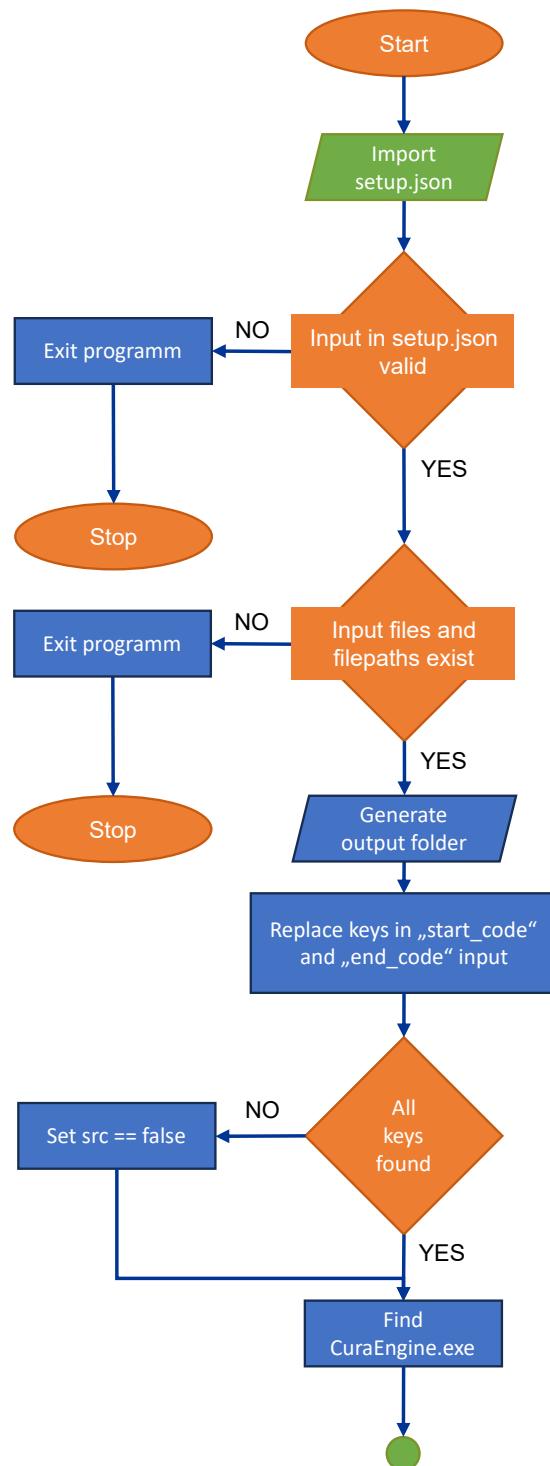
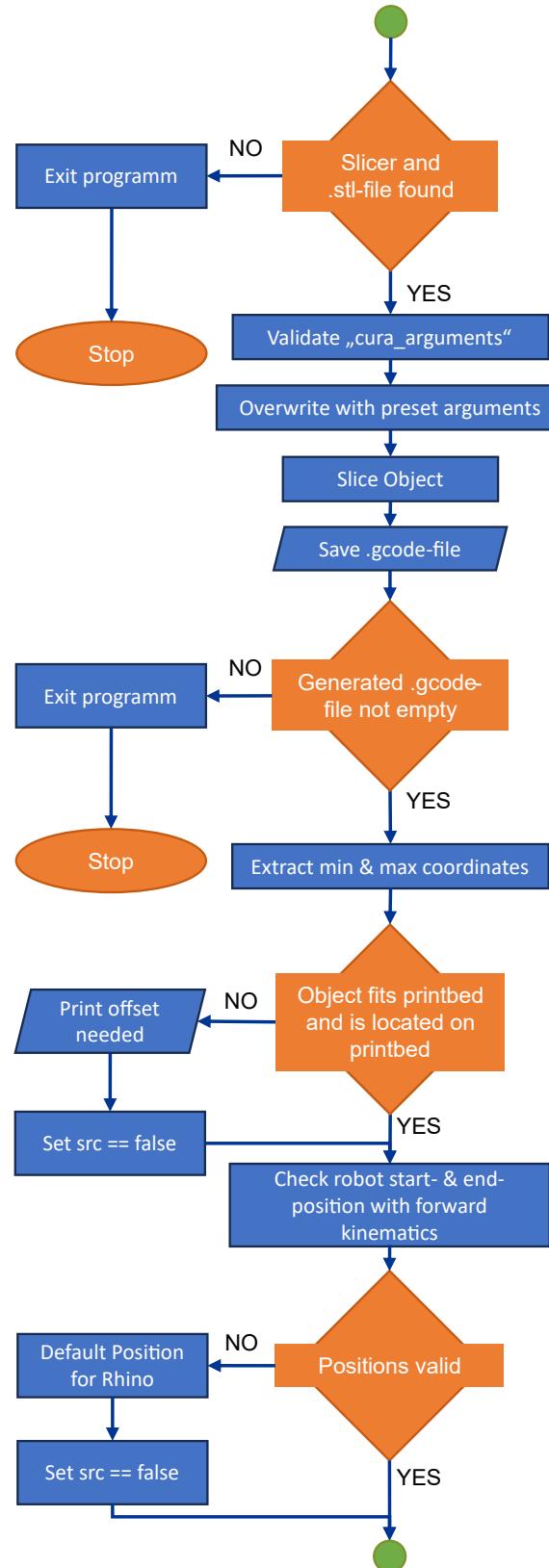


Figure 4.35 Flowchart of *main*-Function (Part 1)

Figure 4.36 Flowchart of *main*-Function (Part 2)

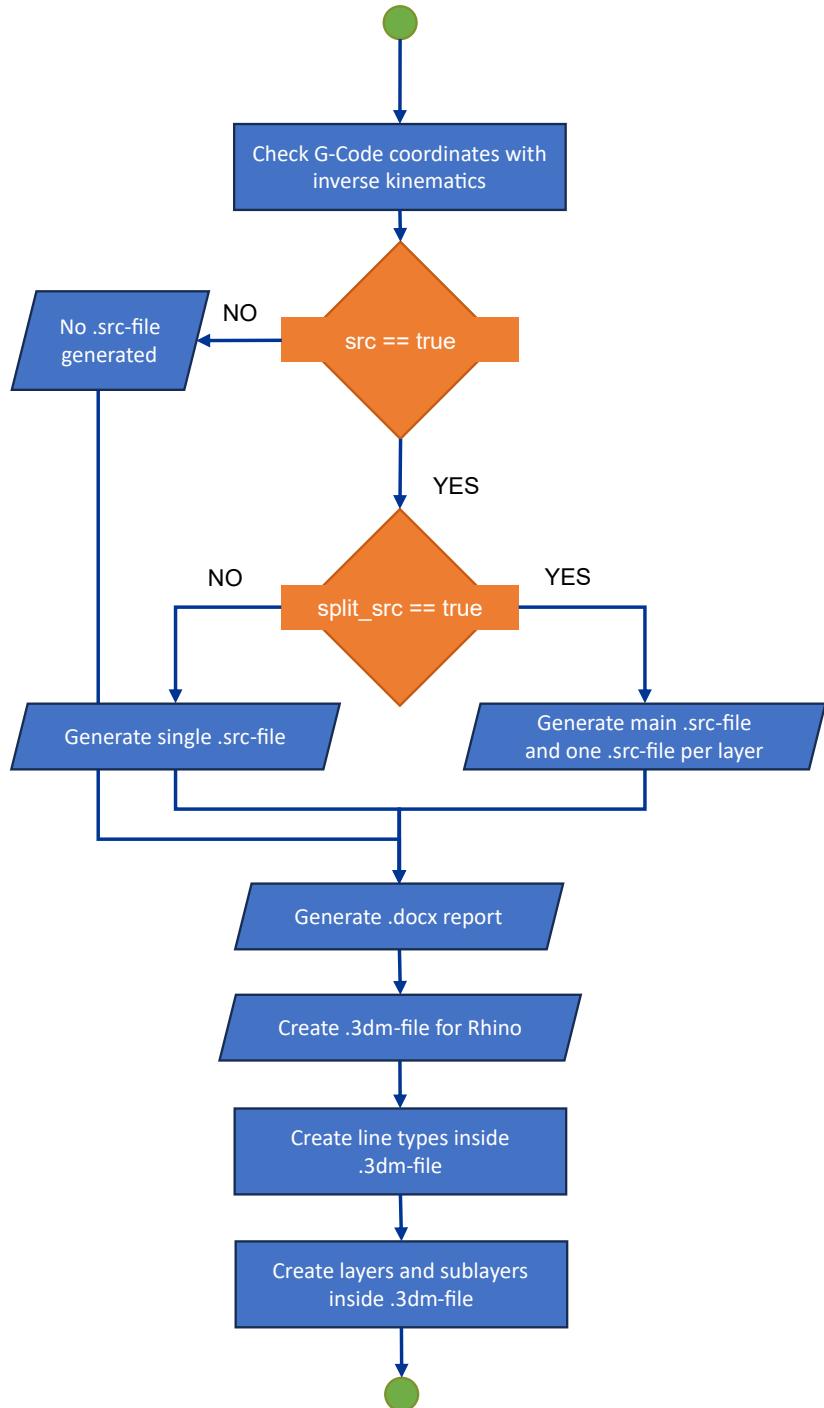


Figure 4.37 Flowchart of *main*-Function (Part 3)

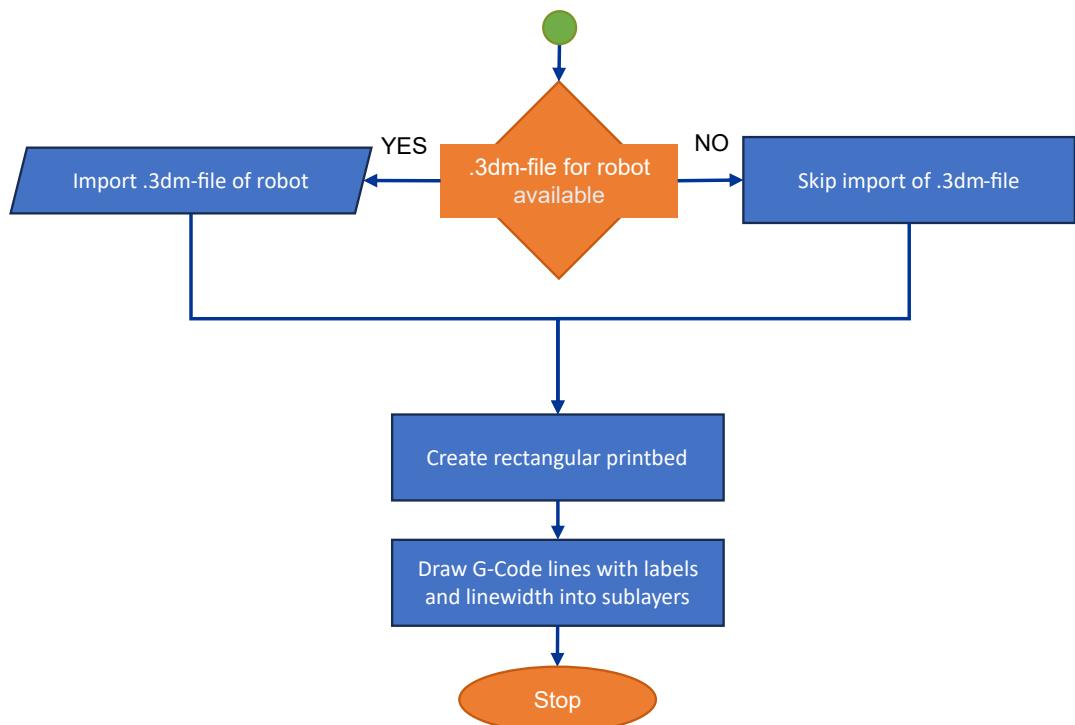


Figure 4.38 Flowchart of *main*-Function (Part 4)

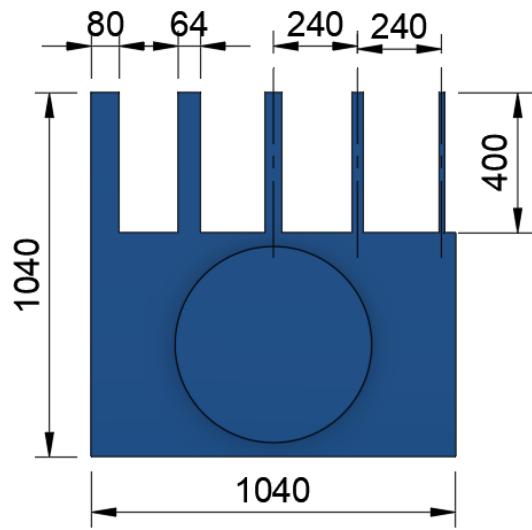
5 Validation and Limitations

5.1 Validation [package: tests]

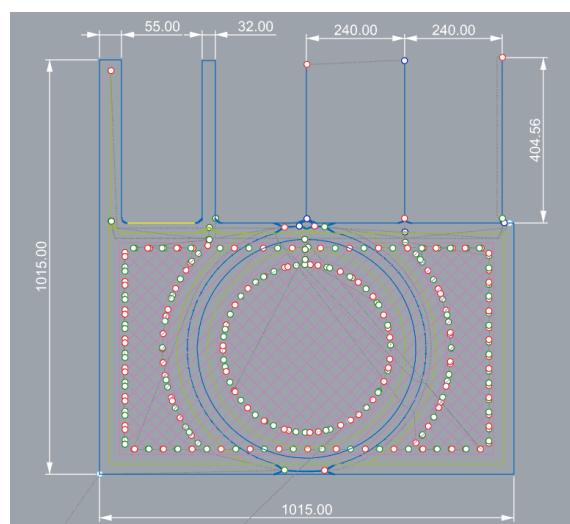
In addition to the actual implementation, validating the program is an essential step in the overall workflow. The goal is to identify and eliminate programming errors using either existing reference solutions or custom examples that have been verified for correctness. The program described in the previous chapters was therefore extensively tested during development, among other methods using the `pytest` library [70].

5.1.1 Validation of Rhino Output

To verify the correct transfer of geometry data, the output in Rhinoceros3D was compared with the dimensions specified in `printer.def.json`. Figure 5.1 shows a comparison between the measured dimensions of the test object in Autodesk Fusion 360 and in Rhinoceros3D after the slicing process. It should be noted that the dimensions measured in Rhino represent only the centerlines of the tool path and do not account for the extrusion width of 25 mm (or 32 mm for the second "finger" from the left).



(a) Top View of Test-Object with Measurements from Fusion 360 [11]



(b) Top View of Test-Object with Measurements from Rhinoceros3D [75]

Figure 5.1 Comparison of Measurements for Sliced Test-Object (Scaled 8000 %)

As the comparison shows, the distance of 240 mm between the "fingers" matches exactly. The overall dimensions of the object also correspond to the specifications when accounting for half the extrusion width of 25 mm and 32 mm respectively on each edge. This also applies to the sizing of the individual finger geometries when considering the line width. Only the length of the finger segments shows a minor deviation, which amounts to approximately 1 % compared to the original design geometry. Due to the very close match of all other dimensions, this deviation is attributed to the slicing process itself rather than to the subsequent processing within the program code.

5.1.2 Validation of Robot Kinematics

In addition to the visual inspection of the results in the Rhino file, the kinematic module developed in this work was subjected to comprehensive validation. For this purpose, approximately 23 000 different positions were generated using the RoboDK software [76] for the implemented model of the KUKA KR 340 R3300. For each position, both a forward and an inverse kinematic solution were calculated and stored in a corresponding `.json` file.

The validation process was carried out in two steps:

- **Forward kinematics:** Based on the given joint angles, the position and orientation of the end-effector were computed. The results were then compared to the reference values stored in the test dataset.
- **Inverse kinematics:** Starting from the predefined position and orientation of the end-effector, a homogeneous transformation matrix was constructed. Based on this matrix, the corresponding joint angles were calculated and compared to the list of solutions provided in the test file.

A test case is considered successful for the forward kinematic case, if all computed values match the reference solutions within a defined tolerance. For the inverse kinematic test all calculated solutions need to be clearly identifiable and need to match with the specified tolerance compared to the expected solution set. Table 5.1 provides a summary of the validation test results for the kinematic module.

Table 5.1 Summary of Validation Tests for the Kinematics Module

Test	Passed / Failed	Tolerance
Forward kinematic	23.196 / 4	$\pm 2 \times 10^{-2}$ [deg]
Inverse kinematic	23.200 / 0	$\pm 10^{-2}$ [mm] and [deg]

The failed test cases in the forward kinematics exclusively involve configurations in which the wrist pitch angle is exactly 90° . This configuration corresponds to the *Gimbal Lock* scenario illustrated in Figure 4.16b, in which no unique solution for the orientation exists. As a result, the computed solution does not match the reference result from RoboDK within the specified tolerance. Such singularities are extremely rare in practical operation and are typically handled internally by the robot's control software. Since the forward kinematics still returns a numerically valid result, the `.src` file is generated even in these cases. The issue is therefore confined to the test definition within the program and is not further addressed. All other test cases passed successfully within the defined tolerance of around 10^{-2} degree or millimeters, which is below the specified accuracy of the robot (see Chapter 2.6.1).

5.2 Limitations

In addition to validating the program's functionality, it is essential to understand its current boundaries and constraints in order to use the application effectively and safely. While several limitations have already been discussed throughout this work, the present chapter provides a concise overview of the most significant restrictions.

- **Validation of input arguments:** At present, only the parameters defined by the user in `setup.json` are validated according to their specified data types. The input values contained in the associated `.def.json` configuration files are not subject to validation. As a result, invalid entries in these files cannot neither be detected nor handled by the program.
- **Robot kinematics:** The application is currently restricted to 6-DOF robots based on the OPW architecture. Alternative joint configurations or other robot types are not supported by the present implementation.
- **Tool frame orientation:** The current version does not support relative rotations between `$TOOL` and `$NULLFRAME`. Consequently, tools must be mounted with an orientation that is parallel to the flange coordinate frame.
- **Singularities in forward kinematics:** For pitch angles (rotation about the Y -axis) of exactly 90° , a singularity occurs. While the computed results mostly agree with those produced by RoboDK, numerical inaccuracies may arise in this configuration. This special case affects only the start and end positions.
- **Singularities in inverse kinematics:** In addition to the already discussed *wrist singularity*, two other singularity types may arise according to [82]. The first is the *shoulder singularity*, which occurs when axes A1 and A6 become collinear. The second is the *elbow singularity*, which results from the alignment of links " c_2 " and " c_3 " (cf. Figure 4.11) in parallel. The former can be ruled out for the present configuration due to the robot's fixed orientation with respect to the print bed during operation. For the latter, further investigation is currently not required given the existing setup. However, should the arrangement of the robot or print bed change in the future, or should the software be adapted for other configurations, this limitation may need to be reconsidered.
- **Limitations in the selection of slicer arguments:** At present, the `CuraEngine.exe` only processes the most deeply nested parameters from the respective `.def.json` files. Higher-level parameters cannot be used, even though they are generally accepted and handled by the program as documented in [93] and mentioned in the Help section accessible via the `CuraEngine help` command in the command line interface.
- **Incomplete validation of slicer arguments:** The complete set of Cura parameters comprises more than 600 individual entries, many of which are internally cross-referenced or duplicated. It was not feasible within the scope of this work to validate all of these options. Therefor ongoing updates and validation can be documented using the Excel files provided in the `Cura` package of the [GitHub project](#).
- **Import of external .3dm files:** When importing external Rhino files, only specific geometry types are supported (Points, Curves, Lines, B-reps, and Meshes). If a file contains unsupported entity types, this may result in warnings or, in some cases, a program interruption.
- **Compatibility with Cura versions:** The program has been specifically developed for use with *Ultimaker Cura 5.8.0*. Using other versions of Cura (either earlier or later) may require adjustments to the program's structure.

- **Software requirement Rhinoceros3D:** During execution, the program accesses Rhinoceros3D in the background. Therefore, a locally installed version of Rhino 8 is required to run the application.
- **Geometric representation of the print path:** To minimize memory consumption and reduce runtime, the print path geometry is not rendered with full geometric fidelity as mentioned in Chapter 4.9 in Rhinoceros3D. As a result, the representation of layer height and the positioning of the printed object on the print bed may significantly deviate from the actual printed geometry.

6 Outlook

While the previous chapter focused on the current limitations of the implementation, this chapter explores potential enhancements and future developments. The aim is to identify opportunities that go beyond the limitations previously described and to outline strategies that could be beneficial for further optimization of the program.

This chapter is intended to serve as an impetus for new ideas and to present perspectives for additional functionalities and possibilities to expand the capabilities of the software.

6.1 Simple Extensions of the Program

Parallel Processes

To reduce program runtime, it would be feasible to parallelize specific sub-processes and distribute them across multiple CPU cores. This approach could be applied, for example, to the kinematic calculations or to the writing of the Rhino file, thereby significantly decreasing execution time for these modules.

Feedrate Input

As described in Chapter 4.4, lines containing only feedrate values are already extracted from the G-code. In future implementations, this information could be used in combination with the feedrate values included in the *general commands* to directly control the travel and print speeds from the G-code itself, rather than specifying it externally in *setup.json*.

6.2 Advanced Extensions of the Program

Calibration of Printing Parameters

In addition to specifying the printing parameters, proper calibration of the printer is essential for achieving consistently high-quality print results. The tests listed in [91] provide an overview of commonly used calibration prints for polymer-based FDM processes. Implementing similar calibration routines tailored to the specific setup described in this work could further enhance print quality and reproducibility. In the long term, an automated calibration procedure based on the available sensors and cameras within the robotic environment may be considered. Ultimately, the goal could be to dynamically and autonomously adjust printing parameters, thereby reducing the manual effort required for setup and monitoring of the process chain.

Graphical User Interface

To simplify input handling and reduce the likelihood of user errors, the implementation of a graphical user interface could be a useful enhancement. The structure of such an interface could follow the layout of Cura's native GUI. In addition to streamlining the input process, it would also significantly facilitate parameter validation compared to the current *.json*-based input format.

Simulation of the Printing Process Using Rhinoceros3D

At present, the printing process can only be followed manually by examining the layer-by-layer geometry rendered in the Rhino file. This visualization is limited to the gradual build-up of the printed object on the print bed.

Although the reachability of individual robot positions is already verified by computing each movement point-by-point, the results are currently used only for simple Boolean checks of reachability.

For future versions of the program, it would be highly beneficial to simulate the complete printing process, including all robotic motion sequences. Therefor it is crucial that the simulated trajectories precisely reflect the actual robotic movements, as any deviation could result in unrealistic or infeasible paths that would not be executable in the real system.

Such a detailed simulation would also enable the extraction of dynamic process variables, such as the actual speed and acceleration of the nozzle. These data could then be used to optimize the control of external components, like robot-pump-interaction by improving the synchronization between material flow and robotic motion.

As a simplified alternative, a so-called *dry-run* could be implemented. In this approach, the generated G-code would be executed without material extrusion in an initial run. During this process, telemetry data such as velocity, position, or acceleration could be recorded. These data could subsequently serve as input parameters for the actual print operation, thereby refining the control of hardware components and reducing process variability.

Non-Planar Slicing

Since the robotic system used is capable of executing complex spatial movements beyond the purely Cartesian motions confined to the X - Y plane of each print layer, more advanced tool path strategies could be implemented from a hardware perspective. On the software side, the so-called *non-planar slicing* method allows for generating print paths that are no longer limited to planar and parallel layer heights. Open-source tools such as the slicer presented in [111] or its more recent extension [15] could serve as the basis for such an implementation.

The main benefit of this approach lies in the ability to eliminate the need for support structures by conforming the printing path to the curvature of the object. This enables the fabrication of complex geometries and undercuts without additional scaffolding (e.g. support). Furthermore, non-planar slicing offers significant potential in the context of load-oriented design, making it a valuable alternative to the conventional layer-by-layer approach currently implemented.

7 Conclusion

The program presented in this work significantly improved the previous software-based workflow for concrete-3D-printing at the Technical University of Munich. By integrating all required process steps, starting with provision of a `.stl` file and ending in the visualization of the generated print path on the computer the sequence of operations is now considerably simplified for the user. Instead of manually executing individual steps of the process chain, the user is only required to input the necessary information once at the beginning of the workflow. All subsequent operations are executed automatically, while erroneous inputs are clearly identified and documented in a transparent manner. This not only improves the overall clarity of the process but also reduces potential sources of error during both setup-phase and execution.

Through the provided [documentation](#) on GitHub and the modular structure of the entire program, the prerequisites for future expansion, validation, and improvements by external users are established. Furthermore, any modifications to the software components, such as Rhinoceros3D or Ultimaker Cura can be implemented efficiently by targeting the relevant packages. The serial structure of the `main`-function also facilitates the straightforward integration of additional functionality.

Taken as a whole, the program provides a robust foundation for continued use in future applications. For reliable and flawless use in research or industrial contexts, however, the limitations outlined in Chapter 5.2 must either be resolved or carefully considered. Among the possible enhancements presented in Chapter 6, *non-planar slicing* appears particularly promising, as it would allow for a more advanced exploitation of geometric freedom in 3D-printing with mineral-based materials such as concrete.

A further development of the current software can therefore be regarded as both valuable and worthwhile.

A G-Code Listings

A.1 Prusa Slicer 2.9.2 – G-Code Listing

```
1 ; generated by PrusaSlicer 2.9.2 on 2025-04-14 at 12:06:16 UTC
2 ;
3 ;
4 ;
5 ; external perimeters extrusion width = 0.42mm
6 ; perimeters extrusion width = 0.44mm
7 ; infill extrusion width = 0.44mm
8 ; solid infill extrusion width = 0.44mm
9 ; top infill extrusion width = 0.40mm
10 ; first layer extrusion width = 0.42mm
11
12 M201 X500 Y500 Z100 E5000 ; sets maximum accelerations, mm/sec^2
13 M203 X500 Y500 Z10 E60 ; sets maximum feedrates, mm / sec
14 M204 S500 T1000 ; sets acceleration (S) and retract acceleration (R), mm/sec^2
15 M205 X8.00 Y8.00 Z0.40 E5.00 ; sets the jerk limits, mm/sec
16 M205 S0 T0 ; sets the minimum extruding and travel feed rate, mm/sec
17
18 ; printing object Slicer Test v3.stl id:0 copy 0
19 ; stop printing object Slicer Test v3.stl id:0 copy 0
20
21 ; TYPE: Custom
22 G90 ; use absolute coordinates
23 M83 ; extruder relative mode
24 M104 S150 ; set temporary nozzle temp to prevent oozing during homing
25 M140 S0 ; set final bed temp
26 G4 S30 ; allow partial nozzle warmup
27 G28 ; home all axis
28 G1 Z50 F240
29 G1 X2.0 Y10 F3000
30 M104 S20 ; set final nozzle temp
31 M190 S0 ; wait for bed temp to stabilize
32 M109 S20 ; wait for nozzle temp to stabilize
33 G1 Z0.28 F240
34 G92 E0
35 G1 X2.0 Y140 E10 F1500 ; prime the nozzle
36 G1 X2.3 Y140 F5000
37 G92 E0
38 G1 X2.3 Y10 E10 F1200 ; prime the nozzle
39 G92 E0
40 G21 ; set units to millimeters
41 G90 ; use absolute coordinates
42 M83 ; use relative distances for extrusion
```

Listing A.1.1 G-Code Generated Using Prusa Slicer [68] (Initialization)

```
43 ; Filament gcode
44 M107
45 ;LAYER_CHANGE
46 ;Z:0.2
47 ;HEIGHT:0.2
48 G1 E-5 F3600
49 G1 Z.2 F9000
50 ; printing object Slicer Test v3.stl id:0 copy 0
51 G1 X116.331 Y116.704
52 G1 Z.2
53 G1 E5 F2400
54 M204 S500
55 ; TYPE: External perimeter
56 ;WIDTH:0.592344
57 G1 F1200
58 G1 X116.055 Y116.724 E.01721
59 ;WIDTH:0.552764
60 G1 X115.779 Y116.743 E.01596
61 ;WIDTH:0.555403
62 G1 X115.709 Y116.763 E.00422
63 ;WIDTH:0.597621
```

```

64 G1 X115.639 Y116.783 E.00457
65 ;WIDTH:0.63984
66 G1 X115.57 Y116.802 E.00483
67 ;WIDTH:0.682058
68 G1 X115.5 Y116.822 E.00527
69 G1 E-3.5 F3600
70 ;WIPE_START
71 G1 F7200
72 G1 X115.57 Y116.802 E-.03458
73 G1 X115.639 Y116.783 E-.03399
74 G1 X115.709 Y116.763 E-.03458
75 G1 X115.779 Y116.743 E-.03458
76 G1 X116.055 Y116.724 E-.13141
77 G1 X116.331 Y116.704 E-.13144
78 ;WIPE_END
79 G1 E-1.09942 F3600
80 G1 X115.5 Y121.481 F9000
81 G1 E5 F2400
82 ;WIDTH:0.6
83 G1 F1200
84 G1 X115.5 Y116.979 E.28384
85 ;WIDTH:0.641029
86 G1 X115.5 Y116.9 E.00535
87 ;WIDTH:0.682058
88 G1 X115.5 Y116.822 E.00564
89 G1 X115.407 Y116.792 E.00707
90 ;WIDTH:0.633492
91 G1 X115.314 Y116.762 E.00653
92 ;WIDTH:0.584925
93 G1 X115.221 Y116.732 E.00599
94 ;WIDTH:0.568366
95 G1 X114.788 Y116.716 E.02577
96 G1 E-3.5 F3600
97 ;WIPE_START
98 G1 F7200
99 G1 X115.221 Y116.732 E-.20582
100 G1 X115.314 Y116.762 E-.04642
101 G1 X115.407 Y116.792 E-.04642
102 G1 X115.5 Y116.822 E-.04642
103 G1 X115.5 Y116.9 E-.03705
104 G1 X115.5 Y116.979 E-.03752
105 G1 X115.5 Y119.253 E-1.08035
106 ;WIPE_END
107 G1 X118.5 Y121.488 F9000
108 G1 E5 F2400
109 ;WIDTH:0.400002
110 G1 F1200
111 G1 X118.5 Y116.981 E.18214
112 G1 E-3.5 F3600
113 ;WIPE_START
114 G1 F7200
115 G1 X118.5 Y120.139 E-1.5
116 ;WIPE_END
117 G1 X121.5 Y121.495 F9000
118 G1 E5 F2400
119 ;WIDTH:0.38292
120 G1 F1200
121 G1 X121.5 Y116.979 E.17378
122 G1 E-3.5 F3600
123 ;WIPE_START
124 G1 F7200
125 G1 X121.5 Y120.137 E-1.5
126 ;WIPE_END
127 G1 X114.813 Y109.281 F9000
128 G1 E5 F2400
129 ;WIDTH:0.562992
130 G1 F1200
131 G1 X115.072 Y109.267 E.01527
132 ;WIDTH:0.534713
133 G1 X115.331 Y109.253 E.01444
134 ;WIDTH:0.536148
135 G1 X116.013 Y109.268 E.03808
136 ;WIDTH:0.542128
137 G1 X116.057 Y109.271 E.00249
138 G1 E-3.5 F3600
139 ;WIPE_START
140 G1 F7200
141 G1 X116.013 Y109.268 E-.02095
142 G1 X115.331 Y109.253 E-.32403
143 G1 X115.072 Y109.267 E-.1232
144 G1 X114.813 Y109.281 E-.1232
145 ;WIPE_END

```

```

146 G1 E-.90862 F3600
147 G1 X112.959 Y109.682 F9000
148 G1 E5 F2400
149 ; TYPE: Perimeter
150 ;WIDTH:0.419999
151 G1 F1200
152 G1 X112.363 Y110.23 E.03455
153 G1 X112.004 Y110.698 E.02517
154 G1 X111.736 Y111.229 E.02538
155 G1 X111.516 Y111.724 E.02312
156 G1 X111.325 Y112.705 E.04265
157 G1 X111.332 Y113.387 E.02911
158 G1 X111.458 Y114.08 E.03006
159 G1 X111.758 Y114.86 E.03566
160 G1 X111.911 Y115.137 E.0135
161 G1 X112.204 Y115.58 E.02267
162 G1 X112.68 Y116.089 E.02974
163 G1 X113.085 Y116.393 E.02161
164 G1 X112.64 Y116.465 E.01924
165 G1 X112.499 Y116.565 E.00738
166 G1 X112.313 Y116.448 E.00938
167 G1 X111.263 Y116.318 E.04515
168 G1 X109.881 Y116.313 E.05898
169 G1 X109.687 Y116.506 E.01168
170 G1 X109.687 Y109.686 E.29105
171 G1 X112.899 Y109.682 E.13708
172 G1 X113.119 Y110.034 F9000
173 ; TYPE: External perimeter
174 G1 F1200
175 G1 X112.662 Y110.459 E.02663
176 G1 X112.284 Y110.96 E.02678
177 G1 X111.885 Y111.8 E.03969
178 G1 X111.702 Y112.705 E.0394
179 G1 X111.704 Y113.326 E.0265
180 G1 X111.81 Y113.945 E.0268
181 G1 X112.145 Y114.805 E.03939
182 G1 X112.483 Y115.327 E.02654
183 G1 X112.906 Y115.787 E.02667
184 G1 X113.379 Y116.142 E.02524
185 G1 X113.694 Y116.304 E.01512
186 G1 X114.325 Y116.513 E.02837
187 ;WIDTH:0.469455
188 G1 X114.479 Y116.581 E.00813
189 ;WIDTH:0.518911
190 G1 X114.634 Y116.648 E.0091
191 ;WIDTH:0.568366
192 G1 X114.788 Y116.716 E.01001
193 G1 X114.588 Y116.739 E.01197
194 ;WIDTH:0.518911
195 G1 X114.388 Y116.762 E.01085
196 ;WIDTH:0.469455
197 G1 X114.188 Y116.786 E.00972
198 ;WIDTH:0.419999
199 G1 X113.291 Y116.753 E.03831
200 G1 X112.789 Y116.812 E.02157
201 G1 X112.689 Y116.979 E.00831
202 G1 X112.689 Y121.474 E.19183
203 G1 X112.658 Y121.719 E.01054
204 G1 X112.342 Y121.719 E.01349
205 G1 X112.311 Y121.486 E.01003
206 G1 X112.311 Y116.98 E.1923
207 G1 X112.259 Y116.85 E.00598
208 G1 X112.144 Y116.793 E.00548
209 G1 X111.261 Y116.695 E.03791
210 G1 X109.879 Y116.69 E.05898
211 ;WIDTH:0.42146
212 G1 X109.689 Y116.878 E.01145
213 G1 X109.689 Y121.689 E.20611
214 G1 X109.311 Y121.689 E.01619
215 G1 X109.31 Y116.118 E.23867
216 ;WIDTH:0.419999
217 G1 X109.31 Y109.31 E.29054
218 G1 X113.012 Y109.302 E.15799
219 ;WIDTH:0.446963
220 G1 X113.527 Y109.284 E.02356
221 G1 X114.025 Y109.21 E.02302
222 ;WIDTH:0.467444
223 G1 X114.287 Y109.234 E.01264
224 ;WIDTH:0.515218
225 G1 X114.55 Y109.258 E.01412
226 ;WIDTH:0.562992
227 G1 X114.813 Y109.281 E.01554

```

```

228 G1 X114.783 Y109.295 E.00195
229 ;WIDTH:0.551372
230 G1 X114.403 Y109.432 E.02324
231 ;WIDTH:0.515765
232 G1 X114.023 Y109.57 E.02164
233 ;WIDTH:0.480157
234 G1 X113.642 Y109.708 E.02005
235 ;WIDTH:0.444549
236 G1 X113.169 Y110.003 E.02534
237 G1 E-3.5 F3600
238 ;WIPE_START
239 G1 F7200
240 G1 X113.119 Y110.034 E.-02794
241 G1 X112.662 Y110.459 E.-29644
242 G1 X112.284 Y110.96 E.-29811
243 G1 X111.885 Y111.8 E.-44172
244 G1 X111.703 Y112.699 E.-43579
245 ;WIPE_END
246 G1 X118.018 Y109.669 F9000
247 G1 E5 F2400
248 ; TYPE: Perimeter
249 ;WIDTH:0.419999
250 G1 F1200
251 G1 X118.499 Y109.686 E.02054
252 G1 X121.313 Y109.687 E.12009
253 G1 X121.319 Y115.532 E.24944
254 G1 X121.395 Y116.406 E.03744
255 G1 X120.972 Y116.347 E.01823
256 G1 X120.202 Y116.319 E.03288
257 G1 X119.065 Y116.352 E.04854
258 G1 X118.451 Y116.43 E.02641
259 G1 X117.953 Y116.376 E.02138
260 G1 X118.521 Y115.896 E.03174
261 G1 X118.899 Y115.443 E.02518
262 G1 X119.423 Y114.46 E.04754
263 G1 X119.61 Y113.78 E.0301
264 G1 X119.68 Y112.783 E.04265
265 G1 X119.589 Y112.107 E.02911
266 G1 X119.423 Y111.541 E.02517
267 G1 X119.29 Y111.267 E.013
268 G1 X118.899 Y110.558 E.03455
269 G1 X118.451 Y110.033 E.02945
270 G1 X118.064 Y109.707 E.02159
271 G1 X117.981 Y110.133 F9000
272 ; TYPE: External perimeter
273 G1 F1200
274 G1 X117.719 Y109.917 E.01449
275 G1 X117.167 Y109.643 E.0263
276 G1 X116.602 Y109.458 E.02537
277 ;WIDTH:0.460709
278 G1 X116.42 Y109.396 E.00909
279 ;WIDTH:0.501419
280 G1 X116.239 Y109.333 E.00995
281 ;WIDTH:0.542128
282 G1 X116.057 Y109.271 E.01086
283 G1 X116.258 Y109.251 E.01141
284 ;WIDTH:0.501419
285 G1 X116.459 Y109.23 E.01049
286 ;WIDTH:0.460709
287 G1 X116.659 Y109.21 E.0095
288 ;WIDTH:0.419999
289 G1 X118.5 Y109.309 E.07868
290 G1 X121.69 Y109.31 E.13614
291 G1 X121.696 Y115.511 E.26464
292 G1 X121.725 Y116.027 E.02206
293 G1 X121.79 Y116.531 E.02169
294 G1 X121.79 Y116.79 E.01105
295 G1 X121.558 Y116.791 E.0099
296 G1 X121.5 Y116.979 E.0084
297 G1 X121.449 Y116.795 E.00815
298 G1 X120.959 Y116.723 E.02114
299 G1 X120.043 Y116.691 E.03912
300 G1 X119.141 Y116.723 E.03852
301 G1 X118.636 Y116.798 E.02179
302 G1 X118.5 Y116.981 E.00973
303 ;WIDTH:0.419171
304 G1 X118.401 Y116.81 E.00841
305 ;WIDTH:0.419999
306 G1 X117.797 Y116.741 E.02594
307 G1 X116.798 Y116.79 E.04269
308 ;WIDTH:0.459296
309 G1 X116.69 Y116.77 E.00518

```

```

310 ;WIDTH:0.498593
311 G1 X116.582 Y116.75 E.00566
312 ;WIDTH:0.53789
313 G1 X116.474 Y116.731 E.00614
314 ;WIDTH:0.577187
315 G1 X116.366 Y116.711 E.00664
316 G1 X116.459 Y116.656 E.00653
317 ;WIDTH:0.53789
318 G1 X116.553 Y116.6 E.00613
319 ;WIDTH:0.498593
320 G1 X116.647 Y116.545 E.00562
321 ;WIDTH:0.459296
322 G1 X116.74 Y116.489 E.00512
323 ;WIDTH:0.419999
324 G1 X117.195 Y116.346 E.02035
325 G1 X117.755 Y116.058 E.02687
326 G1 X118.231 Y115.655 E.02662
327 G1 X118.63 Y115.17 E.0268
328 G1 X119.061 Y114.353 E.03942
329 G1 X119.234 Y113.75 E.02677
330 G1 X119.306 Y112.829 E.03943
331 G1 X119.227 Y112.213 E.0265
332 G1 X119.061 Y111.647 E.02517
333 G1 X118.609 Y110.799 E.04101
334 G1 X118.128 Y110.254 E.03102
335 G1 X118.027 Y110.171 E.00558
336 G1 E-3.5 F3600
337 ;WIPE_START
338 G1 F7200
339 G1 X117.981 Y110.133 E.-0.02834
340 G1 X117.719 Y109.917 E.-16129
341 G1 X117.167 Y109.643 E.-29272
342 G1 X116.602 Y109.458 E.-2824
343 G1 X116.42 Y109.396 E.-09133
344 G1 X116.239 Y109.333 E.-09103
345 G1 X116.057 Y109.271 E.-09133
346 G1 X116.258 Y109.251 E.-09595
347 G1 X116.459 Y109.23 E.-09599
348 G1 X116.659 Y109.21 E.-09547
349 G1 X117.025 Y109.23 E.-17415
350 ;WIPE_END
351 G1 X121.194 Y110.575 F9000
352 G1 E5 F2400
353 ;TYPE: Solid infill
354 ;WIDTH:0.422094
355 G1 F1200
356 G1 X120.596 Y109.977 E.03629
357 G1 X120.059 Y109.977 E.02304
358 G1 X121.024 Y110.942 E.05857
359 G1 X121.025 Y111.478 E.023
360 G1 X119.523 Y109.977 E.09112
361 G1 X118.987 Y109.977 E.023
362 G1 X121.025 Y112.015 E.12368
363 G1 X121.025 Y112.552 E.02304
364 G1 X119.481 Y111.007 E.09373
365 G1 X119.64 Y111.306 E.01453
366 G1 X119.824 Y111.887 E.02615
367 G1 X121.026 Y113.089 E.07295
368 G1 X121.026 Y113.625 E.023
369 G1 X119.934 Y112.533 E.06627
370 G1 X119.975 Y112.849 E.01367
371 G1 X119.955 Y113.09 E.01038
372 G1 X121.027 Y114.162 E.06506
373 G1 X121.027 Y114.699 E.02304
374 G1 X119.914 Y113.586 E.06755
375 G1 X119.89 Y113.882 E.01274
376 G1 X119.842 Y114.05 E.0075
377 G1 X121.028 Y115.235 E.07195
378 G1 X121.047 Y115.791 E.02387
379 G1 X119.723 Y114.467 E.08035
380 G1 X119.638 Y114.698 E.01056
381 G1 X119.562 Y114.842 E.00699
382 G1 X120.769 Y116.049 E.07325
383 G1 X120.214 Y116.03 E.02383
384 G1 X119.375 Y115.191 E.05092
385 G1 X119.184 Y115.537 E.01696
386 G1 X119.683 Y116.036 E.03028
387 G1 X119.166 Y116.055 E.0222
388 G1 X118.828 Y115.716 E.02054
389 G1 E-3.5 F3600
390 ;WIPE_START
391 G1 F7200

```

```

392 | G1 X119.166 Y116.055 E-.22739
393 | G1 X119.683 Y116.036 E-.24574
394 | G1 X119.184 Y115.537 E-.3352
395 | G1 X119.375 Y115.191 E-.18773
396 | G1 X120.125 Y115.941 E-.50394
397 | ;WIPE_END
398 | G1 X112.225 Y110.224 F9000
399 | G1 E5 F2400
400 | ;WIDTH:0.427467
401 | G1 F1200
402 | G1 X111.973 Y109.972 E.01551
403 | G1 X111.43 Y109.973 E.02363
404 | G1 X111.864 Y110.407 E.02671
405 | G1 X111.771 Y110.525 E.00654
406 | G1 X111.65 Y110.737 E.01062
407 | G1 X110.887 Y109.974 E.04696
408 | G1 X110.345 Y109.975 E.02359
409 | G1 X111.474 Y111.104 E.06949
410 | G1 X111.297 Y111.471 E.01773
411 | G1 X109.977 Y110.151 E.08124
412 | G1 X109.977 Y110.695 E.02368
413 | G1 X111.183 Y111.901 E.07423
414 | G1 X111.09 Y112.351 E.02
415 | G1 X109.977 Y111.239 E.06847
416 | G1 X109.977 Y111.783 E.02368
417 | G1 X111.044 Y112.849 E.06564
418 | G1 X111.034 Y113.302 E.01972
419 | G1 X111.049 Y113.398 E.00423
420 | G1 X109.977 Y112.327 E.06595
421 | G1 X109.977 Y112.871 E.02368
422 | G1 X111.147 Y114.04 E.07198
423 | G1 X111.465 Y114.902 E.03999
424 | G1 X109.977 Y113.414 E.09159
425 | G1 X109.977 Y113.958 E.02368
426 | G1 X112.14 Y116.121 E.13313
427 | G1 X111.531 Y116.056 E.02666
428 | G1 X109.977 Y114.502 E.09565
429 | G1 X109.977 Y115.046 E.02368
430 | G1 X110.959 Y116.028 E.06044
431 | G1 X110.416 Y116.028 E.02363
432 | G1 X109.808 Y115.42 E.03742
433 | ;LAYER_CHANGE
434 | ;Z:0.4
435 | ;HEIGHT:0.2
436 | ;BEFORE_LAYER_CHANGE
437 | G92 E0
438 | ;0.4
439
440
441 | G1 E-3.5 F3600
442 | ;WIPE_START
443 | G1 F7200
444 | G1 X110.416 Y116.028 E-.40842
445 | G1 X110.959 Y116.028 E-.25793
446 | G1 X109.977 Y115.046 E-.65966
447 | G1 X109.977 Y114.68 E-.17399
448 | ;WIPE_END
449 | G1 Z.4 F9000
450 | ;AFTER_LAYER_CHANGE
451 | ;0.4
452 | G1 Z.4
453 | G1 X115.779 Y116.743
454 | G1 E5 F2400
455 | ; TYPE: External perimeter
456 | ;WIDTH:0.554771
457 | G1 F1500
458 | G1 X115.709 Y116.764 E.00423
459 | ;WIDTH:0.595398
460 | G1 X115.639 Y116.785 E.00457
461 | ;WIDTH:0.636025
462 | G1 X115.57 Y116.806 E.00484
463 | ;WIDTH:0.676652
464 | G1 X115.5 Y116.827 E.00524
465 | G1 E-3.5 F3600
466 | ;WIPE_START
467 | G1 F7200
468 | G1 X115.57 Y116.806 E-.03471
469 | G1 X115.639 Y116.785 E-.03426
470 | G1 X115.709 Y116.764 E-.03471
471 | G1 X115.779 Y116.743 E-.03471
472 | ;WIPE_END
473 | G1 E-1.36161 F3600

