

A METHOD FOR ADDITIVE MANUFACTURING OF FUNCTIONALLY GRADED
DENTAL CROWNS

A Thesis

by

CLAY MICHAEL MULLINS

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,
Committee Members,
Head of Department,

Jyhwen Wang
Mathew Kuttolamadom
Sarah Wolff
Reza Langari

December 2022

Major Subject: Engineering Technology

Copyright 2022 Clay Mullins

ABSTRACT

Leucite containing ceramic has been proven in past research to have increased abrasive wear resistance in dental composites when compared to commercially available dental composite materials. Using this material as a basis for manufacturability, the additive manufacturing (AM) method of vat photopolymerization (VP) was selected. Functional grading was also incorporated into the manufacturing process to better represent the natural tooth structure. The effect of incorporating functionally graded additive manufacturing (FGAM) into the manufacturing process of a dental crown will result in an internal structure that more closely resembles that of a natural tooth.

The traditional layer-based method of AM technologies poses several challenges for functional grading. This research effort is the first to develop a new method in the form of shell-based manufacturing, since commercially available slicing software is incapable of producing shell-based manufacturing g-code. The focus of this thesis addresses this issue, and details a program that collects layer-based g-code and converts the data into shell-based g-code. The program utilizes the g-code produced by slicing software when given an .STL file of a crown geometry, and converts that g-code into separate and unique g-codes for each shell. After extracting X and Y coordinate data from the original g-code, an algorithm creates concentric regions bounded by the interior and exterior walls of each layer. The concentric regions become the basis for each shell, and the newly created coordinate data is converted into separate and unique g-code for each shell.

The objective of this research is to develop a methodology for the FGAM of dental crowns. VP is the best AM technique because of its high resolution and accuracy, which is necessary for

the production of small and complex geometry like those found in dental crowns. However, the AM method of fused deposition modeling (FDM) is used to prove the concept of converting layer-based g-code to shell-based g-code due to current machine availability. To resolve the issues of resolution and accuracy, the crown geometry was scaled 500%. Four separate shells were created using FDM, which proves the concept that layer-based g-code can be converted to separate shell-based g-code.

ACKNOWLEDGMENTS

This work was supported by the thesis committee chair Dr. Jyhwen Wang, Professor in Manufacturing and Mechanical Engineering Technology, Department of Engineering Technology and Industrial Distribution, Texas A&M University. Additional support and crown .STL file geometry was provided by Dr. Amirali Zandinejad, Associate Professor in the Department of Comprehensive Dentistry, Texas A&M University. I also wish to acknowledge and thank committee members Dr. Mathew Kuttolamadom, Associate Professor in Manufacturing and Mechanical Engineering Technology, Department of Engineering Technology and Industrial Distribution, and Dr. Sarah Wolff, Assistant professor in Industrial and Systems Engineering for their support during the review process.

FDM machine and equipment were provided by the Department of Engineering Technology and Industrial Distribution, Texas A&M University.

Special thanks to Dr. Angie Price, Associate Dean for Undergraduate Programs, College of Engineering, Texas A&M University for selecting me for a TA position in MMET-207 Metallic Materials, which provided the funding for the completion of this master's program.

Much love goes out to the family and friends who provided emotional and financial support during the course of this project. Thanks to Dr. Gail Nicosia DDS and Dr. James Kim DDS for providing answers to late night dental questions. Special thanks to Karen Leuty and Patrick Mullins for being great parents and supporting me during hard times. Special thanks to Marilyn and Ray Leuty for providing financial support during my undergraduate career. Special thanks to my girlfriend Allyson Eoff for all that you have done to make me a better man.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis committee consisting of Dr. Jyhwen Wang, [advisor], Dr. Mathew Kuttolamadom [committee member] of the Engineering Technology and Industrial Distribution Department and Dr. Sarah Wolff [committee member] of the Industrial and Systems Engineering Department.

The work presented for the thesis was conducted independently by the student.

Funding Sources

Financial support for FDM materials and equipment provided by the Department of Engineering Technology and Industrial Distribution, Texas A&M University. Student financial support provided by a graduate teaching assistant position in the Manufacturing and Mechanical Engineering Technology Department, Texas A&M University.