```

```

474 G1 X115.5 Y121.7 F9000
475 G1 E5 F2400
476 ;WIDTH:0.638326
477 G1 F1500
478 G1 X115.5 Y119.264 E.16415
479 ;WIDTH:0.676652
480 G1 X115.5 Y116.827 E.17479
481 G1 X115.399 Y116.805 E.00741
482 ;WIDTH:0.638207
483 G1 X115.297 Y116.783 E.00703
484 ;WIDTH:0.599761
485 G1 X115.196 Y116.761 E.00651
486 ;WIDTH:0.561316
487 G1 X115.095 Y116.739 E.00606
488 G1 E-3.5 F3600
489 ;WIPE_START
490 G1 F7200
491 G1 X115.196 Y116.761 E-.0491
492 G1 X115.297 Y116.783 E-.0491
493 G1 X115.399 Y116.805 E-.04956
494 G1 X115.5 Y116.827 E-.0491
495 G1 X115.5 Y119.264 E-1.15757
496 G1 X115.5 Y119.57 E-.14557
497 ;WIPE_END
498 G1 X118.5 Y121.8 F9000
499 G1 E5 F2400
500 ;WIDTH:0.4
501 G1 F1500
502 G1 X118.5 Y116.979 E.19483
503 G1 E-3.5 F3600
504 ;WIPE_START
505 G1 F7200
506 G1 X118.5 Y120.137 E-1.5
507 ;WIPE_END
508 G1 X121.5 Y121.9 F9000
509 G1 E5 F2400
510 ;WIDTH:0.38292
511 G1 F1500
512 G1 X121.5 Y116.979 E.18936
513 G1 E-3.5 F3600
514 ;WIPE_START
515 G1 F7200
516 G1 X121.5 Y120.137 E-1.5
517 ;WIPE_END
518 G1 X114.526 Y109.314 F9000
519 G1 E5 F2400
520 ;WIDTH:0.627586
521 G1 F1500
522 G1 X114.742 Y109.295 E.01435
523 ;WIDTH:0.590813
524 G1 X114.958 Y109.277 E.01344
525 ;WIDTH:0.55404
526 G1 X115.173 Y109.259 E.01248
527 ;WIDTH:0.535964
528 G1 X116.008 Y109.268 E.0466
529 ;WIDTH:0.542126
530 G1 X116.057 Y109.271 E.00277
531 G1 E-3.5 F3600
532 ;WIPE_START
533 G1 F7200
534 G1 X116.008 Y109.268 E-.02332
535 G1 X115.173 Y109.259 E-.39665
536 G1 X114.958 Y109.277 E-.10248
537 G1 X114.742 Y109.295 E-.10296
538 G1 X114.526 Y109.314 E-.103
539 ;WIPE_END
540 G1 E-.77159 F3600
541 G1 X113.244 Y109.597 F9000
542 G1 E5 F2400
543 ;TYPE: Perimeter
544 ;WIDTH:0.439999
545 G1 F1500
546 G1 X113.254 Y109.597 E.00045
547 G1 X112.974 Y109.78 E.01503
548 G1 X112.418 Y110.306 E.0344
549 G1 X111.96 Y110.948 E.03544
550 G1 X111.601 Y111.754 E.03965
551 G1 X111.415 Y112.713 E.0439
552 G1 X111.422 Y113.381 E.03002
553 G1 X111.546 Y114.06 E.03102
554 G1 X111.918 Y114.983 E.04472
555 G1 X112.44 Y115.718 E.04051

```

```

556 G1 X112.905 Y116.167 E.02905
557 G1 X113.239 Y116.403 E.01838
558 G1 X112.931 Y116.403 E.01384
559 G1 X112.629 Y116.46 E.01381
560 G1 X112.5 Y116.549 E.00704
561 G1 X112.371 Y116.46 E.00704
562 G1 X112.121 Y116.403 E.01152
563 G1 X109.979 Y116.403 E.09626
564 G1 X109.597 Y116.784 E.02425
565 G1 X109.597 Y109.597 E.32298
566 G1 X113.184 Y109.597 E.1612
567 G1 X113.235 Y110.065 F9000
568 ; TYPE: External perimeter
569 ;WIDTH:0.419999
570 G1 F1500
571 G1 X112.733 Y110.531 E.02923
572 G1 X112.257 Y111.205 E.03521
573 G1 X111.98 Y111.833 E.02929
574 G1 X111.802 Y112.713 E.03832
575 G1 X111.804 Y113.318 E.02582
576 G1 X111.907 Y113.921 E.02611
577 G1 X112.237 Y114.763 E.03859
578 G1 X112.709 Y115.44 E.03522
579 G1 X113.2 Y115.91 E.02901
580 G1 X113.708 Y116.247 E.02602
581 G1 X114.471 Y116.562 E.03523
582 ;WIDTH:0.45429
583 G1 X114.679 Y116.621 E.01007
584 ;WIDTH:0.48858
585 G1 X114.887 Y116.68 E.01091
586 ;WIDTH:0.52287
587 G1 X115.095 Y116.739 E.01174
588 G1 X114.88 Y116.756 E.01172
589 ;WIDTH:0.48858
590 G1 X114.664 Y116.773 E.01093
591 ;WIDTH:0.45429
592 G1 X114.449 Y116.79 E.01004
593 ;WIDTH:0.419999
594 G1 X112.797 Y116.809 E.07051
595 ;WIDTH:0.42146
596 G1 X112.689 Y116.979 E.00863
597 G1 X112.689 Y121.789 E.20607
598 G1 X112.311 Y121.789 E.01619
599 G1 X112.311 Y116.979 E.20607
600 G1 X112.252 Y116.843 E.00635
601 ;WIDTH:0.419999
602 G1 X112.121 Y116.79 E.00603
603 G1 X109.979 Y116.79 E.09141
604 ;WIDTH:0.45382
605 G1 X109.899 Y116.853 E.00474
606 ;WIDTH:0.48764
607 G1 X109.819 Y116.916 E.00513
608 ;WIDTH:0.52146
609 G1 X109.739 Y116.979 E.00551
610 G1 X109.739 Y121.739 E.2578
611 G1 X109.261 Y121.739 E.02589
612 G1 X109.261 Y116.979 E.2578
613 G1 X109.244 Y116.66 E.0173
614 ;WIDTH:0.48764
615 G1 X109.227 Y116.34 E.01613
616 ;WIDTH:0.45382
617 G1 X109.21 Y116.021 E.01486
618 ;WIDTH:0.419999
619 G1 X109.21 Y109.21 E.29067
620 G1 X113.902 Y109.21 E.20024
621 ;WIDTH:0.461517
622 G1 X114.027 Y109.231 E.006
623 ;WIDTH:0.503034
624 G1 X114.151 Y109.252 E.00655
625 ;WIDTH:0.544552
626 G1 X114.276 Y109.272 E.00719
627 ;WIDTH:0.586069
628 G1 X114.401 Y109.293 E.00779
629 ;WIDTH:0.627586
630 G1 X114.526 Y109.314 E.00839
631 G1 X114.482 Y109.339 E.00335
632 ;WIDTH:0.600825
633 G1 X114.346 Y109.413 E.00978
634 ;WIDTH:0.555619
635 G1 X114.211 Y109.487 E.00893
636 ;WIDTH:0.510412
637 G1 X114.075 Y109.561 E.00819

```

```

638 ;WIDTH:0.465206
639 G1 X113.94 Y109.635 E.00736
640 ;WIDTH:0.419999
641 G1 X113.286 Y110.034 E.03269
642 G1 E-3.5 F3600
643 ;WIPE_START
644 G1 F7200
645 G1 X113.235 Y110.065 E-.02835
646 G1 X112.733 Y110.531 E-.32535
647 G1 X112.257 Y111.205 E-.39194
648 G1 X111.98 Y111.833 E-.32603
649 G1 X111.802 Y112.713 E-.42647
650 G1 X111.802 Y112.717 E-.00186
651 ;WIPE_END
652 G1 X117.768 Y109.597 F9000
653 G1 E5 F2400
654 ; TYPE: Perimeter
655 ;WIDTH:0.439999
656 G1 F1500
657 G1 X121.403 Y109.597 E.16336
658 G1 X121.403 Y110.753 E.05195
659 G1 X121.403 Y116.403 E.25391
660 G1 X117.766 Y116.403 E.16345
661 G1 X118.188 Y116.063 E.02435
662 G1 X118.626 Y115.642 E.0273
663 G1 X119.056 Y115.031 E.03358
664 G1 X119.339 Y114.426 E.03002
665 G1 X119.519 Y113.787 E.02983
666 G1 X119.584 Y113.289 E.02257
667 G1 X119.59 Y112.786 E.02261
668 G1 X119.494 Y112.103 E.031
669 G1 X119.169 Y111.181 E.04393
670 G1 X118.81 Y110.592 E.031
671 G1 X118.233 Y109.953 E.03869
672 G1 X117.816 Y109.634 E.02359
673 G1 X116.684 Y109.495 F9000
674 ; TYPE: External perimeter
675 ;WIDTH:0.419999
676 G1 F1500
677 G1 X116.603 Y109.458 E.0038
678 ;WIDTH:0.460708
679 G1 X116.421 Y109.396 E.00909
680 ;WIDTH:0.501417
681 G1 X116.239 Y109.333 E.00999
682 ;WIDTH:0.542126
683 G1 X116.057 Y109.271 E.01086
684 G1 X116.248 Y109.251 E.01085
685 ;WIDTH:0.501417
686 G1 X116.44 Y109.23 E.01002
687 ;WIDTH:0.460708
688 G1 X116.631 Y109.21 E.00908
689 ;WIDTH:0.419999
690 G1 X121.79 Y109.21 E.22017
691 G1 X121.79 Y110.753 E.06585
692 G1 X121.79 Y116.79 E.25764
693 G1 X121.548 Y116.793 E.01033
694 G1 X121.5 Y116.979 E.0082
695 ;WIDTH:0.404799
696 G1 X121.467 Y116.842 E.00577
697 ;WIDTH:0.419999
698 G1 X121.421 Y116.79 E.00296
699 G1 X118.639 Y116.794 E.11873
700 G1 X118.5 Y116.979 E.00988
701 G1 X118.396 Y116.805 E.00865
702 G1 X118.321 Y116.79 E.00326
703 G1 X116.544 Y116.79 E.07584
704 ;WIDTH:0.467072
705 G1 X116.161 Y116.766 E.01842
706 ;WIDTH:0.514144
707 G1 X115.779 Y116.743 E.02041
708 G1 X116.169 Y116.648 E.02141
709 ;WIDTH:0.467072
710 G1 X116.56 Y116.552 E.01933
711 ;WIDTH:0.419999
712 G1 X117.191 Y116.301 E.02898
713 G1 X117.68 Y116.001 E.02448
714 G1 X118.312 Y115.415 E.03678
715 G1 X118.708 Y114.863 E.02899
716 G1 X118.968 Y114.317 E.02581
717 G1 X119.135 Y113.738 E.02572
718 G1 X119.2 Y113.24 E.02143
719 G1 X119.206 Y112.833 E.01737

```

```

720 | G1 X119.127 Y112.227 E.02608
721 | G1 X118.835 Y111.377 E.03836
722 | G1 X118.523 Y110.851 E.0261
723 | G1 X117.918 Y110.187 E.03834
724 | G1 X117.422 Y109.828 E.02613
725 | G1 X116.739 Y109.52 E.03197
726 | G1 E-3.5 F3600
727 | ;WIPE_START
728 | G1 F7200
729 | G1 X116.684 Y109.495 E.-.0287
730 | G1 X116.603 Y109.458 E.-.0423
731 | G1 X116.421 Y109.396 E.-.09133
732 | G1 X116.239 Y109.333 E.-.09148
733 | G1 X116.057 Y109.271 E.-.09133
734 | G1 X116.248 Y109.251 E.-.09122
735 | G1 X116.44 Y109.23 E.-.09174
736 | G1 X116.631 Y109.21 E.-.09122
737 | G1 X118.485 Y109.21 E.-.88068
738 | ;WIPE_END
739 | G1 X121.097 Y112.285 F9000
740 | G1 E5 F2400
741 | ; TYPE: Internal infill
742 | ;WIDTH:0.44
743 | G1 F3000
744 | G1 X121.097 Y109.903 E.10705
745 | G1 X119.582 Y111.418 E.09629
746 | G1 X119.823 Y112.174 E.03566
747 | G1 X119.9 Y112.851 E.03062
748 | G1 X119.888 Y113.323 E.02122
749 | G1 X119.74 Y114.124 E.03661
750 | G1 X121.097 Y115.518 E.08743
751 | G1 X120.518 Y116.097 E.0368
752 | G1 X118.607 Y116.105 E.08588
753 | G1 X118.914 Y115.765 E.02059
754 | G1 E-3.5 F3600
755 | ;WIPE_START
756 | G1 F7200
757 | G1 X118.607 Y116.105 E.-.21759
758 | G1 X120.518 Y116.097 E.-.90773
759 | G1 X121.076 Y115.54 E.-.37468
760 | ;WIPE_END
761 | G1 X112.181 Y115.875 F9000
762 | G1 E5 F2400
763 | G1 F3000
764 | G1 X112.447 Y116.168 E.01778
765 | G1 X112.168 Y116.098 E.01293
766 | G1 X110.482 Y116.097 E.07577
767 | G1 X109.903 Y115.518 E.0368
768 | G1 X111.258 Y114.126 E.0873
769 | G1 X111.109 Y113.295 E.03794
770 | G1 X111.109 Y112.707 E.02642
771 | G1 X111.303 Y111.687 E.04666
772 | G1 X111.408 Y111.408 E.0134
773 | G1 X109.903 Y109.903 E.09565
774 | G1 X109.903 Y112.285 E.10705
775 | ;LAYER_CHANGE
776 | ;Z:0.6
777 | ;HEIGHT:0.2
778 | ;BEFORE_LAYER_CHANGE
779 | G92 E0
780 | ;0.6
781 |
782 |
783 | G1 E-3.5 F3600
784 | ;WIPE_START
785 | G1 F7200
786 | G1 X109.903 Y109.903 E-1.13145
787 | G1 X110.452 Y110.452 E.-.36855
788 | ;WIPE_END
789 | G1 Z.6 F9000
790 | ;AFTER_LAYER_CHANGE
791 | ;0.6
792 | G1 Z.6
793 | G1 X115.5 Y121.7
794 | G1 E5 F2400
795 | ; TYPE: External perimeter
796 | ;WIDTH:0.6
797 | G1 F1500
798 | G1 X115.5 Y116.979 E.29765
799 | G1 E-3.5 F3600
800 | ;WIPE_START
801 | G1 F7200

```

```

802 G1 X115.5 Y120.137 E-1.5
803 ;WIPE_END
804 G1 X118.5 Y121.8 F9000
805 G1 E5 F2400
806 ;WIDTH:0.4
807 G1 F1500
808 G1 X118.5 Y116.979 E.19483
809 G1 E-3.5 F3600
810 ;WIPE_START
811 G1 F7200
812 G1 X118.5 Y120.137 E-1.5
813 ;WIPE_END
814 G1 X121.5 Y121.9 F9000
815 G1 E5 F2400
816 ;WIDTH:0.38292
817 G1 F1500
818 G1 X121.5 Y116.979 E.18936
819 G1 X121.155 Y116.403 F9000
820 ; TYPE: Perimeter
821 ;WIDTH:0.439999
822 G1 F1500
823 G1 X118.679 Y116.403 E.11127
824 G1 X118.563 Y116.415 E.00524
825 G1 X118.5 Y116.492 E.00447
826 G1 X118.437 Y116.415 E.00447
827 G1 X118.321 Y116.403 E.00524
828 G1 X115.779 Y116.403 E.11424
829 G1 X115.555 Y116.448 E.01027
830 G1 X115.5 Y116.531 E.00447
831 G1 X115.445 Y116.448 E.00447
832 G1 X115.221 Y116.403 E.01027
833 G1 X112.879 Y116.403 E.10525
834 G1 X112.648 Y116.451 E.0106
835 G1 X112.5 Y116.677 E.01214
836 G1 X112.352 Y116.451 E.01214
837 G1 X112.121 Y116.403 E.0106
838 G1 X109.979 Y116.403 E.09626
839 G1 X109.597 Y116.784 E.02425
840 G1 X109.597 Y109.597 E.32298
841 G1 X121.403 Y109.597 E.53056
842 G1 X121.403 Y113 E.15293
843 G1 X121.403 Y116.403 E.15293
844 G1 X121.215 Y116.403 E.00845
845 G1 X121.487 Y116.924 F9000
846 ; TYPE: External perimeter
847 ;WIDTH:0.404799
848 G1 F1500
849 G1 X121.467 Y116.842 E.00346
850 ;WIDTH:0.419999
851 G1 X121.421 Y116.79 E.00296
852 G1 X118.669 Y116.79 E.11745
853 G1 X118.5 Y116.979 E.01082
854 G1 X118.359 Y116.794 E.00993
855 G1 X118.321 Y116.79 E.00163
856 G1 X115.765 Y116.79 E.10908
857 ;WIDTH:0.465
858 G1 X115.699 Y116.837 E.00387
859 ;WIDTH:0.51
860 G1 X115.633 Y116.885 E.00431
861 ;WIDTH:0.555
862 G1 X115.566 Y116.932 E.00474
863 ;WIDTH:0.6
864 G1 X115.5 Y116.979 E.00511
865 G1 X115.449 Y116.935 E.00425
866 ;WIDTH:0.555
867 G1 X115.397 Y116.892 E.00391
868 ;WIDTH:0.51
869 G1 X115.346 Y116.848 E.00356
870 ;WIDTH:0.465
871 G1 X115.295 Y116.805 E.00319
872 ;WIDTH:0.419999
873 G1 X115.221 Y116.79 E.00322
874 G1 X112.838 Y116.794 E.1017
875 ;WIDTH:0.42146
876 G1 X112.689 Y116.979 E.01018
877 G1 X112.689 Y121.789 E.20607
878 G1 X112.311 Y121.789 E.01619
879 G1 X112.311 Y116.979 E.20607
880 G1 X112.197 Y116.806 E.00888
881 ;WIDTH:0.419999
882 G1 X112.121 Y116.79 E.00331
883 G1 X109.979 Y116.79 E.09141

```

```

884 ;WIDTH:0.45382
885 G1 X109.899 Y116.853 E.00474
886 ;WIDTH:0.48764
887 G1 X109.819 Y116.916 E.00513
888 ;WIDTH:0.52146
889 G1 X109.739 Y116.979 E.00551
890 G1 X109.739 Y121.739 E.2578
891 G1 X109.261 Y121.739 E.02589
892 G1 X109.261 Y116.979 E.2578
893 G1 X109.244 Y116.66 E.0173
894 ;WIDTH:0.48764
895 G1 X109.227 Y116.34 E.01613
896 ;WIDTH:0.45382
897 G1 X109.21 Y116.021 E.01486
898 ;WIDTH:0.419999
899 G1 X109.21 Y109.21 E.29067
900 G1 X121.79 Y109.21 E.53687
901 G1 X121.79 Y113 E.16174
902 G1 X121.79 Y116.79 E.16174
903 G1 X121.548 Y116.793 E.01033
904 G1 X121.501 Y116.975 E.00802
905 G1 E-3.5 F3600
906 ;WIPE_START
907 G1 F7200
908 G1 X121.487 Y116.924 E.-0.02512
909 G1 X121.467 Y116.842 E.-0.04009
910 G1 X121.421 Y116.79 E.-0.03298
911 G1 X118.669 Y116.79 E-1.3072
912 G1 X118.536 Y116.938 E.-0.09461
913 ;WIPE_END
914 G1 X119.54 Y115.716 F9000
915 G1 E5 F2400
916 M204 S250
917 ; TYPE: Bridge infill
918 ;WIDTH:0.395349
919 ;HEIGHT:0.389872
920 G1 F1500
921 G1 X111.702 Y115.716 E.54448
922 G1 X111.419 Y115.271 E.03663
923 G1 X119.58 Y115.271 E.56692
924 G1 X119.797 Y114.826 E.03439
925 G1 X111.2 Y114.826 E.59721
926 G1 X111.039 Y114.38 E.03294
927 G1 X119.962 Y114.38 E.61985
928 G1 X120.077 Y113.935 E.03193
929 G1 X110.926 Y113.935 E.63569
930 G1 X110.856 Y113.49 E.03129
931 G1 X120.144 Y113.49 E.64521
932 G1 X120.169 Y113.044 E.03103
933 G1 X110.829 Y113.044 E.64882
934 G1 X110.846 Y112.599 E.03094
935 G1 X120.152 Y112.599 E.64646
936 G1 X120.092 Y112.154 E.03119
937 G1 X110.907 Y112.154 E.63805
938 G1 X111.013 Y111.708 E.03185
939 G1 X119.989 Y111.708 E.62354
940 G1 X119.836 Y111.263 E.03269
941 G1 X111.166 Y111.263 E.60228
942 G1 X111.369 Y110.818 E.03398
943 G1 X119.628 Y110.818 E.57373
944 G1 X119.361 Y110.372 E.03611
945 G1 X111.64 Y110.372 E.53635
946 G1 X111.983 Y109.927 E.03903
947 G1 X119.277 Y109.927 E.50669
948 M204 S500
949 G1 X120.539 Y109.722 F9000
950 ; TYPE: Top solid infill
951 ;WIDTH:0.407953
952 ;HEIGHT:0.2
953 G1 F1800
954 G1 X121.117 Y110.3 E.03377
955 G1 X121.117 Y110.816 E.02132
956 G1 X120.184 Y109.883 E.05451
957 G1 X119.667 Y109.883 E.02136
958 G1 X121.117 Y111.333 E.08472
959 G1 X121.117 Y111.849 E.02132
960 G1 X120.068 Y110.8 E.06129
961 G1 X120.235 Y111.187 E.01741
962 G1 X120.381 Y111.628 E.01919
963 G1 X121.117 Y112.365 E.04303
964 G1 X121.117 Y112.881 E.02132
965 G1 X120.517 Y112.282 E.03503

```

```

966 G1 X120.567 Y112.848 E.02347
967 G1 X121.117 Y113.398 E.03213
968 G1 X121.117 Y113.914 E.02132
969 G1 X120.556 Y113.352 E.03281
970 G1 X120.502 Y113.814 E.01922
971 G1 X121.117 Y114.43 E.03596
972 G1 X121.117 Y114.946 E.02132
973 G1 X120.413 Y114.242 E.04113
974 G1 X120.295 Y114.64 E.01715
975 G1 X121.117 Y115.463 E.04806
976 G1 X121.117 Y115.979 E.02132
977 G1 X120.151 Y115.012 E.05647
978 G1 X119.984 Y115.361 E.01598
979 G1 X120.739 Y116.117 E.04414
980 G1 X120.223 Y116.117 E.02132
981 G1 X119.795 Y115.689 E.02501
982 G1 X119.587 Y115.997 E.01535
983 G1 X119.868 Y116.278 E.01642
984 G1 E-3.5 F3600
985 ;WIPE_START
986 G1 F7200
987 G1 X119.587 Y115.997 E.-18876
988 G1 X119.795 Y115.689 E.-17654
989 G1 X120.223 Y116.117 E.-28751
990 G1 X120.739 Y116.117 E.-2451
991 G1 X119.984 Y115.361 E.-50751
992 G1 X120.07 Y115.182 E.-09458
993 ;WIPE_END
994 G1 X111.515 Y110.136 F9000
995 G1 E5 F2400
996 ;WIDTH:0.420784
997 G1 F1800
998 G1 X111.262 Y109.883 E.0153
999 G1 X110.728 Y109.883 E.02284
1000 G1 X111.185 Y110.34 E.02764
1001 G1 X110.992 Y110.682 E.01679
1002 G1 X110.193 Y109.883 E.04832
1003 G1 X109.883 Y109.883 E.01326
1004 G1 X109.883 Y110.107 E.00958
1005 G1 X110.822 Y111.046 E.05679
1006 G1 X110.678 Y111.436 E.01778
1007 G1 X109.883 Y110.641 E.04808
1008 G1 X109.883 Y111.175 E.02284
1009 G1 X110.562 Y111.855 E.0411
1010 G1 X110.48 Y112.307 E.01965
1011 G1 X109.883 Y111.71 E.03611
1012 G1 X109.883 Y112.244 E.02284
1013 G1 X110.437 Y112.798 E.03351
1014 G1 X110.442 Y113.338 E.02309
1015 G1 X109.883 Y112.779 E.03381
1016 G1 X109.883 Y113.313 E.02284
1017 G1 X110.522 Y113.952 E.03865
1018 G1 X110.587 Y114.252 E.01313
1019 G1 X110.718 Y114.682 E.01922
1020 G1 X109.883 Y113.847 E.0505
1021 G1 X109.883 Y114.382 E.02288
1022 G1 X111.221 Y115.72 E.08092
1023 G1 X111.504 Y116.12 E.02095
1024 G1 X111.086 Y116.119 E.01788
1025 G1 X109.883 Y114.916 E.07276
1026 G1 X109.883 Y115.45 E.02284
1027 G1 X110.551 Y116.118 E.0404
1028 G1 X110.015 Y116.117 E.02292
1029 G1 X109.722 Y115.824 E.01772
1030 M106 S252.45
1031 ;LAYER_CHANGE
1032 ;Z:0.8
1033 ;HEIGHT:0.2
1034 ;BEFORE_LAYER_CHANGE
1035 G92 E0
1036 ;0.8
1037
1038
1039 G1 E-3.5 F3600
1040 ;WIPE_START
1041 G1 F7200
1042 G1 X110.015 Y116.117 E.-19682
1043 G1 X110.551 Y116.118 E.-2546
1044 G1 X109.883 Y115.45 E.-44873
1045 G1 X109.883 Y114.916 E.-25365
1046 G1 X110.398 Y115.432 E.-3462
1047 ;WIPE_END

```

```

1048 G1 Z.8 F9000
1049 ;AFTER_LAYER_CHANGE
1050 ;0.8
1051 G1 Z.8
1052 G1 X112.908 Y114.29
1053 G1 E5 F2400
1054 ; TYPE: Perimeter
1055 ;WIDTH:0.439999
1056 G1 F860
1057 G1 X112.833 Y114.118 E.00843
1058 G1 X112.64 Y113.483 E.02983
1059 G1 X112.611 Y112.752 E.03288
1060 G1 X112.685 Y112.3 E.02058
1061 G1 X112.861 Y111.802 E.02374
1062 G1 X113.202 Y111.231 E.02989
1063 G1 X113.375 Y111.039 E.01161
1064 G1 X113.873 Y110.599 E.02986
1065 G1 X114.525 Y110.269 E.03284
1066 G1 X114.959 Y110.151 E.02021
1067 G1 X115.689 Y110.106 E.03287
1068 G1 X116.134 Y110.17 E.0202
1069 G1 X116.811 Y110.415 E.03236
1070 G1 X117.408 Y110.816 E.03232
1071 G1 X117.587 Y110.999 E.0115
1072 G1 X117.885 Y111.365 E.02121
1073 G1 X118.203 Y111.948 E.02984
1074 G1 X118.379 Y112.654 E.0327
1075 G1 X118.386 Y112.925 E.01218
1076 G1 X118.393 Y113.171 E.01106
1077 G1 X118.28 Y113.826 E.02987
1078 G1 X118.118 Y114.248 E.02031
1079 G1 X117.722 Y114.863 E.03287
1080 G1 X117.543 Y115.046 E.0115
1081 G1 X117.028 Y115.465 E.02984
1082 G1 X116.362 Y115.769 E.0329
1083 G1 X116.113 Y115.826 E.01148
1084 G1 X115.452 Y115.9 E.02989
1085 G1 X114.729 Y115.795 E.03283
1086 G1 X114.298 Y115.64 E.02058
1087 G1 X113.84 Y115.376 E.02376
1088 G1 X113.35 Y114.944 E.02936
1089 G1 X112.935 Y114.353 E.03245
1090 G1 X112.932 Y114.345 E.00038
1091 G1 X112.583 Y114.513 F9000
1092 ; TYPE: External perimeter
1093 ;WIDTH:0.419999
1094 G1 F860
1095 G1 X112.384 Y114.055 E.02131
1096 G1 X112.255 Y113.538 E.02274
1097 G1 X112.222 Y112.728 E.0346
1098 G1 X112.309 Y112.204 E.02267
1099 G1 X112.509 Y111.637 E.02566
1100 G1 X112.889 Y111 E.03165
1101 G1 X113.246 Y110.604 E.02275
1102 G1 X113.663 Y110.272 E.02275
1103 G1 X114.386 Y109.906 E.03458
1104 G1 X114.896 Y109.767 E.02256
1105 G1 X115.705 Y109.717 E.03459
1106 G1 X116.229 Y109.793 E.0226
1107 G1 X116.987 Y110.067 E.0344
1108 G1 X117.657 Y110.517 E.03444
1109 G1 X118.029 Y110.897 E.02269
1110 G1 X118.337 Y111.336 E.02289
1111 G1 X118.569 Y111.816 E.02275
1112 G1 X118.765 Y112.601 E.03453
1113 G1 X118.773 Y112.914 E.01336
1114 G1 X118.781 Y113.199 E.01217
1115 G1 X118.655 Y113.929 E.03161
1116 G1 X118.465 Y114.424 E.02263
1117 G1 X118.027 Y115.106 E.03459
1118 G1 X117.654 Y115.486 E.02272
1119 G1 X117.224 Y115.801 E.02275
1120 G1 X116.487 Y116.137 E.03457
1121 G1 X115.968 Y116.256 E.02272
1122 G1 X115.436 Y116.288 E.02274
1123 G1 X114.634 Y116.173 E.03458
1124 G1 X114.134 Y115.992 E.02269
1125 G1 X113.614 Y115.692 E.02562
1126 G1 X113.06 Y115.204 E.03151
1127 G1 X112.611 Y114.566 E.03329
1128 G1 E-3.5 F3600
1129 ;WIPE_START

```

```

1130 G1 F7200
1131 G1 X112.583 Y114.513 E-.02847
1132 G1 X112.384 Y114.055 E-.2372
1133 G1 X112.255 Y113.538 E-.2531
1134 G1 X112.222 Y112.728 E-.38507
1135 G1 X112.309 Y112.204 E-.25231
1136 G1 X112.509 Y111.637 E-.28559
1137 G1 X112.572 Y111.532 E-.05826
1138 ;WIPE_END
1139 G1 X114.459 Y110.645 F9000
1140 G1 E5 F2400
1141 ; TYPE: Internal infill
1142 ;WIDTH:0.44
1143 G1 F860
1144 G1 X114.901 Y110.484 E.02114
1145 G1 X112.977 Y112.407 E.12225
1146 G1 X112.924 Y112.924 E.02336
1147 G1 X115.585 Y115.585 E.16912
1148 G1 X115.415 Y115.585 E.00764
1149 G1 X118.08 Y112.92 E.16937
1150 G1 X118.074 Y112.696 E.01007
1151 G1 X117.995 Y112.38 E.01464
1152 G1 X116.098 Y110.482 E.1206
1153 G1 X115.677 Y110.413 E.01917
1154 G1 X115.095 Y110.449 E.02621
1155 ;LAYER_CHANGE
1156 ;Z:1
1157 ;HEIGHT:0.2
1158 ;BEFORE_LAYER_CHANGE
1159 G92 E0
1160 ;1
1161
1162
1163 G1 E-3.5 F3600
1164 ;WIPE_START
1165 G1 F7200
1166 G1 X115.677 Y110.413 E-.27698
1167 G1 X116.098 Y110.482 E-.20264
1168 G1 X117.617 Y112.001 E-1.02038
1169 ;WIPE_END
1170 G1 Z1 F9000
1171 ;AFTER_LAYER_CHANGE
1172 ;1
1173 G1 Z1
1174 G1 X115.053 Y110.156
1175 G1 E5 F2400
1176 ; TYPE: Perimeter
1177 ;WIDTH:0.439999
1178 G1 F1397
1179 G1 X115.376 Y110.104 E.0147
1180 G1 X115.904 Y110.129 E.02375
1181 G1 X116.596 Y110.317 E.03223
1182 G1 X117.215 Y110.663 E.03187
1183 G1 X117.694 Y111.107 E.02935
1184 G1 X117.853 Y111.327 E.0122
1185 G1 X118.119 Y111.774 E.02338
1186 G1 X118.337 Y112.401 E.02983
1187 G1 X118.358 Y112.657 E.01154
1188 G1 X118.396 Y113.121 E.02092
1189 G1 X118.295 Y113.766 E.02934
1190 G1 X118.112 Y114.262 E.02376
1191 G1 X117.879 Y114.659 E.02069
1192 G1 X117.437 Y115.155 E.02986
1193 G1 X117.015 Y115.473 E.02375
1194 G1 X116.607 Y115.681 E.02058
1195 G1 X115.901 Y115.872 E.03287
1196 G1 X115.645 Y115.888 E.01153
1197 G1 X114.982 Y115.853 E.02984
1198 G1 X114.294 Y115.636 E.03242
1199 G1 X113.735 Y115.298 E.02936
1200 G1 X113.353 Y114.949 E.02325
1201 G1 X112.935 Y114.354 E.03268
1202 G1 X112.703 Y113.731 E.02988
1203 G1 X112.616 Y113.204 E.024
1204 G1 X112.637 Y112.54 E.02986
1205 G1 X112.843 Y111.839 E.03284
1206 G1 X113.054 Y111.442 E.0202
1207 G1 X113.522 Y110.88 E.03287
1208 G1 X113.873 Y110.599 E.02021
1209 G1 X114.525 Y110.269 E.03284
1210 G1 X114.778 Y110.2 E.01179
1211 G1 X114.994 Y110.165 E.00983

```

```

1212 G1 X114.939 Y109.784 F9000
1213 ; TYPE: External perimeter
1214 ;WIDTH:0.419999
1215 G1 F1397
1216 G1 X115.364 Y109.716 E.01837
1217 G1 X115.965 Y109.744 E.02568
1218 G1 X116.743 Y109.955 E.0344
1219 G1 X117.445 Y110.348 E.03433
1220 G1 X117.986 Y110.849 E.03147
1221 G1 X118.338 Y111.337 E.02568
1222 G1 X118.57 Y111.819 E.02283
1223 G1 X118.72 Y112.33 E.02273
1224 G1 X118.744 Y112.625 E.01263
1225 G1 X118.786 Y113.135 E.02184
1226 G1 X118.671 Y113.864 E.0315
1227 G1 X118.463 Y114.428 E.02565
1228 G1 X118.194 Y114.887 E.0227
1229 G1 X117.701 Y115.441 E.03165
1230 G1 X117.221 Y115.803 E.02566
1231 G1 X116.746 Y116.044 E.02273
1232 G1 X115.964 Y116.256 E.03458
1233 G1 X115.433 Y116.289 E.02271
1234 G1 X114.903 Y116.234 E.02274
1235 G1 X114.133 Y115.991 E.03446
1236 G1 X113.502 Y115.61 E.03146
1237 G1 X113.061 Y115.206 E.02552
1238 G1 X112.596 Y114.544 E.03453
1239 G1 X112.384 Y114.055 E.02275
1240 G1 X112.254 Y113.534 E.02292
1241 G1 X112.211 Y113 E.02286
1242 G1 X112.254 Y112.469 E.02274
1243 G1 X112.483 Y111.691 E.03461
1244 G1 X112.731 Y111.225 E.02253
1245 G1 X113.249 Y110.602 E.03458
1246 G1 X113.663 Y110.272 E.02259
1247 G1 X114.386 Y109.906 E.03458
1248 G1 X114.676 Y109.827 E.01283
1249 G1 X114.88 Y109.794 E.00882
1250 G1 X116.679 Y110.506 F9000
1251 ; TYPE: Top solid infill
1252 ;WIDTH:0.414325
1253 G1 F1397
1254 G1 X117.504 Y111.331 E.04904
1255 G1 X117.757 Y111.682 E.01819
1256 G1 X117.94 Y112.061 E.01769
1257 G1 X118.036 Y112.389 E.01437
1258 G1 X116.135 Y110.487 E.11304
1259 G1 X115.86 Y110.413 E.01197
1260 G1 X115.518 Y110.397 E.01439
1261 G1 X118.098 Y112.976 E.15334
1262 G1 X118.109 Y113.11 E.00565
1263 G1 X118.054 Y113.457 E.01477
1264 G1 X115.026 Y110.429 E.18
1265 G1 X114.615 Y110.544 E.01794
1266 G1 X117.949 Y113.878 E.19819
1267 G1 X117.853 Y114.139 E.01169
1268 G1 X117.791 Y114.245 E.00516
1269 G1 X114.266 Y110.72 E.20955
1270 G1 X114.029 Y110.841 E.01119
1271 G1 X113.936 Y110.915 E.005
1272 G1 X117.583 Y114.562 E.2168
1273 G1 X117.335 Y114.84 E.01566
1274 G1 X113.658 Y111.162 E.21861
1275 G1 X113.42 Y111.449 E.01567
1276 G1 X117.056 Y115.085 E.21614
1277 G1 X116.863 Y115.23 E.01015
1278 G1 X116.739 Y115.293 E.00585
1279 G1 X113.207 Y111.762 E.20993
1280 G1 X113.109 Y111.947 E.0088
1281 G1 X113.054 Y112.134 E.00819
1282 G1 X116.369 Y115.449 E.19706
1283 G1 X115.956 Y115.561 E.01799
1284 G1 X112.935 Y112.54 E.17959
1285 G1 X112.887 Y113.018 E.02019
1286 G1 X115.481 Y115.612 E.1542
1287 G1 X115.04 Y115.572 E.01861
1288 G1 X114.859 Y115.515 E.00798
1289 G1 X112.971 Y113.627 E.11223
1290 G1 X113.024 Y113.838 E.00914
1291 G1 X113.186 Y114.213 E.01717
1292 G1 X113.548 Y114.729 E.0265
1293 G1 X114.332 Y115.513 E.04661

```

```

1294 ; stop printing object Slicer Test v3.stl id:0 copy 0
1295 G1 E-3.5 F3600
1296 ;WIPE_START
1297 G1 F7200;_WIPE
1298 G1 X113.548 Y114.729 E-.52665
1299 G1 X113.186 Y114.213 E-.2994
1300 G1 X113.024 Y113.838 E-.19404
1301 G1 X112.971 Y113.627 E-.10334
1302 G1 X113.532 Y114.188 E-.37657
1303 ;WIPE_END
1304 M107
1305 ; TYPE: Custom
1306 ; Filament-specific end gcode
1307 ;END gcode for filament

```

Listing A.1.2 G-Code Generated Using Prusa Slicer [68] (Main)

```

1308 G1 Z3 F600 ; Move print head up
1309 G1 X5 Y193.8 F9000 ; present print
1310 G1 Z71 F600 ; Move print head further up
1311 G1 Z150 F600 ; Move print head further up
1312 M140 S0 ; turn off heatbed
1313 M104 S0 ; turn off temperature
1314 M107 ; turn off fan
1315 M84 X Y E ; disable motors
1316 ; objects_info = {"objects": [{"name": "Slicer Test v3.stl id:0 copy 0", "polygon": "[[122.000,117.000],[121.600,117.000],[121.600,122.000],[121.400,122.000],[121.400,117.000],[118.700,117.000]]"}]
1317 ; filament used [mm] = 58.94
1318 ; filament used [cm3] = 0.10
1319 ; total filament used [g] = 0.00
1320 ; total filament cost = 0.00
1321 ; total filament used for wipe tower [g] = 0.00
1322 ; estimated printing time (normal mode) = 1m 52s
1323 ; estimated first layer printing time (normal mode) = 55s
1324
1325 ; prusaslicer_config = begin
1326 ; arc_fitting = disabled
1327 ; autoemit_temperature_commands = 1
1328 ; automatic_extrusion_widths = 0
1329 ; automatic_infill_combination = 0
1330 ; automatic_infill_combination_max_layer_height = 100%
1331 ; avoid_crossing_curved_overhangs = 0
1332 ; avoid_crossing_perimeters = 0
1333 ; avoid_crossing_perimeters_max_detour = 0
1334 ; bed_custom_model =
1335 ; bed_custom_texture =
1336 ; bed_shape = 3x3,228x3,228x228,3x228
1337 ; bed_temperature = 0
1338 ; bed_temperature_extruder = 0
1339 ; before_layer_gcode = ;BEFORE_LAYER_CHANGE\nG92 E0\n;{layer_z}\n\n
1340 ; between_objects_gcode =
1341 ; binary_gcode = 0
1342 ; bottom_fill_pattern = monotonic
1343 ; bottom_solid_layers = 1
1344 ; bottom_solid_min_thickness = 0
1345 ; bridge_acceleration = 250
1346 ; bridge_angle = 0
1347 ; bridge_fan_speed = 100
1348 ; bridge_flow_ratio = 0.95
1349 ; bridge_speed = 25
1350 ; brim_separation = 0
1351 ; brim_type = outer_only
1352 ; brim_width = 0
1353 ; chamber_minimal_temperature = 0
1354 ; chamber_temperature = 0
1355 ; color_change_gcode = M600
1356 ; colorprint_heights =
1357 ; compatible_printers_condition_cummulative = "printer_model=~/(ENDER|CR|SERMOON).*/ and
    nozzle_diameter[0]==0.4";
1358 ; complete_objects = 0
1359 ; cooling = 1
1360 ; cooling_tube_length = 5
1361 ; cooling_tube_retraction = 91.5
1362 ; default_acceleration = 500
1363 ; default_filament_profile = "Generic PLA @CREALITY"
1364 ; default_print_profile = "0.16 mm OPTIMAL (0.4 mm nozzle) @CREALITY"
1365 ; deretract_speed = 40
1366 ; disable_fan_first_layers = 3
1367 ; dont_support_bridges = 1
1368 ; draft_shield = disabled

```

```

1369 ; duplicate_distance = 6
1370 ; elefant_foot_compensation = 0.1
1371 ; enable_dynamic_fan_speeds = 0
1372 ; enable_dynamic_overhang_speeds = 0
1373 ; end_filament_gcode = ""; Filament-specific end gcode \n;END gcode for filament\n"
1374 ; end_gcode = {if max_layer_z < max_print_height}G1 Z{z_offset+min(max_layer_z+2,
1375   max_print_height)} F600 ; Move print head up{endif}\nG1 X5 Y{print_bed_max[1]*0.85} F{
1376   travel_speed*60} ; present print\n{if max_layer_z < max_print_height-10}G1 Z{z_offset+
1377   min(max_layer_z+70, max_print_height-10)} F600 ; Move print head further up{endif}\n{if
1378   max_layer_z < max_print_height*0.6}G1 Z{max_print_height*0.6} F600 ; Move print head
1379   further up{endif}\nM140 S0 ; turn off heatbed\nM104 S0 ; turn off temperature\nM107 ;
1380   turn off fan\nM84 X Y E ; disable motors
1381 ; ensure_vertical_shell_thickness = enabled
1382 ; external_perimeter_acceleration = 0
1383 ; external_perimeter_extrusion_width = 0.42
1384 ; external_perimeter_speed = 25
1385 ; external_perimeters_first = 0
1386 ; extra_loading_move = -2
1387 ; extra_perimeters = 0
1388 ; extra_perimeters_on_overhangs = 0
1389 ; extruder_clearance_height = 20
1390 ; extruder_clearance_radius = 20
1391 ; extruder_colour = #FCE94F
1392 ; extruder_offset = 0x0
1393 ; extrusion_axis = E
1394 ; extrusion_multiplier = 1
1395 ; extrusion_width = 0.44
1396 ; fan_always_on = 0
1397 ; fan_below_layer_time = 60
1398 ; filament_abrasive = 0
1399 ; filament_colour = #29B2B2
1400 ; filament_cooling_final_speed = 3.4
1401 ; filament_cooling_initial_speed = 2.2
1402 ; filament_cooling_moves = 4
1403 ; filament_cost = 0
1404 ; filament_density = 0
1405 ; filament_deretract_speed = nil
1406 ; filament_diameter = 1.5
1407 ; filament_infill_max_crossing_speed = 0
1408 ; filament_infill_max_speed = 0
1409 ; filament_load_time = 0
1410 ; filament_loading_speed = 28
1411 ; filament_loading_speed_start = 3
1412 ; filament_max_volumetric_speed = 0
1413 ; filament_minimal_purge_on_wipe_tower = 15
1414 ; filament_multitool_ramming = 0
1415 ; filament_multitool_ramming_flow = 10
1416 ; filament_multitool_ramming_volume = 10
1417 ; filament_notes = ""
1418 ; filament_purge_multiplier = 100%
1419 ; filament_ramming_parameters = "120 100 6.6 6.8 7.2 7.6 7.9 8.2 8.7 9.4 9.9 10.0| 0.05 6.6
1420   0.45 6.8 0.95 7.8 1.45 8.3 1.95 9.7 2.45 10 2.95 7.6 3.45 7.6 3.95 7.6 4.45 7.6 4.95
1421   7.6"
1422 ; filament_retract_before_travel = nil
1423 ; filament_retract_before_wipe = nil
1424 ; filament_retract_layer_change = nil
1425 ; filament_retract_length = nil
1426 ; filament_retract_length_toolchange = nil
1427 ; filament_retract_lift = nil
1428 ; filament_retract_lift_above = nil
1429 ; filament_retract_lift_below = nil
1430 ; filament_retract_restart_extra = nil
1431 ; filament_retract_restart_extra_toolchange = nil
1432 ; filament_retract_speed = nil
1433 ; filament_seam_gap_distance = nil
1434 ; filament_settings_id = "My Settings"
1435 ; filament_shrinkage_compensation_xy = 0%
1436 ; filament_shrinkage_compensation_z = 0%
1437 ; filament_soluble = 0
1438 ; filament_spool_weight = 0
1439 ; filament_stamping_distance = 0
1440 ; filament_stamping_loading_speed = 20
1441 ; filament_toolchange_delay = 0
1442 ; filament_travel_lift_before_obstacle = nil
1443 ; filament_travel_max_lift = nil
1444 ; filament_travel_ramping_lift = nil
1445 ; filament_travel_slope = nil
1446 ; filament_type = PLA
1447 ; filament_unload_time = 0
1448 ; filament_unloading_speed = 90
1449 ; filament_unloading_speed_start = 100
1450 ; filament_vendor = (Unknown)

```

```

1443 ; filament_wipe = nil
1444 ; fill_angle = 45
1445 ; fill_density = 20%
1446 ; fill_pattern = grid
1447 ; first_layer_acceleration = 0
1448 ; first_layer_acceleration_over_raft = 0
1449 ; first_layer_bed_temperature = 0
1450 ; first_layer_extrusion_width = 0.42
1451 ; first_layer_height = 0.2
1452 ; first_layer_infill_speed = 0
1453 ; first_layer_speed = 20
1454 ; first_layer_speed_over_raft = 30
1455 ; first_layer_temperature = 20
1456 ; full_fan_speed_layer = 0
1457 ; fuzzy_skin = none
1458 ; fuzzy_skin_point_dist = 0.8
1459 ; fuzzy_skin_thickness = 0.3
1460 ; gap_fill_enabled = 1
1461 ; gap_fill_speed = 30
1462 ; gcode_comments = 0
1463 ; gcode_flavor = marlin
1464 ; gcode_label_objects = octoprint
1465 ; gcode_resolution = 0.0125
1466 ; gcode_substitutions =
1467 ; high_current_on_filament_swap = 0
1468 ; host_type = prusalink
1469 ; idle_temperature = nil
1470 ; infill_acceleration = 0
1471 ; infill_anchor = 600%
1472 ; infill_anchor_max = 50
1473 ; infill_every_layers = 1
1474 ; infill_extruder = 1
1475 ; infill_extrusion_width = 0.44
1476 ; infill_first = 0
1477 ; infill_overlap = 23%
1478 ; infill_speed = 50
1479 ; interface_shells = 0
1480 ; interlocking_beam = 0
1481 ; interlocking_beam_layer_count = 2
1482 ; interlocking_beam_width = 0.8
1483 ; interlocking_boundary_avoidance = 2
1484 ; interlocking_depth = 2
1485 ; interlocking_orientation = 22.5
1486 ; ironing = 0
1487 ; ironing_flowrate = 10%
1488 ; ironing_spacing = 0.1
1489 ; ironing_speed = 20
1490 ; ironing_type = top
1491 ; layer_gcode = ;AFTER_LAYER_CHANGE\n;{layer_z}
1492 ; layer_height = 0.2
1493 ; machine_limits_usage = emit_to_gcode
1494 ; machine_max_acceleration_e = 5000
1495 ; machine_max_acceleration_extruding = 500
1496 ; machine_max_acceleration_retracting = 1000
1497 ; machine_max_acceleration_travel = 500
1498 ; machine_max_acceleration_x = 500
1499 ; machine_max_acceleration_y = 500
1500 ; machine_max_acceleration_z = 100
1501 ; machine_max_feedrate_e = 60
1502 ; machine_max_feedrate_x = 500
1503 ; machine_max_feedrate_y = 500
1504 ; machine_max_feedrate_z = 10
1505 ; machine_max_jerk_e = 5
1506 ; machine_max_jerk_x = 8
1507 ; machine_max_jerk_y = 8
1508 ; machine_max_jerk_z = 0.4
1509 ; machine_min_extruding_rate = 0
1510 ; machine_min_travel_rate = 0
1511 ; max_fan_speed = 100
1512 ; max_layer_height = 0.32
1513 ; max_print_height = 250
1514 ; max_print_speed = 100
1515 ; max_volumetric_extrusion_rate_slope_negative = 0
1516 ; max_volumetric_extrusion_rate_slope_positive = 0
1517 ; max_volumetric_speed = 0
1518 ; min_bead_width = 85%
1519 ; min_fan_speed = 35
1520 ; min_feature_size = 25%
1521 ; min_layer_height = 0.06
1522 ; min_print_speed = 10
1523 ; min_skirt_length = 4
1524 ; mmu_segmented_region_interlocking_depth = 0

```

```

1525 ; mmu_segmented_region_max_width = 0
1526 ; multimaterial_purging = 140
1527 ; notes =
1528 ; nozzle_diameter = 0.4
1529 ; nozzle_high_flow = 0
1530 ; only_one_perimeter_first_layer = 0
1531 ; only_retract_when_crossing_perimeters = 0
1532 ; ooze_prevention = 0
1533 ; output_filename_format = {input_filename_base}_{print_time}_{digits(layer_height,1,2)}mm_{temperature[0]}C_{filament_type[0]}_{printer_model}.gcode
1534 ; over_bridge_speed = 0
1535 ; overhang_fan_speed_0 = 0
1536 ; overhang_fan_speed_1 = 0
1537 ; overhang_fan_speed_2 = 0
1538 ; overhang_fan_speed_3 = 0
1539 ; overhang_speed_0 = 15
1540 ; overhang_speed_1 = 15
1541 ; overhang_speed_2 = 20
1542 ; overhang_speed_3 = 25
1543 ; overhangs = 0
1544 ; parking_pos_retraction = 92
1545 ; pause_print_gcode =
1546 ; perimeter_acceleration = 0
1547 ; perimeter_extruder = 1
1548 ; perimeter_extrusion_width = 0.44
1549 ; perimeter_generator = arachne
1550 ; perimeter_speed = 40
1551 ; perimeters = 2
1552 ; physical_printer_settings_id =
1553 ; post_process =
1554 ; prefer_clockwise_movements = 0
1555 ; print_settings_id = 0.20 mm NORMAL (0.4 mm nozzle) @CREALITY
1556 ; printer_model = ENDER3
1557 ; printer_notes = Don't remove the following keywords! These keywords are used in the "compatible printer" condition of the print and filament profiles to link the particular print and filament profiles to this printer profile.\nPRINTER_VENDOR_CREALITY\nPRINTER_MODEL_ENDER3\nPRINTER_HAS_BOWDEN
1558 ; printer_settings_id = Creality Ender-3 (0.4 mm nozzle)
1559 ; printer_technology = FFF
1560 ; printer_variant = 0.4
1561 ; printer_vendor =
1562 ; profile_vendor = Creality
1563 ; profile_version = 1.0.0
1564 ; raft_contact_distance = 0.1
1565 ; raft_expansion = 1.5
1566 ; raft_first_layer_density = 90%
1567 ; raft_first_layer_expansion = 3
1568 ; raft_layers = 0
1569 ; remaining_times = 0
1570 ; resolution = 0
1571 ; retract_before_travel = 2
1572 ; retract_before_wipe = 70%
1573 ; retract_layer_change = 1
1574 ; retract_length = 5
1575 ; retract_length_toolchange = 1
1576 ; retract_lift = 0
1577 ; retract_lift_above = 0.2
1578 ; retract_lift_below = 0
1579 ; retract_restart_extra = 0
1580 ; retract_restart_extra_toolchange = 0
1581 ; retract_speed = 60
1582 ; scarf_seam_entire_loop = 0
1583 ; scarf_seam_length = 20
1584 ; scarf_seam_max_segment_length = 1
1585 ; scarf_seam_on_inner_perimeters = 0
1586 ; scarf_seam_only_on_smooth = 1
1587 ; scarf_seam_placement = nowhere
1588 ; scarf_seam_start_height = 0%
1589 ; seam_gap_distance = 15%
1590 ; seam_position = nearest
1591 ; silent_mode = 0
1592 ; single_extruder_multi_material = 0
1593 ; single_extruder_multi_material_priming = 0
1594 ; skirt_distance = 3
1595 ; skirt_height = 2
1596 ; skirts = 0
1597 ; slice_closing_radius = 0.049
1598 ; slicing_mode = regular
1599 ; slowdown_below_layer_time = 5
1600 ; small_perimeter_speed = 25
1601 ; solid_infill_acceleration = 0
1602 ; solid_infill_below_area = 0

```

```

1603 ; solid_infill_every_layers = 0
1604 ; solid_infill_extruder = 1
1605 ; solid_infill_extrusion_width = 0.44
1606 ; solid_infill_speed = 40
1607 ; spiral_vase = 0
1608 ; staggered_inner_seams = 0
1609 ; standby_temperature_delta = -5
1610 ; start_filament_gcode = "; Filament gcode\n"
1611 ; start_gcode = G90 ; use absolute coordinates\nM83 ; extruder relative mode\nM104 S{is_nil(
    idle_temperature[0]) ? 150 : idle_temperature[0]} ; set temporary nozzle temp to prevent
    oozing during homing\nM140 S{first_layer_bed_temperature[0]} ; set final bed temp\nG4
    S30 ; allow partial nozzle warmup\nG28 ; home all axis\nG1 Z50 F240\nG1 X2.0 Y10 F3000\
    nM104 S{first_layer_temperature[0]} ; set final nozzle temp\nM190 S{
        first_layer_bed_temperature[0]} ; wait for bed temp to stabilize\nM109 S{
            first_layer_temperature[0]} ; wait for nozzle temp to stabilize\nG1 Z0.28 F240\nG92 E0\
            nG1 X2.0 Y140 E10 F1500 ; prime the nozzle\nG1 X2.3 Y140 F5000\nG92 E0\nG1 X2.3 Y10 E10\
            F1200 ; prime the nozzle\nG92 E0
1612 ; support_material = 0
1613 ; support_material_angle = 0
1614 ; support_material_auto = 1
1615 ; support_material_bottom_contact_distance = 0
1616 ; support_material_bottom_interface_layers = -1
1617 ; support_material_buildplate_only = 0
1618 ; support_material_closing_radius = 2
1619 ; support_material_contact_distance = 0.15
1620 ; support_material_enforce_layers = 0
1621 ; support_material_extruder = 0
1622 ; support_material_extrusion_width = 0.36
1623 ; support_material_interface_contact_loops = 0
1624 ; support_material_interface_extruder = 0
1625 ; support_material_interface_layers = 2
1626 ; support_material_interface_pattern = rectilinear
1627 ; support_material_interface_spacing = 0.2
1628 ; support_material_interface_speed = 100%
1629 ; support_material_pattern = rectilinear
1630 ; support_material_spacing = 1
1631 ; support_material_speed = 40
1632 ; support_material_style = grid
1633 ; support_material_synchronize_layers = 0
1634 ; support_material_threshold = 40
1635 ; support_material_with_sheath = 0
1636 ; support_material_xy_spacing = 60%
1637 ; support_tree_angle = 40
1638 ; support_tree_angle_slow = 25
1639 ; support_tree_branch_diameter = 2
1640 ; support_tree_branch_diameter_angle = 5
1641 ; support_tree_branch_diameter_double_wall = 3
1642 ; support_tree_branch_distance = 1
1643 ; support_tree_tip_diameter = 0.8
1644 ; support_tree_top_rate = 15%
1645 ; temperature = 20
1646 ; template_custom_gcode =
1647 ; thick_bridges = 1
1648 ; thin_walls = 0
1649 ; thumbnails =
1650 ; thumbnails_format = PNG
1651 ; toolchange_gcode =
1652 ; top_fill_pattern = monotonic
1653 ; top_infill_extrusion_width = 0.4
1654 ; top_one_perimeter_type = none
1655 ; top_solid_infill_acceleration = 0
1656 ; top_solid_infill_speed = 30
1657 ; top_solid_layers = 1
1658 ; top_solid_min_thickness = 0
1659 ; travel_acceleration = 0
1660 ; travel_lift_before_obstacle = 0
1661 ; travel_max_lift = 0
1662 ; travel_ramping_lift = 0
1663 ; travel_slope = 0
1664 ; travel_speed = 150
1665 ; travel_speed_z = 0
1666 ; use_firmware_retraction = 0
1667 ; use_relative_e_distances = 1
1668 ; use_volumetric_e = 0
1669 ; variable_layer_height = 1
1670 ; wall_distribution_count = 1
1671 ; wall_transition_angle = 10
1672 ; wall_transition_filter_deviation = 25%
1673 ; wall_transition_length = 100%
1674 ; wipe = 1
1675 ; wipe_into_infill = 0
1676 ; wipe_into_objects = 0

```

```

1677 ; wipe_tower = 0
1678 ; wipe_tower_acceleration = 0
1679 ; wipe_tower_bridging = 10
1680 ; wipe_tower_brim_width = 2
1681 ; wipe_tower_cone_angle = 0
1682 ; wipe_tower_extra_flow = 100%
1683 ; wipe_tower_extra_spacing = 100%
1684 ; wipe_tower_extruder = 0
1685 ; wipe_tower_no_sparse_layers = 0
1686 ; wipe_tower_width = 60
1687 ; wiping_volumes_matrix = 0
1688 ; wiping_volumes_use_custom_matrix = 0
1689 ; xy_size_compensation = 0
1690 ; z_offset = 0
1691 ; prusaslicer_config = end

```

Listing A.1.3 G-Code generated using Prusa Slicer [68] (Reset)

A.2 OrcaSlicer 2.2.0 – G-Code Listing

```

1 ; HEADER_BLOCK_START
2 ; generated by OrcaSlicer 2.2.0 on 2025-04-14 at 16:23:04
3 ; total layer number: 5
4 ; filament_density: 1.24
5 ; filament_diameter: 1.75
6 ; max_z_height: 1.00
7 ; HEADER_BLOCK_END
8
9 ; external perimeters extrusion width = 0.45mm
10 ; perimeters extrusion width = 0.45mm
11 ; infill extrusion width = 0.45mm
12 ; solid infill extrusion width = 0.45mm
13 ; top infill extrusion width = 0.40mm
14 ; first layer extrusion width = 0.42mm
15
16 ; EXECUTABLE_BLOCK_START
17 M73 P0 R2
18 M201 X500 Y500 Z100 E5000
19 M203 X500 Y500 Z10 E60
20 M204 P500 R1000 T500
21 M205 X8.00 Y8.00 Z0.40 E5.00 ; sets the jerk limits, mm/sec
22 ; TYPE: Custom
23 G90 ; use absolute coordinates
24 M83 ; extruder relative mode
25 M140 S60 ; set final bed temp
26 M104 S150 ; set temporary nozzle temp to prevent oozing during homing
27 G4 S10 ; allow partial nozzle warmup
28 G28 ; home all axis
29 G1 Z50 F240
30 G1 X2 Y10 F3000
31 M73 P10 R1
32 M104 S220 ; set final nozzle temp
33 M190 S60 ; wait for bed temp to stabilize
34 M109 S220 ; wait for nozzle temp to stabilize
35 M73 P10 R1
36 G1 Z0.28 F240
37 G92 E0
38 G1 Y140 E10 F1500 ; prime the nozzle
39 M73 P21 R1
40 M73 P21 R1
41 G1 X2.3 F5000
42 M73 P25 R1
43 G92 E0
44 M73 P25 R1
45 G1 Y10 E10 F1200 ; prime the nozzle
46 G92 E0

```

Listing A.2.1 G-Code Generated Using Orca Slicer [63] (Initialization)

```

47 G90
48 G21
49 M83 ; use relative distances for extrusion
50 ; filament start gcode
51 M106 S0
52 ;LAYER_CHANGE
53 ;Z:0.2

```

```

54 ;HEIGHT:0.2
55 ;BEFORE_LAYER_CHANGE
56 ;0.2
57 G92 E0
58
59 G1 E-4 F3600
60 M73 P30 R1
61 ;_SET_FAN_SPEED_CHANGING_LAYER
62 ; printing object Slicer Test v3.stl id:0 copy 0
63 M73 P30 R1
64 G1 Z.6 F9000
65 G1 X115.895 Y110.906
66 M73 P31 R1
67 M73 P31 R1
68 G1 Z.6
69 M73 P32 R1
70 M73 P32 R1
71 G1 Z.2
72 G1 E4 F2400
73 ; TYPE: Inner wall
74 ;WIDTH:0.419999
75 G1 F900
76 G1 X115.472 Y110.847 E.01312
77 G1 X114.702 Y110.819 E.0237
78 G1 X113.565 Y110.852 E.03494
79 G1 X112.951 Y110.93 E.01901
80 G1 X112.453 Y110.876 E.0154
81 G1 X113.021 Y110.396 E.02285
82 G1 X113.399 Y109.943 E.01814
83 G1 X113.923 Y108.96 E.03423
84 G1 X114.11 Y108.28 E.02165
85 G1 X114.18 Y107.283 E.03071
86 G1 X114.089 Y106.607 E.02097
87 G1 X113.923 Y106.041 E.01813
88 G1 X113.79 Y105.767 E.00935
89 G1 X113.399 Y105.058 E.02489
90 G1 X112.951 Y104.533 E.02121
91 G1 X112.518 Y104.169 E.01739
92 G1 X112.999 Y104.186 E.01481
93 G1 X115.813 Y104.187 E.08647
94 G1 X115.819 Y110.032 E.1796
95 M73 P33 R1
96 M73 P33 R1
97 G1 X115.892 Y110.866 E.02572
98 G1 X116 Y111.479 F9000
99 ; TYPE: Outer wall
100 G1 F900
101 G1 X115.949 Y111.295 E.00585
102 G1 X115.459 Y111.223 E.01523
103 G1 X114.543 Y111.191 E.02815
104 G1 X113.641 Y111.223 E.02773
105 G1 X113.136 Y111.298 E.01568
106 G1 X113 Y111.481 E.00701
107 ;WIDTH:0.419168
108 G1 X112.901 Y111.31 E.00604
109 ;WIDTH:0.419999
110 G1 X112.297 Y111.241 E.01868
111 G1 X111.298 Y111.29 E.03075
112 ;WIDTH:0.459296
113 G1 X111.19 Y111.27 E.00372
114 ;WIDTH:0.498592
115 G1 X111.082 Y111.25 E.00407
116 ;WIDTH:0.537889
117 G1 X110.974 Y111.231 E.00442
118 ;WIDTH:0.577185
119 G1 X110.866 Y111.211 E.00478
120 G1 X110.959 Y111.156 E.00474
121 ;WIDTH:0.537889
122 G1 X111.053 Y111.1 E.00439
123 ;WIDTH:0.498592
124 G1 X111.147 Y111.045 E.00404
125 ;WIDTH:0.459296
126 G1 X111.24 Y110.989 E.00369
127 ;WIDTH:0.419999
128 G1 X111.695 Y110.846 E.01466
129 G1 X112.255 Y110.558 E.01934
130 G1 X112.731 Y110.155 E.01917
131 G1 X113.131 Y109.669 E.01932
132 G1 X113.561 Y108.853 E.02835
133 M73 P34 R1
134 M73 P34 R1
135 G1 X113.734 Y108.25 E.0193

```

```

136 G1 X113.806 Y107.329 E.02836
137 G1 X113.727 Y106.713 E.01908
138 G1 X113.561 Y106.147 E.01813
139 G1 X113.109 Y105.299 E.02953
140 G1 X112.628 Y104.754 E.02234
141 G1 X112.219 Y104.417 E.01628
142 G1 X111.667 Y104.143 E.01894
143 G1 X111.102 Y103.958 E.01827
144 ;WIDTH:0.460708
145 G1 X110.92 Y103.896 E.00654
146 ;WIDTH:0.501417
147 G1 X110.739 Y103.833 E.00717
148 ;WIDTH:0.542126
149 G1 X110.557 Y103.771 E.00781
150 G1 X110.758 Y103.751 E.00821
151 ;WIDTH:0.501417
152 G1 X110.959 Y103.73 E.00754
153 ;WIDTH:0.460708
154 G1 X111.159 Y103.71 E.00687
155 ;WIDTH:0.419999
156 G1 X113 Y103.809 E.05663
157 G1 X116.19 Y103.81 E.09804
158 G1 X116.196 Y110.011 E.19054
159 G1 X116.29 Y111.29 E.0394
160 M73 P35 R1
161 M73 P35 R1
162 G1 X116.058 Y111.291 E.00714
163 G1 X116.012 Y111.44 E.00479
164 G1 E-2.8 F3600
165 ;WIPE_START
166 G1 F900
167 G1 X115.949 Y111.295 E.-18986
168 G1 X115.459 Y111.223 E.-59469
169 G1 X115.113 Y111.211 E.-41545
170 ;WIPE_END
171 G1 Z.6 F9000
172 G1 X116 Y115.995 Z.6
173 G1 Z.2
174 G1 E4 F2400
175 ;WIDTH:0.38292
176 G1 F900
177 G1 X116 Y111.479 E.12513
178 G1 E-2.8 F3600
179 M73 P36 R1
180 ;WIPE_START
181 M73 P36 R1
182 G1 F900
183 G1 X116 Y112.479 E-1.2
184 ;WIPE_END
185 G1 Z.6 F9000
186 G1 X110.831 Y111.204 Z.6
187 G1 Z.2
188 G1 E4 F2400
189 ;WIDTH:0.592342
190 G1 F900
191 G1 X110.555 Y111.224 E.0124
192 ;WIDTH:0.552763
193 G1 X110.279 Y111.243 E.01151
194 ;WIDTH:0.554051
195 G1 X110.209 Y111.264 E.00303
196 ;WIDTH:0.594918
197 G1 X110.139 Y111.285 E.00327
198 ;WIDTH:0.635785
199 G1 X110.07 Y111.306 E.00351
200 ;WIDTH:0.676652
201 G1 X110 Y111.327 E.00376
202 G1 E-2.8 F3600
203 ;WIPE_START
204 G1 F900
205 G1 X110.07 Y111.306 E-.10332
206 G1 X110.139 Y111.285 E-.10332
207 G1 X110.209 Y111.264 E-.10332
208 G1 X110.279 Y111.243 E-.10332
209 G1 X110.555 Y111.224 E-.39337
210 G1 X110.831 Y111.204 E-.39337
211 ;WIPE_END
212 G1 Z.6 F9000
213 G1 X110 Y115.981 Z.6
214 G1 Z.2
215 M73 P37 R1
216 M73 P37 R1
217 G1 E4 F2400

```

```

218 ;WIDTH:0.638326
219 G1 F900
220 G1 X110 Y113.654 E.11289
221 ;WIDTH:0.676652
222 G1 X110 Y111.327 E.12016
223 G1 X109.899 Y111.305 E.00532
224 ;WIDTH:0.638054
225 G1 X109.799 Y111.283 E.00499
226 ;WIDTH:0.599455
227 G1 X109.698 Y111.261 E.00467
228 ;WIDTH:0.560857
229 G1 X109.598 Y111.239 E.00435
230 G1 E-2.8 F3600
231 ;WIPE_START
232 G1 F900
233 G1 X109.698 Y111.261 E.-12359
234 G1 X109.799 Y111.283 E.-12359
235 G1 X109.899 Y111.305 E.-12359
236 G1 X110 Y111.327 E.-12359
237 G1 X110 Y111.916 E.-70564
238 ;WIPE_END
239 G1 Z.6 F9000
240 G1 X113 Y115.988 Z.6
241 G1 Z.2
242 G1 E4 F2400
243 M73 P38 R1
244 ;WIDTH:0.400002
245 M73 P38 R1
246 G1 F900
247 G1 X113 Y111.481 E.13115
248 G1 E-2.8 F3600
249 ;WIPE_START
250 G1 F900
251 G1 X113 Y112.481 E-1.2
252 ;WIPE_END
253 G1 Z.6 F9000
254 G1 X109.313 Y103.781 Z.6
255 G1 Z.2
256 G1 E4 F2400
257 ;WIDTH:0.562994
258 G1 F900
259 M73 P39 R1
260 M73 P39 R1
261 G1 X109.572 Y103.767 E.01101
262 ;WIDTH:0.534714
263 G1 X109.831 Y103.753 E.01041
264 ;WIDTH:0.536148
265 G1 X110.513 Y103.768 E.02742
266 ;WIDTH:0.542126
267 G1 X110.557 Y103.771 E.00179
268 G1 E-2.8 F3600
269 ;WIPE_START
270 G1 F900
271 G1 X110.513 Y103.768 E.-05267
272 G1 X109.831 Y103.753 E.-81865
273 G1 X109.572 Y103.767 E.-31173
274 G1 X109.558 Y103.768 E.-01695
275 ;WIPE_END
276 G1 Z.6 F9000
277 G1 X107.094 Y111.006 Z.6
278 G1 Z.2
279 G1 E4 F2400
280 ;TYPE: Inner wall
281 ;WIDTH:0.419999
282 G1 F900
283 G1 X106.999 Y111.065 E.00342
284 G1 X106.87 Y110.971 E.00491
285 G1 X106.312 Y110.876 E.01737
286 G1 X105.763 Y110.818 E.01698
287 G1 X104.381 Y110.813 E.04247
288 G1 X104.187 Y111.006 E.0084
289 G1 X104.187 Y104.186 E.20956
290 G1 X107.459 Y104.182 E.10052
291 M73 P40 R1
292 M73 P40 R1
293 G1 X106.863 Y104.729 E.02484
294 G1 X106.504 Y105.198 E.01814
295 G1 X106.236 Y105.729 E.01828
296 G1 X106.016 Y106.224 E.01663
297 G1 X105.825 Y107.205 E.03071
298 G1 X105.832 Y107.887 E.02097
299 G1 X105.958 Y108.581 E.02166

```

```

300 | G1 X106.258 Y109.36 E.02567
301 | G1 X106.411 Y109.637 E.00972
302 | G1 X106.71 Y110.086 E.01655
303 | G1 X107.175 Y110.584 E.02096
304 | G1 X107.584 Y110.893 E.01574
305 | G1 X107.195 Y110.943 E.01203
306 | M73 P41 R1
307 | M73 P41 R1
308 | G1 X107.128 Y110.985 E.00244
309 | G1 X107.289 Y111.312 F9000
310 ; TYPE: Outer wall
311 | G1 F900
312 | G1 X107.189 Y111.479 E.00599
313 | G1 X107.189 Y115.974 E.13812
314 | G1 X107.158 Y116.219 E.00758
315 | G1 X106.842 Y116.219 E.00972
316 | G1 X106.811 Y115.986 E.0072
317 | G1 X106.811 Y111.48 E.13846
318 | G1 X106.71 Y111.313 E.00601
319 | G1 X106.271 Y111.251 E.01363
320 | G1 X105.761 Y111.195 E.01574
321 | G1 X104.379 Y111.19 E.04247
322 ;WIDTH:0.42146
323 | G1 X104.189 Y111.378 E.00826
324 | G1 X104.189 Y116.189 E.1484
325 | G1 X103.811 Y116.189 E.01168
326 | M73 P42 R1
327 | M73 P42 R1
328 | G1 X103.81 Y110.618 E.17184
329 ;WIDTH:0.419999
330 | G1 X103.81 Y103.81 E.20922
331 | G1 X107.512 Y103.802 E.11375
332 ;WIDTH:0.446963
333 | G1 X108.027 Y103.784 E.01697
334 | G1 X108.525 Y103.71 E.01657
335 | M73 P43 R1
336 ;WIDTH:0.467445
337 | M73 P43 R1
338 | G1 X108.778 Y103.738 E.0088
339 ;WIDTH:0.51522
340 | G1 X109.03 Y103.766 E.00979
341 ;WIDTH:0.562994
342 | G1 X109.283 Y103.795 E.01078
343 ;WIDTH:0.551374
344 | G1 X108.903 Y103.932 E.01676
345 ;WIDTH:0.515766
346 | G1 X108.523 Y104.07 E.01558
347 ;WIDTH:0.480158
348 | G1 X108.142 Y104.208 E.01441
349 ;WIDTH:0.444549
350 | G1 X107.619 Y104.534 E.0202
351 ;WIDTH:0.419999
352 | G1 X107.163 Y104.959 E.01914
353 | G1 X106.783 Y105.46 E.01932
354 | G1 X106.385 Y106.3 E.02856
355 | G1 X106.202 Y107.205 E.02836
356 | G1 X106.204 Y107.826 E.01908
357 | G1 X106.31 Y108.445 E.01931
358 | G1 X106.645 Y109.305 E.02836
359 | G1 X106.985 Y109.828 E.01916
360 | G1 X107.466 Y110.338 E.02153
361 | G1 X107.88 Y110.643 E.01581
362 | G1 X108.195 Y110.804 E.01088
363 | G1 X109.043 Y111.084 E.02743
364 ;WIDTH:0.454086
365 | G1 X109.228 Y111.136 E.00643
366 ;WIDTH:0.488172
367 | G1 X109.413 Y111.187 E.00697
368 ;WIDTH:0.522258
369 | G1 X109.598 Y111.239 E.0075
370 | G1 X109.326 Y111.256 E.01065
371 ;WIDTH:0.488172
372 | G1 X109.054 Y111.273 E.00989
373 ;WIDTH:0.454086
374 | G1 X108.782 Y111.29 E.00913
375 ;WIDTH:0.419999
376 | G1 X107.791 Y111.253 E.03047
377 | G1 X107.329 Y111.307 E.01428
378 | G1 E-2.8 F3600
379 ;WIPE_START
380 | G1 F900
381 | G1 X107.189 Y111.479 E-.26622

```

```

382 G1 X107.189 Y112.257 E-.93378
383 ;WIPE_END
384 G1 Z.6 F9000
385 G1 X106.739 Y104.718
386 M73 P44 R1
387 M73 P44 R1
388 G1 Z.6
389 G1 Z.2
390 G1 E4 F2400
391 ; TYPE: Bottom surface
392 ;WIDTH:0.42906
393 G1 F2100
394 G1 X106.485 Y104.464 E.01132
395 G1 X105.94 Y104.465 E.01715
396 G1 X106.378 Y104.902 E.01946
397 G1 X106.277 Y105.03 E.00509
398 G1 X106.161 Y105.232 E.00735
399 G1 X105.395 Y104.466 E.03408
400 G1 X104.85 Y104.467 E.01715
401 G1 X105.983 Y105.601 E.05043
402 G1 X105.806 Y105.969 E.01287
403 G1 X104.47 Y104.633 E.05945
404 G1 X104.47 Y105.179 E.01718
405 G1 X105.691 Y106.4 E.05432
406 G1 X105.597 Y106.852 E.01454
407 G1 X104.47 Y105.725 E.05016
408 G1 X104.47 Y106.271 E.01718
409 M73 P45 R1
410 M73 P45 R1
411 G1 X105.542 Y107.343 E.04771
412 G1 X105.542 Y107.802 E.01444
413 G1 X105.557 Y107.905 E.00327
414 G1 X104.47 Y106.817 E.0484
415 G1 X104.47 Y107.363 E.01718
416 G1 X105.658 Y108.551 E.05286
417 G1 X105.977 Y109.417 E.02902
418 G1 X104.47 Y107.91 E.06706
419 G1 X104.47 Y108.456 E.01718
420 G1 X106.643 Y110.629 E.09672
421 G1 X106.032 Y110.564 E.01935
422 G1 X104.47 Y109.002 E.0695
423 G1 X104.47 Y109.548 E.01718
424 G1 X105.458 Y110.536 E.04395
425 G1 X104.912 Y110.536 E.01718
426 G1 X104.3 Y109.924 E.02721
427 G1 E-2.8 F3600
428 ;WIPE_START
429 G1 F2100
430 G1 X104.912 Y110.536 E-1.03773
431 G1 X105.047 Y110.536 E-.16227
432 M73 P46 R1
433 ;WIPE_END
434 M73 P46 R1
435 G1 Z.6 F9000
436 G1 X115.701 Y105.07 Z.6
437 G1 Z.2
438 G1 E4 F2400
439 ;WIDTH:0.423234
440 G1 F2100
441 G1 X115.101 Y104.47 E.02631
442 G1 X114.563 Y104.469 E.01667
443 G1 X115.532 Y105.438 E.04246
444 G1 X115.532 Y105.977 E.01668
445 G1 X114.025 Y104.469 E.06606
446 G1 X113.487 Y104.469 E.01667
447 G1 X115.533 Y106.515 E.08966
448 G1 X115.533 Y107.053 E.01668
449 G1 X113.956 Y105.476 E.06913
450 G1 X114.193 Y105.957 E.01662
451 M73 P47 R1
452 M73 P47 R1
453 G1 X114.315 Y106.373 E.01346
454 G1 X115.533 Y107.591 E.05339
455 G1 X115.534 Y108.13 E.01668
456 G1 X114.425 Y107.021 E.04859
457 G1 X114.468 Y107.349 E.01026
458 G1 X114.448 Y107.582 E.00723
459 G1 X115.534 Y108.668 E.0476
460 G1 X115.535 Y109.206 E.01668
461 G1 X114.407 Y108.079 E.04941
462 G1 X114.382 Y108.381 E.00939
463 G1 X114.336 Y108.545 E.00529

```

```

464 G1 X115.535 Y109.745 E.05258
465 G1 X115.555 Y110.303 E.0173
466 G1 X114.216 Y108.963 E.0587
467 G1 X114.131 Y109.195 E.00765
468 G1 X114.054 Y109.339 E.00507
469 G1 X115.272 Y110.557 E.05336
470 G1 X114.715 Y110.538 E.01727
471 G1 X113.867 Y109.69 E.03715
472 G1 X113.675 Y110.036 E.01226
473 G1 X114.183 Y110.543 E.02223
474 G1 X113.664 Y110.562 E.0161
475 G1 X113.318 Y110.217 E.01515
476 ; stop printing object Slicer Test v3.stl id:0 copy 0
477 M106 S255
478 ;LAYER_CHANGE
479 ;Z:0.4
480 ;HEIGHT:0.2
481 ;BEFORE_LAYER_CHANGE
482 ;0.4
483 G92 E0
484
485 G1 E-2.8 F3600
486 ;WIPE_START
487 G1 F2100
488 M73 P48 R1
489 M73 P48 R1
490 G1 X113.664 Y110.562 E-.58669
491 G1 X114.174 Y110.544 E-.61331
492 ;WIPE_END
493 ;_SET_FAN_SPEED_CHANGING_LAYER
494 ; printing object Slicer Test v3.stl id:0 copy 0
495 G1 Z.8 F9000
496 G1 X112.354 Y104.132 Z.8
497 G1 Z.4
498 G1 E4 F2400
499 ; TYPE: Inner wall
500 ;WIDTH:0.449999
501 G1 F2077
502 G1 X115.868 Y104.132 E.11656
503 G1 X115.868 Y110.868 E.22344
504 G1 X112.349 Y110.868 E.11672
505 G1 X112.647 Y110.668 E.01188
506 M73 P49 R1
507 M73 P49 R1
508 G1 X113.208 Y110.102 E.02644
509 G1 X113.588 Y109.546 E.02234
510 G1 X113.873 Y108.936 E.02234
511 G1 X114.057 Y108.264 E.02313
512 G1 X114.125 Y107.282 E.03265
513 G1 X114.034 Y106.615 E.02234
514 G1 X113.837 Y105.971 E.02232
515 G1 X113.537 Y105.367 E.02236
516 G1 X113.144 Y104.821 E.02232
517 G1 X112.743 Y104.412 E.01903
518 G1 X112.387 Y104.155 E.01455
519 G1 X110.831 Y103.857 F9000
520 ; TYPE: Outer wall
521 ;WIDTH:0.542128
522 G1 F2077
523 G1 X110.557 Y103.771 E.0117
524 G1 X110.844 Y103.748 E.0117
525 ;WIDTH:0.496064
526 G1 X111.13 Y103.725 E.01062
527 ;WIDTH:0.449999
528 G1 X116.275 Y103.725 E.17066
529 G1 X116.275 Y111.275 E.25045
530 G1 X116.045 Y111.278 E.00763
531 G1 X116.023 Y111.378 E.00341
532 ;WIDTH:0.41646
533 G1 X116 Y111.479 E.00313
534 ;WIDTH:0.4225
535 G1 X115.97 Y111.334 E.00457
536 ;WIDTH:0.449999
537 G1 X115.921 Y111.275 E.00253
538 G1 X113.135 Y111.28 E.09242
539 G1 X113 Y111.479 E.00798
540 G1 X112.903 Y111.292 E.00699
541 G1 X112.821 Y111.275 E.00275
542 G1 X111.131 Y111.275 E.05608
543 ;WIDTH:0.49338
544 G1 X110.844 Y111.252 E.01056
545 ;WIDTH:0.536761

```

```

546 G1 X110.557 Y111.229 E.01158
547 G1 X110.847 Y111.138 E.01225
548 ;WIDTH:0.49338
549 G1 X111.138 Y111.046 E.01117
550 ;WIDTH:0.449999
551 G1 X111.705 Y110.81 E.0204
552 G1 X112.451 Y110.304 E.02991
553 M73 P50 R1
554 M73 P50 R1
555 G1 X112.874 Y109.868 E.02014
556 G1 X113.221 Y109.37 E.02014
557 G1 X113.482 Y108.822 E.02014
558 G1 X113.651 Y108.23 E.0204
559 G1 X113.721 Y107.332 E.02991
560 G1 X113.644 Y106.729 E.02014
561 G1 X113.47 Y106.148 E.02013
562 G1 X113.204 Y105.601 E.02017
563 G1 X112.853 Y105.106 E.02013
564 G1 X112.421 Y104.671 E.02035
565 G1 X111.735 Y104.207 E.02749
566 G1 X111.106 Y103.944 E.02262
567 ;WIDTH:0.496064
568 G1 X110.87 Y103.869 E.00914
569 G1 E-2.8 F3600
570 ;WIPE_START
571 G1 F2400
572 G1 X110.557 Y103.771 E.-.39302
573 G1 X110.844 Y103.748 E.-.34502
574 G1 X111.13 Y103.725 E.-.34502
575 G1 X111.228 Y103.725 E.-.11693
576 ;WIPE_END
577 G1 Z.8 F9000
578 G1 X116 Y116.4 Z.8
579 M73 P50 R0
580 M73 P50 R0
581 G1 Z.4
582 G1 E4 F2400
583 ;WIDTH:0.38292
584 G1 F2077
585 G1 X116 Y111.479 E.13635
586 G1 E-2.8 F3600
587 M73 P51 R0
588 ;WIPE_START
589 M73 P51 R0
590 G1 F2400
591 G1 X116 Y112.479 E-1.2
592 ;WIPE_END
593 G1 Z.8 F9000
594 G1 X110 Y111.332 Z.8
595 G1 Z.4
596 G1 E4 F2400
597 ;WIDTH:0.672612
598 G1 F2077
599 G1 X110.074 Y111.303 E.00409
600 ;WIDTH:0.623385
601 G1 X110.148 Y111.273 E.00377
602 ;WIDTH:0.574158
603 G1 X110.222 Y111.244 E.00345
604 ;WIDTH:0.539606
605 G1 X110.545 Y111.23 E.01305
606 G1 X109.331 Y111.22 F9000
607 ;WIDTH:0.560456
608 G1 F2077
609 G1 X109.721 Y111.24 E.0165
610 ;WIDTH:0.558581
611 G1 X109.791 Y111.263 E.00308
612 ;WIDTH:0.596591
613 G1 X109.861 Y111.286 E.00331
614 ;WIDTH:0.634602
615 G1 X109.93 Y111.309 E.00354
616 ;WIDTH:0.672612
617 G1 X110 Y111.332 E.00376
618 G1 X110 Y111.405 E.00377
619 ;WIDTH:0.636306
620 G1 X110 Y111.479 E.00355
621 ;WIDTH:0.6
622 G1 X110 Y116.2 E.21433
623 G1 E-2.8 F3600
624 ;WIPE_START
625 G1 F2400
626 M73 P52 R0
627 M73 P52 R0

```

```

628 G1 X110 Y115.2 E-1.2
629 ;WIPE_END
630 G1 Z.8 F9000
631 G1 X113 Y116.3 Z.8
632 G1 Z.4
633 G1 E4 F2400
634 ;WIDTH:0.4
635 G1 F2077
636 G1 X113 Y111.479 E.14029
637 G1 E-2.8 F3600
638 ;WIPE_START
639 G1 F2400
640 G1 X113 Y112.479 E-1.2
641 ;WIPE_END
642 G1 Z.8 F9000
643 G1 X110.557 Y103.771 Z.8
644 G1 Z.4
645 M73 P53 R0
646 M73 P53 R0
647 G1 E4 F2400
648 ;WIDTH:0.542128
649 G1 F2077
650 G1 X110.508 Y103.768 E.00198
651 ;WIDTH:0.535964
652 G1 X109.673 Y103.759 E.03356
653 ;WIDTH:0.554041
654 G1 X109.458 Y103.777 E.00902
655 ;WIDTH:0.590814
656 G1 X109.242 Y103.795 E.00967
657 ;WIDTH:0.627586
658 G1 X109.026 Y103.814 E.01031
659 G1 E-2.8 F3600
660 ;WIPE_START
661 G1 F2400
662 G1 X109.242 Y103.795 E-.25979
663 G1 X109.458 Y103.777 E-.25979
664 G1 X109.673 Y103.759 E-.25979
665 G1 X110.024 Y103.763 E-.42063
666 ;WIPE_END
667 G1 Z.8 F9000
668 G1 X104.132 Y111.214 Z.8
669 G1 Z.4
670 G1 E4 F2400
; TYPE: Inner wall
671 ;WIDTH:0.449999
672 G1 F2077
673 M73 P54 R0
674 M73 P54 R0
675 G1 X104.132 Y104.132 E.23493
676 G1 X107.638 Y104.132 E.11631
677 G1 X107.386 Y104.302 E.01009
678 G1 X106.887 Y104.789 E.02313
679 G1 X106.392 Y105.493 E.02854
680 G1 X106.288 Y105.702 E.00775
681 G1 X106.067 Y106.247 E.0195
682 G1 X105.88 Y107.213 E.03265
683 G1 X105.891 Y107.906 E.02299
684 G1 X106.08 Y108.795 E.03014
685 G1 X106.379 Y109.487 E.02499
686 G1 X106.763 Y110.063 E.02297
687 G1 X107.311 Y110.63 E.02616
688 G1 X107.633 Y110.868 E.0133
689 G1 X107.236 Y110.885 E.01319
690 G1 X107 Y111.007 E.00883
691 G1 X106.886 Y110.928 E.00462
692 G1 X106.621 Y110.868 E.00898
693 G1 X104.479 Y110.868 E.07108
694 G1 X104.16 Y111.186 E.01493
695 G1 X104.359 Y111.377 F9000
696 ; TYPE: Outer wall
697 ;WIDTH:0.52146
698 G1 F2077
699 G1 F2077
700 G1 X104.239 Y111.479 E.00612
701 G1 X104.239 Y116.239 E.18564
702 G1 X103.761 Y116.239 E.01866
703 G1 X103.761 Y111.479 E.18564
704 G1 X103.743 Y111 E.01867
705 M73 P55 R0
706 ;WIDTH:0.48573
707 M73 P55 R0
708 G1 X103.725 Y110.521 E.01728
709 ;WIDTH:0.449999

```

```

710 G1 X103.725 Y103.725 E.22545
711 G1 X108.402 Y103.725 E.15514
712 ;WIDTH:0.494396
713 G1 X108.558 Y103.747 E.0058
714 ;WIDTH:0.538793
715 G1 X108.714 Y103.769 E.00637
716 ;WIDTH:0.58319
717 G1 X108.87 Y103.792 E.00694
718 ;WIDTH:0.627586
719 G1 X109.026 Y103.814 E.00751
720 G1 X108.981 Y103.838 E.00242
721 ;WIDTH:0.604692
722 G1 X108.844 Y103.909 E.00709
723 ;WIDTH:0.566019
724 G1 X108.706 Y103.98 E.0066
725 ;WIDTH:0.527346
726 G1 X108.569 Y104.052 E.00612
727 ;WIDTH:0.488673
728 G1 X108.431 Y104.123 E.00563
729 ;WIDTH:0.449999
730 G1 X107.665 Y104.598 E.02991
731 G1 X107.22 Y105.023 E.0204
732 G1 X106.662 Y105.864 E.03348
733 G1 X106.465 Y106.329 E.01676
734 G1 X106.287 Y107.213 E.02991
735 G1 X106.291 Y107.826 E.02035
736 G1 X106.456 Y108.638 E.02749
737 G1 X106.721 Y109.266 E.02262
738 G1 X107.031 Y109.749 E.01901
739 G1 X107.655 Y110.388 E.02964
740 G1 X108.197 Y110.759 E.02176
741 G1 X108.75 Y111.008 E.02013
742 ;WIDTH:0.486818
743 G1 X108.943 Y111.079 E.00746
744 ;WIDTH:0.523637
745 G1 X109.137 Y111.149 E.00807
746 ;WIDTH:0.560456
747 G1 X109.331 Y111.22 E.00869
748 G1 X109.288 Y111.226 E.00181
749 ;WIDTH:0.548669
750 G1 X109.001 Y111.25 E.01187
751 ;WIDTH:0.499334
752 G1 X108.714 Y111.275 E.01072
753 ;WIDTH:0.449999
754 G1 X107.379 Y111.275 E.0443
755 G1 X107.238 Y111.332 E.00504
756 ;WIDTH:0.449612
757 G1 X107.189 Y111.479 E.00512
758 ;WIDTH:0.42146
759 G1 X107.189 Y116.289 E.14839
760 G1 X106.811 Y116.289 E.01168
761 G1 X106.811 Y111.479 E.14839
762 ;WIDTH:0.449612
763 G1 X106.762 Y111.332 E.00512
764 M73 P56 R0
765 ;WIDTH:0.449999
766 M73 P56 R0
767 G1 X106.621 Y111.275 E.00504
768 G1 X104.479 Y111.275 E.07108
769 ;WIDTH:0.48573
770 G1 X104.389 Y111.351 E.00422
771 G1 E-2.8 F3600
772 ;WIPE_START
773 G1 F2400
774 G1 X104.239 Y111.479 E-.23648
775 G1 X104.239 Y112.281 E-.96352
776 ;WIPE_END
777 G1 Z.8 F9000
778 G1 X106.434 Y110.563
779 G1 Z.8
780 G1 Z.4
781 G1 E4 F2400
782 ;TYPE: Sparse infill
783 ;WIDTH:0.45
784 G1 F2077
785 G1 X104.806 Y110.563 E.05401
786 G1 X104.437 Y110.194 E.01728
787 G1 X104.437 Y109.806 E.01289
788 G1 X105.706 Y108.537 E.0595
789 G1 X105.587 Y107.938 E.02026
790 G1 X105.574 Y107.209 E.0242
791 G1 X105.592 Y107.032 E.00588

```

```

792 | G1 X105.769 Y106.18 E.02888
793 | G1 X105.879 Y105.879 E.01062
794 | G1 X104.437 Y104.437 E.06764
795 | G1 X104.437 Y106.066 E.05401
796 | M73 P57 R0
797 | M73 P57 R0
798 | G1 E-3 F3600
799 | ;WIPE_START
800 | G1 X104.437 Y105.066 E-1
801 | ;WIPE_END
802 | G1 Z.8 F9000
803 | G1 X115.563 Y106.066 Z.8
804 | G1 Z.4
805 | G1 E4 F2400
806 | G1 F2077
807 | G1 X115.563 Y104.437 E.05401
808 | G1 X114.125 Y105.875 E.06746
809 | G1 X114.329 Y106.535 E.02293
810 | G1 X114.435 Y107.354 E.02738
811 | G1 X114.351 Y108.374 E.03394
812 | G1 X114.3 Y108.543 E.00586
813 | M73 P58 R0
814 | M73 P58 R0
815 | G1 X115.563 Y109.806 E.05924
816 | G1 X115.563 Y110.194 E.01289
817 | G1 X115.192 Y110.565 E.01737
818 | G1 X113.564 Y110.572 E.05401
819 | ; stop printing object Slicer Test v3.stl id:0 copy 0
820 | ;LAYER_CHANGE
821 | ;Z:0.6
822 | ;HEIGHT:0.2
823 | ;BEFORE_LAYER_CHANGE
824 | ;0.6
825 | G92 E0
826
827 | G1 E-3 F3600
828 | ;WIPE_START
829 | G1 X114.564 Y110.567 E-1
830 | ;WIPE_END
831 | ;_SET_FAN_SPEED_CHANGING_LAYER
832 | ; printing object Slicer Test v3.stl id:0 copy 0
833 | G1 Z1 F9000
834 | G1 X104.132 Y111.214 Z1
835 | G1 Z.6
836 | G1 E4 F2400
837 | ; TYPE: Inner wall
838 | ;WIDTH:0.449999
839 | G1 X104.132 Y104.132 E.23492
840 | G1 X115.868 Y104.132 E.3893
841 | M73 P59 R0
842 | M73 P59 R0
843 | G1 X115.868 Y110.868 E.22345
844 | G1 X113.066 Y110.868 E.09295
845 | G1 X113 Y110.949 E.00347
846 | G1 X112.944 Y110.88 E.00295
847 | G1 X112.821 Y110.868 E.0041
848 | G1 F1200
849 | G1 X110.942 Y110.868 E.06233
850 | G1 F840
851 | G1 X110.279 Y110.868 E.02199
852 | G1 X110.042 Y110.916 E.00802
853 | G1 F960
854 | G1 X110 Y110.979 E.00251
855 | G1 X109.958 Y110.916 E.00251
856 | G1 F840
857 | G1 X109.721 Y110.868 E.00802
858 | G1 X109.052 Y110.868 E.02219
859 | M106 S255
860 | G1 F1200
861 | G1 X107.379 Y110.868 E.0555
862 | G1 F2400
863 | G1 X107.134 Y110.919 E.0083
864 | G1 X107 Y111.124 E.00812
865 | G1 X106.866 Y110.919 E.00812
866 | G1 X106.621 Y110.868 E.0083
867 | G1 X104.479 Y110.868 E.07105
868 | M73 P60 R0
869 | M73 P60 R0
870 | G1 X104.16 Y111.186 E.01494
871 | G1 X104.359 Y111.377 F9000
872 | ; TYPE: Outer wall
873 | ;WIDTH:0.52146

```

```

874 G1 F2400
875 G1 X104.239 Y111.479 E.00612
876 G1 X104.239 Y116.239 E.18564
877 G1 X103.761 Y116.239 E.01866
878 G1 X103.761 Y111.479 E.18564
879 G1 X103.743 Y111 E.01867
880 ;WIDTH:0.48573
881 G1 X103.725 Y110.521 E.01728
882 ;WIDTH:0.449999
883 G1 X103.725 Y103.725 E.22545
884 G1 X116.275 Y103.725 E.41631
885 G1 X116.275 Y111.275 E.25045
886 G1 X116.045 Y111.278 E.00763
887 M73 P61 R0
888 M73 P61 R0
889 G1 X116.023 Y111.378 E.00341
890 ;WIDTH:0.41646
891 G1 X116 Y111.479 E.00313
892 ;WIDTH:0.4225
893 G1 X115.97 Y111.334 E.00457
894 ;WIDTH:0.449999
895 G1 X115.921 Y111.275 E.00253
896 G1 X113.168 Y111.275 E.09133
897 G1 X113 Y111.479 E.00876
898 G1 X112.862 Y111.279 E.00803
899 G1 X110.259 Y111.275 E.08637
900 ;WIDTH:0.4875
901 G1 X110.194 Y111.326 E.00298
902 ;WIDTH:0.525
903 G1 X110.129 Y111.377 E.00323
904 ;WIDTH:0.5625
905 G1 X110.065 Y111.428 E.00348
906 ;WIDTH:0.6
907 G1 X110 Y111.479 E.00374
908 G1 X109.95 Y111.432 E.00311
909 ;WIDTH:0.5625
910 G1 X109.9 Y111.385 E.0029
911 ;WIDTH:0.525
912 G1 X109.85 Y111.338 E.00269
913 ;WIDTH:0.4875
914 G1 X109.8 Y111.291 E.00248
915 ;WIDTH:0.449999
916 G1 X109.721 Y111.275 E.00267
917 G1 X107.338 Y111.275 E.07905
918 G1 X107.189 Y111.479 E.00837
919 ;WIDTH:0.42146
920 G1 X107.189 Y116.289 E.14839
921 G1 X106.811 Y116.289 E.01168
922 G1 X106.811 Y111.479 E.14839
923 ;WIDTH:0.449999
924 G1 X106.703 Y111.292 E.00714
925 G1 X106.621 Y111.275 E.00277
926 G1 X104.479 Y111.275 E.07108
927 ;WIDTH:0.48573
928 G1 X104.389 Y111.351 E.00422
929 G1 E-2.8 F3600
930 ;WIPE_START
931 G1 F2400
932 G1 X104.239 Y111.479 E-.23648
933 G1 X104.239 Y112.281 E-.96352
934 ;WIPE_END
935 G1 Z1 F9000
936 M73 P62 R0
937 M73 P62 R0
938 G1 X116 Y116.4 Z1
939 G1 Z.6
940 G1 E4 F2400
941 ;WIDTH:0.38292
942 G1 X116 Y111.479 E.13635
943 G1 E-2.8 F3600
944 ;WIPE_START
945 G1 F2400
946 G1 X116 Y112.479 E-1.2
947 ;WIPE_END
948 G1 Z1 F9000
949 G1 X113 Y116.3 Z1
950 G1 Z.6
951 M73 P63 R0
952 M73 P63 R0
953 G1 E4 F2400
954 ;WIDTH:0.4
955 G1 X113 Y111.479 E.14029