TABLE OF CONTENTS

ABSTRACT.....	ii
ACKNOWLEDGMENTS	iv
CONTRIBUTORS AND FUNDING SOURCES	v
TABLE OF CONTENTS.....	vi
NOMENCLATURE	viii
LIST OF FIGURES	ix
CHAPTER 1 INTRODUCTION	1
1.1: Natural Tooth Structure and Functional Grading.....	2
1.2: Selection of a Manufacturing Technique	4
1.3: Research objective and approach	6
CHAPTER 2 LITERATURE REVIEW	7
2.1: Leucite containing ceramic.....	7
2.2: Functional Grading.....	7
2.3: Functional Grading of Ceramics using Additive Manufacturing.....	8
2.3.1: SLA of dental crowns	9
CHAPTER 3 METHOD OF MANUFACTURING.....	10
3.1: AM Technology Selection.....	10
3.2 Slicing Software	12
3.3: Achieving Functional Grading Through Changes in Particle Density.....	15
3.4: Modifications to the crown .STL file	15
3.5: Block Diagram of Manufacturing Process	21
3.6: Adjustments made to process for printing with FDM	22
CHAPTER 4 PROGRAM DEVELOPMENT	23
4.1: Using imported g-code to extrapolate relevant X and Y coordinate data	23
4.2: Using ceiling and roof layers to find the start of overhang for each shell.....	26
4.3: Manipulating X and Y coordinates to create new interior shell coordinates	29

4.3.1: Corresponding exterior points to interior points.....	29
4.3.2: Creating new shell points.....	33
4.4: Creating new g-code files for each shell	34
4.4.1: Initializing g-code	34
4.4.2: Keeping track of extrusion and height values.....	34
4.4.3: Writing g-code from X and Y coordinates	35
4.4.4: Writing g-code for build plate adhesion	36
4.4.5: Writing g-code for support structure	37
4.4.6: Writing g-code for interior and exterior walls.....	41
4.4.7: Writing g-code for infill.....	42
4.4.8: Writing g-code in region above shell ceiling.....	45
4.4.9: Finalizing g-code	46
4.6: Block diagram of full program.....	47
CHAPTER 5 PRINTING PROCEDURE.....	48
5.1: FDM printer details	48
5.2: Slicing software settings.....	49
5.3: Modifications to manufacturing process to accommodate FDM	49
5.4: Post processing	49
CHAPTER 6 CONCLUSIONS AND RECOMMENDATIONS	57
Recommendations for Future Work.....	57
REFERENCES	59
APPENDIX A CURA SLICING SOFTWARE SETTINGS.....	63

NOMENCLATURE

ABBREVIATIONS

AM	Additive Manufacturing
CODE	Ceramic On-Demand Extrusion
DLP	Digital Light Projection
EDJ	Enamel Dentin Junction
FGC	Functionally Graded Ceramic
FDM	Fusion Deposition Modeling
FGAM	Functionally Graded Additive Manufacturing
FGM	Functionally Graded Material
HIP	Hot Isostatic Pressing
LAM	Laser Additive Manufacturing
LDED	Laser Directed Energy Deposition
SLA	StereoLithography
SLS	Selective Laser Sintering
VP	Vat Photopolymerization

LIST OF FIGURES

Figure 1: Weight loss of the composites vs. time [2]	2
Figure 2: Abrasion rate vs. microhardness of the composites [2].....	2
Figure 3: Zones of Vickers hardness testing on human tooth[2]	3
Figure 4: Vickers hardness values for Enamel and Dentin structures. EDJ is the intermediate region between enamel and dentin [2]	4
Figure 5: AM manufacturing techniques for ceramic materials [3]	5
Figure 6: Method of printing in shells	6
Figure 7: A. Digital design of anatomically contoured crown and B. Splinted crown [5]	9
Figure 8: Order of print features in g-code	15
Figure 9: Isometric view of crown geometry .STL.....	17
Figure 10: Front view of crown geometry .STL	17
Figure 11: Interior view of crown geometry .STL.....	18
Figure 12: Tooth preparation type chamfer in orange outline	18
Figure 13: Types of tooth preparation[11].....	19
Figure 14: Shells of .STL crown geometry.....	19
Figure 15: Section view of crown geometry after modification	20
Figure 16: Isometric view of crown geometry after modification	20
Figure 17: Block Diagram of Manufacturing Process	21
Figure 18: Block diagram for extracting X and Y coordinate values	25
Figure 19: g-code of file_ceiling saved as is	26
Figure 20: Complex geometry is starting to print on layer 89	27
Figure 21: No complex geometry found on layer 88	28
Figure 22: Roof and ceiling layers.....	28
Figure 23: Quadrant naming convention	31
Figure 24: Block diagram of interior points function	32
Figure 25: ratio of extrusion to movement	35
Figure 26: Build plate adhesion of shell3	37
Figure 27: Block diagram of infill_zig for even layers.....	41