```

```

956 G1 E-2.8 F3600
957 ;WIPE_START
958 G1 F2400
959 G1 X113 Y112.479 E-1.2
960 ;WIPE_END
961 G1 Z1 F9000
962 G1 X110 Y116.2 Z1
963 G1 Z.6
964 G1 E4 F2400
965 ;WIDTH:0.6
966 G1 X110 Y111.479 E.21433
967 M73 P64 R0
968 M73 P64 R0
969 G1 E-2.8 F3600
970 ;WIPE_START
971 G1 F2400
972 G1 X110 Y112.479 E-1.2
973 ;WIPE_END
974 G1 Z1 F9000
975 G1 X113.873 Y110.58
976 G1 Z1
977 G1 Z.6
978 G1 E4 F2400
979 ; TYPE: Bridge
980 ;WIDTH:0.429646
981 G1 F1500
982 G1 X106.351 Y110.577 E.23705
983 G1 X106.059 Y110.193 E.01519
984 G1 X113.943 Y110.193 E.24847
985 M73 P65 R0
986 M73 P65 R0
987 G1 X114.18 Y109.807 E.01428
988 G1 X105.819 Y109.807 E.26347
989 G1 X105.627 Y109.42 E.01361
990 G1 X114.37 Y109.42 E.27553
991 G1 X114.52 Y109.033 E.01307
992 G1 X105.477 Y109.033 E.285
993 G1 X105.363 Y108.646 E.0127
994 M73 P66 R0
995 M73 P66 R0
996 G1 X114.634 Y108.646 E.29213
997 G1 X114.712 Y108.26 E.01244
998 G1 X105.285 Y108.26 E.2971
999 G1 X105.238 Y107.873 E.01227
1000 M73 P67 R0
1001 M73 P67 R0
1002 G1 X114.759 Y107.873 E.30001
1003 G1 X114.773 Y107.486 E.0122
1004 G1 X105.224 Y107.486 E.30093
1005 G1 X105.237 Y107.16 E.01028
1006 G1 X105.241 Y107.1 E.00192
1007 G1 X114.756 Y107.1 E.29987
1008 G1 X114.708 Y106.713 E.01228
1009 M73 P68 R0
1010 M73 P68 R0
1011 G1 X105.289 Y106.713 E.29682
1012 G1 X105.37 Y106.326 E.01245
1013 G1 X114.627 Y106.326 E.2917
1014 G1 X114.511 Y105.939 E.01272
1015 G1 X105.486 Y105.939 E.2844
1016 G1 X105.64 Y105.553 E.01311
1017 M73 P69 R0
1018 M73 P69 R0
1019 G1 X114.358 Y105.553 E.27475
1020 G1 X114.165 Y105.166 E.01363
1021 G1 X105.835 Y105.166 E.26249
1022 G1 X106.078 Y104.779 E.01439
1023 G1 X114.132 Y104.779 E.25382
1024 G1 X114.982 Y104.252 F9000
1025 M73 P70 R0
1026 ; TYPE: Top surface
1027 ;WIDTH:0.421038
1028 M73 P70 R0
1029 G1 F1800
1030 M106 S255
1031 G1 X115.748 Y105.018 E.0334
1032 G1 X115.748 Y105.553 F9000
1033 G1 F1800
1034 G1 X114.447 Y104.252 E.0567
1035 G1 X113.912 Y104.252 F9000
1036 G1 F1800
1037 G1 X115.748 Y106.088 E.08

```

```

1038 G1 X115.748 Y106.622 F9000
1039 G1 F1800
1040 G1 X114.52 Y105.394 E.05352
1041 G1 X114.82 Y106.229 F9000
1042 G1 F1800
1043 G1 X115.748 Y107.157 E.04044
1044 G1 X115.748 Y107.692 F9000
1045 G1 F1800
1046 G1 X114.95 Y106.894 E.03476
1047 G1 X114.985 Y107.463 F9000
1048 G1 F1800
1049 G1 X115.748 Y108.227 E.03328
1050 G1 X115.748 Y108.761 F9000
1051 G1 F1800
1052 G1 X114.962 Y107.975 E.03425
1053 G1 X114.896 Y108.444 F9000
1054 G1 F1800
1055 G1 X115.748 Y109.296 E.03713
1056 G1 X115.748 Y109.831 F9000
1057 M73 P71 R0
1058 M73 P71 R0
1059 G1 F1800
1060 G1 X114.794 Y108.876 E.0416
1061 G1 X114.66 Y109.277 F9000
1062 G1 F1800
1063 G1 X115.748 Y110.366 E.04742
1064 G1 X115.596 Y110.748 F9000
1065 G1 F1800
1066 G1 X114.499 Y109.651 E.04779
1067 G1 X114.315 Y110.001 F9000
1068 G1 F1800
1069 G1 X115.062 Y110.748 E.03255
1070 G1 X114.527 Y110.748 F9000
1071 G1 F1800
1072 G1 X114.107 Y110.329 E.01829
1073 G1 E-2.81299 F3600
1074 ;WIPE_START
1075 G1 F1800
1076 G1 X114.527 Y110.748 E-1.18701
1077 ;WIPE_END
1078 G1 Z1 F9000
1079 G1 X105.894 Y104.674 Z1
1080 G1 Z.6
1081 G1 E4 F2400
1082 ;WIDTH:0.420974
1083 G1 F1800
1084 M73 P72 R0
1085 M73 P72 R0
1086 G1 X105.472 Y104.252 E.01838
1087 G1 X104.938 Y104.252 F9000
1088 G1 F1800
1089 G1 X105.687 Y105.001 E.03265
1090 G1 X105.502 Y105.351 F9000
1091 G1 F1800
1092 G1 X104.403 Y104.252 E.0479
1093 G1 X104.252 Y104.635 F9000
1094 G1 F1800
1095 G1 X105.342 Y105.725 E.0475
1096 G1 X105.208 Y106.126 F9000
1097 G1 F1800
1098 G1 X104.252 Y105.17 E.04168
1099 G1 X104.252 Y105.704 F9000
1100 G1 F1800
1101 G1 X105.105 Y106.558 E.03717
1102 G1 X105.036 Y107.024 F9000
1103 G1 F1800
1104 G1 X104.252 Y106.239 E.03419
1105 G1 X104.252 Y106.774 F9000
1106 G1 F1800
1107 G1 X105.013 Y107.535 E.03317
1108 G1 X105.053 Y108.109 F9000
1109 G1 F1800
1110 G1 X104.252 Y107.308 E.0349
1111 G1 X104.252 Y107.843 F9000
1112 G1 F1800
1113 G1 X105.178 Y108.769 E.04034
1114 G1 X105.483 Y109.609 F9000
1115 G1 F1800
1116 G1 X104.252 Y108.378 E.05366
1117 G1 X104.252 Y108.912 F9000
1118 M73 P73 R0
1119 M73 P73 R0

```

```

1120 G1 F1800
1121 G1 X106.091 Y110.752 E.08014
1122 G1 X105.555 Y110.751 F9000
1123 G1 F1800
1124 G1 X104.252 Y109.447 E.0568
1125 G1 X104.252 Y109.982 F9000
1126 G1 F1800
1127 G1 X105.02 Y110.749 E.03346
1128 ; stop printing object Slicer Test v3.stl id:0 copy 0
1129 ;LAYER_CHANGE
1130 ;Z:0.8
1131 ;HEIGHT:0.2
1132 ;BEFORE_LAYER_CHANGE
1133 ;0.8
1134 G92 E0
1135
1136 G1 E-2.8 F3600
1137 ;WIPE_START
1138 G1 F1800
1139 G1 X104.312 Y110.042 E-1.2
1140 ;WIPE_END
1141 ;_SET_FAN_SPEED_CHANGING_LAYER
1142 ; printing object Slicer Test v3.stl id:0 copy 0
1143 G1 Z1.2 F9000
1144 G1 X107.865 Y109.411 Z1.2
1145 G1 Z.8
1146 G1 E4 F2400
1147 ; TYPE: Inner wall
1148 ;WIDTH:0.449999
1149 G1 F600
1150 G1 X107.591 Y109.052 E.01496
1151 G1 X107.352 Y108.588 E.01732
1152 G1 X107.171 Y107.956 E.02182
1153 M73 P74 R0
1154 M73 P74 R0
1155 G1 X107.135 Y107.512 E.01477
1156 G1 X107.224 Y106.794 E.02402
1157 G1 X107.368 Y106.366 E.01496
1158 G1 X107.62 Y105.909 E.01732
1159 G1 X108.036 Y105.416 E.02141
1160 G1 X108.612 Y104.994 E.02369
1161 G1 X108.842 Y104.889 E.00838
1162 G1 X109.473 Y104.683 E.02201
1163 G1 X109.993 Y104.637 E.01732
1164 G1 X110.647 Y104.709 E.02182
1165 G1 X111.08 Y104.846 E.01505
1166 G1 X111.542 Y105.088 E.01732
1167 G1 X112.05 Y105.5 E.02169
1168 G1 X112.372 Y105.898 E.01699
1169 G1 X112.576 Y106.287 E.01457
1170 G1 X112.739 Y106.69 E.01442
1171 G1 X112.862 Y107.629 E.03142
1172 G1 X112.831 Y107.88 E.00838
1173 G1 X112.737 Y108.317 E.01482
1174 G1 X112.472 Y108.949 E.02275
1175 G1 X112.045 Y109.494 E.02297
1176 G1 X111.663 Y109.823 E.01671
1177 M73 P75 R0
1178 M73 P75 R0
1179 G1 X111.094 Y110.148 E.02173
1180 G1 X110.395 Y110.337 E.02402
1181 G1 X109.945 Y110.365 E.01496
1182 G1 X109.427 Y110.305 E.01732
1183 G1 X108.801 Y110.102 E.02182
1184 G1 X108.413 Y109.885 E.01476
1185 G1 X107.896 Y109.437 E.02269
1186 G1 X107.567 Y109.691 F9000
1187 ; TYPE: Outer wall
1188 G1 F600
1189 G1 X107.246 Y109.271 E.01754
1190 G1 X106.972 Y108.739 E.01986
1191 G1 X106.769 Y108.029 E.02449
1192 G1 X106.726 Y107.504 E.01748
1193 G1 X106.825 Y106.702 E.02678
1194 G1 X106.994 Y106.202 E.01754
1195 G1 X107.283 Y105.677 E.01986
1196 G1 X107.757 Y105.116 E.02436
1197 G1 X108.405 Y104.641 E.02668
1198 G1 X108.887 Y104.421 E.01757
1199 G1 X109.401 Y104.281 E.01767
1200 G1 X109.998 Y104.228 E.01986
1201 M73 P76 R0

```

```

1202 M73 P76 R0
1203 G1 X110.731 Y104.309 E.02449
1204 G1 X111.237 Y104.468 E.01757
1205 G1 X111.767 Y104.746 E.01986
1206 G1 X112.34 Y105.21 E.02446
1207 G1 X112.714 Y105.673 E.01976
1208 G1 X113.057 Y106.327 E.02449
1209 G1 X113.205 Y106.833 E.01748
1210 G1 X113.271 Y107.638 E.02678
1211 G1 X113.206 Y108.163 E.01757
1212 G1 X113.056 Y108.676 E.01772
1213 G1 X112.825 Y109.156 E.01768
1214 G1 X112.518 Y109.593 E.01772
1215 G1 X112.144 Y109.975 E.01771
1216 G1 X111.714 Y110.29 E.01771
1217 G1 X111.241 Y110.53 E.01758
1218 G1 X110.462 Y110.741 E.02678
1219 G1 X109.934 Y110.774 E.01754
1220 G1 X109.34 Y110.704 E.01986
1221 G1 X108.637 Y110.477 E.02449
1222 G1 X108.177 Y110.22 E.01748
1223 M73 P77 R0
1224 M73 P77 R0
1225 G1 X107.597 Y109.717 E.02546
1226 G1 E-2.8 F3600
1227 ;WIPE_START
1228 G1 F2400
1229 G1 X107.246 Y109.271 E-.6815
1230 G1 X107.048 Y108.887 E-.5185
1231 ;WIPE_END
1232 G1 Z1.2 F9000
1233 G1 X108.762 Y105.263
1234 G1 Z1.2
1235 G1 Z.8
1236 G1 E4 F2400
1237 ; TYPE: Sparse infill
1238 ;WIDTH:0.45
1239 G1 F600
1240 G1 X109.158 Y105.085 E.01439
1241 G1 X107.595 Y106.648 E.07329
1242 G1 X107.523 Y106.862 E.0075
1243 G1 X107.45 Y107.45 E.01967
1244 G1 X110.053 Y110.053 E.12207
1245 M73 P78 R0
1246 M73 P78 R0
1247 G1 X109.943 Y110.057 E.00365
1248 G1 X112.542 Y107.458 E.12192
1249 G1 X112.501 Y106.964 E.01646
1250 G1 X112.41 Y106.653 E.01075
1251 G1 X110.85 Y105.093 E.07316
1252 G1 X110.584 Y105.009 E.00927
1253 G1 X109.99 Y104.944 E.01982
1254 G1 X109.527 Y104.985 E.01542
1255 G1 X109.354 Y105.032 E.00595
1256 ; stop printing object Slicer Test v3.stl id:0 copy 0
1257 ;LAYER_CHANGE
1258 ;Z:1
1259 ;HEIGHT:0.2
1260 ;BEFORE_LAYER_CHANGE
1261 ;1
1262 G92 E0
1263
1264 G1 E-3 F3600
1265 M73 P79 R0
1266 ;WIPE_START
1267 M73 P79 R0
1268 G1 X109.527 Y104.985 E.-.1793
1269 G1 X109.99 Y104.944 E.-.46483
1270 G1 X110.344 Y104.983 E.-.35587
1271 ;WIPE_END
1272 ;_SET_FAN_SPEED_CHANGING_LAYER
1273 ; printing object Slicer Test v3.stl id:0 copy 0
1274 G1 Z1.4 F9000
1275 G1 X108.401 Y109.875 Z1.4
1276 G1 Z1
1277 G1 E4 F2400
1278 ; TYPE: Inner wall
1279 ;WIDTH:0.449999
1280 G1 F911
1281 G1 X107.91 Y109.457 E.0214
1282 G1 X107.741 Y109.241 E.00908
1283 G1 X107.474 Y108.833 E.01617

```

```

1284 | G1 X107.22 Y108.195 E.02281
1285 | G1 X107.144 Y107.505 E.02301
1286 | G1 X107.186 Y107.01 E.01648
1287 | G1 X107.366 Y106.372 E.02198
1288 | G1 X107.695 Y105.802 E.02182
1289 | G1 X108.031 Y105.419 E.01692
1290 | G1 X108.612 Y104.994 E.02389
1291 | G1 X108.842 Y104.889 E.00838
1292 | G1 X109.467 Y104.685 E.02181
1293 | G1 X110.123 Y104.64 E.02182
1294 | M73 P80 R0
1295 | M73 P80 R0
1296 | G1 X110.641 Y104.707 E.01732
1297 | G1 X111.074 Y104.844 E.01505
1298 | G1 X111.65 Y105.161 E.02182
1299 | G1 X112.042 Y105.491 E.01699
1300 | G1 X112.443 Y106.007 E.02169
1301 | G1 X112.559 Y106.241 E.00866
1302 | G1 X112.735 Y106.676 E.01555
1303 | G1 X112.862 Y107.629 E.03191
1304 | G1 X112.831 Y107.88 E.00838
1305 | G1 X112.678 Y108.517 E.02173
1306 | G1 X112.343 Y109.145 E.02361
1307 | G1 X111.867 Y109.673 E.0236
1308 | G1 X111.508 Y109.936 E.01475
1309 | G1 X110.912 Y110.214 E.02182
1310 | G1 X110.648 Y110.278 E.00901
1311 | G1 X110.164 Y110.352 E.01622
1312 | G1 X109.678 Y110.338 E.01614
1313 | G1 X109.038 Y110.198 E.02173
1314 | G1 X108.437 Y109.894 E.02236
1315 | G1 X108.173 Y110.217 F9000
1316 | ; TYPE: Outer wall
1317 | G1 F911
1318 | G1 X107.615 Y109.74 E.02435
1319 | G1 X107.245 Y109.269 E.01986
1320 | G1 X106.995 Y108.799 E.01767
1321 | G1 X106.823 Y108.294 E.01771
1322 | G1 X106.737 Y107.766 E.01772
1323 | G1 X106.737 Y107.234 E.01768
1324 | G1 X106.823 Y106.706 E.01772
1325 | G1 X106.994 Y106.204 E.01761
1326 | G1 X107.362 Y105.564 E.02449
1327 | G1 X107.755 Y105.117 E.01973
1328 | G1 X108.405 Y104.641 E.02675
1329 | G1 X108.887 Y104.421 E.01757
1330 | M73 P81 R0
1331 | M73 P81 R0
1332 | G1 X109.399 Y104.282 E.01761
1333 | G1 X110.136 Y104.231 E.02449
1334 | G1 X110.73 Y104.308 E.01986
1335 | G1 X111.235 Y104.468 E.01757
1336 | G1 X111.882 Y104.823 E.02449
1337 | G1 X112.337 Y105.207 E.01976
1338 | G1 X112.789 Y105.79 E.02446
1339 | G1 X113.056 Y106.325 E.01986
1340 | G1 X113.205 Y106.833 E.01755
1341 | G1 X113.271 Y107.638 E.02678
1342 | G1 X113.206 Y108.163 E.01757
1343 | G1 X113.057 Y108.672 E.01758
1344 | G1 X112.679 Y109.381 E.02665
1345 | G1 X112.141 Y109.977 E.02664
1346 | G1 X111.716 Y110.288 E.01747
1347 | G1 X111.047 Y110.6 E.02449
1348 | G1 X110.465 Y110.741 E.01986
1349 | G1 X109.933 Y110.774 E.01767
1350 | G1 X109.402 Y110.719 E.01771
1351 | G1 X108.891 Y110.58 E.01758
1352 | G1 X108.209 Y110.235 E.02535
1353 | G1 E-2.8 F3600
1354 | ;WIPE_START
1355 | G1 F2400
1356 | G1 X107.615 Y109.74 E.-.92763
1357 | G1 X107.475 Y109.561 E.-.27237
1358 | ;WIPE_END
1359 | G1 Z1.4 F9000
1360 | G1 X111.19 Y105.044
1361 | M73 P82 R0
1362 | M73 P82 R0
1363 | G1 Z1.4
1364 | G1 Z1
1365 | G1 E4 F2400

```

```

1366 ; TYPE: Top surface
1367 ;WIDTH:0.410992
1368 G1 F911
1369 G1 X112.464 Y106.318 E.05403
1370 G1 X112.696 Y107.071 F9000
1371 G1 F911
1372 G1 X110.425 Y104.8 E.09633
1373 G1 X109.881 Y104.776 F9000
1374 G1 F911
1375 G1 X112.741 Y107.636 E.12128
1376 G1 X112.678 Y108.094 F9000
1377 G1 F911
1378 M73 P83 R0
1379 M73 P83 R0
1380 G1 X109.409 Y104.825 E.13865
1381 G1 X109.008 Y104.944 F9000
1382 G1 F911
1383 G1 X112.556 Y108.492 E.15048
1384 G1 X112.375 Y108.832 F9000
1385 G1 F911
1386 G1 X108.654 Y105.111 E.15781
1387 G1 X108.354 Y105.331 F9000
1388 G1 F911
1389 G1 X112.175 Y109.153 E.16209
1390 G1 X111.928 Y109.426 F9000
1391 M73 P84 R0
1392 M73 P84 R0
1393 G1 F911
1394 G1 X108.065 Y105.563 E.16388
1395 G1 X107.821 Y105.84 F9000
1396 G1 F911
1397 G1 X111.659 Y109.677 E.16277
1398 G1 X111.342 Y109.881 F9000
1399 G1 F911
1400 G1 X107.625 Y106.164 E.15769
1401 G1 X107.447 Y106.507 F9000
1402 M73 P85 R0
1403 M73 P85 R0
1404 G1 F911
1405 G1 X110.987 Y110.047 E.15016
1406 G1 X110.589 Y110.169 F9000
1407 G1 F911
1408 G1 X107.325 Y106.904 E.13847
1409 G1 X107.263 Y107.364 F9000
1410 G1 F911
1411 G1 X110.133 Y110.234 E.12173
1412 G1 X109.588 Y110.209 F9000
1413 M73 P86 R0
1414 M73 P86 R0
1415 G1 F911
1416 G1 X107.295 Y107.916 E.09725
1417 G1 X107.516 Y108.657 F9000
1418 G1 F911
1419 G1 X108.804 Y109.946 E.05466
1420 ; stop printing object Slicer Test v3.stl id:0 copy 0
1421 G1 E-2.8 F3600
1422 ;WIPE_START
1423 G1 F1800
1424 G1 X108.097 Y109.239 E-1.2
1425 ;WIPE_END
1426 M106 S0
1427 ; TYPE: Custom
1428 ; filament end gcode

```

Listing A.2.2 G-Code Generated Using Orca Slicer [63] (Main)

```

1429 G1 Z3 F600 ; Move print head up
1430 G1 X5 Y176 F9000 ; present print
1431 G1 Z71 F600 ; Move print head further up
1432 M73 P87 R0
1433 M73 P87 R0
1434 G1 Z150 F600 ; Move print head further up
1435 M73 P93 R0
1436 M140 S0 ; turn off heatbed
1437 M104 S0 ; turn off temperature
1438 M107 ; turn off fan
1439 M84 X Y E ; disable motors
1440 M73 P100 R0
1441 ; EXECUTABLE_BLOCK_END
1442 ; filament used [mm] = 46.37

```

```

1444 ; filament used [cm3] = 0.11
1445 ; filament used [g] = 0.14
1446 ; filament cost = 0.00
1447 ; total filament used [g] = 0.14
1448 ; total filament cost = 0.00
1449 ; total layers count = 5
1450 ; estimated printing time (normal mode) = 2m 0s
1451
1452 ; CONFIG_BLOCK_START
1453 ; accel_to_decel_enable = 1
1454 ; accel_to_decel_factor = 50%
1455 ; activate_airfiltration = 0
1456 ; activate_chamber_temp_control = 0
1457 ; adaptive_bed_mesh_margin = 0
1458 ; adaptive_pressure_advance = 0
1459 ; adaptive_pressure_advance_bridges = 0
1460 ; adaptive_pressure_advance_model = "0,0,0\n0,0,0"
1461 ; adaptive_pressure_advance_overhangs = 0
1462 ; additional_cooling_fan_speed = 70
1463 ; alternate_extra_wall = 0
1464 ; auxiliary_fan = 0
1465 ; bbl_calib_mark_logo = 1
1466 ; bbl_use_printhonst = 0
1467 ; bed_custom_model =
1468 ; bed_custom_texture =
1469 ; bed_exclude_area = 0x0
1470 ; bed_mesh_max = 99999,99999
1471 ; bed_mesh_min = -99999,-99999
1472 ; bed_mesh_probe_distance = 50,50
1473 ; before_layer_change_gcode = ;BEFORE_LAYER_CHANGE\n;[layer_z]\nG92 E0\n
1474 ; best_object_pos = 0.5,0.5
1475 ; bottom_shell_layers = 1
1476 ; bottom_shell_thickness = 0
1477 ; bottom_solid_infill_flow_ratio = 1
1478 ; bottom_surface_pattern = monotonic
1479 ; bridge_acceleration = 50%
1480 ; bridge_angle = 0
1481 ; bridge_density = 100%
1482 ; bridge_flow = 0.95
1483 ; bridge_no_support = 0
1484 ; bridge_speed = 25
1485 ; brim_ears_detection_length = 1
1486 ; brim_ears_max_angle = 125
1487 ; brim_object_gap = 0
1488 ; brim_type = auto_brim
1489 ; brim_width = 0
1490 ; chamber_temperature = 0
1491 ; change_extrusion_role_gcode =
1492 ; change_filament_gcode = M600
1493 ; close_fan_the_first_x_layers = 1
1494 ; complete_print_exhaust_fan_speed = 80
1495 ; cool_plate_temp = 60
1496 ; cool_plate_temp_initial_layer = 60
1497 ; cooling_tube_length = 5
1498 ; cooling_tube_retraction = 91.5
1499 ; counterbore_hole_bridging = none
1500 ; curr_bed_type = High Temp Plate
1501 ; default_acceleration = 0
1502 ; default_filament_colour = ""
1503 ; default_filament_profile = "Creality Generic PLA"
1504 ; default_jerk = 0
1505 ; default_print_profile = 0.20mm Standard @Creality Ender3
1506 ; deretraction_speed = 40
1507 ; detect_narrow_internal_solid_infill = 1
1508 ; detect_overhang_wall = 1
1509 ; detect_thin_wall = 1
1510 ; different_settings_to_system = bottom_shell_layers;skirt_loops;sparse_infill_density;;
    sparse_infill_pattern;top_shell_layers;top_shell_thickness;;
1511 ; disable_m73 = 0
1512 ; dont_filter_internal_bridges = disabled
1513 ; dont_slow_down_outer_wall = 0
1514 ; draft_shield = disabled
1515 ; during_print_exhaust_fan_speed = 60
1516 ; elefant_foot_compensation = 0.1
1517 ; elefant_foot_compensation_layers = 1
1518 ; emit_machine_limits_to_gcode = 1
1519 ; enable_arc_fitting = 0
1520 ; enable_filament_ramming = 1
1521 ; enable_long_retraction_when_cut = 0
1522 ; enable_overhang_bridge_fan = 1
1523 ; enable_overhang_speed = 1
1524 ; enable_pressure_advance = 0

```

```

1525 ; enable_prime_tower = 0
1526 ; enable_support = 0
1527 ; enforce_support_layers = 0
1528 ; eng_plate_temp = 60
1529 ; eng_plate_temp_initial_layer = 60
1530 ; ensure_vertical_shell_thickness = ensure_all
1531 ; exclude_object = 0
1532 ; extra_loading_move = -2
1533 ; extra_perimeters_on_overhangs = 0
1534 ; extruder_clearance_height_to_lid = 25
1535 ; extruder_clearance_height_to_rod = 25
1536 ; extruder_clearance_radius = 47
1537 ; extruder_colour = #26A69A
1538 ; extruder_offset = 0x0
1539 ; fan_cooling_layer_time = 100
1540 ; fan_kickstart = 0
1541 ; fan_max_speed = 100
1542 ; fan_min_speed = 100
1543 ; fan_speedup_overhangs = 1
1544 ; fan_speedup_time = 0
1545 ; filament_colour = #26A69A
1546 ; filament_cooling_final_speed = 3.4
1547 ; filament_cooling_initial_speed = 2.2
1548 ; filament_cooling_moves = 4
1549 ; filament_cost = 20
1550 ; filament_density = 1.24
1551 ; filament_diameter = 1.75
1552 ; filament_end_gcode = "; filament end gcode \n"
1553 ; filament_flow_ratio = 0.98
1554 ; filament_ids = GFL99
1555 ; filament_is_support = 0
1556 ; filament_loading_speed = 28
1557 ; filament_loading_speed_start = 3
1558 ; filament_max_volumetric_speed = 12
1559 ; filament_minimal_purge_on_wipe_tower = 15
1560 ; filament_multitool_ramming = 0
1561 ; filament_multitool_ramming_flow = 10
1562 ; filament_multitool_ramming_volume = 10
1563 ; filament_notes = ""
1564 ; filament_ramming_parameters = "120 100 6.6 6.8 7.2 7.6 7.9 8.2 8.7 9.4 9.9 10.0| 0.05 6.6
    0.45 6.8 0.95 7.8 1.45 8.3 1.95 9.7 2.45 10 2.95 7.6 3.45 7.6 3.95 7.6 4.45 7.6 4.95
    7.6"
1565 ; filament_settings_id = "Creality Generic PLA"
1566 ; filament_shrink = 100%
1567 ; filament_shrinkage_compensation_z = 100%
1568 ; filament_soluble = 0
1569 ; filament_stamping_distance = 0
1570 ; filament_stamping_loading_speed = 0
1571 ; filament_start_gcode = "; filament start gcode\n"
1572 ; filament_toolchange_delay = 0
1573 ; filament_type = PLA
1574 ; filament_unloading_speed = 90
1575 ; filament_unloading_speed_start = 100
1576 ; filament_vendor = Generic
1577 ; filename_format = {input_filename_base}_{filament_type[initial_tool]}_{print_time}.gcode
1578 ; filter_out_gap_fill = 0
1579 ; first_layer_print_sequence = 0
1580 ; flush_into_infill = 0
1581 ; flush_into_objects = 0
1582 ; flush_into_support = 1
1583 ; flush_multiplier = 0
1584 ; flush_volumes_matrix = 0
1585 ; flush_volumes_vector = 140,140
1586 ; full_fan_speed_layer = 0
1587 ; fuzzy_skin = none
1588 ; fuzzy_skin_first_layer = 0
1589 ; fuzzy_skin_point_distance = 0.8
1590 ; fuzzy_skin_thickness = 0.3
1591 ; gap_fill_target = nowhere
1592 ; gap_infill_speed = 30
1593 ; gcode_add_line_number = 0
1594 ; gcode_comments = 0
1595 ; gcode_flavor = marlin
1596 ; gcode_label_objects = 1
1597 ; has_scarf_joint_seam = 0
1598 ; head_wrap_detect_zone =
1599 ; high_current_on_filament_swap = 0
1600 ; hole_to_polyhole = 0
1601 ; hole_to_polyhole_threshold = 0.01
1602 ; hole_to_polyhole_twisted = 1
1603 ; host_type = octoprint
1604 ; hot_plate_temp = 60

```

```

1605 ; hot_plate_temp_initial_layer = 60
1606 ; idle_temperature = 0
1607 ; independent_support_layer_height = 1
1608 ; infill_anchor = 400%
1609 ; infill_anchor_max = 20
1610 ; infill_combination = 0
1611 ; infill_combination_max_layer_height = 100%
1612 ; infill_direction = 45
1613 ; infill_jerk = 9
1614 ; infill_wall_overlap = 25%
1615 ; initial_layer_acceleration = 0
1616 ; initial_layer_infill_speed = 35
1617 ; initial_layer_jerk = 9
1618 ; initial_layer_line_width = 0.42
1619 ; initial_layer_min_bead_width = 85%
1620 ; initial_layer_print_height = 0.2
1621 ; initial_layer_speed = 15
1622 ; initial_layer_travel_speed = 100%
1623 ; inner_wall_acceleration = 0
1624 ; inner_wall_jerk = 9
1625 ; inner_wall_line_width = 0.45
1626 ; inner_wall_speed = 40
1627 ; interface_shells = 0
1628 ; interlocking_beam = 0
1629 ; interlocking_beam_layer_count = 2
1630 ; interlocking_beam_width = 0.8
1631 ; interlocking_boundary_avoidance = 2
1632 ; interlocking_depth = 2
1633 ; interlocking_orientation = 22.5
1634 ; internal_bridge_flow = 1
1635 ; internal_bridge_speed = 150%
1636 ; internal_solid_infill_acceleration = 100%
1637 ; internal_solid_infill_line_width = 0
1638 ; internal_solid_infill_pattern = monotonic
1639 ; internal_solid_infill_speed = 50
1640 ; ironing_angle = -1
1641 ; ironing_flow = 15%
1642 ; ironing_pattern = zig-zag
1643 ; ironing_spacing = 0.1
1644 ; ironing_speed = 15
1645 ; ironing_type = no ironing
1646 ; is_infill_first = 0
1647 ; layer_change_gcode =
1648 ; layer_height = 0.2
1649 ; line_width = 0.45
1650 ; long_retractions_when_cut = 0
1651 ; machine_end_gcode = {if max_layer_z < printable_height}G1 Z{z_offset+min(max_layer_z+2,
    printable_height)} F600 ; Move print head up{endif}\nG1 X5 Y{print_bed_max[1]*0.8} F{
    travel_speed*60} ; present print\n{if max_layer_z < printable_height-10}G1 Z{z_offset+
    min(max_layer_z+70, printable_height-10)} F600 ; Move print head further up{endif}\n{if
    max_layer_z < max_print_height*0.6}G1 Z{printable_height*0.6} F600 ; Move print head
    further up{endif}\nM140 S0 ; turn off heatbed\nM104 S0 ; turn off temperature\nM107 ;
    turn off fan\nM84 X Y E ; disable motors
1652 ; machine_load_filament_time = 0
1653 ; machine_max_acceleration_e = 5000,5000
1654 ; machine_max_acceleration_extruding = 500,500
1655 ; machine_max_acceleration_retracting = 1000,1000
1656 ; machine_max_acceleration_travel = 1500,1250
1657 ; machine_max_acceleration_x = 500,500
1658 ; machine_max_acceleration_y = 500,500
1659 ; machine_max_acceleration_z = 100,100
1660 ; machine_max_jerk_e = 5,5
1661 ; machine_max_jerk_x = 8,8
1662 ; machine_max_jerk_y = 8,8
1663 ; machine_max_jerk_z = 0.4,0.4
1664 ; machine_max_speed_e = 60,60
1665 ; machine_max_speed_x = 500,500
1666 ; machine_max_speed_y = 500,500
1667 ; machine_max_speed_z = 10,10
1668 ; machine_min_extruding_rate = 0,0
1669 ; machine_min_travel_rate = 0,0
1670 ; machine_pause_gcode = M25
1671 ; machine_start_gcode = G90 ; use absolute coordinates\nM83 ; extruder relative mode\nM140 S[
    bed_temperature_initial_layer_single] ; set final bed temp\nM104 S150 ; set temporary
    nozzle temp to prevent oozing during homing\nG4 S10 ; allow partial nozzle warmup\nG28 ;
    home all axis\nG1 Z50 F240\nG1 X2 Y10 F3000\nM104 S[nozzle_temperature_initial_layer] ;
    set final nozzle temp\nM190 S[bed_temperature_initial_layer_single] ; wait for bed temp
    to stabilize\nM109 S[nozzle_temperature_initial_layer] ; wait for nozzle temp to
    stabilize\nG1 Z0.28 F240\nG92 E0\nG1 Y140 E10 F1500 ; prime the nozzle\nG1 X2.3 F5000\
    nG92 E0\nG1 Y10 E10 F1200 ; prime the nozzle\nG92 E0
1672 ; machine_tool_change_time = 0
1673 ; machine_unload_filament_time = 0

```

```

1674 ; make_overhang_printable = 0
1675 ; make_overhang_printable_angle = 55
1676 ; make_overhang_printable_hole_size = 0
1677 ; manual_filament_change = 0
1678 ; max_bridge_length = 10
1679 ; max_layer_height = 0.36
1680 ; max_travel_detour_distance = 0
1681 ; max_volumetric_extrusion_rate_slope = 0
1682 ; max_volumetric_extrusion_rate_slope_segment_length = 3
1683 ; min_bead_width = 85%
1684 ; min_feature_size = 25%
1685 ; min_layer_height = 0.08
1686 ; min_length_factor = 0.5
1687 ; min_skirt_length = 0
1688 ; min_width_top_surface = 300%
1689 ; minimum_sparse_infill_area = 10
1690 ; mmu_segmented_region_interlocking_depth = 0
1691 ; mmu_segmented_region_max_width = 0
1692 ; notes =
1693 ; nozzle_diameter = 0.4
1694 ; nozzle_height = 2.5
1695 ; nozzle_hrc = 0
1696 ; nozzle_temperature = 220
1697 ; nozzle_temperature_initial_layer = 220
1698 ; nozzle_temperature_range_high = 230
1699 ; nozzle_temperature_range_low = 190
1700 ; nozzle_type = undefined
1701 ; nozzle_volume = 0
1702 ; only_one_wall_first_layer = 0
1703 ; only_one_wall_top = 0
1704 ; ooze_prevention = 0
1705 ; other_layers_print_sequence = 0
1706 ; other_layers_print_sequence_nums = 0
1707 ; outer_wall_acceleration = 500
1708 ; outer_wall_jerk = 9
1709 ; outer_wall_line_width = 0.45
1710 ; outer_wall_speed = 40
1711 ; overhang_1_4_speed = 0
1712 ; overhang_2_4_speed = 20
1713 ; overhang_3_4_speed = 15
1714 ; overhang_4_4_speed = 10
1715 ; overhang_fan_speed = 100
1716 ; overhang_fan_threshold = 50%
1717 ; overhang_reverse = 0
1718 ; overhang_reverse_internal_only = 0
1719 ; overhang_reverse_threshold = 50%
1720 ; overhang_speed_classic = 0
1721 ; parking_pos_retraction = 92
1722 ; pellet_flow_coefficient = 0.4157
1723 ; pellet_modded_printer = 0
1724 ; post_process =
1725 ; precise_outer_wall = 0
1726 ; precise_z_height = 0
1727 ; preferred_orientation = 0
1728 ; preheat_steps = 1
1729 ; preheat_time = 30
1730 ; pressure_advance = 0.02
1731 ; prime_tower_brim_width = 3
1732 ; prime_tower_width = 60
1733 ; prime_volume = 45
1734 ; print_compatible_printers = "Creality Ender-3 0.4 nozzle"
1735 ; print_flow_ratio = 1
1736 ; print_order = default
1737 ; print_sequence = by layer
1738 ; print_settings_id = 0.20mm Standard @Creality Ender3
1739 ; printable_area = 0x0,220x0,220x220,0x220
1740 ; printable_height = 250
1741 ; printer_model = Creality Ender-3
1742 ; printer_notes =
1743 ; printer_settings_id = Creality Ender-3 0.4 nozzle
1744 ; printer_structure = i3
1745 ; printer_technology = FFFF
1746 ; printer_variant = 0.4
1747 ; printhost_authorization_type = key
1748 ; printhost_ssl_ignore_revocate = 0
1749 ; printing_by_object_gcode =
1750 ; purge_in_prime_tower = 1
1751 ; raft_contact_distance = 0.1
1752 ; raft_expansion = 1.5
1753 ; raft_first_layer_density = 90%
1754 ; raft_first_layer_expansion = 2
1755 ; raft_layers = 0

```

```

1756 ; reduce_crossing_wall = 0
1757 ; reduce_fan_stop_start_freq = 1
1758 ; reduce_infill_retraction = 1
1759 ; required_nozzle_HRC = 0
1760 ; resolution = 0.012
1761 ; retract_before_wipe = 70%
1762 ; retract_length_toolchange = 1
1763 ; retract_lift_above = 0
1764 ; retract_lift_below = 0
1765 ; retract_lift_enforce = All Surfaces
1766 ; retract_restart_extra = 0
1767 ; retract_restart_extra_toolchange = 0
1768 ; retract_when_changing_layer = 1
1769 ; retraction_distances_when_cut = 18
1770 ; retraction_length = 4
1771 ; retraction_minimum_travel = 2
1772 ; retraction_speed = 60
1773 ; role_based_wipe_speed = 1
1774 ; rotate_solid_infill_direction = 1
1775 ; scan_first_layer = 0
1776 ; scarf_angle_threshold = 155
1777 ; scarf_joint_flow_ratio = 1
1778 ; scarf_joint_speed = 100%
1779 ; scarf_overhang_threshold = 40%
1780 ; seam_gap = 10%
1781 ; seam_position = aligned
1782 ; seam_slope_conditional = 0
1783 ; seam_slope_entire_loop = 0
1784 ; seam_slope_inner_walls = 0
1785 ; seam_slope_min_length = 20
1786 ; seam_slope_start_height = 0
1787 ; seam_slope_steps = 10
1788 ; seam_slope_type = none
1789 ; silent_mode = 0
1790 ; single_extruder_multi_material = 1
1791 ; single_extruder_multi_material_priming = 0
1792 ; skirt_distance = 2
1793 ; skirt_height = 2
1794 ; skirt_loops = 0
1795 ; skirt_speed = 50
1796 ; skirt_start_angle = -135
1797 ; skirt_type = combined
1798 ; slice_closing_radius = 0.049
1799 ; slicing_mode = regular
1800 ; slow_down_for_layer_cooling = 1
1801 ; slow_down_layer_time = 8
1802 ; slow_down_layers = 0
1803 ; slow_down_min_speed = 10
1804 ; slowdown_for_curled_perimeters = 1
1805 ; small_area_infill_flow_compensation = 0
1806 ; small_area_infill_flow_compensation_model = 0,0;"\n0.2,0.4444";"\n0.4,0.6145";"\n0
    .6,0.7059";"\n0.8,0.7619";"\n1.5,0.8571";"\n2,0.8889";"\n3,0.9231";"\n5,0.9520";"\n10,1"
1807 ; small_perimeter_speed = 50%
1808 ; small_perimeter_threshold = 0
1809 ; solid_infill_direction = 45
1810 ; solid_infill_filament = 1
1811 ; sparse_infill_acceleration = 100%
1812 ; sparse_infill_density = 20%
1813 ; sparse_infill_filament = 1
1814 ; sparse_infill_line_width = 0.45
1815 ; sparse_infill_pattern = grid
1816 ; sparse_infill_speed = 60
1817 ; spiral_mode = 0
1818 ; spiral_mode_max_xy_smoothing = 200%
1819 ; spiral_mode_smooth = 0
1820 ; staggered_inner_seams = 0
1821 ; standby_temperature_delta = -5
1822 ; start_end_points = 30x-3,54x245
1823 ; support_airfiltration = 0
1824 ; support_angle = 0
1825 ; support_base_pattern = rectilinear
1826 ; support_base_pattern_spacing = 0.2
1827 ; support_bottom_interface_spacing = 0.5
1828 ; support_bottom_z_distance = 0.2
1829 ; support_chamber_temp_control = 0
1830 ; support_critical_regions_only = 0
1831 ; support_expansion = 0
1832 ; support_filament = 0
1833 ; support_interface_bottom_layers = -1
1834 ; support_interface_filament = 0
1835 ; support_interface_loop_pattern = 0
1836 ; support_interface_not_for_body = 1

```

```

1837 ; support_interface_pattern = auto
1838 ; support_interface_spacing = 0.2
1839 ; support_interface_speed = 80
1840 ; support_interface_top_layers = 3
1841 ; support_line_width = 0.38
1842 ; support_material_interface_fan_speed = -1
1843 ; support_multi_bed_types = 0
1844 ; support_object_xy_distance = 0.35
1845 ; support_on_build_plate_only = 0
1846 ; support_remove_small_overhang = 1
1847 ; support_speed = 40
1848 ; support_style = grid
1849 ; support_threshold_angle = 30
1850 ; support_top_z_distance = 0.15
1851 ; support_type = normal(auto)
1852 ; temperature_vitrification = 60
1853 ; template_custom_gcode =
1854 ; textured_cool_plate_temp = 40
1855 ; textured_cool_plate_temp_initial_layer = 40
1856 ; textured_plate_temp = 60
1857 ; textured_plate_temp_initial_layer = 60
1858 ; thick_bridges = 0
1859 ; thick_internal_bridges = 1
1860 ; thumbnails =
1861 ; thumbnails_format = PNG
1862 ; time_cost = 0
1863 ; time_lapse_gcode =
1864 ; timelapse_type = 0
1865 ; top_bottom_infill_wall_overlap = 25%
1866 ; top_shell_layers = 1
1867 ; top_shell_thickness = 0.2
1868 ; top_solid_infill_flow_ratio = 1
1869 ; top_surface_acceleration = 0
1870 ; top_surface_jerk = 9
1871 ; top_surface_line_width = 0.4
1872 ; top_surface_pattern = monotonicline
1873 ; top_surface_speed = 30
1874 ; travel_acceleration = 0
1875 ; travel_jerk = 12
1876 ; travel_slope = 3
1877 ; travel_speed = 150
1878 ; travel_speed_z = 0
1879 ; tree_support_adaptive_layer_height = 1
1880 ; tree_support_angle_slow = 25
1881 ; tree_support_auto_brim = 1
1882 ; tree_support_branch_angle = 40
1883 ; tree_support_branch_angle_organic = 40
1884 ; tree_support_branch_diameter = 5
1885 ; tree_support_branch_diameter_angle = 5
1886 ; tree_support_branch_diameter_double_wall = 3
1887 ; tree_support_branch_diameter_organic = 2
1888 ; tree_support_branch_distance = 5
1889 ; tree_support_branch_distance_organic = 1
1890 ; tree_support_brim_width = 3
1891 ; tree_support_tip_diameter = 0.8
1892 ; tree_support_top_rate = 30%
1893 ; tree_support_wall_count = 0
1894 ; upward_compatible_machine =
1895 ; use_firmware_retraction = 0
1896 ; use_relative_e_distances = 1
1897 ; wall_direction = auto
1898 ; wall_distribution_count = 1
1899 ; wall_filament = 1
1900 ; wall_generator = arachne
1901 ; wall_loops = 2
1902 ; wall_sequence = inner wall/outer wall
1903 ; wall_transition_angle = 10
1904 ; wall_transition_filter_deviation = 25%
1905 ; wall_transition_length = 100%
1906 ; wipe = 1
1907 ; wipe_before_external_loop = 0
1908 ; wipe_distance = 1
1909 ; wipe_on_loops = 0
1910 ; wipe_speed = 80%
1911 ; wipe_tower_bridging = 10
1912 ; wipe_tower_cone_angle = 0
1913 ; wipe_tower_extra_flow = 100%
1914 ; wipe_tower_extra_spacing = 100%
1915 ; wipe_tower_filament = 0
1916 ; wipe_tower_max_purge_speed = 90
1917 ; wipe_tower_no_sparse_layers = 0
1918 ; wipe_tower_rotation_angle = 0

```

```

1919 ; wipe_tower_x = 0.000
1920 ; wipe_tower_x = 0
1921 ; wipe_tower_y = 250.000
1922 ; wipe_tower_y = 250
1923 ; wiping_volumes_extruders = 70,70,70,70,70,70,70,70,70,70
1924 ; xy_contour_compensation = 0
1925 ; xy_hole_compensation = 0
1926 ; z_hop = 0.4
1927 ; z_hop_types = Normal Lift
1928 ; z_offset = 0
1929 ; first_layer_bed_temperature = 60
1930 ; bed_shape = 0x0,220x0,220x220,0x220
1931 ; first_layer_temperature = 220
1932 ; first_layer_height = 0.200
1933 ; CONFIG_BLOCK_END

```

Listing A.2.3 G-Code Generated Using Orca Slicer [63] (Reset)

A.3 Ultimaker Cura 5.8.0 – G-Code Listing

```

1 ;FLAVOR:Marlin
2 ;TIME:78
3 ;Filament used: 0.0294967m
4 ;Layer height: 0.2
5 ;MINX:111.2
6 ;MINY:111.2
7 ;MINZ:0.2
8 ;MAXX:123.8
9 ;MAXY:124.653
10 ;MAXZ:1
11 ;TARGET_MACHINE.NAME:Creatality Ender-3
12 ;Generated with Cura_SteamEngine 5.10.0
13 M140 S50
14 M105
15 M190 S50
16 M104 S200
17 M105
18 M109 S200
19 ; Ender 3 Custom Start G-code
20 G92 E0 ; Reset Extruder
21 G28 ; Home all axes
22 G1 Z2.0 F3000 ; Move Z Axis up little to prevent scratching of Heat Bed
23 G1 X0.1 Y20 Z0.3 F5000.0 ; Move to start position
24 G1 X0.1 Y200.0 Z0.3 F1500.0 E15 ; Draw the first line
25 G1 X0.4 Y200.0 Z0.3 F5000.0 ; Move to side a little
26 G1 X0.4 Y20 Z0.3 F1500.0 E30 ; Draw the second line
27 G92 E0 ; Reset Extruder
28 G1 Z2.0 F3000 ; Move Z Axis up little to prevent scratching of Heat Bed
29 G1 X5 Y20 Z0.3 F5000.0 ; Move over to prevent blob squish

```

Listing A.3.1 G-Code Generated Using Ultimaker Cura [97] (Initialization)

```

30 M82 ;absolute extrusion mode
31 G92 E0
32 G92 E0
33 G1 F2700 E-5
34 ;LAYER_COUNT:5
35 ;LAYER:0
36 M107
37 ;MESH:Slicer Test v3.stl
38 G0 F6000 X111.6 Y118.8 Z0.2
39 ; TYPE: WALL-INNER
40 G1 F2700 E0
41 G1 F1200 X112 Y118.4 E0.01881
42 G1 X114.159 Y118.403 E0.09062
43 G1 X114.331 Y118.445 E0.09651
44 G1 X114.5 Y118.558 E0.10327
45 G1 X114.639 Y118.458 E0.10897
46 G1 X114.846 Y118.401 E0.11611
47 G1 X115.209 Y118.399 E0.12818
48 G1 X114.917 Y118.185 E0.14022
49 G1 X114.663 Y117.96 E0.15151
50 G1 X114.447 Y117.734 E0.16191
51 G1 X114.263 Y117.518 E0.17135
52 G1 X114.06 Y117.233 E0.18298
53 G1 X113.867 Y116.902 E0.19573

```

```

54 G1 X113.724 Y116.602 E0.20678
55 G1 X113.62 Y116.334 E0.21634
56 G1 X113.511 Y115.962 E0.22924
57 G1 X113.447 Y115.634 E0.24035
58 G1 X113.412 Y115.314 E0.25106
59 G1 X113.4 Y114.979 E0.26221
60 G1 X113.415 Y114.638 E0.27356
61 G1 X113.447 Y114.362 E0.2828
62 G1 X113.511 Y114.036 E0.29385
63 G1 X113.603 Y113.712 E0.30505
64 G1 X113.724 Y113.394 E0.31637
65 G1 X113.867 Y113.094 E0.32742
66 G1 X114.06 Y112.767 E0.34005
67 G1 X114.259 Y112.487 E0.35148
68 G1 X114.471 Y112.236 E0.36241
69 G1 X114.663 Y112.036 E0.37163
70 G1 X114.917 Y111.811 E0.38291
71 G1 X115.192 Y111.604 E0.39436
72 G1 X111.601 Y111.601 E0.5138
73 G1 X111.6 Y118.8 E0.75324
74 G1 F2700 E-4.24676
75 G0 F6000 X111.49 Y118.8
76 G0 X115.937 Y118.586
77 G0 X116.533 Y117.666
78 G0 X118.576 Y117.622
79 G0 X120.5 Y118.437
80 G1 F2700 E0.75324
81 G1 F1200 X120.642 Y118.402 E0.7581
82 G1 X123.4 Y118.4 E0.84983
83 G1 X123.4 Y111.6 E1.076
84 G1 X119.81 Y111.6 E1.19541
85 G1 X120.195 Y111.911 E1.21187
86 G1 X120.444 Y112.144 E1.22321
87 G1 X120.669 Y112.399 E1.23452
88 G1 X120.865 Y112.659 E1.24535
89 G1 X121.031 Y112.92 E1.25564
90 G1 X121.267 Y113.351 E1.27198
91 G1 X121.425 Y113.814 E1.28825
92 G1 X121.508 Y114.146 E1.29963
93 G1 X121.564 Y114.466 E1.31044
94 G1 X121.594 Y114.81 E1.32192
95 G1 X121.594 Y115.191 E1.3346
96 G1 X121.565 Y115.523 E1.34568
97 G1 X121.516 Y115.816 E1.35556
98 G1 X121.425 Y116.172 E1.36778
99 G1 X121.269 Y116.635 E1.38403
100 G1 X121.009 Y117.119 E1.40231
101 G1 X120.861 Y117.343 E1.41124
102 G1 X120.673 Y117.597 E1.42175
103 G1 X120.437 Y117.856 E1.4334
104 G1 X120.168 Y118.11 E1.44571
105 G1 X119.804 Y118.399 E1.46116
106 G1 X120.32 Y118.4 E1.47833
107 G1 X120.5 Y118.437 E1.48444
108 G0 F6000 X120.5 Y119
109 ; TYPE: WALL_OUTER
110 G1 F1200 X120.625 Y118.8 E1.49228
111 G1 X123.44 Y118.804 E1.58591
112 G1 F1297.3 X123.5 Y119 E1.59222
113 G1 X123.56 Y118.804 E1.59852
114 G1 F1200 X123.8 Y118.8 E1.60651
115 G1 X123.8 Y111.2 E1.85928
116 G1 X118.991 Y111.2 E2.01923
117 G1 F1165 X118.664 Y111.212 E2.03044
118 G1 F948.6 X118.321 Y111.294 E2.04528
119 G1 X118.621 Y111.46 E2.05971
120 G1 F1165 X118.908 Y111.578 E2.07034
121 G1 F1200 X119.182 Y111.705 E2.08038
122 G1 X119.44 Y111.85 E2.09022
123 G1 X119.697 Y112.023 E2.10053
124 G1 X119.924 Y112.205 E2.11021
125 G1 X120.146 Y112.41 E2.12026
126 G1 X120.342 Y112.63 E2.13006
127 G1 X120.529 Y112.875 E2.14031
128 G1 X120.673 Y113.099 E2.14916
129 G1 X120.906 Y113.531 E2.16549
130 G1 X121.051 Y113.96 E2.18055
131 G1 X121.122 Y114.254 E2.19061
132 G1 X121.172 Y114.551 E2.20063
133 G1 X121.196 Y114.843 E2.21037
134 G1 X121.196 Y115.157 E2.22082
135 G1 X121.17 Y115.457 E2.23083

```

```

136 G1 X121.122 Y115.747 E2.24061
137 G1 X121.046 Y116.044 E2.2508
138 G1 X120.906 Y116.46 E2.2654
139 G1 X120.675 Y116.899 E2.2819
140 G1 X120.52 Y117.132 E2.29121
141 G1 X120.353 Y117.357 E2.30053
142 G1 X120.134 Y117.594 E2.31126
143 G1 X119.915 Y117.8 E2.32126
144 G1 X119.682 Y117.988 E2.33122
145 G1 X119.428 Y118.154 E2.34131
146 G1 X119.184 Y118.294 E2.35067
147 G1 X118.894 Y118.424 E2.36124
148 G1 X118.605 Y118.529 E2.37147
149 G1 X118.309 Y118.609 E2.38166
150 G1 F1048 X117.911 Y118.742 E2.39764
151 G1 X118.326 Y118.799 E2.4136
152 G1 F1200 X120.378 Y118.805 E2.48185
153 G1 X120.5 Y119 E2.4895
154 G1 X120.5 Y123.8 E2.64915
155 G1 F2700 E-2.35085
156 G0 F6000 X119.635 Y123.8
157 G0 X118.465 Y123.7
158 G0 X117.5 Y123.7
159 G1 F2700 E2.64915
160 G1 F781.8 X117.5 Y122.079 E2.73191
161 G1 F747.7 X117.5 Y120.458 E2.81844
162 G1 F716.4 X117.5 Y118.837 E2.90875
163 G1 F800 X117.214 Y118.742 E2.92378
164 G1 F890.5 X116.824 Y118.719 E2.94129
165 G0 F6000 X117.5 Y118.836
166 G1 F800 X117.751 Y118.747 E2.95458
167 G1 F930.2 X117.911 Y118.742 E2.96145
168 G1 F2700 E-2.03855
169 G0 F6000 X119.063 Y118.586
170 G0 X123.5 Y119
171 G1 F2700 E2.96145
172 G1 F1411.8 X123.5 Y123.9 E3.09997
173 G1 F2700 E-1.90003
174 G0 F6000 X122.735 Y123.9
175 G0 X121.353 Y124.653
176 G0 X116.547 Y124.653
177 G0 X115.565 Y123.8
178 G0 X114.7 Y123.8
179 G1 F2700 E3.09997
180 G1 F1200 X114.7 Y119 E3.25962
181 G1 X114.758 Y118.858 E3.26472
182 G1 X114.882 Y118.8 E3.26928
183 G1 X116.167 Y118.8 E3.31202
184 G1 F1165 X116.483 Y118.788 E3.32285
185 G1 F975.6 X116.824 Y118.719 E3.33708
186 G1 X116.515 Y118.578 E3.35098
187 G1 F1165 X115.953 Y118.363 E3.37159
188 G1 F1200 X115.701 Y118.235 E3.38099
189 G1 X115.43 Y118.066 E3.39161
190 G1 X115.18 Y117.883 E3.40192
191 G1 X114.951 Y117.683 E3.41203
192 G1 X114.751 Y117.474 E3.42165
193 G1 X114.567 Y117.258 E3.43109
194 G1 X114.379 Y116.989 E3.44201
195 G1 X114.227 Y116.728 E3.45205
196 G1 X114.097 Y116.457 E3.46205
197 G1 X113.993 Y116.189 E3.47161
198 G1 X113.903 Y115.884 E3.48219
199 G1 X113.846 Y115.599 E3.49185
200 G1 X113.812 Y115.297 E3.50196
201 G1 X113.8 Y114.995 E3.51201
202 G1 X113.813 Y114.684 E3.52237
203 G1 X113.845 Y114.408 E3.53161
204 G1 X113.907 Y114.1 E3.54206
205 G1 X113.993 Y113.807 E3.55221
206 G1 X114.097 Y113.539 E3.56178
207 G1 X114.227 Y113.268 E3.57177
208 G1 X114.379 Y113.01 E3.58173
209 G1 X114.566 Y112.743 E3.59257
210 G1 X114.759 Y112.513 E3.60256
211 G1 X114.951 Y112.313 E3.61178
212 G1 X115.194 Y112.102 E3.62249
213 G1 X115.444 Y111.92 E3.63277
214 G1 X115.7 Y111.766 E3.64271
215 G1 X115.972 Y111.626 E3.65288
216 G1 F1162.2 X116.515 Y111.417 E3.67286
217 G1 F973.6 X116.816 Y111.28 E3.68642

```

```

218 G1 X116.484 Y111.214 E3.7003
219 G1 F1162.2 X116.194 Y111.201 E3.71026
220 G1 F1200 X111.201 Y111.201 E3.87633
221 G1 X111.2 Y118 E4.10247
222 G1 F1066.7 X111.25 Y119 E4.13993
223 G1 F960 X111.25 Y123.75 E4.33741
224 G1 X111.75 Y123.75 E4.3582
225 G1 X111.75 Y119 E4.55568
226 G1 F1066.7 X112 Y118.8 E4.56766
227 G1 F1200 X114.12 Y118.801 E4.63818
228 G1 X114.242 Y118.858 E4.64265
229 G1 X114.3 Y119 E4.64776
230 G1 X114.3 Y123.8 E4.8074
231 G1 X114.7 Y123.8 E4.82071
232 G1 F2700 E-0.17929
233 G0 F6000 X116.816 Y111.28
234 G1 F2700 E4.82071
235 G1 F892.2 X117.126 Y111.258 E4.83461
236 G1 F944.9 X117.433 Y111.25 E4.84758
237 G1 F948.6 X117.867 Y111.256 E4.86585
238 G1 F872.7 X118.321 Y111.294 E4.88668
239 G1 F2700 E-0.11332
240 G0 F6000 X118.138 Y112.238
241 G0 X119.818 Y116.633
242 G0 X120.748 Y118.043
243 ; TYPE: SKIN
244 G1 F2700 E4.88668
245 G1 F1200 X120.954 Y117.819 E4.8968
246 G1 X121.169 Y117.527 E4.90886
247 G1 X121.339 Y117.271 E4.91908
248 G1 X121.602 Y116.769 E4.93793
249 G1 X121.766 Y116.285 E4.95493
250 G1 X121.87 Y115.875 E4.969
251 G1 X121.918 Y115.581 E4.97891
252 G1 X121.951 Y115.219 E4.991
253 G1 X121.956 Y114.843 E5.0035
254 G1 X121.92 Y114.411 E5.01792
255 G1 X121.869 Y114.121 E5.02772
256 G1 X121.762 Y113.691 E5.04245
257 G1 X121.602 Y113.219 E5.05903
258 G1 X121.344 Y112.743 E5.07704
259 G1 X121.17 Y112.465 E5.08795
260 G1 X120.951 Y112.178 E5.09995
261 G1 X120.76 Y111.96 E5.10959
262 G1 X123.04 Y111.96 E5.18543
263 G1 X123.04 Y118.04 E5.38765
264 G1 X120.748 Y118.043 E5.46388
265 G0 F6000 X121.44 Y117.48
266 G1 F1200 X121.801 Y117.841 E5.48086
267 G0 F6000 X122.366 Y117.84
268 G1 F1200 X121.644 Y117.119 E5.5148
269 G0 F6000 X121.822 Y116.731
270 G1 F1200 X122.84 Y117.749 E5.56268
271 G0 F6000 X122.84 Y117.183
272 G1 F1200 X121.965 Y116.308 E5.60384
273 G0 F6000 X122.076 Y115.853
274 G1 F1200 X122.84 Y116.617 E5.63977
275 G0 F6000 X122.84 Y116.052
276 G1 F1200 X122.139 Y115.35 E5.67277
277 G0 F6000 X122.152 Y114.798
278 G1 F1200 X122.84 Y115.486 E5.70513
279 G0 F6000 X122.84 Y114.92
280 G1 F1200 X122.078 Y114.159 E5.74095
281 G0 F6000 X122.84 Y114.355
282 G1 F1200 X121.868 Y113.383 E5.78667
283 G0 F6000 X122.84 Y113.223
284 G1 F1200 X121.777 Y112.16 E5.83667
285 G0 F6000 X121.221 Y112.17
286 G1 F1200 X122.84 Y113.789 E5.91282
287 G0 F6000 X123.3 Y113.789
288 G0 X123.3 Y112.658
289 G0 X122.84 Y112.658
290 G1 F1200 X122.342 Y112.16 E5.93625
291 G1 F2700 E0.93625
292 G0 F6000 X122.342 Y111.7
293 G0 X120.332 Y111.786
294 G0 X121.168 Y113.589
295 G0 X120.147 Y113.984
296 G0 X115.922 Y117.355
297 G0 X114.235 Y118.044
298 G1 F2700 E5.93625
299 G1 F1200 X111.96 Y118.04 E6.01191

```

```

300 G1 X111.96 Y111.96 E6.21414
301 G1 X114.225 Y111.963 E6.28947
302 G1 X113.983 Y112.25 E6.30196
303 G1 X113.727 Y112.617 E6.31684
304 G1 X113.538 Y112.943 E6.32937
305 G1 X113.295 Y113.471 E6.34871
306 G1 X113.155 Y113.951 E6.36534
307 G1 X113.084 Y114.324 E6.37796
308 G1 X113.052 Y114.656 E6.38906
309 G1 X113.04 Y114.967 E6.39941
310 G1 X113.052 Y115.327 E6.41139
311 G1 X113.093 Y115.714 E6.42433
312 G1 X113.155 Y116.032 E6.43511
313 G1 X113.307 Y116.564 E6.45351
314 G1 X113.539 Y117.051 E6.47145
315 G1 X113.727 Y117.383 E6.48414
316 G1 X113.927 Y117.671 E6.49581
317 G1 X114.235 Y118.044 E6.51189
318 G0 F6000 X114.662 Y118.107
319 G0 X114.539 Y118.334
320 G0 X112.749 Y118.301
321 G0 X112.75 Y117.841
322 G1 F1200 X112.16 Y117.251 E6.53965
323 G0 F6000 X112.16 Y116.685
324 G1 F1200 X113.317 Y117.843 E6.59409
325 G0 F6000 X113.636 Y117.595
326 G1 F1200 X112.16 Y116.12 E6.66349
327 G0 F6000 X112.16 Y115.554
328 G1 F1200 X113.071 Y116.465 E6.70634
329 G0 F6000 X112.892 Y115.721
330 G1 F1200 X112.16 Y114.988 E6.7408
331 G0 F6000 X112.16 Y114.423
332 G1 F1200 X112.846 Y115.108 E6.77304
333 G0 F6000 X112.861 Y114.558
334 G1 F1200 X112.16 Y113.857 E6.80602
335 G0 F6000 X112.16 Y113.291
336 G1 F1200 X112.932 Y114.063 E6.84233
337 G0 F6000 X113.046 Y113.612
338 G1 F1200 X112.16 Y112.726 E6.884
339 G0 F6000 X112.16 Y112.16
340 G1 F1200 X113.201 Y113.201 E6.93297
341 G0 F6000 X113.381 Y112.815
342 G1 F1200 X112.726 Y112.161 E6.96375
343 G0 F6000 X113.292 Y112.161
344 G1 F1200 X113.592 Y112.461 E6.97787
345 ;MESH:NONMESH
346 G0 F300 X113.592 Y112.461 Z0.4
347 G0 F6000 X113.968 Y112.726
348 G0 X113.294 Y115.696
349 G0 X114.699 Y118.257
350 G0 X111.785 Y118.436
351 G0 X111.786 Y118.144
352 G0 X111.49 Y118.8
353 G0 X111.6 Y118.8
354 ;TIME_ELAPSED:22.884674
355 ;LAYER:1
356 ;note -- min layer time used
357 M106 S85
358 ;TYPE:WALL-INNER
359 ;MESH:Slicer Test v3.stl
360 G1 F1201.7 X112 Y118.4 E6.99668
361 G1 X114.159 Y118.403 E7.06849
362 G1 X114.331 Y118.445 E7.07438
363 G1 X114.5 Y118.558 E7.08114
364 G1 X114.639 Y118.458 E7.08683
365 G1 X114.846 Y118.401 E7.09398
366 G1 X115.209 Y118.399 E7.10605
367 G1 X114.917 Y118.185 E7.11809
368 G1 X114.663 Y117.96 E7.12938
369 G1 X114.447 Y117.734 E7.13977
370 G1 X114.263 Y117.518 E7.14921
371 G1 X114.06 Y117.233 E7.16085
372 G1 X113.867 Y116.902 E7.17359
373 G1 X113.724 Y116.602 E7.18465
374 G1 X113.62 Y116.334 E7.19421
375 G1 X113.511 Y115.962 E7.2071
376 G1 X113.447 Y115.634 E7.21822
377 G1 X113.412 Y115.314 E7.22892
378 G1 X113.4 Y114.979 E7.24007
379 G1 X113.415 Y114.638 E7.25142
380 G1 X113.447 Y114.362 E7.26067
381 G1 X113.511 Y114.036 E7.27172

```

```

382 G1 X113.603 Y113.712 E7.28292
383 G1 X113.724 Y113.394 E7.29423
384 G1 X113.867 Y113.094 E7.30529
385 G1 X114.06 Y112.767 E7.31792
386 G1 X114.259 Y112.487 E7.32934
387 G1 X114.471 Y112.236 E7.34027
388 G1 X114.663 Y112.036 E7.34949
389 G1 X114.917 Y111.811 E7.36078
390 G1 X115.192 Y111.604 E7.37223
391 G1 X111.601 Y111.601 E7.49166
392 G1 X111.6 Y118.8 E7.7311
393 G1 F2700 E2.7311
394 G0 F7500 X111.49 Y118.8
395 G0 X115.937 Y118.586
396 G0 X116.533 Y117.666
397 G0 X118.576 Y117.622
398 G0 X119.063 Y118.586
399 G0 X120.534 Y118.127
400 G0 X120.5 Y118.437
401 G1 F2700 E7.7311
402 G1 F1201.7 X120.642 Y118.402 E7.73597
403 G1 X123.4 Y118.4 E7.8277
404 G1 X123.4 Y111.6 E8.05387
405 G1 X119.81 Y111.6 E8.17327
406 G1 X120.195 Y111.911 E8.18973
407 G1 X120.444 Y112.144 E8.20107
408 G1 X120.669 Y112.399 E8.21239
409 G1 X120.865 Y112.659 E8.22321
410 G1 X121.031 Y112.92 E8.2335
411 G1 X121.267 Y113.351 E8.24985
412 G1 X121.425 Y113.814 E8.26612
413 G1 X121.508 Y114.146 E8.2775
414 G1 X121.564 Y114.466 E8.2883
415 G1 X121.594 Y114.81 E8.29979
416 G1 X121.594 Y115.191 E8.31246
417 G1 X121.565 Y115.523 E8.32355
418 G1 X121.516 Y115.816 E8.33343
419 G1 X121.425 Y116.172 E8.34565
420 G1 X121.269 Y116.635 E8.3619
421 G1 X121.009 Y117.119 E8.38017
422 G1 X120.861 Y117.343 E8.3891
423 G1 X120.673 Y117.597 E8.39961
424 G1 X120.437 Y117.856 E8.41127
425 G1 X120.168 Y118.11 E8.42357
426 G1 X119.804 Y118.399 E8.43903
427 G1 X120.32 Y118.4 E8.45619
428 G1 X120.5 Y118.437 E8.4623
429 G0 F7500 X120.5 Y119
430 ; TYPE: WALL-OUTER
431 G1 F1201.7 X120.5 Y123.8 E8.62195
432 G1 F2700 E3.62195
433 G0 F7500 X121.365 Y123.8
434 G0 X122.735 Y123.9
435 G0 X123.5 Y123.9
436 G1 F2700 E8.62195
437 G1 F1413.7 X123.5 Y119 E8.76048
438 G1 F2700 E3.76048
439 G0 F7500 X123.5 Y118.49
440 G0 X120.5 Y118.49
441 G0 X120.5 Y119
442 G1 F2700 E8.76048
443 G1 F1201.7 X120.625 Y118.8 E8.76833
444 G1 X123.44 Y118.804 E8.86195
445 G1 F1299.1 X123.469 Y118.9 E8.86504
446 G1 X123.5 Y119 E8.86826
447 G1 X123.531 Y118.9 E8.87148
448 G1 X123.56 Y118.804 E8.87457
449 G1 F1201.7 X123.8 Y118.8 E8.88255
450 G1 X123.8 Y111.2 E9.13533
451 G1 X118.991 Y111.2 E9.29527
452 G1 F1166.7 X118.664 Y111.212 E9.30648
453 G1 F949.9 X118.321 Y111.294 E9.32132
454 G1 X118.621 Y111.46 E9.33575
455 G1 F1166.7 X118.908 Y111.578 E9.34638
456 G1 F1201.7 X119.182 Y111.705 E9.35642
457 G1 X119.44 Y111.85 E9.36627
458 G1 X119.697 Y112.023 E9.37657
459 G1 X119.924 Y112.205 E9.38625
460 G1 X120.146 Y112.41 E9.3963
461 G1 X120.342 Y112.63 E9.4061
462 G1 X120.529 Y112.875 E9.41635
463 G1 X120.673 Y113.099 E9.42521

```

```

464 G1 X120.906 Y113.531 E9.44153
465 G1 X121.051 Y113.96 E9.45659
466 G1 X121.122 Y114.254 E9.46665
467 G1 X121.172 Y114.551 E9.47667
468 G1 X121.196 Y114.843 E9.48641
469 G1 X121.196 Y115.157 E9.49686
470 G1 X121.17 Y115.457 E9.50687
471 G1 X121.122 Y115.747 E9.51665
472 G1 X121.046 Y116.044 E9.52685
473 G1 X120.906 Y116.46 E9.54145
474 G1 X120.675 Y116.899 E9.55794
475 G1 X120.52 Y117.132 E9.56725
476 G1 X120.353 Y117.357 E9.57657
477 G1 X120.134 Y117.594 E9.5873
478 G1 X119.915 Y117.8 E9.5973
479 G1 X119.682 Y117.988 E9.60726
480 G1 X119.428 Y118.154 E9.61735
481 G1 X119.184 Y118.294 E9.62671
482 G1 X118.894 Y118.424 E9.63728
483 G1 X118.605 Y118.529 E9.64751
484 G1 X118.309 Y118.609 E9.65771
485 G1 F1049.5 X117.911 Y118.742 E9.67369
486 G1 X118.326 Y118.799 E9.68964
487 G1 F1201.7 X120.378 Y118.805 E9.75789
488 G1 X120.5 Y119 E9.76554
489 G1 F2700 E4.76554
490 G0 F7500 X120.5 Y118.49
491 G0 X119.063 Y118.586
492 G0 X117.911 Y118.742
493 G1 F2700 E9.76554
494 G1 F931.5 X117.751 Y118.747 E9.77241
495 G1 F801.1 X117.5 Y118.836 E9.7857
496 G0 F7500 X116.824 Y118.719
497 G1 F891.8 X117.214 Y118.742 E9.8032
498 G1 F801.1 X117.5 Y118.836 E9.81822
499 G1 F717.4 X117.5 Y120.457 E9.90853
500 G1 F748.7 X117.5 Y122.078 E9.99506
501 G1 F782.9 X117.5 Y123.699 E10.07782
502 G1 F2700 E5.07782
503 G0 F7500 X116.535 Y123.699
504 G0 X115.565 Y123.8
505 G0 X114.7 Y123.8
506 G1 F2700 E10.07782
507 G1 F1201.7 X114.7 Y119 E10.23747
508 G1 X114.758 Y118.858 E10.24257
509 G1 X114.882 Y118.8 E10.24713
510 G1 X116.167 Y118.8 E10.28987
511 G1 F1166.7 X116.483 Y118.788 E10.3007
512 G1 F977 X116.824 Y118.719 E10.31493
513 G1 X116.515 Y118.578 E10.32883
514 G1 F1166.7 X115.953 Y118.363 E10.34944
515 G1 F1201.7 X115.701 Y118.235 E10.35884
516 G1 X115.43 Y118.066 E10.36946
517 G1 X115.18 Y117.883 E10.37977
518 G1 X114.951 Y117.683 E10.38988
519 G1 X114.751 Y117.474 E10.3995
520 G1 X114.567 Y117.258 E10.40894
521 G1 X114.379 Y116.989 E10.41986
522 G1 X114.227 Y116.728 E10.4299
523 G1 X114.097 Y116.457 E10.4399
524 G1 X113.993 Y116.189 E10.44946
525 G1 X113.903 Y115.884 E10.46004
526 G1 X113.846 Y115.599 E10.4697
527 G1 X113.812 Y115.297 E10.47981
528 G1 X113.8 Y114.995 E10.48986
529 G1 X113.813 Y114.684 E10.50022
530 G1 X113.845 Y114.408 E10.50946
531 G1 X113.907 Y114.1 E10.51991
532 G1 X113.993 Y113.807 E10.53006
533 G1 X114.097 Y113.539 E10.53963
534 G1 X114.227 Y113.268 E10.54962
535 G1 X114.379 Y113.01 E10.55958
536 G1 X114.566 Y112.743 E10.57042
537 G1 X114.759 Y112.513 E10.58041
538 G1 X114.951 Y112.313 E10.58963
539 G1 X115.194 Y112.102 E10.60034
540 G1 X115.444 Y111.92 E10.61062
541 G1 X115.7 Y111.766 E10.62056
542 G1 X115.972 Y111.626 E10.63073
543 G1 F1163.8 X116.515 Y111.417 E10.65071
544 G1 F975 X116.816 Y111.28 E10.66427
545 G1 X116.484 Y111.214 E10.67815

```

```

546 G1 F1163.8 X116.194 Y111.201 E10.68811
547 G1 F1201.7 X111.201 Y111.201 E10.85418
548 G1 X111.2 Y118 E11.08032
549 G1 F1068.2 X111.25 Y119 E11.11778
550 G1 F961.3 X111.25 Y123.75 E11.31526
551 G1 X111.75 Y123.75 E11.33605
552 G1 X111.75 Y119 E11.53353
553 G1 F1068.2 X112 Y118.8 E11.54551
554 G1 F1201.7 X114.12 Y118.801 E11.61603
555 G1 X114.242 Y118.858 E11.6205
556 G1 X114.3 Y119 E11.62561
557 G1 X114.3 Y123.8 E11.78525
558 G1 X114.7 Y123.8 E11.79856
559 G1 F2700 E6.79856
560 G0 F7500 X116.816 Y111.28
561 G1 F2700 E11.79856
562 G1 F893.4 X117.126 Y111.258 E11.81246
563 G1 F946.2 X117.433 Y111.25 E11.82543
564 G1 F949.9 X117.867 Y111.256 E11.8437
565 G1 F873.9 X118.321 Y111.294 E11.86453
566 G1 F2700 E6.86453
567 G0 F7500 X119.063 Y111.414
568 G0 X121.008 Y112.334
569 G0 X120.92 Y112.4
570 ; TYPE: FILL
571 G1 F2700 E11.86453
572 G1 F1201.7 X122.598 Y112.4 E11.92034
573 G0 F7500 X122.8 Y112.2
574 G1 F1201.7 X123.32 Y111.68 E11.94448
575 G0 F7500 X123.32 Y112.006
576 G1 F1201.7 X122.993 Y111.68 E11.96016
577 G0 F7500 X122.993 Y111.91
578 G0 X121.645 Y113.501
579 G0 X121.426 Y113.573
580 G1 F1201.7 X122.8 Y112.2 E12.02476
581 G1 X122.8 Y117.142 E12.18913
582 G1 X123.32 Y117.663 E12.21362
583 G0 F7500 X123.056 Y117.6
584 G1 F1201.7 X120.92 Y117.6 E12.28466
585 G0 F7500 X121.007 Y117.667
586 G0 X122.335 Y118.09
587 G0 X122.335 Y118.322
588 G1 F1201.7 X123.32 Y117.337 E12.33099
589 G0 F7500 X122.8 Y117.142
590 G1 F1201.7 X121.572 Y115.915 E12.38873
591 G1 F2700 E7.38873
592 G0 F7500 X121.799 Y115.972
593 G0 X121.168 Y116.411
594 G0 X120.147 Y116.016
595 G0 X115.556 Y117.062
596 G0 X114.808 Y117.862
597 G0 X113.986 Y117.662
598 G0 X114.075 Y117.6
599 G1 F2700 E12.38873
600 G1 F1201.7 X111.944 Y117.6 E12.45961
601 G0 F7500 X111.681 Y117.662
602 G1 F1201.7 X112.4 Y116.942 E12.49345
603 G0 F7500 X111.68 Y117.337
604 G1 F1201.7 X112.664 Y118.321 E12.53973
605 G0 F7500 X112.664 Y118.09
606 G0 X113.199 Y115.966
607 G0 X113.421 Y115.922
608 G1 F1201.7 X112.4 Y116.942 E12.58774
609 G1 X112.4 Y112.401 E12.73877
610 G1 X111.681 Y111.681 E12.77261
611 G0 F7500 X111.681 Y112.005
612 G1 F1201.7 X112.005 Y111.681 E12.78785
613 G0 F7500 X112.401 Y112.4
614 G1 F1201.7 X114.076 Y112.4 E12.84356
615 G0 F7500 X113.987 Y112.337
616 G0 X113.36 Y113.49
617 G0 X113.571 Y113.571
618 G1 F1201.7 X112.4 Y112.401 E12.89862
619 G1 F2700 E7.89862
620 ; MESH: NONMESH
621 G0 F300 X112.4 Y112.401 Z0.6
622 G0 F7500 X113.584 Y114.68
623 G0 X114.665 Y115
624 G0 X117.441 Y117.834
625 G0 X117.5 Y118.535
626 ; TIME_ELAPSED:39.150134
627 ; LAYER: 2

```

```

628 M106 S170
629 ; TYPE: WALL-INNER
630 ;MESH:Slicer Test v3.stl
631 G1 F2700 E12.89862
632 G1 F1500 X117.553 Y118.452 E12.9019
633 G1 X117.732 Y118.403 E12.90807
634 G1 X120.424 Y118.403 E12.99761
635 G1 X120.5 Y118.498 E13.00165
636 G1 X120.575 Y118.403 E13.00568
637 G1 X123.4 Y118.4 E13.09964
638 G1 X123.4 Y111.6 E13.32581
639 G1 X111.6 Y111.6 E13.71828
640 G1 X111.6 Y118.8 E13.95775
641 G1 X112 Y118.4 E13.97656
642 G1 X114.169 Y118.403 E14.0487
643 G1 X114.351 Y118.455 E14.055
644 G1 X114.5 Y118.688 E14.0642
645 G1 X114.649 Y118.455 E14.0734
646 G1 X114.831 Y118.403 E14.07969
647 G1 X117.268 Y118.403 E14.16075
648 G1 X117.447 Y118.452 E14.16692
649 G1 X117.5 Y118.535 E14.1702
650 G0 F9000 X117.5 Y119
651 ; TYPE: WALL-OUTER
652 G1 F1000 X117.5 Y123.7 E14.40468
653 G1 F2700 E9.40468
654 G0 F9000 X118.465 Y123.7
655 G0 X119.635 Y123.8
656 G0 X120.5 Y123.8
657 G1 F2700 E14.40468
658 G1 F1500 X120.5 Y119 E14.56433
659 G1 F2700 E9.56433
660 G0 F9000 X120.5 Y118.49
661 G0 X123.5 Y118.49
662 G0 X123.5 Y119
663 G1 F2700 E14.56433
664 G1 F1764.7 X123.5 Y123.9 E14.70286
665 G1 F2700 E9.70286
666 G0 F9000 X122.735 Y123.9
667 G0 X122.747 Y119.653
668 G0 X117.5 Y119
669 G1 F2700 E14.70286
670 G1 F1200 X117.737 Y118.8 E14.71575
671 G1 F1500 X120.345 Y118.805 E14.80249
672 G1 X120.5 Y119 E14.81078
673 G1 X120.655 Y118.805 E14.81906
674 G1 X123.44 Y118.804 E14.91169
675 G1 F1621.6 X123.5 Y119 E14.918
676 G1 X123.56 Y118.804 E14.9243
677 G1 F1500 X123.8 Y118.8 E14.93229
678 G1 X123.8 Y111.2 E15.18507
679 G1 X111.2 Y111.2 E15.60414
680 G1 X111.2 Y118 E15.83031
681 G1 F1333.3 X111.25 Y119 E15.86778
682 G1 F1200 X111.25 Y123.75 E16.06526
683 G1 X111.75 Y123.75 E16.08605
684 G1 X111.75 Y119 E16.28353
685 G1 F1333.3 X112 Y118.8 E16.29551
686 G1 F1500 X114.174 Y118.802 E16.36782
687 G1 X114.3 Y119 E16.37562
688 G1 X114.3 Y123.8 E16.53527
689 G1 X114.7 Y123.8 E16.54857
690 G1 X114.7 Y119 E16.70822
691 G1 X114.827 Y118.801 E16.71607
692 G1 X117.263 Y118.8 E16.7971
693 G1 F1200 X117.5 Y119 E16.80999
694 G0 F9000 X117.5 Y118.094
695 ; TYPE: SKIN
696 G1 F1500 X117.495 Y118.095 E16.81016
697 G1 X117.214 Y118.04 E16.81968
698 G1 X114.795 Y118.045 E16.90014
699 G1 X114.477 Y118.13 E16.91109
700 G1 X114.245 Y118.042 E16.91934
701 G1 X111.96 Y118.04 E16.99534
702 G1 X111.96 Y111.96 E17.19756
703 G1 X123.04 Y111.96 E17.56608
704 G1 X123.04 Y118.04 E17.7683
705 G1 X117.73 Y118.04 E17.94492
706 G1 X117.5 Y118.094 E17.95277
707 G0 F9000 X117.56 Y118.348
708 G0 X115.081 Y118.403
709 G0 X112.749 Y118.3

```

```

710 | G0 X112.749 Y117.84
711 | G1 F1500 X112.16 Y117.251 E17.98048
712 | G0 F9000 X112.16 Y116.685
713 | G1 F1500 X113.315 Y117.841 E18.03483
714 | G0 F9000 X113.882 Y117.841
715 | G1 F1500 X112.16 Y116.12 E18.1158
716 | G0 F9000 X112.16 Y115.554
717 | G1 F1500 X114.517 Y117.911 E18.22667
718 | G0 F9000 X115.019 Y117.847
719 | G1 F1500 X112.16 Y114.988 E18.36115
720 | G0 F9000 X112.16 Y114.423
721 | G1 F1500 X115.583 Y117.846 E18.52216
722 | G0 F9000 X116.147 Y117.844
723 | G1 F1500 X112.16 Y113.857 E18.70969
724 | G0 F9000 X112.16 Y113.291
725 | G1 F1500 X116.712 Y117.843 E18.9238
726 | G0 F9000 X117.284 Y117.85
727 | G1 F1500 X112.16 Y112.726 E19.16482
728 | G0 F9000 X112.16 Y112.16
729 | G1 F1500 X117.841 Y117.841 E19.43204
730 | G0 F9000 X118.406 Y117.841
731 | G1 F1500 X112.726 Y112.16 E19.69923
732 | G0 F9000 X113.291 Y112.16
733 | G1 F1500 X118.972 Y117.841 E19.96645
734 | G0 F9000 X119.537 Y117.84
735 | G1 F1500 X113.857 Y112.16 E20.23362
736 | G0 F9000 X114.423 Y112.16
737 | G1 F1500 X120.103 Y117.84 E20.50079
738 | G0 F9000 X120.668 Y117.84
739 | G1 F1500 X114.988 Y112.16 E20.76796
740 | G0 F9000 X115.554 Y112.16
741 | G1 F1500 X121.233 Y117.839 E21.03508
742 | G0 F9000 X121.799 Y117.839
743 | G1 F1500 X116.12 Y112.16 E21.3022
744 | G0 F9000 X116.685 Y112.16
745 | G1 F1500 X122.365 Y117.839 E21.56935
746 | G0 F9000 X122.84 Y117.749
747 | G1 F1500 X117.251 Y112.16 E21.83224
748 | G0 F9000 X117.817 Y112.16
749 | G1 F1500 X122.84 Y117.183 E22.0685
750 | G0 F9000 X122.84 Y116.617
751 | G1 F1500 X118.382 Y112.16 E22.27817
752 | G0 F9000 X118.948 Y112.16
753 | G1 F1500 X122.84 Y116.052 E22.46124
754 | G0 F9000 X122.84 Y115.486
755 | G1 F1500 X119.514 Y112.16 E22.61768
756 | G0 F9000 X120.079 Y112.16
757 | G1 F1500 X122.84 Y114.92 E22.74753
758 | G0 F9000 X122.84 Y114.355
759 | G1 F1500 X120.645 Y112.16 E22.85078
760 | G0 F9000 X121.211 Y112.16
761 | G1 F1500 X122.84 Y113.789 E22.9274
762 | G0 F9000 X122.84 Y113.223
763 | G1 F1500 X121.777 Y112.16 E22.9774
764 | G0 F9000 X122.342 Y112.16
765 | G1 F1500 X122.84 Y112.658 E23.00082
766 | ;MESH:NONMESH
767 | G0 F300 X122.84 Y112.658 Z0.8
768 | G0 F9000 X123.3 Y112.658
769 | G0 X123.228 Y118.228
770 | G0 X117.607 Y118.337
771 | G0 X117.5 Y117.888
772 | ;TIME_ELAPSED:57.168930
773 | ;LAYER:3
774 | ;note -- min layer time used
775 | M106 S255
776 | ;TYPE:WALL-INNER
777 | ;MESH:Slicer Test v3.stl
778 | G1 F600 X117.67 Y117.892 E23.00648
779 | G1 X118.262 Y117.809 E23.02636
780 | G1 X118.813 Y117.583 E23.04617
781 | G1 X119.298 Y117.287 E23.06507
782 | G1 X119.727 Y116.854 E23.08534
783 | G1 X120.001 Y116.464 E23.10119
784 | G1 X120.202 Y116.045 E23.11665
785 | G1 X120.308 Y115.724 E23.12789
786 | G1 X120.4 Y115.215 E23.1451
787 | G1 X120.376 Y114.643 E23.16414
788 | G1 X120.332 Y114.381 E23.17298
789 | G1 X120.204 Y113.955 E23.18777
790 | G1 X119.997 Y113.528 E23.20355
791 | G1 X119.726 Y113.141 E23.21927

```

```

792 | G1 X119.321 Y112.733 E23.23839
793 | G1 X118.928 Y112.469 E23.25413
794 | G1 X118.458 Y112.252 E23.27135
795 | G1 X117.779 Y112.102 E23.29448
796 | G1 X117.207 Y112.114 E23.31351
797 | G1 X116.62 Y112.223 E23.33337
798 | G1 X116.087 Y112.464 E23.35282
799 | G1 X115.582 Y112.808 E23.37315
800 | G1 X115.113 Y113.332 E23.39654
801 | G1 X114.836 Y113.85 E23.41607
802 | G1 X114.684 Y114.301 E23.4319
803 | G1 X114.611 Y114.77 E23.44769
804 | G1 X114.611 Y115.229 E23.46296
805 | G1 X114.684 Y115.699 E23.47878
806 | G1 X114.836 Y116.146 E23.49448
807 | G1 X115.113 Y116.664 E23.51402
808 | G1 X115.512 Y117.11 E23.53392
809 | G1 X115.974 Y117.48 E23.55361
810 | G1 X116.517 Y117.727 E23.57345
811 | G1 X117.078 Y117.879 E23.59278
812 | G1 X117.5 Y117.888 E23.60682
813 | G0 F9000 X117.5 Y118.294
814 | ; TYPE: WALL_OUTER
815 | G1 F600 X117.695 Y118.292 E23.6133
816 | G1 X117.972 Y118.261 E23.62257
817 | G1 X118.236 Y118.216 E23.63148
818 | G1 X118.494 Y118.145 E23.64038
819 | G1 X118.74 Y118.054 E23.64911
820 | G1 X118.993 Y117.941 E23.65832
821 | G1 X119.227 Y117.809 E23.66726
822 | G1 X119.449 Y117.662 E23.67611
823 | G1 X119.655 Y117.493 E23.68498
824 | G1 X119.857 Y117.303 E23.6942
825 | G1 X120.034 Y117.111 E23.70288
826 | G1 X120.2 Y116.892 E23.71202
827 | G1 X120.347 Y116.666 E23.72099
828 | G1 X120.471 Y116.429 E23.72989
829 | G1 X120.576 Y116.188 E23.73863
830 | G1 X120.697 Y115.822 E23.75145
831 | G1 X120.773 Y115.402 E23.76565
832 | G1 X120.796 Y115.127 E23.77483
833 | G1 X120.796 Y114.861 E23.78367
834 | G1 X120.773 Y114.593 E23.79262
835 | G1 X120.706 Y114.19 E23.80621
836 | G1 X120.577 Y113.809 E23.81959
837 | G1 X120.471 Y113.572 E23.82822
838 | G1 X120.343 Y113.326 E23.83745
839 | G1 X120.196 Y113.099 E23.84644
840 | G1 X120.034 Y112.885 E23.85537
841 | G1 X119.857 Y112.693 E23.86405
842 | G1 X119.656 Y112.503 E23.87325
843 | G1 X119.445 Y112.334 E23.88224
844 | G1 X119.227 Y112.188 E23.89097
845 | G1 X118.993 Y112.059 E23.89986
846 | G1 X118.739 Y111.941 E23.90917
847 | G1 X118.495 Y111.852 E23.91781
848 | G1 X118.236 Y111.784 E23.92672
849 | G1 X117.97 Y111.735 E23.93571
850 | G1 X117.695 Y111.704 E23.94492
851 | G1 X117.433 Y111.701 E23.95363
852 | G1 X117.165 Y111.716 E23.96256
853 | G1 X116.893 Y111.757 E23.97171
854 | G1 X116.636 Y111.814 E23.98047
855 | G1 X116.38 Y111.893 E23.98938
856 | G1 X116.123 Y111.999 E23.99862
857 | G1 X115.891 Y112.115 E24.00725
858 | G1 X115.657 Y112.262 E24.01644
859 | G1 X115.441 Y112.421 E24.02536
860 | G1 X115.238 Y112.594 E24.03423
861 | G1 X115.053 Y112.786 E24.0431
862 | G1 X114.875 Y112.999 E24.05233
863 | G1 X114.721 Y113.216 E24.06118
864 | G1 X114.589 Y113.447 E24.07003
865 | G1 X114.468 Y113.692 E24.07912
866 | G1 X114.374 Y113.945 E24.0881
867 | G1 X114.295 Y114.205 E24.09714
868 | G1 X114.246 Y114.46 E24.10577
869 | G1 X114.212 Y114.737 E24.11505
870 | G1 X114.201 Y115.003 E24.12391
871 | G1 X114.212 Y115.263 E24.13256
872 | G1 X114.246 Y115.537 E24.14175
873 | G1 X114.295 Y115.795 E24.15048

```

```

874 G1 X114.374 Y116.051 E24.15939
875 G1 X114.468 Y116.304 E24.16837
876 G1 X114.589 Y116.549 E24.17746
877 G1 X114.721 Y116.78 E24.18631
878 G1 X114.875 Y116.997 E24.19516
879 G1 X115.053 Y117.21 E24.20439
880 G1 X115.238 Y117.402 E24.21326
881 G1 X115.442 Y117.575 E24.22215
882 G1 X115.657 Y117.738 E24.23113
883 G1 X115.89 Y117.88 E24.2402
884 G1 X116.123 Y117.997 E24.24888
885 G1 X116.381 Y118.103 E24.25815
886 G1 X116.633 Y118.182 E24.26694
887 G1 X116.893 Y118.243 E24.27582
888 G1 X117.164 Y118.28 E24.28492
889 G1 X117.433 Y118.295 E24.29388
890 G1 X117.5 Y118.294 E24.29611
891 G0 F9000 X117.5 Y117.527
892 ; TYPE: SKIN
893 G1 F600 X117.122 Y117.516 E24.30868
894 G1 X116.536 Y117.357 E24.32888
895 G1 X116.084 Y117.109 E24.34603
896 G1 X115.755 Y116.842 E24.36012
897 G1 X115.41 Y116.456 E24.37734
898 G1 X115.169 Y116.004 E24.39437
899 G1 X115.031 Y115.603 E24.40848
900 G1 X114.968 Y115.204 E24.42191
901 G1 X114.968 Y114.792 E24.43562
902 G1 X115.031 Y114.393 E24.44905
903 G1 X115.169 Y113.992 E24.46316
904 G1 X115.406 Y113.544 E24.48002
905 G1 X115.766 Y113.143 E24.49794
906 G1 X116.173 Y112.82 E24.51522
907 G1 X116.729 Y112.569 E24.53551
908 G1 X117.241 Y112.472 E24.55284
909 G1 X117.744 Y112.461 E24.56958
910 G1 X118.263 Y112.576 E24.58726
911 G1 X118.657 Y112.739 E24.60144
912 G1 X119.07 Y112.99 E24.61751
913 G1 X119.446 Y113.373 E24.63536
914 G1 X119.687 Y113.715 E24.64928
915 G1 X119.865 Y114.077 E24.6627
916 G1 X120.009 Y114.566 E24.67965
917 G1 X120.039 Y115.186 E24.7003
918 G1 X119.957 Y115.626 E24.71518
919 G1 X119.865 Y115.909 E24.72508
920 G1 X119.687 Y116.278 E24.73871
921 G1 X119.393 Y116.693 E24.75562
922 G1 X118.988 Y117.054 E24.77367
923 G1 X118.657 Y117.256 E24.78656
924 G1 X118.159 Y117.457 E24.80443
925 G1 X117.64 Y117.531 E24.82186
926 G1 X117.5 Y117.527 E24.82652
927 G0 F9000 X117.469 Y117.787
928 G0 X118.214 Y117.654
929 G0 X118.741 Y117.439
930 G0 X119.116 Y117.211
931 G0 X119.547 Y116.829
932 G0 X119.726 Y116.576
933 G0 X120.221 Y115.641
934 G0 X119.766 Y115.559
935 G1 F600 X118.056 Y117.269 E24.90695
936 G0 F9000 X117.433 Y117.326
937 G1 F600 X119.826 Y114.934 E25.01949
938 G0 F9000 X119.761 Y114.433
939 G1 F600 X116.935 Y117.259 E25.15242
940 G0 F9000 X116.512 Y117.116
941 G1 F600 X119.611 Y114.017 E25.29818
942 G0 F9000 X119.403 Y113.66
943 G1 F600 X116.154 Y116.909 E25.45101
944 G0 F9000 X115.849 Y116.648
945 G1 F600 X119.145 Y113.352 E25.60604
946 G0 F9000 X118.844 Y113.087
947 G1 F600 X115.582 Y116.349 E25.75948
948 G0 F9000 X115.384 Y115.981
949 G1 F600 X118.483 Y112.883 E25.90522
950 G0 F9000 X118.063 Y112.736
951 G1 F600 X115.231 Y115.569 E26.03845
952 G0 F9000 X115.169 Y115.065
953 G1 F600 X117.569 Y112.665 E26.15134
954 G0 F9000 X116.935 Y112.733
955 G1 F600 X115.226 Y114.442 E26.23173