Figure 28: Block diagram of gcode_infill.....	44
Figure 29: Result of gcode_shell and infill_zig in regions above the shell ceiling	45
Figure 30: Block diagram of full program.....	47
Figure 31: LULZBOT TAZ 6.....	48
Figure 32: Support removal with small screwdriver.....	50
Figure 33: Printed part after support removal.....	51
Figure 34: DREMEL with sanding and polishing accessories	51
Figure 35: Interior ceiling after grinding away support structure.....	52
Figure 36: Sanding vertically along the taper of the part until layers are no longer visible.....	52
Figure 37: Sanding along the width of the part with respect to the taper	53
Figure 38: Buildup of filament near the build platform due to reduced print speed	54
Figure 39: Before and after post processing	54
Figure 40: Isometric view of printed shells	55
Figure 41: Side view of printed shells	55
Figure 42: Side view of combined shells.....	56
Figure 43: Underside view of shells combined shells	56
Figure A.1: CURA settings for Quality	63
Figure A.2: CURA settings for shell.....	63
Figure A.3: CURA settings for infill	64
Figure A.4: CURA settings for material.....	65
Figure A.5: CURA settings for speed.....	66
Figure A.6: CURA settings for travel.....	66
Figure A.7: CURA settings for cooling	67
Figure A.8: CURA settings for support	68
Figure A.9: CURA settings for build plate adhesion.....	68
Figure A.10: CURA settings for mesh fixes.....	68
Figure A.11: CURA settings for special modes.....	68
Figure A.12: CURA settings for experimental	69

CHAPTER 1 INTRODUCTION

A material called leucite containing ceramic has been shown to have an increased abrasive wear resistance in dental composites when compared to composites made with commercially available materials. An abrasive wear test was conducted on three different dental composite materials, the research showed that leucite containing ceramic had the lowest weight loss percentage as shown in Figure 1. The figure shows that percentage of weight loss vs. time is linear, and that the leucite containing ceramic had the smallest slope. This indicates that the material had a better wear resistance when compared to the material alternatives: TetricCeram and glass-based. This effect has been attributed to the higher surface hardness of the material (Figure 2). This figure compares the wear rate found in the first figure vs the Vickers microhardness and shows that the hardness of leucite containing ceramic is higher than that of the material alternatives. Abrasive wear resistance is a significant material property when determining which material to use for a dental composite, as abrasion and attrition have been identified as the main clinical wear mechanisms for dental composites [1].

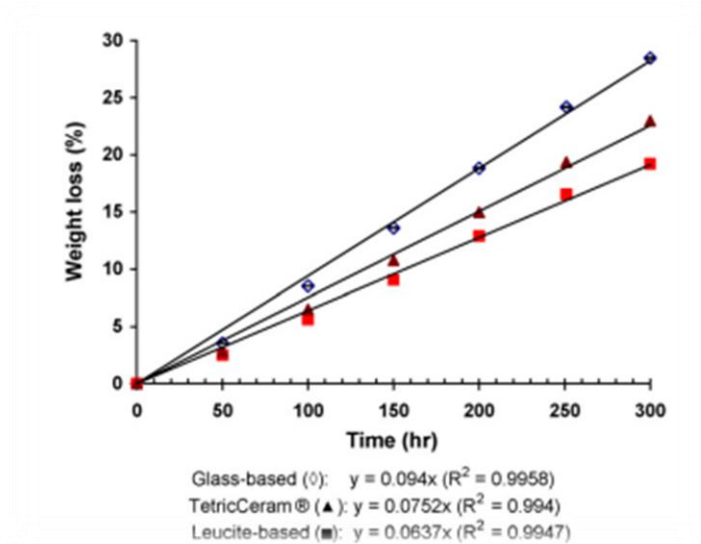


Figure 1: Weight loss of the composites vs. time [2]