```

```

956 ; MESH:NONMESH
957 G0 F300 X115.226 Y114.442 Z1
958 G0 F9000 X114.775 Y114.371
959 G0 X114.771 Y115.215
960 G0 X114.84 Y115.66
961 G0 X114.984 Y116.081
962 G0 X115.243 Y116.57
963 G0 X115.623 Y116.993
964 G0 X115.974 Y117.275
965 G0 X116.366 Y117.494
966 G0 X117.469 Y117.787
967 G0 X117.5 Y117.888
968 ; TIME_ELAPSED:68.095268
969 ; LAYER:4
970 ;note -- min layer time used
971 ; TYPE: WALL-INNER
972 ;MESH:Slicer Test v3.stl
973 G1 F600 X117.67 Y117.892 E26.23738
974 G1 X118.262 Y117.809 E26.25726
975 G1 X118.813 Y117.583 E26.27707
976 G1 X119.298 Y117.287 E26.29597
977 G1 X119.727 Y116.854 E26.31624
978 G1 X120.001 Y116.464 E26.3321
979 G1 X120.202 Y116.045 E26.34755
980 G1 X120.308 Y115.724 E26.3588
981 G1 X120.4 Y115.215 E26.376
982 G1 X120.376 Y114.643 E26.39504
983 G1 X120.332 Y114.381 E26.40388
984 G1 X120.204 Y113.955 E26.41867
985 G1 X119.997 Y113.528 E26.43446
986 G1 X119.726 Y113.141 E26.45017
987 G1 X119.321 Y112.733 E26.46929
988 G1 X118.928 Y112.469 E26.48504
989 G1 X118.458 Y112.252 E26.50225
990 G1 X117.779 Y112.102 E26.52538
991 G1 X117.207 Y112.114 E26.54441
992 G1 X116.62 Y112.223 E26.56427
993 G1 X116.087 Y112.464 E26.58373
994 G1 X115.582 Y112.808 E26.60405
995 G1 X115.113 Y113.332 E26.62744
996 G1 X114.836 Y113.85 E26.64698
997 G1 X114.684 Y114.301 E26.6628
998 G1 X114.611 Y114.77 E26.67859
999 G1 X114.611 Y115.229 E26.69386
1000 G1 X114.684 Y115.699 E26.70968
1001 G1 X114.836 Y116.146 E26.72538
1002 G1 X115.113 Y116.664 E26.74492
1003 G1 X115.512 Y117.11 E26.76482
1004 G1 X115.974 Y117.48 E26.78451
1005 G1 X116.517 Y117.727 E26.80435
1006 G1 X117.078 Y117.879 E26.82368
1007 G1 X117.5 Y117.888 E26.83772
1008 G0 F9000 X117.5 Y118.294
1009 ; TYPE: WALL-OUTER
1010 G1 F600 X117.695 Y118.292 E26.84421
1011 G1 X117.972 Y118.261 E26.85348
1012 G1 X118.236 Y118.216 E26.86238
1013 G1 X118.494 Y118.145 E26.87128
1014 G1 X118.74 Y118.054 E26.88001
1015 G1 X118.993 Y117.941 E26.88922
1016 G1 X119.227 Y117.809 E26.89816
1017 G1 X119.449 Y117.662 E26.90702
1018 G1 X119.655 Y117.493 E26.91588
1019 G1 X119.857 Y117.303 E26.9251
1020 G1 X120.034 Y117.111 E26.93379
1021 G1 X120.2 Y116.892 E26.94293
1022 G1 X120.347 Y116.666 E26.95189
1023 G1 X120.471 Y116.429 E26.96079
1024 G1 X120.576 Y116.188 E26.96953
1025 G1 X120.697 Y115.822 E26.98235
1026 G1 X120.773 Y115.402 E26.99655
1027 G1 X120.796 Y115.127 E27.00573
1028 G1 X120.796 Y114.861 E27.01458
1029 G1 X120.773 Y114.593 E27.02352
1030 G1 X120.706 Y114.19 E27.03711
1031 G1 X120.577 Y113.809 E27.05049
1032 G1 X120.471 Y113.572 E27.05912
1033 G1 X120.343 Y113.326 E27.06835
1034 G1 X120.196 Y113.099 E27.07734
1035 G1 X120.034 Y112.885 E27.08627
1036 G1 X119.857 Y112.693 E27.09496
1037 G1 X119.656 Y112.503 E27.10416

```

```

1038 G1 X119.445 Y112.334 E27.11315
1039 G1 X119.227 Y112.188 E27.12187
1040 G1 X118.993 Y112.059 E27.13076
1041 G1 X118.739 Y111.941 E27.14008
1042 G1 X118.495 Y111.852 E27.14871
1043 G1 X118.236 Y111.784 E27.15762
1044 G1 X117.97 Y111.735 E27.16662
1045 G1 X117.695 Y111.704 E27.17582
1046 G1 X117.433 Y111.701 E27.18454
1047 G1 X117.165 Y111.716 E27.19346
1048 G1 X116.893 Y111.757 E27.20261
1049 G1 X116.636 Y111.814 E27.21137
1050 G1 X116.38 Y111.893 E27.22028
1051 G1 X116.123 Y111.999 E27.22952
1052 G1 X115.891 Y112.115 E27.23815
1053 G1 X115.657 Y112.262 E27.24734
1054 G1 X115.441 Y112.421 E27.25626
1055 G1 X115.238 Y112.594 E27.26514
1056 G1 X115.053 Y112.786 E27.274
1057 G1 X114.875 Y112.999 E27.28324
1058 G1 X114.721 Y113.216 E27.29209
1059 G1 X114.589 Y113.447 E27.30093
1060 G1 X114.468 Y113.692 E27.31002
1061 G1 X114.374 Y113.945 E27.319
1062 G1 X114.295 Y114.205 E27.32804
1063 G1 X114.246 Y114.46 E27.33667
1064 G1 X114.212 Y114.737 E27.34596
1065 G1 X114.201 Y115.003 E27.35481
1066 G1 X114.212 Y115.263 E27.36347
1067 G1 X114.246 Y115.537 E27.37265
1068 G1 X114.295 Y115.795 E27.38138
1069 G1 X114.374 Y116.051 E27.3903
1070 G1 X114.468 Y116.304 E27.39927
1071 G1 X114.589 Y116.549 E27.40836
1072 G1 X114.721 Y116.78 E27.41721
1073 G1 X114.875 Y116.997 E27.42606
1074 G1 X115.053 Y117.21 E27.43529
1075 G1 X115.238 Y117.402 E27.44416
1076 G1 X115.442 Y117.575 E27.45306
1077 G1 X115.657 Y117.738 E27.46203
1078 G1 X115.89 Y117.88 E27.47111
1079 G1 X116.123 Y117.997 E27.47978
1080 G1 X116.381 Y118.103 E27.48905
1081 G1 X116.633 Y118.182 E27.49784
1082 G1 X116.893 Y118.243 E27.50672
1083 G1 X117.164 Y118.28 E27.51582
1084 G1 X117.433 Y118.295 E27.52478
1085 G1 X117.5 Y118.294 E27.52701
1086 G0 F9000 X117.5 Y117.527
1087 ; TYPE: SKIN
1088 G1 F600 X117.122 Y117.516 E27.53959
1089 G1 X116.536 Y117.357 E27.55978
1090 G1 X116.084 Y117.109 E27.57693
1091 G1 X115.755 Y116.842 E27.59102
1092 G1 X115.41 Y116.456 E27.60824
1093 G1 X115.169 Y116.004 E27.62528
1094 G1 X115.031 Y115.603 E27.63938
1095 G1 X114.968 Y115.204 E27.65282
1096 G1 X114.968 Y114.792 E27.66652
1097 G1 X115.031 Y114.393 E27.67996
1098 G1 X115.169 Y113.992 E27.69406
1099 G1 X115.406 Y113.544 E27.71092
1100 G1 X115.766 Y113.143 E27.72884
1101 G1 X116.173 Y112.82 E27.74612
1102 G1 X116.729 Y112.569 E27.76641
1103 G1 X117.241 Y112.472 E27.78374
1104 G1 X117.744 Y112.461 E27.80048
1105 G1 X118.263 Y112.576 E27.81816
1106 G1 X118.657 Y112.739 E27.83234
1107 G1 X119.07 Y112.99 E27.84842
1108 G1 X119.446 Y113.373 E27.86627
1109 G1 X119.687 Y113.715 E27.88018
1110 G1 X119.865 Y114.077 E27.8936
1111 G1 X120.009 Y114.566 E27.91055
1112 G1 X120.039 Y115.186 E27.9312
1113 G1 X119.957 Y115.626 E27.94609
1114 G1 X119.865 Y115.909 E27.95598
1115 G1 X119.687 Y116.278 E27.96961
1116 G1 X119.393 Y116.693 E27.98652
1117 G1 X118.988 Y117.054 E28.00457
1118 G1 X118.657 Y117.256 E28.01747
1119 G1 X118.159 Y117.457 E28.03533

```

```

1120 G1 X117.64 Y117.531 E28.05277
1121 G1 X117.5 Y117.527 E28.05742
1122 G0 F9000 X116.595 Y117.161
1123 G1 F600 X115.35 Y115.915 E28.11601
1124 G0 F9000 X115.168 Y115.168
1125 G1 F600 X117.322 Y117.322 E28.21733
1126 G0 F9000 X117.863 Y117.297
1127 G1 F600 X115.196 Y114.63 E28.34277
1128 G0 F9000 X115.314 Y114.183
1129 G1 F600 X118.31 Y117.179 E28.4837
1130 G0 F9000 X118.695 Y116.998
1131 G1 F600 X115.497 Y113.8 E28.63412
1132 G0 F9000 X115.736 Y113.474
1133 G1 F600 X119.02 Y116.757 E28.78857
1134 G0 F9000 X119.303 Y116.475
1135 G1 F600 X116.023 Y113.194 E28.94287
1136 G0 F9000 X116.352 Y112.958
1137 G1 F600 X119.533 Y116.139 E29.0925
1138 G0 F9000 X119.708 Y115.748
1139 G1 F600 X116.743 Y112.783 E29.23196
1140 G0 F9000 X117.208 Y112.682
1141 G1 F600 X119.816 Y115.291 E29.35466
1142 G0 F9000 X119.816 Y114.725
1143 G1 F600 X117.762 Y112.671 E29.45127
1144 G0 F9000 X118.588 Y112.931
1145 G1 F600 X119.553 Y113.896 E29.49666
1146 ;TIME_ELAPSED:78.759930
1147 G1 F2700 E24.49666
1148 M140 S0
1149 M107
1150 G91 ;Relative positioning

```

Listing A.3.2 G-Code Generated Using Ultimaker Cura [97] (Main)

```

1151 G1 E-2 F2700 ;Retract a bit
1152 G1 E-2 Z0.2 F2400 ;Retract and raise Z
1153 G1 X5 Y5 F3000 ;Wipe out
1154 G1 Z10 ;Raise Z more
1155 G90 ;Absolute positioning
1156
1157 G1 X0 Y235 ;Present print
1158 M106 S0 ;Turn-off fan
1159 M104 S0 ;Turn-off hotend
1160 M140 S0 ;Turn-off bed
1161
1162 M84 X Y E ;Disable all steppers but Z
1163
1164 M82 ;absolute extrusion mode
1165 M104 S0
1166 ;End of Gcode
1167 ;SETTING_3 {"global_quality": "[general]"\nversion = 4\nname = Standard Quality
1168 ;SETTING_3 #2\ndefinition = creality_ender3\n[metadata]\n\ntype = quality_c
1169 ;SETTING_3 hanges\nquality_type = standard\nsetting_version = 25\n\n[values]
1170 ;SETTING_3 \nadhesion_type = none\n", "extruder_quality": "[general]"\nver
1171 ;SETTING_3 sion = 4\nname = Standard Quality #2\ndefinition = creality_ender3\
1172 ;SETTING_3 \n[metadata]\n\ntype = quality_changes\nquality_type = standard\n
1173 ;SETTING_3 setting_version = 25\nposition = 0\n\n[values]\nfill_outline_gaps
1174 ;SETTING_3 = True\n\nfill_pattern = grid\ninitial_bottom_layers = 1\n\ntop_lay
1175 ;SETTING_3 ers = 1\n\n"]}

```

Listing A.3.3 G-Code Generated Using Ultimaker Cura [97] (Reset)

B Fixed Arguments for Cura

Listing B.1 Fixed arguments specified in *preset_arguments.py*

```
25 preset = {
26     "machine_name": machine_name, # match "id" from Robot.setup.
27         json
28     "machine_width": bed_size["X"], # match "bed_size" from Robot.
29         setup.json
30     "machine_depth": bed_size["Y"], # match "bed_size" from Robot.
31         setup.json
32     "machine_height": bed_size["Z"], # match "bed_size" from Robot.
33         .setup.json
34     "machine_show_variants": False, # allows only one variant of
35         machine
36     "machine_start_gcode": "", # no custom G-Code possible for
37         start sequence
38     "machine_end_gcode": "", # no custom G-Code possible for end
39         sequence
40     "machine_center_is_zero": False, # to fix origin to print bed
41         corner matching $BASE
42     "machine_extruder_start_code": "", # no custom G-Code possible
43         for extruder start
44     "machine_extruder_end_code": "", # no custom G-Code possible
45         for extruder stop
46     "machine_g_code_flavor": "RepRap (Marlin/Sprinter)", # Fixed
47         to Marlin as specified
48     "material_guid": "",
49     "material_type": "", # no material specified
50     "material_brand": "",
51     "machine_shape": "rectangular", # circular shape not
52         compatible with Rhino implementation and Report
53     "machine_extruder_count": 1, # no dual extruder possible
54     "extruders_enabled_count": 0,
55     "machine_firmware_retract": False, # No G10 or G11 commands to
56         specify retractions
57     "initial_layer_line_width_factor": 100, # line width for layer
58         0 fixed to 100 %
59     "ironing_enabled": False, # not possible to differentiate
60         between surface and ironing; therefor disabled
61     "roofing_layer_count": "0", # unresolved issue if not
62         specified here
63     "wall_0_material_flow": flow[
64         "wall_outer"
65     ], # to match "linetype_flow" input from Pump.setup.json
66     "wall_x_material_flow": flow[
67         "wall_inner"
68     ], # to match "linetype_flow" input from Pump.setup.json
69     "wall_0_material_flow_roofing": flow[
70         "wall_outer"
71     ], # to match "linetype_flow" input from Pump.setup.json
72     "wall_x_material_flow_roofing": flow[
73         "wall_inner"
74     ], # to match "linetype_flow" input from Pump.setup.json
75     "skin_material_flow": flow[
76         "surface"
77     ], # to match "linetype_flow" input from Pump.setup.json
78     "skin_material_flow": flow[
79         "surface"
80     ], # to match "linetype_flow" input from Pump.setup.json
```

```

62 "roofing_material_flow": flow[
63     "surface"
64 ], # to match "linetype_flow" input from Pump.setup.json
65 "infill_material_flow": flow[
66     "infill"
67 ], # to match "linetype_flow" input from Pump.setup.json
68 "skirt_brim_material_flow": flow[
69     "curb"
70 ], # to match "linetype_flow" input from Pump.setup.json
71 "support_material_flow": flow[
72     "support"
73 ], # to match "linetype_flow" input from Pump.setup.json
74 "support_roof_material_flow": flow[
75     "support"
76 ], # to match "linetype_flow" input from Pump.setup.json
77 "support_bottom_material_flow": flow[
78     "support"
79 ], # to match "linetype_flow" input from Pump.setup.json
80 "prime_tower_flow": 100, # as default flow for not specified
81     line types is 100% and prime tower has not been tested
82 "material_flow_layer_0": 100, # flow for layer 0 fixed to
83     100%; no bed adhesion problems for concrete 3D-printing
84 "retraction_enable": pump_retract, # to match "retract" input
85     from Pump.setup.json
86 "retraction_hop_enabled": False, # could disturb layer_height
87     calculation; therefor deactivated
88 "layer_0_z_overlap": 0, # could disturb layer_height
     calculation; therefor deactivated
     "wipe_hop_enable": False, # could disturb layer_height
     calculation; therefor deactivated
     "mesh_rotation_matrix": scaling, # calculated for "
     cura_scaling" from Cura.setup.json via module scaling_matrix
     .py
     "filament_diameter": filament_dia, # to match "
     filament_diameter" input from Pump.setup.json and
     recalculate E-value correctly
     }

```

C List of Usable and Validated Commands from *printer.def.json* With Comments

Parent Key	Key	Description	Unit	Type	Value	Usable	Comment
machine_settings	machine_name	The name of your 3D printer model.		str	Unknown	yes	setup.json[Robot.id]
machine_settings	machine_show_variants	Whether to show the different variants of this machine, which are described in separate json files.		bool	False	fixed	
machine_settings	machine_start_gcode	G-code commands to be executed at the very start - separated by \n.		str		fixed	
machine_settings	machine_end_gcode	G-code commands to be executed at the very end - separated by \n.		str		fixed	
machine_settings	material_guid	GUID of the material. This is set automatically.		str		fixed	
machine_settings	material_type	The type of material used.		str		fixed	
machine_settings	material_brand	The brand of material used.		str		fixed	
machine_settings	material_diameter	Adjusts the diameter of the filament used. Match this value with the diameter of the used filament.	mm	float	25	yes	setup.json[Pump.filament_diameter]
machine_settings	material_bed_temp_wait	Whether to insert a command to wait until the build plate temperature is reached at the start.		bool	True	no	
machine_settings	material_print_temp_wait	Whether to wait until the nozzle temperature is reached at the start.		bool	True	no	
machine_settings	material_print_temp_prepend	Whether to include nozzle temperature commands at the start of the gcode. When the start_gcode already contains nozzle temperature commands Cura frontend will automatically disable this setting.		bool	True	no	
machine_settings	material_bed_temp_prepend	Whether to include build plate temperature commands at the start of the gcode. When the start_gcode already contains build plate temperature commands Cura frontend will automatically disable this setting.		bool	True	no	
machine_settings	machine_width	The width (X-direction) of the printable area.		float	1200	yes	setup.json[Robot.bed_size("X")]
machine_settings	machine_depth	The depth (Y-direction) of the printable area.		float	4500	yes	setup.json[Robot.bed_size("Y")]
machine_settings	machine_height	The height (Z-direction) of the printable area.		float	2000	yes	setup.json[Robot.bed_size("Z")]
machine_settings	machine_shape	The shape of the build plate without taking unprintable areas into account.		key	rectangular	fixed	only rectangular possible
machine_settings	machine_buildplate_type	The material of the build plate installed on the printer.		key	aluminum	no	
machine_settings	machine_heated_bed	Whether the machine has a heated build plate present.		bool	False	no	
machine_settings	machine_heated_build_volume	Whether the machine is able to stabilize the build volume temperature.		bool	False	no	
machine_settings	machine_always_write_active_tool	Write active tool after sending temp commands to inactive tool. Required for Dual Extruder printing with Smoothie or other firmware with modal tool commands.		bool	False	no	
machine_settings	machine_center_is_zero	Whether the X/Y coordinates of the zero position of the printer is at the center of the printable area.		bool	False	yes	
machine_settings	machine_extruder_count	Number of extruder trains. An extruder train is the combination of a feeder, bowden tube, and nozzle.		int	1	fixed	only singel extruder possible
machine_settings	extruders_enabled_count	Number of extruder trains that are enabled; automatically set in software		int	1	fixed	only singel extruder possible
machine_settings	machine_nozzle_tip_outer_diameter	The outer diameter of the tip of the nozzle.	mm	float	30	yes	for collision avoidance
machine_settings	machine_nozzle_head_distance	The height difference between the tip of the nozzle and the lowest part of the print head.	mm	float	10	yes	for collision avoidance
machine_settings	machine_nozzle_expansion_angle	The angle between the horizontal plane and the conical part right above the tip of the nozzle.	°	int	45	yes	for collision avoidance

machine_settings	machine_heat_zone_length	The distance from the tip of the nozzle in which heat from the nozzle is transferred to the filament.	mm	float	16	no
machine_settings	machine_nozzle_temp_enabled	Whether to control temperature from Cura. Turn this off to control nozzle temperature from outside of Cura.	bool	True	no	
machine_settings	machine_nozzle_heat_up_speed	The speed (°C/s) by which the nozzle heats up averaged over the window of normal printing temperatures and the standby temperature.	°C/s	float	2	no
machine_settings	machine_nozzle_cool_down_speed	The speed (°C/s) by which the nozzle cools down averaged over the window of normal printing temperatures and the standby temperature.	°C/s	float	2	no
machine_settings	machine_min_col_heat_time_window	The minimal time an extruder has to be inactive before the nozzle is cooled. Only when an extruder is not used for longer than this time will it be allowed to cool down to the standby temperature.	s	float	50	no
machine_settings	machine_gcode_flavor	The type of g-code to be generated.	key	RePRap (Marlin/Sprinter)	fixed	fixed to Marlin
machine_settings	machine_firmware_retract	Whether to use firmware retract commands (G10/G11) instead of using the E property in G1 commands to retract the material.	bool	False	fixed	
machine_settings	machine_extruders_share_heater	Whether the extruders share a single heater rather than each extruder having its own heater.	bool	False	no	
machine_settings	machine_extruders_share_nozzle	Whether the extruders share a single nozzle rather than each extruder having its own nozzle. When set to true, it is expected that the printer-start gcode script properly sets up all extruders in an initial retraction state that is known and mutually compatible (either zero or one filament not retracted); in that case the initial retraction status is described, per extruder, by the 'machine_extruders_shared_nozzle_initial_retraction' parameter.	bool	False	no	
machine_settings	machine_extruders_shared_nozzle_initial_retraction	How much the filament of each extruder is assumed to have been retracted from the shared nozzle tip at the completion of the printer-start gcode script; the value should be equal to or greater than the length of the common part of the nozzle's ducts.	mm	float	0	no
machine_settings	machine_disallowed_areas	A list of polygons with areas the print head is not allowed to enter.	list[float]	[]	yes	
machine_settings	nozzle_disallowed_areas	A list of polygons with areas the nozzle is not allowed to enter.	list[float]	[]	yes	
machine_settings	machine_head_with_fans_polygon	The shape of the print head. These are coordinates relative to the position of the print head, which is usually the position of its first extruder. The dimensions left and in front of the print head must be negative coordinates.	[list[float]]	[[-150, 150], [150, 150], [150, -150], [-150, -150]]	yes	
machine_settings	gantry_height	The height difference between the tip of the nozzle and the gantry system (X and Y axes).	float	9999999999	yes	
machine_settings	machine_nozzle_id	The nozzle ID for an extruder train, such as "AA 0.4" and "BB 0.8".	str	unknown	no	for multi-head configurations
machine_settings	machine_nozzle_size	The inner diameter of the nozzle. Change this setting when using a non-standard nozzle size.	mm	25	yes	
machine_settings	machine_use_extruder_offset_to_offset_coords	Apply the extruder offset to the coordinate system. Affects all extruders.	bool	True	yes	

machine_settings	extruder_prime_pos_z	The Z coordinate of the position where the nozzle primes at the start of printing.	mm	float	0	possibly	priming not tested
machine_settings	extruder_prime_pos_abs	Make the extruder prime position absolute rather than relative to the last-known location of the head.	bool	False	possibly		
machine_settings	machine_max_feedrate_x	The maximum speed for the motor of the X-direction.	mm/s	float	2000	no	speed set in .src file (not taken from G-Code)
machine_settings	machine_max_feedrate_y	The maximum speed for the motor of the Y-direction.	mm/s	float	2000	no	speed set in .src file (not taken from G-Code)
machine_settings	machine_max_feedrate_z	The maximum speed for the motor of the Z-direction.	mm/s	float	2000	no	speed set in .src file (not taken from G-Code)
machine_settings	machine_max_feedrate_e	The maximum speed of the filament.	mm/s	float	2000	no	speed set in .src file (not taken from G-Code)
machine_settings	machine_max_acceleration_x	Maximum acceleration for the motor of the X-direction.	mm/s ²	float	9000	no	
machine_settings	machine_max_acceleration_y	Maximum acceleration for the motor of the Y-direction.	mm/s ²	float	9000	no	
machine_settings	machine_max_acceleration_z	Maximum acceleration for the motor of the Z-direction.	mm/s ²	float	100	no	
machine_settings	machine_max_acceleration_e	Maximum acceleration for the motor of the filament.	mm/s ²	float	10000	no	
machine_settings	machine_acceleration	The default acceleration of print head movement.	mm/s ²	float	4000	no	
machine_settings	machine_max_jerk_xy	Default jerk for movement in the horizontal plane.	mm/s	float	20	no	
machine_settings	machine_max_jerk_z	Default jerk for the motor of the Z-direction.	mm/s	float	20	no	
machine_settings	machine_max_jerk_e	Default jerk for the motor of the filament.	mm/s	float	5	no	
machine_settings	machine_steps_per_mm_x	How many steps of the stepper motor will result in one millimeter of movement in the X-direction.	float	1	no		
machine_settings	machine_steps_per_mm_y	How many steps of the stepper motor will result in one millimeter of movement in the Y direction.	float	1	no		
machine_settings	machine_steps_per_mm_z	How many steps of the stepper motor will result in one millimeter of movement in the Z direction.	float	1	no		
machine_settings	machine_steps_per_mm_e	How many steps of the stepper motor will result in moving the feeder wheel by one millimeter around its circumference.	float	1	no		for calculation of steps in X3G format https://github.com/Ultimaker/Cura/issues/5756
machine_settings	machine_endstop_positive_direction_x	Whether the endstop of the X axis is in the positive direction (high X coordinate) or negative (low X coordinate).	bool	False	no		
machine_settings	machine_endstop_positive_direction_y	Whether the endstop of the Y axis is in the positive direction (high Y coordinate) or negative (low Y coordinate).	bool	False	no		
machine_settings	machine_endstop_positive_direction_z	Whether the endstop of the Z axis is in the positive direction (high Z coordinate) or negative (low Z coordinate).	bool	True	no		
machine_settings	machine_minimum_feedrate	The minimal movement speed of the print head.	mm/s	float	0	no	speed set in .src file (not taken from G-Code)
machine_settings	machine_feeder_wheel_diameter	The diameter of the wheel that drives the material in the feeder.	mm	float	10	no	
machine_settings	machine_scale_fan_speed_zero_to_one	Scale the fan speed to be between 0 and 1 instead of between 0 and 256.	bool	False	no		
resolution	layer_height	The height of each layer in mm. Higher values produce faster prints in lower resolution, lower values produce slower prints in higher resolution.	mm	float	15	yes	
resolution	layer_height_0	The height of the initial layer in mm. A thicker initial layer makes adhesion to the build plate easier.	mm	float	15	yes	

wall_line_width	wall_line_width_0	Width of the outermost wall line. By lowering this value, higher levels of detail can be printed.	mm	float	25	yes
wall_line_width	wall_line_width_x	Width of a single wall line for all wall lines except the outermost one.	mm	float	25	yes
line_width	skin_line_width	Width of a single top/bottom line.	mm	float	25	yes
line_width	infill_line_width	Width of a single infill line.	mm	float	25	yes
line_width	skirt_brim_line_width	Width of a single skirt or brim line.	mm	float	25	yes
line_width	support_line_width	Width of a single support structure line.	mm	float	25	yes
support_interface	support_roof_line_width	Width of a single support roof line.	mm	float	25	yes
support_interface	support_bottom_line_width	Width of a single support floor line.	mm	float	25	yes
line_width	prime_tower_line_width	Width of a single prime tower line.	mm	float	25	possibly fixed
resolution	initial_layer_line_width_factor	Multiplier of the line width on the first layer. Increasing this could improve bed adhesion.	%	float	100	fixed
wall_extruder_nr	wall_0_extruder_nr	The extruder train used for printing the outer wall. This is used in multi-extrusion.	int	0	no	
wall_extruder_nr	wall_x_extruder_nr	The extruder train used for printing the inner walls. This is used in multi-extrusion.	int	0	no	
wall_thickness	wall_line_count	The number of walls. When calculated by the wall thickness, this value is rounded to a whole number.	int	2	yes	
shell	wall_transition_length	When transitioning between different numbers of walls as the part becomes thinner, a certain amount of space is allotted to split or join the wall lines.	mm	float	25	yes
shell	wall_distribution_count	The number of walls, counted from the center, over which the variation needs to be spread. Lower values mean that the outer walls don't change in width.	int	1	yes	
shell	wall_transition_angle	When to create transitions between even and odd numbers of walls. A wedge shape with an angle greater than this setting will not have transitions and no wall will be printed in the center to fill the remaining space. Reducing this setting reduces the number and length of these center walls, but may leave gaps or overextrude.	°	float	10	yes
shell	wall_transition_filter_distance	If it would be transitioning back and forth between different numbers of walls in quick succession, don't transition at all. Remove transitions if they are closer together than this distance.	mm	float	100	yes
shell	wall_transition_filter_deviation	Prevent transitioning back and forth between one extra wall and one less. This margin extends the range of line widths which follow to [Minimum Wall Line Width - Margin, 2 * Minimum Wall Line Width + Margin]. Increasing this margin reduces the number of transitions, which reduces the number of extrusion starts/stops and travel time. However, large line width variation can lead to under- or overextrusion problems.	mm	float	0.1	yes
shell	wall_0_wipe_dist	Distance of a travel move inserted after the outer wall, to hide the Z seam better.	mm	float	12.5	yes

shell	wall_0_inset	Inset applied to the path of the outer wall. If the outer wall is smaller than the nozzle, and printed after the inner walls, use this offset to get the hole in the nozzle to overlap with the inner walls instead of the outside of the model.	mm	float	0	yes
shell	optimize_wall_printing_order	Optimize the order in which walls are printed so as to reduce the number of retractions and the distance travelled. Most parts will benefit from this being enabled but some may actually take longer so please compare the print time estimates with and without optimization. First layer is not optimized when choosing brim as build plate adhesion type.	bool	True	yes	
shell	inset_direction	Determines the order in which walls are printed. Printing outer walls earlier helps with dimensional accuracy, as faults from inner walls cannot propagate to the outside. However printing them later allows them to stack better when overhangs are printed. When there is an uneven amount of total inner walls, the 'center last line' is always printed last.	key	outside_in	yes	
shell	alternate_extra_perimeter	Prints an extra wall at every other layer. This way infill gets caught between these extra walls, resulting in stronger prints.	bool	False	yes	
min_wall_line_width	min_even_wall_line_width	The minimum line width for normal polygonal walls. This setting determines at which model thickness we switch from printing a single thin wall line, to printing two wall lines. A higher Minimum Even Wall Line Width leads to a higher maximum odd wall line width. The maximum even wall line width is calculated as Outer Wall Line Width + 0.5 * Minimum Odd Wall Line Width.	mm	float	25	yes
min_wall_line_width	min_odd_wall_line_width	The minimum line width for middle line gap filler polyline walls. This setting determines at which model thickness we switch from printing two wall lines, to printing two outer walls and a single central wall in the middle. A higher Minimum Odd Wall Line Width leads to a higher maximum even wall line width. The maximum odd wall line width is calculated as 2 * Minimum Even Wall Line Width.	mm	float	25	yes
shell	fill_outline_gaps	Print pieces of the model which are horizontally thinner than the nozzle size.	bool	True	yes	
shell	min_feature_size	Minimum thickness of thin features. Model features that are thinner than this value will be printed while features thicker than the Minimum Feature Size will be widened to the Minimum Wall Line Width.	mm	float	15	yes
shell	min_bead_width	Width of the wall that will replace thin features (according to the Minimum Feature Size) of the model. If the Minimum Wall Line Width is thinner than the thickness of the feature, the wall will become as thick as the feature itself.	mm	float	12.5	yes
shell	xy_offset	Amount of offset applied to all polygons in each layer. Positive values can compensate for too big holes; negative values can compensate for too small holes.	mm	float	0	yes

shell	xy_offset_layer_0	Amount of offset applied to all polygons in the first layer. A negative value can compensate for squishing of the first layer known as "elephant's foot".	mm	float	0	yes
shell	hole_xy_offset	When greater than zero, the Hole Horizontal Expansion is the amount of offset applied to all holes in each layer. Positive values increase the size of the holes, negative values reduce the size of the holes. When this setting is enabled it can be further tuned with Hole Horizontal Expansion Max Diameter.	mm	float	0	yes
shell	hole_xy_offset_max_diameter	When greater than zero, the Hole Horizontal Expansion is gradually applied on small holes (small holes are expanded more). When set to zero the Hole Horizontal Expansion will be applied to all holes. Holes larger than the Hole Horizontal Expansion Max Diameter are not expanded.	mm	float	0	yes
shell	z_seam_type	Starting point of each path in a layer. When paths in consecutive layers start at the same point a vertical seam may show on the print. When aligning these near a user specified location, the seam is easiest to remove. When placed randomly the inaccuracies at the paths' start will be less noticeable. When taking the shortest path the print will be quicker.	key	sharpest_corner	yes	
shell	z_seam_on_vertex	Place the z-seam on a polygon vertex. Switching this off can place the seam between vertices as well. (Keep in mind that this won't override the restrictions on placing the seam on an unsupported overhang.)	bool	False	yes	
z_seam_position	z_seam_x	The X coordinate of the position near where to start printing each part in a layer.	mm	float	100	yes
z_seam_position	z_seam_y	The Y coordinate of the position near where to start printing each part in a layer.	mm	float	100	yes
shell	z_seam_corner	Control whether corners on the model outline influence the position of the seam. None means that corners have no influence on the seam position. Hide Seam makes the seam more likely to occur on an inside corner. Expose Seam makes the seam more likely to occur on an outside corner. Hide or Expose Seam makes the seam more likely to occur at an inside or outside corner. Smart Hiding allows both inside and outside corners, but chooses inside corners more frequently, if appropriate.	key	z_seam_corner_inner	yes	
shell	z_seam_relative	When enabled, the z seam coordinates are relative to each part's centre. When disabled, the coordinates define an absolute position on the build plate.	bool	False	yes	
top_bottom	roofing_extruder_nr	The extruder train used for printing the top most skin. This is used in multi-extrusion.	int	0	no	
roofing_layer_count	roofing_line_width	Width of a single line of the areas at the top of the print.	mm	float	25	yes
roofing_layer_count	roofing_pattern	The pattern of the top most layers.	key	lines	yes	

roofing_layer_count	roofing_monotonic	Print top surface lines in an ordering that causes them to always overlap with adjacent lines in a single direction. This takes slightly more time to print, but makes flat surfaces look more consistent.	bool	True	yes
roofing_layer_count	roofing_angles	A list of integer line directions to use when the top surface skin layers use the lines or zig zag pattern. Elements from the list are used sequentially as the layers progress and when the end of the list is reached, it starts at the beginning again. The list items are separated by commas and the whole list is contained in square brackets. Default is an empty list which means use the traditional default angles (45 and 135 degrees).	list[float]	[45, 135]	yes
top_bottom	top_bottom_extruder_nr	The extruder train used for printing the top and bottom skin. This is used in multi-extrusion.	int	0	no
top_thickness	top_layers	The number of top layers. When calculated by the top thickness, this value is rounded to a whole number.	int	3	yes
bottom_thickness	bottom_layers	The number of bottom layers. When calculated by the bottom thickness, this value is rounded to a whole number.	int	3	yes
bottom_thickness	initial_bottom_layers	The number of initial bottom layers from the build-plate upwards. When calculated by the bottom thickness, this value is rounded to a whole number.	int	3	yes
top_bottom	top_bottom_pattern	The pattern of the top/bottom layers.	str	lines	yes
top_bottom	top_bottom_pattern_0	The pattern on the bottom of the print on the first layer.	str	lines	yes
top_bottom	connect_skin_polygons	Connect top/bottom skin paths where they run next to each other. For the concentric pattern enabling this setting greatly reduces the travel time, but because the connections can happen midway over infill this feature can reduce the top surface quality.	bool	False	yes
top_bottom	skin_monotonic	Print top/bottom lines in an ordering that causes them to always overlap with adjacent lines in a single direction. This takes slightly more time to print, but makes flat surfaces look more consistent.	bool	False	yes
top_bottom	skin_angles	A list of integer line directions to use when the top/bottom layers use the lines or zig zag pattern. Elements from the list are used sequentially as the layers progress and when the end of the list is reached, it starts at the beginning again. The list items are separated by commas and the whole list is contained in square brackets. Default is an empty list which means use the traditional default angles (45 and 135 degrees).	list[float]	[]	yes
top_bottom	small_skin_width	Small top/bottom regions are filled with walls instead of the default top/bottom pattern. This helps to avoid jerky motions. Off for the topmost (air-exposed) layer by default (see 'Small Top/Bottom On Surface').	float	1	yes

top_bottom	small_skin_on_surface	Enable small (up to 'Small Top/Bottom Width') regions on the top/bottom skinned layer (exposed to air) to be filled with walls instead of the default pattern.	bool	False	yes
top_bottom	skin_no_small_gaps_heuristic	When the model has small vertical gaps of only a few layers, there should normally be skin around those layers in the narrow space. Enable this setting to not generate skin if the vertical gap is very small. This improves printing time and slicing time, but technically leaves infill exposed to the air.	bool	False	yes
top_bottom	skin_outline_count	Replaces the outermost part of the top/bottom pattern with a number of concentric lines. Using one or two lines improves roofs that start on infill material.	int	1	yes
top_bottom	ironing_enabled	Go over the top surface one additional time, but this time extruding very little material. This is meant to melt the plastic on top further, creating a smoother surface. The pressure in the nozzle chamber is kept high so that the creases in the surface are filled with material.	bool	False	fixed due to no possibility to determine the ironing layer for flow calculation
top_bottom	ironing_only_highest_layer	Only perform ironing on the very last layer of the mesh. This saves time if the lower layers don't need a smooth surface finish.	bool	True	no
top_bottom	ironing_pattern	The pattern to use for ironing top surfaces.	key	zigzag	no
top_bottom	ironing_monotonic	Print ironing lines in an ordering that causes them to always overlap with adjacent lines in a single direction. This takes slightly more time to print, but makes flat surfaces look more consistent.	bool	False	no
top_bottom	ironing_line_spacing	The distance between the lines of ironing.	mm	7.5	no
top_bottom	ironing_flow	The amount of material, relative to a normal skin line, to extrude during ironing. Keeping the nozzle filled helps filling some of the crevices of the top surface, but too much results in overextrusion and blips on the side of the surface.	float	10	no
top_bottom	ironing_inset	A distance to keep from the edges of the model. Ironing all the way to the edge of the mesh may result in a jagged edge on your print.	mm	float	12.5 no
top_bottom	speed_ironing	The speed at which to pass over the top surface.	mm/s	float	200 no
top_bottom	acceleration_ironing	The acceleration with which ironing is performed.	mm/s ²	float	3000 no
top_bottom	jerk_ironing	The maximum instantaneous velocity change while performing ironing.	mm/s	float	20 no
skin_overlap	skin_overlap_mm	Adjust the amount of overlap between the walls and (the endpoints of) the skin-centerlines. A slight overlap allows the walls to connect firmly to the skin. Note that, given an equal skin and wall line-width, any value over half the width of the wall may already cause any skin to go past the wall, because at that point the position of the nozzle of the skin-extruder may already reach past the middle of the wall.	mm	float	1 yes

skin_preshrink	top_skin_preshrink	The largest width of top skin areas which are to be removed. Every skin area smaller than this value will disappear. This can help in limiting the amount of time and material spent on printing top skin at slanted surfaces in the model.	mm	float	1	yes
skin_preshrink	bottom_skin_preshrink	The largest width of bottom skin areas which are to be removed. Every skin area smaller than this value will disappear. This can help in limiting the amount of time and material spent on printing bottom skin at slanted surfaces in the model.	mm	float	1	yes
expand_skins_expand_distance	top_skin_expand_distance	The distance the top skins are expanded into the infill. Higher values makes the skin attach better to the infill pattern and makes the walls on the layer above adhere better to the skin. Lower values save amount of material used.	mm	float	1	yes
expand_skins_expand_distance	bottom_skin_expand_distance	The distance the bottom skins are expanded into the infill. Higher values makes the skin attach better to the infill pattern and makes the skin adhere better to the walls on the layer below. Lower values save amount of material used.	mm	float	1	yes
max_skin_angle_for_expansion	min_skin_width_for_expansion	Skin areas narrower than this are not expanded. This avoids expanding the narrow skin areas that are created when the model surface has a slope close to the vertical.	mm	float	0	yes
infill	infill_extruder_nr	The extruder train used for printing infill. This is used in multi-extrusion.	int	0	no	
infill_sparse_density	infill_line_distance	Distance between the printed infill lines. This setting is calculated by the infill density and the infill line width.	mm	float	2	yes
infill	infill_pattern	The pattern of the infill material of the print. The line and zig zag infill swap direction on alternate layers, reducing material cost. The grid, triangle, tri-hexagon, cubic, octet, quarter cubic, cross and concentric patterns are fully printed every layer. Gyroid, cubic, quarter cubic and octet infill change with every layer to provide a more equal distribution of strength over each direction. Lighting infill tries to minimize the infill, by only supporting the ceiling of the object.	key	grid	yes	
infill	zig_zagify_infill	Connect the ends where the infill pattern meets the inner wall using a line which follows the shape of the inner wall. Enabling this setting can make the infill adhere to the walls better and reduce the effects of infill on the quality of vertical surfaces. Disabling this setting reduces the amount of material used.	bool	False	yes	
infill	connect_infill_polygons	Connect infill paths where they run next to each other. For infill patterns which consist of several closed polygons, enabling this setting greatly reduces the travel time.	bool	True	yes	

infill	infill_angles	A list of integer line directions to use. Elements from the list are used sequentially as the layers progress and when the end of the list is reached, it starts at the beginning again. The list items are separated by commas and the whole list is contained in square brackets. Default is an empty list which means use the traditional default angles (45 and 135 degrees for the lines and zig zag patterns and 45 degrees for all other patterns).	list[float]	[]	yes
infill	infill_offset_x	The infill pattern is moved this distance along the X axis.	float	0	yes
infill	infill_offset_y	The infill pattern is moved this distance along the Y axis.	float	0	yes
infill	infill_randomize_start_location	Randomize which infill line is printed first. This prevents one segment becoming the strongest, but it does so at the cost of an additional travel move.	bool	False	yes
infill	infill_multiplier	Convert each infill line to this many lines. The extra lines do not cross over each other, but avoid each other. This makes the infill stiffer, but increases print time and material usage.	int	1	yes
infill	infill_wall_line_count	Add extra walls around the infill area. Such walls can make top/bottom skin lines sag down less which means you need less top/bottom skin layers for the same quality at the cost of some extra material. This feature can combine with the Connect Infill Polygons to connect all the infill into a single extrusion path without the need for travels or retractions if configured right.	int	0	yes
infill	sub_div_rad_add	An addition to the radius from the center of each cube to check for the boundary of the model, as to decide whether this cube should be subdivided. Larger values lead to a thicker shell of small cubes near the boundary of the model.	mm	float	25 possibly
infill_overlap	infill_overlap_mm	The amount of overlap between the infill and the walls. A slight overlap allows the walls to connect firmly to the infill.	mm	float	2.5 yes
infill	infill_wipe_dist	Distance of a travel move inserted after every infill line, to make the infill stick to the walls better. This option is similar to infill overlap, but without extrusion and only on one end of the infill line.	mm	float	2.5 yes
infill	infill_sparse_thickness	The thickness per layer of infill material. This value should always be a multiple of the layer height and is otherwise rounded.	mm	float	15 yes
infill	gradual_infill_steps	Number of times to reduce the infill density by half when getting further below top surfaces. Areas which are closer to top surfaces get a higher density up to the infill Density.	int	0	yes
infill	gradual_infill_step_height	The height of infill of a given density before switching to half the density.	mm	float	1.5 yes

infill	infill_before_walls	Print the infill before printing the walls. Printing the walls first may lead to more accurate walls, but overhangs print worse. Printing the infill first leads to sturdier walls, but the infill pattern might sometimes show through the surface.	bool	True	yes
infill	min_infill_area	Don't generate areas of infill smaller than this (use skin instead). Print infill structures only where tops of the model should be supported. Enabling this reduces print time and material usage, but leads to uniform object strength.	mm ²	0	yes
infill	infill_support_enabled	The minimum angle of internal overhangs for which infill is added. At a value of 0° objects are totally filled with infill, 30° will not provide any infill.	bool	False	yes
infill	infill_support_angle	The number of infill layers that supports skin edges.	int	4	yes
skin_edge_support_thickness	skin_edge_support_layers	Determines when a lightning infill layer has to support the model above it. Measured in the angle given the thickness.	float	40	yes
lightning_infill_support_angle	lightning_infill_overhang_angle	The endpoints of infill lines are shortened to save on material. This setting is the angle of overhang of the endpoints of these lines.	float	40	yes
lightning_infill_support_angle	lightning_infill_prune_angle	The infill lines are straightened out to save on printing time. This is the maximum angle of overhang allowed across the length of the infill line.	float	40	yes
material	default_material_print_temperature	The default temperature used for printing. This should be the "base" temperature of a material. All other print temperatures should use offsets based on this value	°C	20	no
material	build_volume_temperature	The temperature of the environment to print in. If this is 0, the build volume temperature will not be adjusted.	float	0	no
material	material_print_temperature	The temperature used for printing the first layer.	°C	20	no
material	material_print_temperature_layer_0	The minimal temperature while heating up to the Printing Temperature at which printing can already start.	°C	20	no
material	material_initial_print_temperature	The temperature to which to already start cooling down just before the end of printing.	°C	20	no
material	material_final_print_temperature	The extra speed by which the nozzle cools while extruding. The same value is used to signify the heat up speed lost when heating up while extruding.	°C/s	1	no
material	material_extrusion_cool_down_speed	The default temperature used for the heated build plate. This should be the "base" temperature of a build plate. All other print temperatures should use offsets based on this value.	°C	20	no
material	default_material_bed_temperature	The temperature used for the heated build plate. If this is 0, the build plate is left unheated.	°C	20	no
material	material_bed_temperature	The temperature used for the heated build plate at the first layer. If this is 0, the build plate is left unheated during the first layer..	°C	20	no
material	material_adhesion_tendency	Surface adhesion tendency.	int	10	no

			Surface energy.						
material	material_surface_energy		To compensate for the shrinkage of the material as it cools down, the model will be scaled with this factor in the XY-direction (horizontally).	%	int	100	no		
material_shrinkage_percentage	material_shrinkage_percentage_xy		To compensate for the shrinkage of the material as it cools down, the model will be scaled with this factor in the Z-direction (vertically).	%	float	100	yes		
material_shrinkage_percentage	material_shrinkage_percentage_z		Is this material the type that breaks off cleanly when heated (crystalline) or is it the type that produces long intertwined polymer chains (non-crystalline)?	%	float	100	yes		
material	material_crystallinity			bool	False	no			
material	material_anti_ozone_retracted_position		How far the material needs to be retracted before it stops oziling.	mm	float	0	no		
material	material_anti_ozone_retraction_speed		How fast the material needs to be retracted during a filament switch to prevent oziling.	mm/s	float	250	no		
material	material_break_preparation_retracted_position		How far the filament can be stretched before it breaks, while heated.	mm	float	0	no		
material	material_break_preparation_speed		How fast the filament needs to be retracted just before breaking it off in a retraction.	mm/s	float	250	no		
material	material_break_preparation_temperature		The temperature used to purge material should be roughly equal to the highest possible printing temperature.	°C	float	20	no		
material	material_break_retracted_position		How far to retract the filament in order to break it cleanly.	mm	float	0	no		
material	material_break_speed		The speed at which to retract the filament in order to break it cleanly.	mm/s	float	250	no		
material	material_break_temperature		The temperature at which the filament is broken for a clean break.	°C	float	20	no		
material	material_flush_purge_speed		How fast to prime the material after switching to a different material.		float	50	no		
material	material_flush_purge_length		How much material to use to purge the previous material out of the nozzle (in length of filament) when switching to a different material.		float	100	no		
material	material_end_of_filament_purge_speed		How fast to prime the material after replacing an empty spool with a fresh spool of the same material.		float	50	no		
material	material_end_of_filament_purge_length		How much material to use to purge the previous material out of the nozzle (in length of filament) when replacing an empty spool with a fresh spool of the same material.		float	100	no		
material	material_maximum_park_duration		How long the material can be kept out of dry storage safely.		float	300	no		
material	material_no_load_move_factor		A factor indicating how much the filament gets compressed between the feeder and the nozzle chamber, used to determine how far to move the material for a filament switch.		float	0.940860215	no		
wall_material_flow	wall_0_material_flow		Flow compensation on the outermost wall line.	%	float	100	yes	setup.json [Pump.linetype_flow ("wall_outer")]	
wall_material_flow	wall_x_material_flow		Flow compensation on wall lines for all wall lines except the outermost one.	%	float	100	yes	setup.json [Pump.linetype_flow ("wall_inner")]	
wall_material_flow	wall_0_material_flow_roofing		Flow compensation on the top surface outermost wall line.	%	float	100	no	setup.json [Pump.linetype_flow ("wall_outer")]	
wall_material_flow	wall_x_material_flow_roofing		Flow compensation on top surface wall lines for all wall lines except the outermost one.	%	float	100	no	setup.json [Pump.linetype_flow ("wall_inner")]	
material_flow	skin_material_flow		Flow compensation on top/bottom lines.	%	float	100	partially	setup.json [Pump.linetype_flow ("surface")]	
material_flow	roofing_material_flow		Flow compensation on lines of the areas at the top of the print.	%	float	100	partially	setup.json [Pump.linetype_flow ("surface")]	

material_flow	infill_material_flow	Flow compensation on infill lines.	%	float	100	yes	setup.json [Pump.linetype_flow ("infill")]
material_flow	skirt_brim_material_flow	Flow compensation on skirt or brim lines.	%	float	100	yes	setup.json [Pump.linetype_flow ("curb")]
material_flow	support_material_flow	Flow compensation on support structure lines.	%	float	100	yes	setup.json [Pump.linetype_flow ("support")]
support_interface_material_flow	support_roof_material_flow	Flow compensation on support roof lines.	%	float	100	no	setup.json [Pump.linetype_flow ("support")]
support_interface_material_flow	support_bottom_material_flow	Flow compensation on support floor lines.	%	float	100	no	setup.json [Pump.linetype_flow ("support")]
support_interface_material_flow	prime_tower_flow	Flow compensation on prime tower lines.	%	float	100	no	Multi-material not supported
material	material_flow_layer_0	Flow compensation for the first layer: the amount of material extruded on the initial layer is multiplied by this value.	%	float	100	fixed	
material	wall_x_material_flow_layer_0	Flow compensation on wall lines for all wall lines except the outermost one, but only for the first layer.	%	float	100	no	setup.json [Pump.linetype_flow ("wall_outer")]
material	wall_0_material_flow_layer_0	Flow compensation on the outermost wall line of the first layer.	%	float	100	no	setup.json [Pump.linetype_flow ("wall_inner")]
material	skin_material_flow_layer_0	Flow compensation on bottom lines of the first layer.	%	float	100	no	setup.json [Pump.linetype_flow ("surface")]
material	material_standby_temperature	The temperature of the nozzle when another nozzle is currently used for printing.	°C	float	20	no	
material	material_is_support_material	Is this material typically used as a support material during printing.		bool	False	no	
speed_print	speed_infill	The speed at which infill is printed.	mm/s	float	60	no	
speed_wall	speed_wall_0	The speed at which the outermost walls are printed. Printing the outer wall at a lower speed improves the final skin quality. However, having a large difference between the inner wall speed and the outer wall speed will affect quality in a negative way.	mm/s	float	30	no	
speed_wall	speed_wall_x	The speed at which all inner walls are printed. Printing the inner wall faster than the outer wall will reduce printing time. It works well to set this in between the outer wall speed and the infill speed.	mm/s	float	60	no	
speed_wall	speed_wall_0_roofing	The speed at which the top surface outermost wall is printed.	mm/s	float	30	no	
speed_wall	speed_wall_x_roofing	The speed at which the top surface inner walls are printed.	mm/s	float	60	no	
speed_print	speed_roofing	The speed at which top surface skin layers are printed.	mm/s	float	25	no	
speed_print	speed_topbottom	The speed at which top/bottom layers are printed.	mm/s	float	30	no	
speed_support	speed_support_infill	The speed at which the infill of support is printed. Printing the infill at lower speeds improves stability.	mm/s	float	60	no	
speed_support_interface	speed_support_roof	The speed at which the roofs of support are printed. Printing them at lower speeds can improve overhang quality.	mm/s	float	40	no	
speed_support_interface	speed_support_bottom	The speed at which the floor of support is printed. Printing it at lower speed can improve adhesion of support on top of your model.	mm/s	float	40	no	
speed_print	speed_prime_tower	The speed at which the prime tower is printed. Printing the prime tower slower can make it more stable when the adhesion between the different filaments is suboptimal.	mm/s	float	60	no	
speed	speed_travel	The speed at which travel moves are made.	mm/s	float	120	no	

speed_layer_0	speed_print_layer_0	The speed of printing for the initial layer. A lower value is advised to improve adhesion to the build plate.	mm/s	float	30	no
speed_layer_0	speed_travel_layer_0	The speed of travel moves in the initial layer. A lower value is advised to prevent pulling previously printed parts away from the build plate. The value of this setting can automatically be calculated from the ratio between the Travel Speed and the Print Speed.	mm/s	float	60	no
speed	skirt_brim_speed	The speed at which the skirt and brim are printed. Normally this is done at the initial layer speed, but sometimes you might want to print the skirt or brim at a different speed.	mm/s	float	30	no
speed	speed_z_hop	The speed at which the vertical Z movement is made for Z Hops. This is typically lower than the print speed since the build plate or machine's gantry is harder to move.	mm/s	float	10	no
speed	speed_slowdown_layers	The first few layers are printed slower than the rest of the model, to get better adhesion to the build plate and improve the overall success rate of prints. The speed is gradually increased over these layers.	int	2	no	
speed	speed_equalize_flow_width_factor	Extrusion width based correction factor on the speed. At 0% the movement speed is kept constant at the Print Speed. At 100% the movement speed is adjusted so that the flow (in mm ³ /s) is kept constant, i.e. lines half the normal Line Width are printed twice as fast, and lines twice as wide are printed half as fast. A value larger than 100% can help to compensate for the higher pressure required to extrude wide lines.	%	float	100	no
speed	acceleration_enabled	Enables adjusting the print head acceleration. Increasing the accelerations can reduce printing time at the cost of print quality.	bool	False	no	
speed	acceleration_travel_enabled	Use a separate acceleration rate for travel moves. If disabled, travel moves will use the acceleration value of the printed line at their destination.	bool	True	no	
acceleration_print	acceleration_infill	The acceleration with which infill is printed.	mm/s ²	float	3000	no
acceleration_wall	acceleration_wall_0	The acceleration with which the outermost walls are printed.	mm/s ²	float	3000	no
acceleration_wall	acceleration_wall_x	The acceleration with which all inner walls are printed.	mm/s ²	float	3000	no
acceleration_wall	acceleration_wall_0_roofing	The acceleration with which the top surface outermost walls are printed.	mm/s ²	float	3000	no
acceleration_wall	acceleration_wall_x_roofing	The acceleration with which the top surface inner walls are printed.	mm/s ²	float	3000	no
acceleration_print	acceleration_roofing	The acceleration with which top surface skin layers are printed.	mm/s ²	float	3000	no
acceleration_print	acceleration_topbottom	The acceleration with which top/bottom layers are printed.	mm/s ²	float	3000	no
acceleration_support	acceleration_support_infill	The acceleration with which the infill of support is printed.	mm/s ²	float	3000	no
acceleration_support	acceleration_support_roof	The acceleration with which the roofs of support are printed. Printing them at lower acceleration can improve overhang quality.	mm/s ²	float	3000	no
acceleration_support_interface						

acceleration_support_interface	acceleration_support_bottom	The acceleration with which the floors of support are printed. Printing them at lower acceleration can improve adhesion of support on top of your model.	mm/s ²	float	3000	no
acceleration_print	acceleration_prime_tower	The acceleration with which the prime tower is printed.	mm/s ²	float	3000	no
speed	acceleration_travel	The acceleration with which travel moves are made.	mm/s ²	float	5000	no
acceleration_layer_0	acceleration_print_layer_0	The acceleration during the printing of the initial layer.	mm/s ²	float	3000	no
acceleration_layer_0	acceleration_travel_layer_0	The acceleration for travel moves in the initial layer.	mm/s ²	float	3000	no
speed	acceleration_skirt_brim	The acceleration with which the skirt and brim are printed. Normally this is done with the initial layer acceleration, but sometimes you might want to print the skirt or brim at a different acceleration.	mm/s ²	float	3000	no
speed	jerk_enabled	Enables adjusting the jerk of print head when the velocity in the X or Y axis changes. Increasing the jerk can reduce printing time at the cost of print quality.	bool	False	no	
speed	jerk_travel_enabled	Use a separate jerk rate for travel moves. If disabled, travel moves will use the jerk value of the printed line at their destination.	bool	True	no	
jerk_print	jerk_infill	The maximum instantaneous velocity change with which infill is printed.	mm/s	float	20	no
jerk_wall	jerk_wall_0	The maximum instantaneous velocity change with which the outermost walls are printed.	mm/s	float	20	no
jerk_wall	jerk_wall_x	The maximum instantaneous velocity change with which all inner walls are printed.	mm/s	float	20	no
jerk_wall	jerk_wall_0_roofing	The maximum instantaneous velocity change with which the top surface outermost walls are printed.	mm/s	float	20	no
jerk_wall	jerk_wall_x_roofing	The maximum instantaneous velocity change with which the top surface inner walls are printed.	mm/s	float	20	no
jerk_print	jerk_roofing	The maximum instantaneous velocity change with which top surface skin layers are printed.	mm/s	float	20	no
jerk_print	jerk_topbottom	The maximum instantaneous velocity change with which top/bottom layers are printed.	mm/s	float	20	no
jerk_support	jerk_support_infill	The maximum instantaneous velocity change with which the infill of support is printed.	mm/s	float	20	no
jerk_support_interface	jerk_support_roof	The maximum instantaneous velocity change with which the roofs of support are printed.	mm/s	float	20	no
jerk_support_interface	jerk_support_bottom	The maximum instantaneous velocity change with which the floors of support are printed.	mm/s	float	20	no
jerk_print	jerk_prime_tower	The maximum instantaneous velocity change with which the prime tower is printed.	mm/s	float	20	no
speed	jerk_travel	The maximum instantaneous velocity change with which travel moves are made.	mm/s	float	30	no
jerk_layer_0	jerk_print_layer_0	The maximum instantaneous velocity change during the printing of the initial layer.	mm/s	float	20	no
jerk_layer_0	jerk_travel_layer_0	The acceleration for travel moves in the initial layer.	mm/s	float	20	no
speed	jerk_skirt_brim	The maximum instantaneous velocity change with which the skirt and brim are printed.	mm/s	float	20	no
travel	retraction_enable	Retract the filament when the nozzle is moving over a non-printed area.	bool	True	yes	

travel	retract_at_layer_change	Retract the filament when the nozzle is moving to the next layer.		bool	False	yes
travel	retraction_amount	The length of material retracted during a retraction move.	mm	float	100	yes
retraction_speed	retraction_retract_speed	The speed at which the filament is retracted during a retraction move.	mm/s	float	25	no
retraction_speed	retraction_prime_speed	The speed at which the filament is primed during a retraction move.	mm/s	float	25	no
travel	retraction_extra_prime_amount	Some material can ooze away during a travel move, which can be compensated for here.	mm³	float	0	no
travel	retraction_min_travel	The minimum distance of travel needed for a retraction to happen at all. This helps to get fewer retractions in a small area.	mm	float	50	yes
travel	retraction_count_max	This setting limits the number of retractions occurring within the minimum extrusion distance window. Further retractions within this window will be ignored. This avoids retracting repeatedly on the same piece of filament, as that can flatten the filament and cause grinding issues.	int	5	yes	
travel	retraction_extrusion_window	The window in which the maximum retraction count is enforced. This value should be approximately the same as the retraction distance, so that effectively the number of times a retraction passes the same patch of material is limited.	mm	float	50	yes
travel	retraction_combing	Combing keeps the nozzle within already printed areas when traveling. This results in slightly longer travel moves but reduces the need for retractions. If combing is off, the material will retract and the nozzle moves in a straight line to the next point. It is also possible to avoid combing over top/bottom skin areas or to only comb within the infill.	key	all	yes	
travel	retraction_combing_max_distance	When greater than zero, combing travel moves that are longer than this distance will use retraction. If set to zero, there is no maximum and combing moves will not use retraction.	mm	float	0	yes
travel	travel_retract_before_outer_wall	Always retract when moving to start an outer wall.	bool	False	yes	
travel	travel_avoid_other_parts	The nozzle avoids already printed parts when traveling. This option is only available when combing is enabled.	bool	True	yes	
travel	travel_avoid_supports	The nozzle avoids already printed supports when traveling. This option is only available when combing is enabled.	bool	False	yes	
travel	travel_avoid_distance	The distance between the nozzle and already printed parts when avoiding during travel moves.	mm	float	5	yes
travel	layer_start_x	The X coordinate of the position near where to find the part to start printing each layer.	mm	float	0	yes
travel	layer_start_y	The Y coordinate of the position near where to find the part to start printing each layer.	mm	float	0	yes
travel	retraction_hop_enabled	Whenever a retraction is done, the build plate is lowered to create clearance between the nozzle and the print. It prevents the nozzle from hitting the print during travel moves, reducing the chance to knock the print from the build plate.	bool	False	fixed	due to layer height calculation a z-hop can not be enabled

travel	retraction_hop_only_when_collides	Only perform a Z Hop when moving over printed parts which cannot be avoided by horizontal motion by Avoid Printed Parts when Traveling.	bool	False	no
travel	retraction_hop	The height difference when performing a Z Hop. After the machine switched from one extruder to the other, the build plate is lowered to create clearance between the nozzle and the print. This prevents the nozzle from leaving oozed material on the outside of a print.	mm	float	1 no
travel	retraction_hop_after_extruder_switch	After the machine switched from one extruder to the other, the build plate is lowered to create clearance between the nozzle and the print. This prevents the nozzle from leaving oozed material on the outside of a print.	bool	True	no
travel	retraction_hop_after_extruder_switch_height	The height difference when performing a Z Hop after extruder switch.	mm	float	1 no
cooling	cool_fan_enabled	Enables the print cooling fans while printing. The fans improve print quality on layers with short layer times and bridging/overhangs.	bool	True	no
cool_fan_speed	cool_fan_speed_min	The speed at which the fans spin before hitting the threshold. When a layer prints faster than the threshold, the fan speed gradually inclines towards the maximum fan speed.	%	float	100 no
cool_fan_speed	cool_fan_speed_max	The speed at which the fans spin on the minimum layer time. The fan speed gradually increases between the regular fan speed and maximum fan speed when the threshold is hit.	%	float	100 no
cooling	cool_min_layer_time_fan_speed_max	The layer time which sets the threshold between regular fan speed and maximum fan speed. Layers that print slower than this time use regular fan speed. For faster layers the fan speed gradually increases towards the maximum fan speed.	s	float	10 no
cooling	cool_fan_speed_0	The speed at which the fans spin at the start of the print. In subsequent layers the fan speed is gradually increased up to the layer corresponding to Regular Fan Speed at Height.	%	float	0 no
cool_fan_full_at_height	cool_fan_full_layer	The layer at which the fans spin on regular fan speed. If regular fan speed at height is set, this value is calculated and rounded to a whole number. The minimum time spent in a layer. This forces the printer to slow down, to at least spend the time set here in one layer. This allows the printed material to cool down properly before printing the next layer. Layers may still take shorter than the minimal layer time if Lift Head is disabled and if the Minimum Speed would otherwise be violated.	s	int	2 no
cooling	cool_min_layer_time	The minimum print speed, despite slowing down due to the minimum layer time. When the printer would slow down too much, the pressure in the nozzle would be too low and result in bad print quality.	mm/s	float	5 no
cooling	cool_min_speed	When the minimum speed is hit because of minimum layer time, lift the head away from the print and wait the extra time until the minimum layer time is reached.	mm/s	float	10 no
cooling	cool_lift_head	Gradually reduce to this temperature when printing at reduced speeds because of minimum layer time.	°C	bool	False no
cooling	cool_min_temperature	Gradually reduce to this temperature when printing at reduced speeds because of minimum layer time.	°C	float	0 no

		<html>Whether to activate the cooling fans during a nozzle switch. This can help reducing oozing by cooling the nozzle				
cooling	cool_during_extruder_switch	faster. <i>Unchanged</i> keep the fans as they were previouslyOnly last extruder: turn on the fan of the last used extruder, but turn the others off (if any). This is useful if you have completely separate extruders.All fans: turn on all fans during nozzle switch. This is useful if you have a single cooling fan, or multiple fans that stay close to each other.</html>	key	unchanged	no	
support	support_enable	Generate structures to support parts of the model which have overhangs. Without these structures, such parts would collapse during printing.	bool	False	possibly	
support_extruder_nr	support_infill_extruder_nr	The extruder train to use for printing the infill of the support. This is used in multi-extrusion.	int	0	no	
support_extruder_nr	support_extruder_nr_layer_0	The extruder train to use for printing the first layer of support infill. This is used in multi-extrusion.	int	0	no	
support_interface_extruder_nr	support_roof_extruder_nr	The extruder train to use for printing the roofs of the support. This is used in multi-extrusion.	int	0	no	
support_interface_extruder_nr	support_bottom_extruder_nr	The extruder train to use for printing the floors of the support. This is used in multi-extrusion.	int	0	no	
support	support_structure	Chooses between the techniques available to generate support. "Normal" support creates a support structure directly below the overhanging parts and drops those areas straight down. "Tree" support creates branches towards the overhanging areas that support the model on the tips of those branches, and allows the branches to crawl around the model to support it from the build plate as much as possible.	key	normal	possibly	
support	support_tree_angle	The maximum angle of the branches while they grow around the model. Use a lower angle to make them more vertical and more stable. Use a higher angle to be able to have more reach.	float	60	possibly	
support	support_tree_branch_diameter	The diameter of the thinnest branches of tree support. Thicker branches are more sturdy. Branches towards the base will be thicker than this.	float	5	possibly	
support	support_tree_max_diameter	The diameter of the widest branches of tree support. A thicker trunk is more sturdy, a thinner trunk takes up less space on the build plate.	float	25	possibly	
support	support_tree_branch_diameter_angle	The angle of the branches' diameter as they gradually become thicker towards the bottom. An angle of 0 will cause the branches to have uniform thickness over their length. A bit of an angle can increase stability of the tree support.	float	7	possibly	
support_z_seam_away_from_model	support_z_seam_min_distance	The distance between the model and its support structure at the z-axis seam.	mm	50	possibly	
support	support_type	Adjusts the placement of the support structures. The placement can be set to touching build plate or everywhere. When set to everywhere the support structures will also be printed on the model.	key	everywhere	possibly	

support	<code>support_tree_angle_slow</code>	The preferred angle of the branches, when they do not have to avoid the model. Use a lower angle to make them more vertical and more stable. Use a higher angle for branches to merge faster.	°	float	50	possibly
support	<code>support_tree_max_diameter_increase_by_merges_when_support_to_model</code>	The most the diameter of a branch that has to connect to the model may increase by merging with branches that could reach the buildplate. Increasing this reduces print time, but increases the area of support that rests on model	mm	float	1	possibly
support	<code>support_tree_min_height_to_model</code>	How tall a branch has to be if it is placed on the model. Prevents small blobs of support. This setting is ignored when a branch is supporting a support roof.	mm	float	50	possibly
support	<code>support_tree_bp_diameter</code>	Diameter every branch tries to achieve when reaching the buildplate. Improves bed adhesion. Adjusts the density of the support structure used to generate the tips of the branches. A higher value results in better overhangs, but the supports are harder to remove. Use <code>Support_Roof</code> for very high values or ensure support density is similarly high at the top.	mm	float	50	possibly
support	<code>support_tree_top_rate</code>		%	float	15	possibly
support	<code>support_tree_tip_diameter</code>	The diameter of the top of the tip of the branches of tree support.	mm	float	25	possibly
support	<code>support_tree_limit_branch_reach</code>	Limit how far each branch should travel from the point it supports. This can make the support more sturdy, but will increase the amount of branches (and because of that material usage/print time)	bool	True	possibly	
support	<code>support_tree_branch_reach_limit</code>	A recommendation to how far branches can move from the points they support. Branches can violate this value to reach their destination (buildplate or a flat part of the model). Lowering this value will make the support more sturdy, but increase the amount of branches (and because of that material usage/print time)	mm	float	30	possibly
support	<code>support_tree_rest_preference</code>	The preferred placement of the support structures. If structures can't be placed at the preferred location, they will be placed elsewhere, even if that means placing them on the model.	key	buildplate	possibly	
support	<code>support_angle</code>	The minimum angle of overhangs for which support is added. At a value of 0° all overhangs are supported. 90° will not provide any support.	°	float	50	possibly
support	<code>support_pattern</code>	The pattern of the support structures of the print. The different options available result in sturdy or easy to remove support.	key	zigzag	possibly	
support	<code>support_wall_count</code>	The number of walls with which to surround support infill. Adding a wall can make support print more reliably and can support overhangs better, but increases print time and material used.	int	1	possibly	
support_interface	<code>support_roof_wall_count</code>	The number of walls with which to surround support interface roof. Adding a wall can make support print more reliably and can support overhangs better, but increases print time and material used.	int	0	possibly	

support_interface _wall_count	support_bottom_wall_count	The number of walls with which to surround support interface floor. Adding a wall can make support print more reliably and can support overhangs better, but increases print time and material used.	int	0	possibly
support	zig_zagify_support	Connect the ends of the support lines together. Enabling this setting can make your support more sturdy and reduce underextrusion, but it will cost more material.	bool	False	possibly
support	support_connect_zigzags	Connect the ZigZags. This will increase the strength of the zig zag support structure.	bool	True	possibly
support_infill_rate	support_line_distance	Distance between the printed support structure lines. This setting is calculated by the support density.	mm	float	possibly
support	support_initial_layer_line_distance	Distance between the printed initial layer support structure lines. This setting is calculated by the support density.	mm	float	possibly
support	support_infill_density_multiplier_initial_layer	Multiplier for the infill on the initial layers of the support. Increasing this may help for bed adhesion.	int	1	possibly
support	support_infill_angles	A list of integer line directions to use. Elements from the list are used sequentially as the layers progress and when the end of the list is reached, it starts at the beginning again. The list items are separated by commas and the whole list is contained in square brackets. Default is an empty list, which means use the default angle 0 degrees.	list[int]	[]	possibly
support	support_brim_enable	Generate a brim within the support infill regions of the first layer. This brim is printed underneath the support, not around it. Enabling this setting increases the adhesion of support to the build plate.	bool	True	possibly
support_brim_width	support_brim_line_count	The number of lines used for the support brim. More brim lines enhance adhesion to the build plate, at the cost of some extra material.	int	3	possibly
support_z_distance	support_top_distance	Distance from the top of the support to the print.	mm	float	possibly
support_z_distance	support_bottom_distance	Distance from the print to the bottom of the support. Note that this is rounded up to the next layer height.	mm	float	possibly
support	support_xy_distance	Distance of the support structure from the print in the X/Y directions.	mm	float	possibly
support	support_xy_overrides_z	Whether the Support X/Y Distance overrides the Support Z Distance or vice versa. When X/Y overrides Z the X/Y distance can push away the support from the model, influencing the actual Z distance to the overhang. We can disable this by not applying the X/Y distance around overhangs.	key	z_overrides_xy	possibly
support	support_xy_distance_overhang	Distance of the support structure from the overhang in the X/Y directions.	mm	float	possibly
support	support_bottom_stair_step_height	The height of the steps of the stair-like bottom of support resting on the model. A low value makes the support harder to remove, but too high values can lead to unstable support structures. Set to zero to turn off the stair-like behaviour.	mm	float	possibly

support	support_bottom_stair_step_width	The maximum width of the steps of the stair-like bottom of support resting on the model. A low value makes the support harder to remove, but too high values can lead to unstable support structures.	mm	float	5	possibly
support	support_bottom_stair_step_min_slope	The minimum slope of the area for stair-stepping to take effect. Low values should make support easier to remove on shallower slopes, but really low values may result in some very counter-intuitive results on other parts of the model.	°	float	10	possibly
support	support_join_distance	The maximum distance between support structures in the X/Y directions. When separate structures are closer together than this value, the structures merge into one.	mm	float	2	possibly
support	support_offset	Amount of offset applied to all support polygons in each layer. Positive values can smooth out the support areas and result in more sturdy support.	mm	float	50	possibly
support	support_infill_sparse_thickness	The thickness per layer of support infill material. This value should always be a multiple of the layer height and is otherwise rounded.	mm	float	15	possibly
support	gradual_support_infill_steps	Number of times to reduce the support infill density by half when getting further below top surfaces. Areas which are closer to top surfaces get a higher density, up to the Support Infill Density.	int	0	possibly	
support	gradual_support_infill_step_height	The height of support infill of a given density before switching to half the density.	mm	float	1	possibly
support	minimum_support_area	Minimum area size for support polygons. Polygons which have an area smaller than this value will not be generated.	mm²	float	0	possibly
support_interface_enable	support_roof_enable	Generate a dense slab of material between the top of support and the model. This will create a skin between the model and support.	bool	False	possibly	
support_interface_enable	support_bottom_enable	Generate a dense slab of material between the bottom of the support and the model. This will create a skin between the model and support.	bool	False	possibly	
support_interface_height	support_roof_height	The thickness of the support roofs. This controls the amount of dense layers at the top of the support on which the model rests.	mm	float	1	possibly
support_interface_height	support_bottom_height	The thickness of the support floors. This controls the number of dense layers that are printed on top of places of a model on which support rests.	mm	float	1	possibly
support_roof_density	support_roof_line_distance	Distance between the printed support roof lines. This setting is calculated by the Support Roof Density, but can be adjusted separately.	mm	float	25	possibly
support_bottom_density	support_bottom_line_distance	Distance between the printed support floor lines. This setting is calculated by the Support Floor Density, but can be adjusted separately.	mm	float	25	possibly
support_interface_pattern	support_roof_pattern	The pattern with which the roofs of the support are printed.	key	concentric	possibly	
support_interface_pattern	support_bottom_pattern	The pattern with which the floors of the support are printed.	key	concentric	possibly	
minimum_interface_area	minimum_roof_area	Minimum area size for the roofs of the support. Polygons which have an area smaller than this value will be printed as normal support.	mm²	float	1	possibly

minimum_interface_area	minimum_bottom_area	Minimum area size for the floors of the support. Polygons which have an area smaller than this value will be printed as normal support.	mm ²	float	1	possibly
support_interface_offset	support_roof_offset	Amount of offset applied to the roofs of the support.	mm	float	0	possibly
support_interface_offset	support_bottom_offset	Amount of offset applied to the floors of the support.	mm	float	0	possibly
support	support_interface_priority	How support interface and support will interact when they overlap. Currently only implemented for support roof.	key	interface_area_overwrite_support_area	possibly	
support_interface_angles	support_roof_angles	A list of integer line directions to use. Elements from the list are used sequentially as the layers progress and when the end of the list is reached, it starts at the beginning again. The list items are separated by commas and the whole list is contained in square brackets. Default is an empty list which means use the default angles (alternates between 45 and 135 degrees if interfaces are quite thick or 90 degrees).	list[int]	[]	possibly	
support_interface_angles	support_bottom_angles	A list of integer line directions to use. Elements from the list are used sequentially as the layers progress and when the end of the list is reached, it starts at the beginning again. The list items are separated by commas and the whole list is contained in square brackets. Default is an empty list which means use the default angles (alternates between 45 and 135 degrees if interfaces are quite thick or 90 degrees).	list[int]	[]	possibly	
support	support_fan_enable	When enabled, the print cooling fan speed is altered for the skin regions immediately above the support.	bool	False	no	
support	support_supported_skin_fan_speed	Percentage fan speed to use when printing the skin regions immediately above the support. Using a high fan speed can make the support easier to remove.	%	float	100	no
support	support_use_towers	Use specialized towers to support tiny overhang areas. These towers have a larger diameter than the region they support. Near the overhang the towers' diameter decreases, forming a roof.	bool	True	possibly	
support	support_tower_diameter	The diameter of a special tower.	mm	float	3	possibly
support	support_tower_maximum_supported_diameter	Maximum diameter in the X/Y directions of a small area which is to be supported by a specialized support tower.	mm	float	3	possibly
support	support_tower_roof_angle	The angle of a rooftop of a tower. A higher value results in pointed tower roofs, a lower value results in flattened tower roofs.	°	int	65	possibly
support	support_mesh_drop_down	Make support everywhere below the support mesh, so that there's no overhang in the support mesh.	bool	True	possibly	
support	support_meshes_present	There are support meshes present in the scene. This setting is controlled by Cura.	bool	False	possibly	
platform_adhesion	prime_blob_enable	Whether to prime the filament with a blob before printing. Turning this setting on will ensure that the extruder will have material ready at the nozzle before printing. Printing Brim or Skirt can act like priming too, in which case turning this setting off saves some time.	bool	False	possibly	

platform_adhesion	extruder_prime_pos_x	The X coordinate of the position where the nozzle primes at the start of printing.	mm	float	0	possibly
platform_adhesion	extruder_prime_pos_y	The Y coordinate of the position where the nozzle primes at the start of printing.	mm	float	0	possibly
platform_adhesion	adhesion_type	Different options that help to improve both priming your extrusion and adhesion to the build plate. Brim adds a single layer flat area around the base of your model to prevent warping. Raft adds a thick grid with a roof below the model. Skirt is a line printed around the model, but not connected to the model.	key	none	partially	none, skirt, brim tested raft not tested and adjusted
adhesion_extruder_nr	skirt_brim_extruder_nr	The extruder train to use for printing the skirt or brim. This is used in multi-extrusion.	int	0	no	
adhesion_extruder_nr	raft_base_extruder_nr	The extruder train to use for printing the first layer of the raft. This is used in multi-extrusion.	int	0	no	
adhesion_extruder_nr	raft_interface_extruder_nr	The extruder train to use for printing the middle layer(s) of the raft. This is used in multi-extrusion.	int	0	no	
adhesion_extruder_nr	raft_surface_extruder_nr	The extruder train to use for printing the top layer(s) of the raft. This is used in multi-extrusion.	int	0	no	
platform_adhesion	skirt_line_count	Multiple skirt lines help to prime your extrusion better for small models. Setting this to 0 will disable the skirt.	int	5	yes	
platform_adhesion	skirt_height	Printing the innermost skirt line with multiple layers makes it easy to remove the skirt.	int	1	yes	
platform_adhesion	skirt_gap	The horizontal distance between the skirt and the first layer of the print. This is the minimum distance. Multiple skirt lines will extend outwards from this distance.	mm	float	20	yes
platform_adhesion	skirt_brim_minimal_length	The minimum length of the skirt or brim. If this length is not reached by all skirt or brim lines together, more skirt or brim lines will be added until the minimum length is reached. Note: if the line count is set to 0 this is ignored.	mm	float	250	yes
brim_width	brim_line_count	The number of lines used for a brim. More brim lines enhance adhesion to the build plate, but also reduces the effective print area.	int	10	yes	
platform_adhesion	brim_gap	The horizontal distance between the first brim line and the outline of the first layer of the print. A small gap can make the brim easier to remove while still providing the thermal benefits. Enforce brim to be printed around the model even if that space would otherwise be occupied by support. This replaces some regions of the first layer of support by brim regions.	mm	float	5	yes
platform_adhesion	brim_location	Print a brim on the outside of the model, inside, or both. Depending on the model, this helps reducing the amount of brim you need to remove afterwards, while ensuring a proper bed adhesion.	key	outside	yes	
platform_adhesion	brim_inside_margin	A brim around a model may touch an other model where you don't want it. This removes all brim within this distance from brimless models.	bool	True	yes	
platform_adhesion	brim_smart_ordering	Swap print order of the innermost and second innermost brim lines. This improves brim removal.	bool	True	yes	

raft_margin	raft_base_margin	If the raft base is enabled, this is the extra raft area around the model which is also given a stronger raft while using more material and leaving less area for your print.	mm	float	15	possibly
raft_margin	raft_interface_margin	If the raft middle is enabled, this is the extra raft area around the model which is also given a stronger raft while using more material and leaving less area for your print.	mm	float	15	possibly
raft_margin	raft_surface_margin	If the raft top is enabled, this is the extra raft area around the model which is also given a stronger raft while using more material and leaving less area for your print.	mm	float	15	possibly
raft_remove_inside_corners	raft_base_remove_inside_corners	Remove inside corners from the raft base, causing the raft to become convex.	bool	False	possibly	
raft_remove_inside_corners	raft_interface_remove_inside_corners	Remove inside corners from the raft middle part, causing the raft to become convex.	bool	False	possibly	
raft_remove_inside_corners	raft_surface_remove_inside_corners	Remove inside corners from the raft top part, causing the raft to become convex.	bool	False	possibly	
raft_smoothing	raft_base_smoothing	This setting controls how much inner corners in the raftbase outline are rounded. Inward corners are rounded to a semi circle with a radius equal to the value given here. This setting also removes holes in the raft outline which are smaller than such a circle.	mm	float	5	possibly
raft_smoothing	raft_interface_smoothing	This setting controls how much inner corners in the raft middle outline are rounded. Inward corners are rounded to a semi circle with a radius equal to the value given here. This setting also removes holes in the raft outline which are smaller than such a circle.	mm	float	5	possibly
raft_smoothing	raft_surface_smoothing	This setting controls how much inner corners in the raft top outline are rounded. Inward corners are rounded to a semi circle with a radius equal to the value given here. This setting also removes holes in the raft outline which are smaller than such a circle.	mm	float	5	possibly
platform_adhesion	raft_airgap	The gap between the final raft layer and the first layer of the model. Only the first layer is raised by this amount to lower the bonding between the raft layer and the model. Makes it easier to peel off the raft.	mm	float	0	possibly
platform_adhesion	layer_0_z_overlap	Make the first and second layer of the model overlap in the Z direction to compensate for the filament lost in the airgap. All models above the first model layer will be shifted down by this amount. It may be noted that sometimes the second layer is printed below initial layer because of this setting. This is intended behavior.	mm	float	0	fixed due to layer height calculation layer shift in z direction is not possible
platform_adhesion	raft_base_thickness	Layer thickness of the base raft layer. This should be a thick layer which sticks firmly to the printer build plate.	mm	float	18	possibly

platform_adhesion	raft_base_line_width	Width of the lines in the base raft layer. These should be thick lines to assist in build plate adhesion.	mm	float	25	possibly
platform_adhesion	raft_base_line_spacing	The distance between the raft lines for the base raft layer. Wide spacing makes for easy removal of the raft from the build plate.	mm	float	50	possibly
raft_base_infill_overlap_mm	raft_base_infill_overlap_mm	The amount of overlap between the infill and the walls of the raft base. A slight overlap allows the walls to connect firmly to the infill.	mm	float	0	possibly
platform_adhesion	raft_interface_layers	The number of layers between the base and the surface of the raft. These comprise the main thickness of the raft. Increasing this creates a thicker, sturdier raft.	int	1	possibly	
platform_adhesion	raft_interface_thickness	Layer thickness of the middle raft layer.	mm	float	15	possibly
platform_adhesion	raft_interface_line_width	Width of the lines in the middle raft layer. Making the second layer extrude more causes the lines to stick to the build plate.	mm	float	30	possibly
platform_adhesion	raft_interface_line_spacing	The distance between the raft lines for the middle raft layer. The spacing of the middle should be quite wide, while being dense enough to support the top raft layers.	mm	float	50	possibly
platform_adhesion	raft_interface_z_offset	When printing the first layer of the raft interface, translate by this offset to customize the adhesion between base and interface. A negative offset should improve the adhesion.	mm	float	0	possibly
raft_interface_infill_overlap_mm	raft_interface_infill_overlap_mm	The amount of overlap between the infill and the walls of the raft interface. A slight overlap allows the walls to connect firmly to the infill.	mm	float	0	possibly
platform_adhesion	raft_surface_layers	The number of top layers on top of the 2nd raft layer. These are fully filled layers that the model sits on. 2 layers result in a smoother top surface than 1.	int	2	possibly	
platform_adhesion	raft_surface_thickness	Layer thickness of the top raft layers.	mm	float	15	possibly
platform_adhesion	raft_surface_line_width	Width of the lines in the top surface of the raft. These can be thin lines so that the top of the raft becomes smooth.	mm	float	25	possibly
platform_adhesion	raft_surface_line_spacing	The distance between the raft lines for the top raft layers. The spacing should be equal to the line width, so that the surface is solid.	mm	float	25	possibly
platform_adhesion	raft_surface_z_offset	When printing the first layer of the raft surface, translate by this offset to customize the adhesion between interface and surface. A negative offset should improve the adhesion.	mm	float	0	possibly
platform_adhesion	raft_surface_monotonic	Print raft top surface lines in an ordering that causes them to always overlap with adjacent lines in a single direction. This takes slightly more time to print, but makes the surface look more consistent, which is also visible on the model bottom surface.	bool	False	possibly	
raft_surface_infill_overlap_mm	raft_surface_infill_overlap_mm	The amount of overlap between the infill and the walls of the raft surface. A slight overlap allows the walls to connect firmly to the infill.	mm	float	0	possibly
raft_wall_count	raft_base_wall_count	The number of contours to print around the linear pattern in the base layer of the raft.	int	1	possibly	
raft_wall_count	raft_interface_wall_count	The number of contours to print around the linear pattern in the middle layers of the raft.	int	0	possibly	

raft_wall_count	raft_surface_wall_count	The number of contours to print around the linear pattern in the top layers of the raft..	int	0	possibly
raft_speed	raft_base_speed	The speed at which the base raft layer is printed. This should be printed quite slowly, as the volume of material coming out of the nozzle is quite high.	mm/s	float	15 no
raft_speed	raft_interface_speed	The speed at which the middle raft layer is printed. This should be printed quite slowly, as the volume of material coming out of the nozzle is quite high.	mm/s	float	15 no
raft_acceleration	raft_base_acceleration	The speed at which the top raft layers are printed. These should be printed a bit lower, so that the nozzle can slowly smooth out adjacent surface lines.	mm/s ²	float	20 no
raft_acceleration	raft_interface_acceleration	The acceleration with which the base raft layer is printed.	mm/s ²	float	3000 no
raft_acceleration	raft_surface_acceleration	The acceleration with which the middle raft layer is printed.	mm/s ²	float	3000 no
raft_jerk	raft_base_jerk	The acceleration with which the top raft layers are printed.	mm/s ³	float	3000 no
raft_jerk	raft_interface_jerk	The jerk with which the middle raft layer is printed.	mm/s ³	float	20 no
raft_jerk	raft_surface_jerk	The jerk with which the top raft layers are printed.	mm/s ³	float	20 no
raft_fan_speed	raft_base_fan_speed	The fan speed for the base raft layer.	%	float	0 no
raft_fan_speed	raft_interface_fan_speed	The fan speed for the middle raft layer.	%	float	0 no
raft_fan_speed	raft_surface_fan_speed	The fan speed for the top raft layers.	%	float	0 no
raft_flow	raft_base_flow	The amount of material, relative to normal extrusion line, to extrude during raft base printing. Having an increased flow may improve adhesion and raft structural strength.	%	float	100 yes
raft_flow	raft_interface_flow	The amount of material, relative to normal extrusion line, to extrude during raft interface printing. Having an increased flow may improve adhesion and raft structural strength.	%	float	100 no
raft_flow	raft_surface_flow	The amount of material after each nozzle switch. <html>Normal create a bucket in towerInterleaved create a prime tower as sparse as possible. This will save time and filament, but is only possible if the used materials adhere to each other</html>	bool	False	no
dual	prime_tower_mode	Print a tower next to the print which serves to prime the material after each nozzle switch.	key	normal	no
dual	prime_tower_size	Print a tower next to the print which serves to prime the material after each nozzle switch.	mm	float	20 no
dual	prime_tower_min_volume	The width of the prime tower.	mm ³	float	6 no
dual	prime_tower_max_bridging_distance	The minimum volume for each layer of the prime tower in order to purge enough material. The maximum length of the branches which may be printed over the air.	mm	float	5 no

dual	<code>prime_tower_min_shell_thickness</code>	The minimum thickness of the prime tower shell. You may increase it to make the prime tower stronger.	mm	float	25	no
dual	<code>prime_tower_position_x</code>	The x coordinate of the position of the prime tower.	mm	float	200	no
dual	<code>prime_tower_position_y</code>	The y coordinate of the position of the prime tower.	mm	float	200	no
dual	<code>prime_tower_wipe_enabled</code>	After printing the prime tower with one nozzle, wipe the ooze material from the other nozzle off on the prime tower.	bool	True	no	
dual	<code>prime_tower_brim_enable</code>	By enabling this setting, your prime-tower will get a brim, even if the model doesn't. If you want a sturdier base for a high tower, you can increase the base height.	bool	False	no	
dual	<code>prime_tower_brim_size</code>	The width of the prime tower brim/base. A larger base enhances adhesion to the build plate, but also reduces the effective print area.	mm	float	50	no
dual	<code>prime_tower_base_height</code>	The height of the prime tower base. Increasing this value will result in a more sturdy prime tower because the base will be wider. If this setting is too low, the prime tower will not have a sturdy base.	mm	float	0	no
dual	<code>prime_tower_base_curve_magnitude</code>	The magnitude factor used for the slope of the prime tower base. If you increase this value, the base will become slimmer. If you decrease it, the base will become thicker.	float	4	no	
dual	<code>prime_tower_raft_base_line_spacing</code>	The distance between the raft lines for the unique prime tower raft layer. Wide spacing makes for easy removal of the raft from the build plate.	mm	float	50	no
dual	<code>ooze_shield_enabled</code>	Enable exterior ooze shield. This will create a shell around the model which is likely to wipe a second nozzle if it's at the same height as the first nozzle.	bool	False	possibly	
dual	<code>ooze_shield_angle</code>	The maximum angle a part in the ooze shield will have. With 0 degrees being vertical, and 90 degrees being horizontal. A smaller angle leads to less failed ooze shields, but more material.	°	float	60	possibly
dual	<code>ooze_shield_dist</code>	Distance of the ooze shield from the print in the X/Y directions.	mm	float	2	possibly
dual	<code>switch_extruder_retraction_amount</code>	The amount of retraction when switching extruders. Set to 0 for no retraction at all. This should generally be the same as the length of the heat zone.	mm	float	20	no
<code>switch_extruder_retraction_speeds</code>	<code>switch_extruder_retraction_speed</code>	The speed at which the filament is retracted during a nozzle switch retract.	mm/s	float	20	no
<code>switch_extruder_retraction_speeds</code>	<code>switch_extruder_prime_speed</code>	The speed at which the filament is pushed back after a nozzle switch retraction.	mm/s	float	20	no
dual	<code>switch_extruder_extra_prime_amount</code>	Extra material to prime after nozzle switching. Ignore the internal geometry arising from overlapping volumes within a mesh and print the volumes as one. This may cause unintended internal cavities to disappear.	mm ³	float	0	yes
meshfix	<code>meshfix_union_all_remove_holes</code>	Remove the holes in each layer and keep only the outside shape. This will ignore any invisible internal geometry. However, it also ignores layer holes which can be viewed from above or below.	bool	True	yes	
meshfix			bool	False	yes	

meshfix	<code>meshfix_extensive_stitching</code>	Extensive stitching tries to stitch up open holes in the mesh by closing the hole with touching polygons. This option can introduce a lot of processing time.	bool	False	yes
meshfix	<code>meshfix_keep_open_polygons</code>	Normally Cura tries to stitch up small holes in the mesh and remove parts of a layer with big holes. Enabling this option keeps those parts which cannot be stitched. This option should be used as a last resort option when everything else fails to produce proper g-code.	bool	False	yes
meshfix	<code>multiple_mesh_overlap</code>	Make meshes which are touching each other overlap a bit. This makes them bond together better. Remove areas where multiple meshes are overlapping with each other. This may be used if merged dual material objects overlap with each other.	mm	0.5	yes
meshfix	<code>carve_multiple_volumes</code>	Switch to which mesh intersecting volumes will belong with every layer, so that the overlapping meshes become interwoven. Turning this setting off will cause one of the meshes to obtain all of the volume in the overlap, while it is removed from the other meshes.	bool	True	yes
meshfix	<code>alternate_carve_order</code>	Remove empty layers beneath the first printed layer if they are present. Disabling this setting can cause empty first layers if the Slicing Tolerance setting is set to Exclusive or Middle.	bool	True	yes
meshfix	<code>remove_empty_first_layers</code>	The minimum size of a line segment after slicing. If you increase this, the mesh will have a lower resolution. This may allow the printer to keep up with the speed it has to process g-code and will increase slice speed by removing details of the mesh that it can't process anyway.	mm	0.5	yes
meshfix	<code>meshfix_maximum_resolution</code>	The minimum size of a travel line segment after slicing. If you increase this, the travel moves will have less smooth corners. This may allow the printer to keep up with the speed it has to process g-code, but it may cause model avoidance to become less accurate.	mm	1	yes
meshfix	<code>meshfix_maximum_travel_resolution</code>	The maximum deviation allowed when reducing the resolution for the Maximum Resolution setting. If you increase this, the print will be less accurate, but the g-code will be smaller. Maximum Deviation is a limit for Maximum Resolution, so if the two conflict the Maximum Deviation will always be held true.	mm	0.025	yes
meshfix	<code>meshfix_maximum_extrusion_area_deviation</code>	The maximum extrusion area deviation allowed when removing intermediate points from a straight line. An intermediate point may serve as width-changing point in a long straight line. Therefore, if it is removed, it will cause the line to have a uniform width and, as a result, lose (or gain) a bit of extrusion area. If you increase this you may notice slight under- (or over-) extrusion in between straight parallel walls, as more intermediate width-changing points will be allowed to be removed. Your print will be less accurate, but the g-code will be smaller.	µm ²	50000	yes