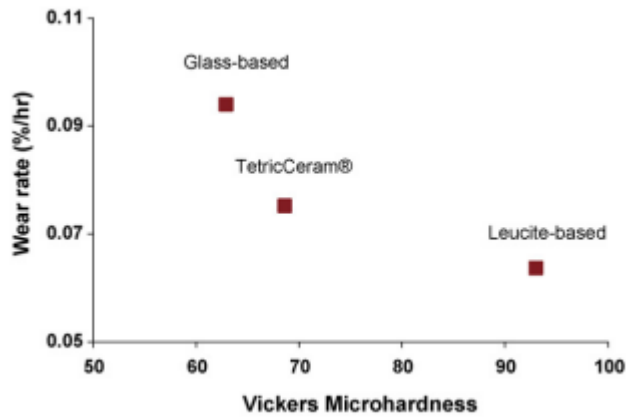


Figure 2: Abrasion rate vs. microhardness of the composites [2]

1.1: Natural Tooth Structure and Functional Grading

The human tooth is a natural functionally grading structure that allows the tooth to resist the occlusal forces found in chewing or mastication. The exterior structure of the tooth, enamel, is a hard, mineral rich human tissue that resists abrasion and corrosion. Dentin is a similarly

composed structure found further into the internal tooth structure with a much lower hardness than the enamel. Both enamel and dentin have the same general hardness throughout the structure, except for the intermediate region between enamel and dentin, known as the enamel dentin junction (EDJ) [2]. Figure 3 and Figure 4 show the hardness of a human tooth at different testing locations towards the interior. Figure 3 shows a cross section of a human tooth and the various zones that are hardness tested. Figure 4 shows the Vickers hardness values for each of the zones.

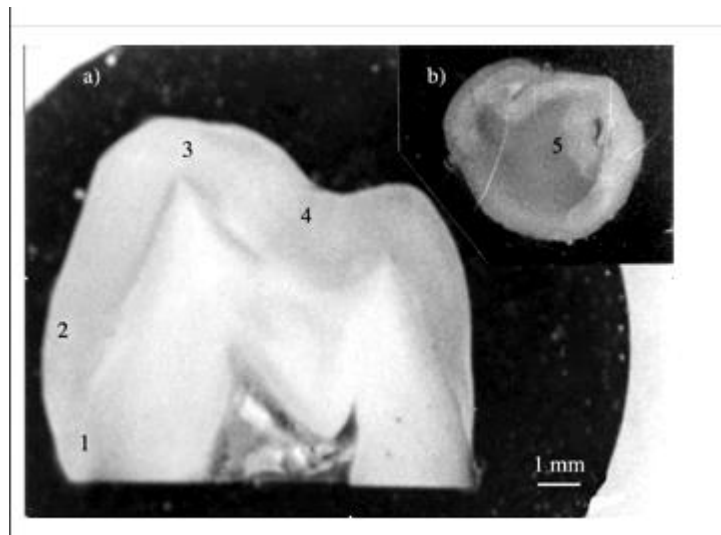


Figure 3: Zones of Vickers hardness testing on human tooth[2]

	Zone 2	Zone 3	Zone 4	Zone 5
			297	350
		297	339	350
		274	339	350
		297	290	350
Enamel	290	322	322	383
	290	322	322	383
	290	322	322	383
	290	322	322	383
	290	322	322	350
	274	322	397	350
	290	322	297	350
	254	297	322	350
	274	297	297	350
EDJ	193	160	206	236
	46.7	51.5	51.5	57.5
Dentin	46.7	51.5	49.8	55.5
	46.7	51.5	49.8	55.5

Figure 4: Vickers hardness values for Enamel and Dentin structures. EDJ is the intermediate region between enamel and dentin [2]

This functional grading allows the tooth to resist abrasion and corrosion near the surface, while allowing the tooth to also flex and deform under applied stress. Traditional dental composites, however, are composed of a single material without functional grading. Therefore, the creation of a dental composite with functional grading must be investigated.

1.2: Selection of a Manufacturing Technique

Additive Manufacturing (AM) has been shown to have the ability to manufacture materials with functional grading. AM also has the potential to produce geometries that are too complex for conventional manufacturing methods. As a whole, AM technologies can be used to produce a wide range of materials, however the field becomes narrowed when only considering ceramic materials. Figure 5 shows the possible additive manufacturing methods that can be used with ceramics.