meshfix	<code>meshfix_fluid_motion_enabled</code>	When enabled tool paths are corrected for printers with smooth motion planners. Small movements that deviate from the general tool path direction are smoothed to improve fluid motions.	bool	True	yes
meshfix	<code>meshfix_fluid_motion_shift_distance</code>	Distance points are shifted to smooth the path	mm	float	0.1
meshfix	<code>meshfix_fluid_motion_small_distance</code>	Distance points are shifted to smooth the path	mm	float	0.01
meshfix	<code>meshfix_fluid_motion_angle</code>	If a toolpath-segment deviates more than this angle from the general motion it is smoothed	°	float	15
blackmagic	<code>print_sequence</code>	Whether to print all models one layer at a time or to wait for one model to finish, before moving on to the next. One at a time mode is possible if a) only one extruder is enabled and b) all models are separated in such way that the whole print head can move in between and all models are lower than the distance between the nozzle and the XY axes.	key	<code>all_at_once</code>	yes
blackmagic	<code>user_defined_print_order_enabled</code>	Allows you to order the object list to manually set the print sequence. First object from the list will be printed first.	bool	False	yes
blackmagic	<code>infill_mesh</code>	Use this mesh to modify the infill of other meshes with which it overlaps. Replaces infill regions of other meshes with regions for this mesh. It's suggested to only print one Wall and no Top/Bottom Skin for this mesh.	bool	False	yes
blackmagic	<code>infill_mesh_order</code>	Determines the priority of this mesh when considering multiple overlapping infill meshes. Areas where multiple infill meshes overlap will take on the settings of the mesh with the highest rank. An infill mesh with a higher rank will modify the infill of infill meshes with lower rank and normal meshes.	int	0	yes
blackmagic	<code>cutting_mesh</code>	Limit the volume of this mesh to within other meshes. You can use this to make certain areas of one mesh print with different settings and with a whole different extruder.	bool	False	yes
blackmagic	<code>mold_enabled</code>	Print models as a mold, which can be cast in order to get a model which resembles the models on the build plate.	bool	False	possibly
blackmagic	<code>mold_width</code>	The minimal distance between the outside of the mold and the outside of the model.	mm	float	5
blackmagic	<code>mold_top_height</code>	The height above horizontal parts in your model which to print mold.	mm	float	0.5
blackmagic	<code>mold_angle</code>	The angle of overhang of the outer walls created for the mold. 0° will make the outer shell of the mold vertical, while 90° will make the outside of the model follow the contour of the model.	°	float	40
blackmagic	<code>support_mesh</code>	Use this mesh to specify support areas. This can be used to generate support structure.	bool	False	possibly
blackmagic	<code>anti_overhang_mesh</code>	Use this mesh to specify where no part of the model should be detected as overhang. This can be used to remove unwanted support structure.	bool	False	possibly

blackmagic	<code>magic_mesh_surface_mode</code>	Treat the model as a surface only, a volume, or volumes with loose surfaces. The normal print mode only prints enclosed volumes. "Surface" prints a single wall tracing the mesh surface with no infill and no top/bottom skin. "Both" prints enclosed volumes like normal and any remaining polygons as surfaces.	key	normal	yes
blackmagic	<code>magic_spiralize</code>	Spiralize smooths out the Z move of the outer edge. This will create a steady Z increase over the whole print. This feature turns a solid model into a single walled print with a solid bottom. This feature should only be enabled when each layer only contains a single part.	bool	False	yes
blackmagic	<code>smooth_spiralized_contours</code>	Smooths the spiralized contours to reduce the visibility of the Z seam (the Z seam should be barely visible on the print but will still be visible in the layer view). Note that smoothing will tend to blur fine surface details.	bool	True	yes
blackmagic	<code>relative_extrusion</code>	Use relative extrusion rather than absolute extrusion. Using relative E-steps makes for easier post-processing of the g-code. However, it's not supported by all printers and it may produce very slight deviations in the amount of deposited material compared to absolute E-steps. Irrespective of this setting, the extrusion mode will always be set to absolute before any g-code script is output.	bool	False	yes
experimental	<code>slicing_tolerance</code>	Vertical tolerance in the sliced layers. The contours of a layer are normally generated by taking cross sections through the middle of each layer's thickness (Middle). Alternatively each layer can have the areas which fall inside of the volume throughout the entire thickness of the layer (Exclusive) or a layer has the areas which fall inside anywhere within the layer (Inclusive). Inclusive retains the most details, Exclusive makes for the best fit and Middle stays closest to the original surface.	key	middle	yes
experimental	<code>infill_enable_travel_optimization</code>	When enabled, the order in which the infill lines are printed is optimized to reduce the distance travelled. The reduction in travel time achieved very much depends on the model being sliced, infill pattern, density, etc. Note that, for some models that have many small areas of infill, the time to slice the model may be greatly increased.	bool	False	yes
experimental	<code>material_flow_temp_graph</code>	Data linking material flow (in mm ³ per second) to temperature (Degrees Celsius).	[[mm ³ , °C]]	str	[[3.5, 200], [7.0, 240]] no
experimental	<code>minimum_polygon_circumference</code>	Polygons in sliced layers that have a circumference smaller than this amount will be filtered out. Lower values lead to higher resolution mesh at the cost of slicing time. It is meant mostly for high resolution SLA printers and very tiny 3D models with a lot of details.	mm	float	1 yes

experimental	interlocking_enable	At the locations where models touch, generate an interlocking beam structure. This improves the adhesion between models, especially models printed in different materials.	bool	False	possibly
experimental	interlocking_beam_width	The width of the interlocking structure beams.	float	0.8	possibly
experimental	interlocking_orientation	The height of the beams of the interlocking structure, measured in number of layers. Less layers is stronger, but more prone to defects.	°	float	22.5
experimental	interlocking_beam_layer_count	The height of the beams of the interlocking structure, measured in number of layers. Less layers is stronger, but more prone to defects.	mm	float	possibly
experimental	interlocking_depth	The distance from the boundary between models to generate interlocking structure, measured in cells. Too few cells will result in poor adhesion.	int	2	possibly
experimental	interlocking_boundary_avoidance	The distance from the outside of a model where interlocking structures will not be generated, measured in cells.	int	2	possibly
experimental	support_skip_some_zags	Skip some support line connections to make the support structure easier to break away. This setting is applicable to the Zig Zag support infill pattern.	bool	False	yes
experimental	support_zag_skip_count	Skip one in every N connection lines to make the support structure easier to break away.	int	5	yes
experimental	draft_shield_enabled	This will create a wall around the model, which traps (hot) air and shields against exterior airflow. Especially useful for materials which warp easily.	bool	False	yes
experimental	draft_shield_delt	Distance of the draft shield from the print, in the X/Y directions.	mm	float	10
experimental	draft_shield_height_limitation	Set the height of the draft shield. Choose to print the draft shield at the full height of the model or at a limited height.	key	full	yes
experimental	draft_shield_height	Height limitation of the draft shield. Above this height no draft shield will be printed.	mm	float	10
experimental	conical_overhang_enabled	Change the geometry of the printed model such that minimal support is required. Steep overhangs will become shallow overhangs. Overhanging areas will drop down to become more vertical.	bool	False	yes
experimental	conical_overhang_angle	The maximum angle of overhangs after they have been made printable. At a value of 0° all overhangs are replaced by a piece of model connected to the build plate, 90° will not change the model in any way.	°	float	50
experimental	conical_overhang_hole_size	The maximum area of a hole in the base of the model before it's removed by Make Overhang Printable. Holes smaller than this will be retained. A value of 0 mm ² will fill all holes in the models base.	mm ²	float	0
experimental	coasting_enable	Coasting replaces the last part of an extrusion path with a travel path. The oozed material is used to print the last piece of the extrusion path in order to reduce stringing.	bool	False	possibly
experimental	coasting_volume	The volume otherwise oozed. This value should generally be close to the nozzle diameter cubed.	mm ³	float	15625

experimental	coasting_min_volume	The smallest volume an extrusion path should have before allowing coasting. For smaller extrusion paths, less pressure has been built up in the bowden tube and so the coasted volume is scaled linearly. This value should always be larger than the Coasting Volume.	mm³	float	625	possibly
experimental	coasting_speed	The speed by which to move during coasting, relative to the speed of the extrusion path. A value slightly under 100% is advised, since during the coasting move the pressure in the bowden tube drops.	%	float	90	no
experimental	cross_infill_pocket_size	The size of pockets at four-way crossings in the cross 3D pattern at heights where the pattern is touching itself.	mm	float	2	yes
experimental	cross_infill_density_image	The file location of an image of which the brightness values determine the minimal density at the corresponding location in the infill of the print.	str			possibly
experimental	cross_support_density_image	The file location of an image of which the brightness values determine the minimal density at the corresponding location in the support.	str			possibly
experimental	support_conical_enabled	Make support areas smaller at the bottom than at the overhang.	bool	False	yes	
experimental	support_conical_angle	The angle of the tilt of conical support. With 0 degrees being vertical, and 90 degrees being horizontal. Smaller angles cause the support to be more sturdy, but consist of more material. Negative angles cause the base of the support to be wider than the top.	°	float	30	yes
experimental	support_conical_min_width	Minimum width to which the base of the conical support area is reduced. Small widths can lead to unstable support structures.	mm	float	5	yes
experimental	magic_fuzzy_skin_enabled	Randomly jitter while printing the outer wall, so that the surface has a rough and fuzzy look.	bool	False		possibly
experimental	magic_fuzzy_skin_outside_only	Jitter only the parts' outlines and not the parts' holes.	bool	False		possibly
experimental	magic_fuzzy_skin_thickness	The width within which to jitter. It's advised to keep this below the outer wall width, since the inner walls are unaltered.	mm	float	5	possibly
magic_fuzzy_skin_point_density	magic_fuzzy_skin_point_dist	The average distance between the random points introduced on each line segment. Note that the original points of the polygon are discarded, so a high smoothness in a reduction of the resolution. This value must be higher than half the Fuzzy Skin Thickness.	mm	float	10	possibly
experimental	flow_rate_max_extrusion_offset	The maximum distance in mm to move the filament to compensate for changes in flow rate.	mm	float	0	no
experimental	flow_rate_extrusion_offset_factor	How far to move the filament in order to compensate for changes in flow rate, as a percentage of how far the filament would move in one second of extrusion.	%	float	100	no
experimental	adaptive_layer_height_enabled	Adaptive layers computes the layer heights depending on the shape of the model.	bool	False	possibly	adaptive layer height not tested
experimental	adaptive_layer_height_variation	The maximum allowed height different from the base layer height.	mm	float	10	possibly

experimental	adaptive_layer_height_variation_step	The difference in height of the next layer height compared to the previous one.	mm	float	1	possibly
experimental	adaptive_layer_height_threshold	Target horizontal distance between two adjacent layers. Reducing this setting causes thinner layers to be used to bring the edges of the layers closer together.	mm	float	12.5	possibly
experimental	wall_overhang_angle	Walls that overhang more than this angle will be printed using overhanging wall settings. When the value is 90, no walls will be treated as overhanging. Overhang that gets supported by support will not be treated as overhang either.	°	float	90	yes
experimental	seam_overhang_angle	Try to prevent seams on walls that overhang more than this angle. When the value is 90, no walls will be treated as overhanging.	°	float	90	yes
experimental	wall_overhang_speed_factor	Overhanging walls will be printed at this percentage of their normal print speed.	%	float	100	yes
experimental	bridge_settings_enabled	Detect bridges and modify print speed, flow and fan settings while bridges are printed.	bool	False	yes	
experimental	bridge_wall_min_length	Unsupported walls shorter than this will be printed using the normal wall settings. Longer unsupported walls will be printed using the bridge wall settings.	mm	float	5	yes
experimental	bridge_skin_support_threshold	If a skin region is supported for less than this percentage of its area, print it using the bridge settings. Otherwise it is printed using the normal skin settings.	%	float	50	yes
experimental	bridge_sparse_infill_max_density	Maximum density of infill considered to be sparse. Skin over sparse infill is considered to be unsupported and so may be treated as a bridge skin.	%	float	0	yes
experimental	bridge_wall_coast	This controls the distance the extruder should coast immediately before a bridge wall begins. Coasting before the bridge starts can reduce the pressure in the nozzle and may produce a flatter bridge.	%	float	100	yes
experimental	bridge_wall_speed	The speed at which the bridge walls are printed.	mm/s	float	15	no
experimental	bridge_wall_material_flow	When printing bridge walls, the amount of material extruded is multiplied by this value.	%	float	50	yes
experimental	bridge_skin_speed	The speed at which bridge skin regions are printed.	mm/s	float	15	no
experimental	bridge_skin_material_flow	When printing bridge skin regions, the amount of material extruded is multiplied by this value.	%	float	60	yes
experimental	bridge_skin_density	The density of the bridge skin layer. Values less than 100 will increase the gaps between the skin lines.	%	float	100	yes
experimental	bridge_enable_more_layers	If enabled, the second and third layers above the air are printed using the following settings. Otherwise, those layers are printed using the normal settings.	bool	True	no	
experimental	bridge_skin_speed_2	Percentage fan speed to use when printing bridge walls and skin.	%	float	100	no
experimental	bridge_skin_material_flow_2	Print speed to use when printing the second bridge skin layer.	mm/s	float	25	no
experimental		When printing the second bridge skin layer, the amount of material extruded is multiplied by this value.	%	float	100	no

experimental	bridge_skin_density_2	The density of the second bridge skin layer. Values less than 100 will increase the gaps between the skin lines.	%	float	75	no
experimental	bridge_fan_speed_2	Percentage fan speed to use when printing the second bridge skin layer.	%	float	0	no
experimental	bridge_skin_speed_3	Print speed to use when printing the third bridge skin layer.	mm/s	float	15	no
experimental	bridge_skin_material_flow_3	When printing the third bridge skin layer, the amount of material extruded is multiplied by this value.	mm/s	float	110	no
experimental	bridge_skin_density_3	The density of the third bridge skin layer. Values less than 100 will increase the gaps between the skin lines.	%	float	80	no
experimental	bridge_fan_speed_3	Percentage fan speed to use when printing the third bridge skin layer.	%	float	0	no
experimental	clean_between_layers	Whether to include nozzle wipe G-Code between layers (maximum 1 per layer). Enabling this setting could influence behavior of retract at layer change. Please use Wipe Retraction settings to control retraction at layers where the wipe script will be working.	bool	False	no	no
experimental	max_extrusion_before_wipe	Maximum material that can be extruded before another nozzle wipe is initiated. If this value is less than the volume of material required in a layer, the setting has no effect in this layer, i.e. it is limited to one wipe per layer.	mm³	float	10	no
experimental	wipe_retraction_enable	Retract the filament when the nozzle is moving over a non-printed area.	bool	True	yes	yes
experimental	wipe_retraction_amount	Amount to retract the filament so it does not ooze during the wipe sequence.	mm	float	1	no
experimental	wipe_retraction_extra_prime_amount	Some material can ooze away during a wipe travel moves, which can be compensated for here.	mm³	float	0	no
experimental	wipe_retraction_retract_speed	The speed at which the filament is retracted during a wipe retraction move.	mm/s	float	3	no
experimental	wipe_retraction_prime_speed	The speed at which the filament is primed during a wipe retraction move.	mm/s	float	2	no
experimental	wipe_pause	Pause after the unretract.	s	float	0	no
experimental	wipe_hop_enable	When wiping, the build plate is lowered to create clearance between the nozzle and the print. It prevents the nozzle from hitting the print during travel moves, reducing the chance to knock the print from the build plate.	bool	True	possibly	could interfere with layer height calculation
experimental	wipe_hop_amount	The height difference when performing a Z Hop.	mm	float	1	possibly
experimental	wipe_hop_speed	Speed to move the z-axis during the hop.	mm/s	float	10	no
experimental	wipe_brush_pos_x	X location where wipe script will start.	mm	float	100	no
experimental	wipe_repeat_count	Number of times to move the nozzle across the brush.	int	5	no	no
experimental	wipe_move_distance	The distance to move the head back and forth across the brush.	mm	float	20	no
small_hole_max_size	small_feature_max_length	Feature outlines that are shorter than this length will be printed using Small Feature Speed.	mm	float	0	yes
experimental	small_feature_speed_factor	Small features will be printed at this percentage of their normal print speed. Slower printing can help with adhesion and accuracy.	%	float	50	no

experimental	small_feature_speed_factor_0	Small features on the first layer will be printed at this percentage of their normal print speed. Slower printing can help with adhesion and accuracy.	%	float	50	no
experimental	material_alternate_walls	Alternate wall directions every other layer and inset. Useful for materials that can build up stress, like for metal printing.	bool	False	yes	
experimental	group_outer_walls	Outer walls of different islands in the same layer are printed in sequence. When enabled the amount of flow changes is limited because walls are printed one type at a time, when disabled the number of travels between islands is reduced because walls in the same islands are grouped.	bool	True	yes	
ppr	ppr_enable	Enable print process reporting for setting threshold values for possible fault detection.	bool	False	no	
ppr	flow_warn_limit	Limit on the flow warning for detection.	%	float	15	no
ppr	flow_anomaly_limit	Limit on flow anomaly for detection.	%	float	25	no
ppr	print_temp_warn_limit	Limit on Print temperature warning for detection.	°C	float	3	no
ppr	print_temp_anomaly_limit	Limit on Print Temperature anomaly for detection.	°C	float	7	no
ppr	bv_temp_warn_limit	Limit on Build Volume Temperature warning for detection.	°C	float	7.5	no
ppr	bv_temp_anomaly_limit	Limit on Build Volume Temperature Anomaly for detection.	°C	float	10	no
command_line_settings	center_object	Whether to center the object on the middle of the build platform (0,0). Instead of using the coordinate system in which the object was saved.	bool	True	yes	
command_line_settings	mesh_position_x	Offset applied to the object in the x direction.	float	0	yes	
command_line_settings	mesh_position_y	Offset applied to the object in the y direction.	float	0	yes	
command_line_settings	mesh_position_z	Offset applied to the object in the z direction. With this you can perform what was used to be called 'Object Sink'.	float	0	yes	
command_line_settings	mesh_rotation_matrix	Transformation matrix to be applied to the model when loading it from file.	str	[[1,0,0],[0,1,0],[0,0,1]]	yes	setup.json ["cura_scaling"]

D List of Usable and Validated Commands from *extruder.def.json* With Comments

Parent Key	Key	Description	Unit	Type	Value	Usable	Comment
machine_settings	extruder_nr	The extruder train used for printing. This is used in multi-extrusion.			0	no	
machine_settings	extruder_prime_pos_z	The Z coordinate of the position where the nozzle primes at the start of printing.	mm	float	0	possibly	
machine_settings	machine_extruder_cooling_fan_number	The number of the print cooling fan associated with this extruder. Only change this from the default value of 0 when you have a different print cooling fan for each extruder.	int	int	0	no	
machine_settings	machine_extruder_end_code	End g-code to execute when switching away from this extruder.		str		no	
machine_settings	machine_extruder_end_code_duration	The time it takes to execute the end g-code, when switching away from this extruder.		float	0	no	
machine_settings	machine_extruder_end_pos_abs	Make the extruder ending position absolute rather than relative to the last-known location of the head.		bool	False	no	
machine_settings	machine_extruder_end_pos_x	The x-coordinate of the ending position when turning the extruder off.	mm	float	0	no	
machine_settings	machine_extruder_end_pos_y	The y-coordinate of the ending position when turning the extruder off.	mm	float	0	no	
machine_settings	machine_extruder_start_code	Start g-code to execute when switching to this extruder.		str		no	
machine_settings	machine_extruder_start_code_duration	The time it'll take to execute the start g-code, when switching to this extruder.		float	0	no	
machine_settings	machine_extruder_start_pos_abs	Make the extruder starting position absolute rather than relative to the last-known location of the head.		bool	False	no	
machine_settings	machine_extruder_start_pos_x	The x-coordinate of the starting position when turning the extruder on.	mm	float	0	no	
machine_settings	machine_extruder_start_pos_y	The y-coordinate of the starting position when turning the extruder on.	mm	float	0	no	
machine_settings	machine_nozzle_id	The nozzle ID for an extruder train, such as "AA 0.4" and "BB 0.8".		str	unknown	no	
machine_settings	machine_nozzle_offset_x	The x-coordinate of the offset of the nozzle.	mm	float	0	possibly	
machine_settings	machine_nozzle_offset_y	The y-coordinate of the offset of the nozzle.	mm	float	0	possibly	
machine_settings	machine_nozzle_size	The inner diameter of the nozzle. Change this setting when using a non-standard nozzle size.	mm	float	25	yes	

material	material_diameter	Adjusts the diameter of the filament used. Match this value with the diameter of the used filament.	mm	float	25	fixed
platform_adhesion	extruder_prime_pos_x	The X coordinate of the position where the nozzle primes at the start of printing.	mm	float	0	possibly
platform_adhesion	extruder_prime_pos_y	The Y coordinate of the position where the nozzle primes at the start of printing.	mm	float	0	possibly

Bibliography

- [1] 3D Printhuset A/S. *D-Shape: Besøgsrapport*. Visit Report. Visit date: April 6, 2016. Visited company: Monolite UK (Dinitech SpA). Copenhagen, Denmark: 3D Printhuset A/S, 2016. URL: <http://3dprintetbyggeri.dk/pdf/bes%C3%B8gsrapporter/D-Shape.pdf>.
- [2] 3D Systems, Inc. *3D Systems – Additive Manufacturing Solutions*. Accessed: 8 April 2025. n.d. URL: <https://www.3dsystems.com/>.
- [3] ABB Robotics. *ABB Robotics and Porsche Consulting collaborate to automate the construction industry*. Accessed: 20 Mai 2025. 2023. URL: <https://new.abb.com/news/de/detail/107956/prsrl-abb-robotics-and-porsche-consulting-collaborate-to-automate-the-construction-industry>.
- [4] Acrome. *The Basics of the Stewart Platform*. Accessed: 9 April 2025. 2024. URL: <https://acrome.net/post/the-basics-of-the-stewart-platform>.
- [5] F. A. Adnan et al. “Real-time slicing algorithm for Stereolithography (STL) CAD model applied in additive manufacturing industry”. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 342. IOP Publishing, 2018. DOI: 10.1088/1757-899X/342/1/012016. URL: <https://doi.org/10.1088/1757-899X/342/1/012016>.
- [6] Hyejin Ahn. *Computer-aided design — Week 6*. Accessed: 13 April 2025. 2020. URL: <https://fabacademy.org/2020/labs/waag/students/hyejin-ahn/week6.html>.
- [7] Alessandro Ranellucci. *Slic3r: Open Source 3D Printing Toolbox*. Accessed: 19 April 2025. 2025. URL: <https://slic3r.org/>.
- [8] Alberto Alvares, Efrain Rodriguez, and Cristhian Riaño Jaimes. “Development of the Linear Delta Robot for Additive Manufacturing”. In: *2018 International Conference on Control, Decision and Information Technologies (CoDIT)*. Accessed: 3 April 2025. 2018, pp. 187–192. DOI: 10.1109/CoDIT.2018.8394869. URL: <https://doi.org/10.1109/CoDIT.2018.8394869>.
- [9] Robert McNeel Associates. *Rhino.Inside*. Version 0.6.0, Available at GitHub: <https://github.com/mcneel/rhino.inside>. 2021. URL: <https://pypi.org/project/rhinoinside/>.
- [10] Autodesk. *The History of 3D Printing: From the 80s to Today*. Accessed: 3 April 2025. 2023. URL: <https://www.autodesk.com/design-make/articles/history-of-3d-printing>.
- [11] Autodesk Inc. *Fusion 360 – Integrated CAD, CAM, and CAE Software*. Accessed: 8 April 2025. Autodesk Inc. 2024. URL: <https://www.autodesk.com/products/fusion-360/overview>.
- [12] Bayerische Ingenieurekammer-Bau. *3D-Druck: Recycelbare Brücke ohne Armierungsstahl und Mörtel*. Accessed: 20 Mai 2025. 2021. URL: https://www.bayika.de/de/aktuelles/meldungen/2021-07-30_3D-Druck-Recycelbare-Bruecke-ohne-Armierungsstahl-und-Moertel.php.
- [13] Bayerische Ingenieurekammer-Bau. *Deutschland braucht 400 Milliarden Euro für Infrastruktur*. Accessed: 20 Mai 2025. 2024. URL: https://www.bayika.de/de/aktuelles/meldungen/2024-11-08_Deutschland-braucht-400-Milliarden-Euro-fuer-Infrastruktur.php.
- [14] BCN3D Technologies. *What is Infill?* Accessed: 13 April 2025. 2023. URL: <https://support.bcn3d.com/knowledge/infill>.
- [15] Joshua Bird. *S4 Slicer*. URL: https://github.com/jyjblrd/S4_Slicer.

- [16] Mathias Brandstötter, Arthur Angerer, and Michael Hofbaur. "An Analytical Solution of the Inverse Kinematics Problem of Industrial Serial Manipulators with an Ortho-parallel Basis and a Spherical Wrist". In: *Proceedings of the Austrian Robotics Workshop 2014 (Revised Version)*. 22–23 May, 2014. Linz, Austria, 2014.
- [17] Bundesanstalt für Straßenwesen (BASt). *Zustandsbewertung der Brücken im Zuge von Bundesfernstraßen*. Accessed: 20 Mai 2025. 2025. URL: https://www.bast.de/DE/Ingenieurbau/Fachthemen/brueckenstatistik/bruecken_hidden_node.html.
- [18] COBOD International A/S. *COBOD - World leader in 3D construction printing solutions*. Accessed: 27 April 2025. 2025. URL: <https://cobod.com/>.
- [19] Collaborative Research Centre TRR 277. *Additive Manufacturing in Construction (TRR277)*. Accessed: 2 April 2025. 2025. URL: <https://amc-trr277.de/>.
- [20] StackExchange Community. *How is the E argument calculated for a given G1 command?* Accessed: 29 April 2025. 2018. URL: <https://3dprinting.stackexchange.com/questions/6289/how-is-the-e-argument-calculated-for-a-given-g1-command>.
- [21] Dr. D-Flo. *How to Build a 3D Printer – Extruder*. Accessed: 12 April 2025. 2023. URL: <https://www.drdflo.com/pages/Guides/How-to-Build-a-3D-Printer/Extruder.html>.
- [22] DB InfraGO AG. *InfraGO-Zustandsbericht*. Accessed: 20 Mai 2025. 2024. URL: <https://www.dbinfrago.com/web/unternehmen/zielbild-infrastruktur/InfraGO-Zustandsbericht-12636112>.
- [23] J. Denavit and R. S. Hartenberg. "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices". In: *Journal of Applied Mechanics* 22.2 (1955), pp. 215–221. ISSN: 0021-8936. DOI: 10.1115/1.4011045. URL: <https://doi.org/10.1115/1.4011045>.
- [24] James Diebel. *Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors*. Tech. rep. Technical Report. Stanford, California, USA: Stanford University, 2006. URL: https://www.astro.rug.nl/software/kapteyn-beta/_downloads/attitude.pdf.
- [25] Kathrin Dörfler et al. "Advancing construction in existing contexts: Prospects and barriers of 3D printing with mobile robots for building maintenance and repair". In: *Cement and Concrete Research* 186 (2024), p. 107656. ISSN: 0008-8846. DOI: 10.1016/j.cemconres.2024.107656. URL: <https://www.sciencedirect.com/science/article/pii/S0008884624002370>.
- [26] Eindhoven University of Technology. *3D Concrete Printing – Structural Engineering and Design Research Group*. Accessed: 3 April 2025. 2025. URL: <https://www.tue.nl/en/research/research-groups/structural-engineering-and-design/3d-concrete-printing>.
- [27] Khaled Elashry and Ruairí Glynn. "An Approach to Automated Construction Using Adaptive Programing". In: Springer, 2014, pp. 51–66. ISBN: 978-3-319-04662-4. DOI: 10.1007/978-3-319-04663-1_4.
- [28] ETH Zürich, Departement Bau, Umwelt und Geomatik. *Departement Bau, Umwelt und Geomatik – ETH Zürich*. Accessed: 4 April 2025. 2025. URL: <https://baug.ethz.ch/>.
- [29] Fischer Holding GmbH Co. KG. *Baubot – Produktivität und Sicherheit durch Digitalisierung und Automatisierung*. <https://www.fischer.de/de-de/produkte/innovationen/baubot>. Accessed: 15 April 2025. 2025.
- [30] Formlabs. *Vergleich der 3D-Drucktechnologien: FDM, SLA und SLS*. Accessed: 4 April 2025. 2025. URL: <https://formlabs.com/de/blog/3d-drucktechnologie-vergleich-fdm-sla-sls/>.
- [31] PERI GmbH. *MAXIMO – Wandschalung für schnelles Schalen*. Accessed: 26 April 2025. 2024. URL: <https://www.peri.de/produkte/maximo-wandschalung-fuer-schnelles-schalen.html#vorteile> (visited on 04/13/2025).

- [32] Hauptverband der Deutschen Bauindustrie e.V. *Produktivität im Bauhauptgewerbe*. Accessed: 20 Mai 2025. 2023. URL: <https://www.bauindustrie.de/zahlen-fakten/publikationen/brancheninfo-bau/produktivitaet-im-bauhauptgewerbe>.
- [33] Hideo Kodama. *JPS56144478A*. 1980. URL: <https://patents.google.com/patent/US4041476A/en> (visited on 04/02/2025).
- [34] Hilti AG. *Semi-autonomer Baustellenroboter Jaibot*. Accessed: 15 April 2025. 2025. URL: <https://www.hilti.de/content/hilti/E3/DE/de/business/business/productivity/semi-autonomer-baustellenroboter-jaibot.html>.
- [35] Andreas Huber. *Technische Mechanik 3 – Dynamik*. 1st ed. Springer Vieweg Berlin, Heidelberg, 2024. ISBN: 978-3-662-68665-2. DOI: 10.1007/978-3-662-68666-9. URL: <https://doi.org/10.1007/978-3-662-68666-9>.
- [36] ICON Technology, Inc. *ICON - Intelligent Machines Building Humanity's Future*. Accessed: 27 April 2025. 2025. URL: <https://iconbuild.com/>.
- [37] Redaktion ingenieur.de. *Deutschlands bröckelnde Brücken: Der Weg aus der Krise*. Accessed: 20 Mai 2025. 2023. URL: <https://www.ingenieur.de/technik/fachbereiche/bau/deutschlands-broeckelnde-bruecken-der-weg-aus-der-krise/>.
- [38] Instatiq GmbH. *Instatiq – Plan, Print, Progress*. Accessed: 17 April 2025. 2024. URL: <https://instatiq.com/>.
- [39] Behrokh Khoshnevis. “Automated construction by contour crafting—related robotics and information technologies”. In: *Automation in Construction* 13.1 (2004). The best of ISARC 2002, pp. 5–19. ISSN: 0926-5805. DOI: <https://doi.org/10.1016/j.autcon.2003.08.012>. URL: <https://www.sciencedirect.com/science/article/pii/S0926580503000736>.
- [40] Harald Kloft, Norman Hack, and Hendrik Lindemann. “Shotcrete 3D Printing (SC3DP) - 3D-Drucken von großformatigen Betonbauteilen”. In: *Deutsche BauZeitschrift* 1 (Feb. 2019), pp. 54–57.
- [41] KUKA AG. *KUKA System Software 8.3 V4. Operating and Programming Instructions for System Integrators*. Internal documentation, not publicly available. 2015.
- [42] KUKA AG. *Datenblatt KR 340 R3300*. https://www.kuka.com/-/media/kuka-downloads/imported/8350ff3ca11642998dbdc81dcc2ed44c/0000233240_de.pdf. Accessed: 3 April 2025. 2020.
- [43] KUKA AG. *Robotersysteme – Produkte und Lösungen*. Accessed: 8 April 2025. n.d. URL: <https://www.kuka.com/de-de/produkte-leistungen/robotersysteme>.
- [44] Ena Lloret-Fritschi et al. “From Smart Dynamic Casting to a Growing Family of Digital Casting Systems”. In: *Cement and Concrete Research* 134 (Aug. 2020). Accessed: 3 April 2025. DOI: [10.1016/j.cemconres.2020.106071](https://doi.org/10.1016/j.cemconres.2020.106071). URL: <https://doi.org/10.1016/j.cemconres.2020.106071>.
- [45] Dirk Lowke et al. “Particle-bed 3D printing in concrete construction – Possibilities and challenges”. In: *Cement and Concrete Research* 112 (2018), pp. 50–65. DOI: <https://doi.org/10.1016/j.cemconres.2018.05.018>. URL: <https://www.sciencedirect.com/science/article/pii/S000888461731267X>.
- [46] Guowei Ma, Li Wang, and Yuyan Ju. “State-of-the-art of 3D printing technology of cementitious material—An emerging technique for construction”. In: *Science China Technological Sciences* 61.4 (2018). Accessed: 5 April 2025, pp. 475–495. DOI: [10.1007/s11431-016-9077-7](https://doi.org/10.1007/s11431-016-9077-7). URL: <https://doi.org/10.1007/s11431-016-9077-7>.
- [47] Marlin Firmware. *Marlin 3D Printer Firmware*. <https://marlinfw.org/>. Accessed: 8 April 2025. 2024.

- [48] Marlin Firmware. *Marlin G-code Documentation*. Accessed: 8 April 2025. 2025. URL: <https://marlinfw.org/meta/gcode/>.
- [49] Matplotlib Development Team. *Matplotlib - Visualization with Python*. Version 3.9.2, Accessed: 29 April 2025. 2025. URL: <https://matplotlib.org/>.
- [50] McKinsey Global Institute. *Reinventing Construction Through a Productivity Revolution*. Tech. rep. Accessed: 20 Mai 2025. McKinsey & Company, Feb. 2017. URL: <https://www.mckinsey.com/capabilities/operations/our-insights/reinventing-construction-through-a-productivity-revolution>.
- [51] V. Mechtcherine et al. "Extrusion-based additive manufacturing with cement-based materials – Production steps, processes, and their underlying physics: A review". In: *Cement and Concrete Research* 132 (2020). DOI: <https://doi.org/10.1016/j.cemconres.2020.106037>. URL: <https://www.sciencedirect.com/science/article/pii/S0008884619317132>.
- [52] Viktor Mechtcherine and Venkatesh Naidu Nerella. "3-D-Druck mit Beton: Sachstand, Entwicklungstendenzen, Herausforderungen". In: *Bautechnik* 95.4 (2018), pp. 275–287. DOI: <https://doi.org/10.1002/bate.201800001>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/bate.201800001>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/bate.201800001>.
- [53] Viktor Mechtcherine et al. "Large-scale digital concrete construction – CONPrint3D concept for on-site, monolithic 3D-printing". In: *Automation in Construction* 107 (2019). ISSN: 0926-5805. DOI: <https://doi.org/10.1016/j.autcon.2019.102933>. URL: <https://www.sciencedirect.com/science/article/pii/S0926580519304601>.
- [54] Valeria Montjoy. *Infographic: The Evolution of 3D Printing in Architecture Since 1939*. Accessed: 3 April 2025. 2023. URL: <https://www.archdaily.com/1005043/infographic-the-evolution-of-3d-printing-in-architecture-since-1939>.
- [55] Shawn Moylan et al. "An Additive Manufacturing Test Artifact". In: *Journal of Research of the National Institute of Standards and Technology* 119 (2014), pp. 429–459. DOI: [10.6028/jres.119.017](https://doi.org/10.6028/jres.119.017). URL: <https://doi.org/10.6028/jres.119.017>.
- [56] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. Accessed: 8 April 2025. Boca Raton: CRC Press, 1994. URL: <https://www.cit.tum.de/fileadmin/w00cgn/rm/pdf/murray-li-sastry-94-complete.pdf>.
- [57] Pooyan Nayyeri, Kourosh Zareinia, and Habiba Bougerara. "Planar and nonplanar slicing algorithms for fused deposition modeling technology: a critical review". In: *The International Journal of Advanced Manufacturing Technology* 119.5 (2022), pp. 2785–2810. ISSN: 1433-3015. DOI: [10.1007/s00170-021-08347-x](https://doi.org/10.1007/s00170-021-08347-x). URL: <https://doi.org/10.1007/s00170-021-08347-x>.
- [58] Saeed B. Niku. *Introduction to Robotics: Analysis, Control, Applications*. 3rd. 528 pages. Wiley, 2019. ISBN: 978-1-119-52760-2.
- [59] Deutsches Institut für Normung. *DIN 66025: Programmieren numerisch gesteuerter Werkzeugmaschinen - Zeichencodierte Steuerungssätze*. Beuth Verlag, Berlin. 1982.
- [60] Deutsches Institut für Normung. *DIN EN ISO/ASTM 52900:2022-03 – Additive manufacturing – General principles – Terminology*. Beuth Verlag, Berlin. Accessed: 3 April 2025. 2022. URL: <https://www.dinmedia.de/en/standard/din-en-iso-astm-52900/344258696>.
- [61] Open Continuum Robotics. *Continuum Robotics 101*. Accessed: 12 April 2025. Oct. 2022. URL: <https://www.opencontinuumrobotics.com/101/2022/10/21/cr-intro.html>.
- [62] OpenAI. *ChatGPT (GPT-4o)*. Accessed: 21 April 2025. 2025. URL: <https://chat.openai.com/>.
- [63] OrcaSlicer Community. *OrcaSlicer – Powerful and Open-Source Slicer for FDM and Resin 3D Printers*. Version 2.2.0. Accessed: 8 April 2025. 2025. URL: <https://orca-slicer.com/>.

- [64] Zoran Pandilov and Vladimir Dukovski. "Comparison of the Characteristics between Serial and Parallel Robots". In: *Acta Technica Corviniensis – Bulletin of Engineering* 7.1 (2014), pp. 23–28. ISSN: 2067-3809.
- [65] PFT - Technische Förderanlagen GmbH. *PFT HM 24 – Der clevere Mischer für Sack- und Silowaren*. Accessed: 8 April 2025. 2025. URL: <https://www.pft.net/de/mischen-foerdern/durchlaufmischer/fuer-sackware/hm-24.html>.
- [66] PFT - Technische Förderanlagen GmbH. *PFT SWING L – Förderpumpe für pastöse Materialien*. Accessed: 12 April 2025. 2025. URL: <https://www.pft.net/de/230v-technik/foerderpumpen/swing-1/>.
- [67] Prusa Research. *Printer Troubleshooting*. Accessed: 12 April 2025. 2025. URL: https://help.prusa3d.com/category/printer-troubleshooting_227.
- [68] Prusa Research. *PrusaSlicer – Open-Source Slicer Software*. Version 2.9.2. Accessed: 8 April 2025. 2025. URL: https://www.prusa3d.com/de/page/prusaslicer_424/.
- [69] Putzmeister. *InstaTIQ – Putzmeister Robotic Arm for 3D Concrete Printing*. Accessed: 11 April 2025. 2025. URL: <https://bauma.putzmeister.com/de/institiq>.
- [70] pytest Development Team. *pytest - Simple powerful testing with Python*. Version 8.3.4, Accessed: 29 April 2025. 2025. URL: <https://pypi.org/project/pytest/>.
- [71] Python Software Foundation. *Python Programming Language, Version 3.12*. <https://www.python.org/>. Version 3.12, Accessed: 17 April 2025. 2025.
- [72] R. Rao. *What are Robot Gantry? Types, Applications, Advantages, Selection Criteria, and more*. Accessed: 11 April 2025. 2023. URL: <https://www.wevolver.com/article/gantry-robot>.
- [73] Prusa Research. *Extrusion stopped mid-print (heat creep)*. 2022. URL: https://help.prusa3d.com/article/extrusion-stopped-mid-print-heat-creep_1948 (visited on 04/11/2025).
- [74] Research Nester. *Marktanalyse und Prognose des 3D-Drucks im Bauwesen – Trends, Chancen und Zukunftsausblick bis 2036*. Accessed: 20 Mai 2025. 2024. URL: <https://researchnester.com/de/reports/3d-printing-construction-market/4593>.
- [75] Robert McNeel & Associates. *Rhinoceros 3D - Modeling Tools for Designers*. Version 8, Accessed: 29 April 2025. 2025. URL: <https://www.rhino3d.com/de/>.
- [76] RoboDK Software S.L. *RoboDK - Simulation and Programming for Industrial Robots*. Accessed: 27 April 2025. 2025. URL: <https://robodk.com/de/>.
- [77] Robotplace. *Apis Cor 3D Printing System*. Accessed: 8 April 2025. no date. URL: <https://robotplace.io/product/apis-cor-3d-printing-system/>.
- [78] SciPy Community. *SciPy - Fundamental algorithms for scientific computing in Python*. Version 1.15.2, Accessed: 29 April 2025. 2025. URL: <https://scipy.org/>.
- [79] Pshtiwan Shakor et al. "Review of binder jetting 3D printing in the construction industry". In: *Progress in Additive Manufacturing* 7 (2022). Published: 21 January 2022, pp. 643–669. DOI: 10.1007/s40964-021-00248-9. URL: <https://link.springer.com/article/10.1007/s40964-021-00248-9>.
- [80] Shenzhen Creality 3D Technology Co., Ltd. *Creality – Official Website of 3D Printers and Accessories*. Accessed: 8 April 2025. 2025. URL: <https://www.creality.com/>.
- [81] Tharmalingam Sivarupan et al. "A review on the progress and challenges of binder jet 3D printing of sand moulds for advanced casting". In: *Additive Manufacturing* 40 (2021), p. 101889. ISSN: 2214-8604. DOI: <https://doi.org/10.1016/j.addma.2021.101889>. URL: <https://www.sciencedirect.com/science/article/pii/S2214860421000543>.

- [82] RoboDK Software S.L. *Robot Singularities: What Are They and How to Beat Them*. Accessed: 15 Mai 2025. 2022. URL: <https://robodk.com/blog/robot-singularities/>.
- [83] Gregory G. Slabaugh. *Computing Euler Angles from a Rotation Matrix*. No publication date provided; Accessed: 22 April 2025. n.d. URL: <https://eecs.qmul.ac.uk/~gslabaugh/publications/euler.pdf>.
- [84] Steve Canny. *python-docx: Create and update Microsoft Word .docx files*. Version 0.2.4, Accessed: 29 April 2025. 2025. URL: <https://pypi.org/project/python-docx/>.
- [85] Steve Scott Crump. *US5121329A*. 1989. URL: <https://patents.google.com/patent/US5121329A/en> (visited on 04/02/2025).
- [86] Straits Research. *Marktgröße, Marktanteil und Trendanalyse für 3D-Druck im Bauwesen*. 2025. URL: <https://straitsresearch.com/report/3d-printing-construction-market>.
- [87] Stratasys Ltd. *3D-Drucker Übersicht*. Accessed: 2 April 2025. 2025. URL: <https://www.stratasys.com/de/3d-printers/>.
- [88] Julius Sustarevas, Dimitrios Kanoulas, and Simon Julier. “Autonomous Mobile 3D Printing of Large-Scale Trajectories”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022. DOI: 10.1109/IROS47612.2022.9982274.
- [89] Tagesschau.de. *Ein Haus in 140 Stunden*. Accessed: 20 Mai 2025. 2024. URL: <https://www.tagesschau.de/wirtschaft/digitales/3d-drucker-groesstes-haus-europas-100.html>.
- [90] Peter Guthrie Tait. “On the Rotation of a Rigid Body about a Fixed Point”. In: *Proceedings of the Royal Society of Edinburgh* 6 (1868), pp. 430–434. URL: <https://zenodo.org/record/2093148/files/article.pdf>.
- [91] Teaching Tech. *3D Printer Calibration Guide*. Accessed: 27 April 2025. 2025. URL: <https://teachingtechyt.github.io/calibration.html>.
- [92] Tuong Phuoc Tho and Truong Thinh Nguyen. “Using a Cable-Driven Parallel Robot with Applications in 3D Concrete Printing”. In: *Applied Sciences* 11.2 (2021), p. 563. DOI: 10.3390/app11020563. URL: <https://doi.org/10.3390/app11020563>.
- [93] Ultimaker. *CuraEngine Command Line Reference*. Accessed: 28 April 2025. 2020. URL: <https://manpages.ubuntu.com/manpages/focal/en/man1/CuraEngine.1.html>.
- [94] Ultimaker. *The Complete History of 3D Printing*. Accessed: 1 April 2025. 2023. URL: <https://ultimaker.com/de/learn/the-complete-history-of-3d-printing/>.
- [95] Ultimaker. *Ultimaker Cura User Manual*. 2023. URL: <https://support.ultimaker.com>.
- [96] Ultimaker. *CuraEngine Wiki: Slicing Overview*. Accessed: 27 April 2025. 2025. URL: <https://github.com/Ultimaker/CuraEngine/wiki/Slicing>.
- [97] Ultimaker. *Ultimaker Cura – Advanced 3D Printing Slicer Software*. Version Version 5.10.0. Accessed: 8 April 2025. 2025. URL: <https://ultimaker.com/de/software/ultimaker-cura/>.
- [98] Ultimaker Community User. *Python or Bash Cura Interface?* Forum post, Accessed: 26 April 2025. 2019. URL: <https://community.ultimaker.com/topic/30295-python-or-bash-cura-interface/>.
- [99] Universal Robots. *What Is Robotic Arm Design?* Accessed: 6 April 2025. n.d. URL: <https://www.universal-robots.com/in/blog/robotic-arm-design/>.
- [100] K. Vasilic. “Additive Fertigung im Betonbau: aktueller Stand”. In: *Beton- und Stahlbetonbau* 115.11 (2020). Accessed: 3 April 2025, pp. 1006–1012. DOI: 10.1002/best.202000079. URL: <https://doi.org/10.1002/best.202000079>.

- [101] Verified Market Reports. *Globaler Markt für 3D -Drucker des Bauwesens nach Typ (Beton, Metall), nach Anwendung (Gebäude, Infrastruktur), nach geografischem Umfang und Prognose*. Accessed: 20 Mai 2025. 2025. URL: <https://www.verifiedmarketreports.com/de/product/construction-3d-printing-market/>.
- [102] W3Schools. *Python RegEx*. Accessed: 17 April 2025. 2025. URL: https://www.w3schools.com/python/python_regex.asp.
- [103] WASP - World's Advanced Saving Project. *Concrete 3D Printer - Delta WASP 3MT CONCRETE*. Accessed: 5 April 2025. 2025. URL: <https://www.3dwasp.com/en/concrete-3d-printer-delta-wasp-3mt-concrete/>.
- [104] Daniel Weger. "Additive Fertigung von Betonstrukturen mit der Selective Paste Intrusion – SPI". Accessed: 3 April 2025. Ph.D. thesis. Technische Universität München, 2020. URL: <https://mediatum.ub.tum.de/doc/1540336/document.pdf>.
- [105] Thomas Wehrli. "Wie William E. Urschel mit der Mauerbaumaschine in die Zukunft fuhr". In: *Baublatt* 21 (2021). Accessed: 2 April 2025, pp. 44–47. URL: https://www.baublatt.ch/file/1376/BB_2121_Urschel_0001.pdf.
- [106] William E Urschel. *US2339892A*. Accessed: 3 April 2025. 1941. URL: <https://patents.google.com/patent/US2339892A/en>.
- [107] Institut der deutschen Wirtschaft Köln. *Wohnungsbedarf: Jährlich müssten 372.600 Wohnungen gebaut werden*. Accessed: 3 Mai 2025. 2024. URL: <https://www.iwkoeln.de/presse/pressemitteilungen/jaehrlich-muessen-372600-wohnungen-gebaut-werden.html>.
- [108] Shaun Wu. *6-Axis Gantry Robot*. Accessed: 8 April 2025. 2019. URL: <https://shaunwu25.github.io/2019/06/15/6-Axis-Gantry-Robot.html>.
- [109] Wyn Kelly Swainson. *US4041476A*. 1971. URL: [\(visited on 04/02/2025\)](https://patents.google.com/patent/US4041476A/en).
- [110] Julian Zeitlhöfler. "Nominal and Observation-Based Attitude Realization for Precise Orbit Determination of the Jason Satellites". PhD thesis. Technical University of Munich, 2019.
- [111] Tianyu Zhang et al. "S3-Slicer: A General Slicing Framework for Multi-Axis 3D Printing". In: *ACM Trans. Graph.* 41.6 (2022). Software implementation available at https://github.com/zhangty019/S3_DeformFDM. ISSN: 0730-0301. DOI: 10.1145/3550454.3555516. URL: <https://doi.org/10.1145/3550454.3555516>.