Taking into account the requirement of the ability to produce functionally graded materials (FGM), the field becomes even narrower. Functionally graded ceramics (FGC) have been produced in past research using various AM methods that will be explored in section 2.2 of the literature review. Ultimately, digital light projection (DLP) using vat photopolymerization was selected as the AM method to create dental composites with functional grading. The two AM methods that meet the resolution and accuracy requirements of producing small and complex geometry are VP and NanoParticle Jetting (NPJ). VP was chosen as the method to be used in this research because of its ability to manufacture shells rather than the layer-based method seen in NPJ.

Direct Additive Manufacturing of Advanced Ceramics								
Single-step processes		Multi-step processes						
Bedless	Bed	Bed			Bedless			
Directed Energy Deposition	Powder Bed	Fusion	Binder Jetting	Sheet Lamination	Material Extrusion		Material Jetting	Vat Photopolymerisation
LENS	Powder- dLS	Powder-iLS	Powder-BJ	LOM	Wax-based	Water-based	Solvent-DIP	SL
	Slurry- dLS	Slurry-iLS	Slurry-BJ	CAM-LEM	FDC	RC / DIW	Wax-DIP	DLP / LCM
					MJS	FEF		SPPW
					T3DP	CODE		2PP
					PHASE*	3DGP		

<p>LENS: Laser Engineered Net Shaping dLS/iLS: direct/indirect Laser Sintering BJ: Binder Jetting LOM: Laminated Object Manufacturing CAM-LEM: Computer-Aided Manufacturing of Laminated Engineering Materials SL: Stereolithography DLP: Digital Light Projection LCM: Lithography-based Ceramic Manufacturing SPPW: Self-Propagating Photopolymer Waveguide 2PP: Two-Photon Photopolymerisation</p>	<p>FDC: Fused Deposition of Ceramics MJS: Multiphase Jet Solidification T3DP: Thermoplastic 3D Printing PHASE*: PHotopolymerisation-ASsisted Extrusion - *suggested acronym RC: Robocasting DIW: Direct Ink Writing FEF: Freeze-Form Extrusion Fabrication CODE: Ceramic On-Demand Extrusion 3DGP: 3D Gel Printing DIP: Direct Inkjet Printing</p>
--	---

Figure 5: AM manufacturing techniques for ceramic materials [3]

1.3: Research objective and approach

The objective of this research is to develop a manufacturing method to produce dental composites using FGAM. The devised solution involves a shell-based manufacturing approach rather than the traditional layer-based method. The first step of this approach, and the focus of this research is in developing a program that can convert layer-based g-code into shell-based. AM machines use a language called g-code to control their movement during the manufacturing process, creating full geometry one layer at a time. Figure 6 shows what the shell-based approach would look like with a dental crown geometry. Functional grading will be implemented via changing the density of leucite containing ceramic particles, where each shell has a higher density of ceramic particles than the more interior shell.

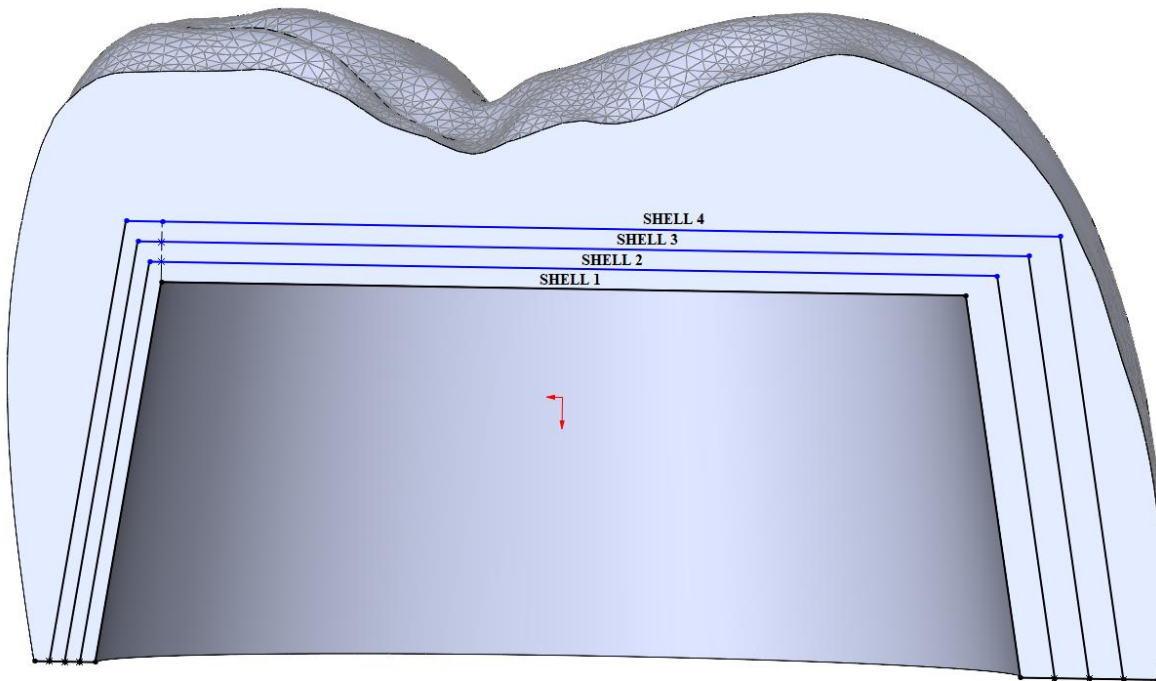


Figure 6: Method of printing in shells

CHAPTER 2 LITERATURE REVIEW

2.1: Leucite containing ceramic

Leucite containing ceramic has been shown to have significantly improved abrasive wear resistance when compared to commercially available dental composite materials. The materials tested were aluminum barium silicate glass filler, a commonly used dental composite material, and TetricCeram, a commercially available dental composite material. Vickers surface microhardness, flexural strength, and flexural modulus for the three materials were measured. The results showed that the leucite containing ceramic had a significantly higher hardness measurement than the other two materials. There was no significant difference between the flexural strength and modulus for the leucite containing ceramic and TetricCeram [4].

2.2: Functional Grading

FGM are composite materials that involve gradual variations in their compositions. They are commonly found in nature, so researchers are investigating ways in which they can emulate this technology in modern applications. Since AM technologies are no longer restricted to single materials, AM has been presented as a method to produce FGMs. FGAM has been developed in past research and can create the kind of FGM that is the focus of this research, a single-phase material with a gradual variation in density [3].

However, there are still issues with the current state of this technology. The main concern for this research is that commercially available AM technologies are mainly suited for homogeneous material composition. AM machines must be modified or used in unique ways to

produce functional grading. Another concern is that there is poor dimensional control when producing objects made with this technology. Current research in FGM has primarily been focused on producing testing specimens with a simple geometry in order to analyze them for their microstructural accuracy [3]. Therefore, more research is required to utilize FGAM in production of a part with complex geometry, like that of a dental crown.

2.3: Functional Grading of Ceramics using Additive Manufacturing

Ceramic materials benefit from an AM process because they can produce complex structures without the need for high-cost molds [8]. VP is unique in the field of ceramic AM in that it has high resolution and accuracy and can produce small and complex geometries with a high surface finish [6]. The ceramic VP process involves the creation of a slurry, where ceramic particles are suspended in a solution with monomers and photoinitiators [9]. To avoid defects during post-processing, proper creation of a ceramic suspension is required [7]. After the VP process, the resulting part is considered a “green part” and requires additional post processing which involves debinding and sintering. The debinding process burns away the monomers or polymers, leaving behind the ceramic particles. The sintering process converts the debinded particles into a fully dense ceramic [9].

Other forms of AM technologies used to manufacture ceramics with functional grading include hot isostatic pressing (HIP) [16], laser directed energy deposition (LDED) [14], a novel extrusion based AM process[18] laser additive manufacturing (LAM)[19], selective laser sintering (SLS)[20], and ceramic on-demand extrusion (CODE) [15]. A method which addresses the resolution and accuracy concerns with small and complex geometry is found in NanoParticle Jetting, a binder jetting type of AM that is capable of producing accurate parts with smooth

surfaces[21]. This AM method is incapable of implementing a shell-based manufacturing method as existing shell geometry would collide with the binder jetting nozzle.

2.3.1: SLA of dental crowns

StereoLithography (SLA), another name for VP, is an AM process which involves the use of a light source and a photocuring material resin. Previous research has shown that FGC can be manufactured using this method[22]. In Revilla-León M et al. Crowns were manufactured using AM and the traditional subtractive manufacturing method. Crowns manufactured with AM resulted in a significant number of marginal and internal discrepancies. The amount of discrepancies from AM dental crowns was considered clinically unacceptable. However, for the splinted crown manufactured with SLA, the amount of discrepancies was considered clinically acceptable. This effect was possibly attributed to the volumetric difference between the two crown geometries as seen in Figure 7. The trueness of both methods to the digital file was analyzed and showed that there were no significant differences [5]. As it relates to this thesis, the previous research conducted shows that complex crown geometry can be accurately manufactured using SLA, however the debinding and sintering process must be improved to eliminate marginal and internal discrepancies.

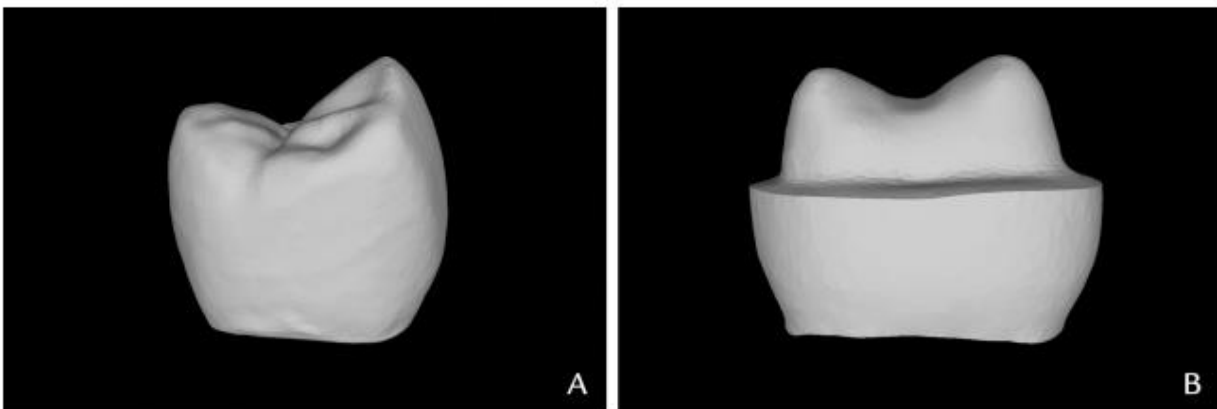


Figure 7: A. Digital design of anatomically contoured crown and B. Splinted crown [5]

CHAPTER 3 METHOD OF MANUFACTURING

3.1: AM Technology Selection

So far, the requirements for which AM technology to use for this research are the ability to print using ceramic material, and good resolution and accuracy since the object to be produced will have small and complex geometry. Further requirements will be discussed in this section.

The typical method of AM involves printing layer-by-layer to produce the final geometry. In a layer-by-layer of FGAM, each layer must undergo a material change multiple times to form an outwards gradient. As the total number of layers increases, the number of material changes required would increase multiplicatively. A method for making material changes quickly and efficiently in FGAM could be explored in future research, however, this research explores a shell-based method. Instead of printing the full geometry layer by layer, the method is to print in shells, where each shell will be created using differing material densities. This method would resemble a matryoshka doll, except the shells would be fused together from the printing process.

Since a shell-based method of manufacturing will be used in this research, a new requirement for this capability in AM technologies arises. The remaining AM technologies that meet the updated requirements are FDM and VP. The final selection for which AM technology to utilize in this research is VP because of its ability to print using ceramic material, the very high resolution and accuracy associated with VP, the ability to change the material density for a functional grade, and the ability for this technology to produce shells.

FDM was not selected as the AM technology because of several issues. Firstly, commercially available extrusion nozzles for ceramic FDM would not be able to satisfy the

resolution and accuracy requirements for producing the small and complex geometries found in dental crowns. Smaller extrusion nozzles could be created for this application in future research. Commercially available FDM printers are built to produce parts in the traditional layer-by-layer format. Significant modifications to an FDM printer would be required in order to print with a shell format due to collision issues with the printhead and existing shells. It is possible to modify an FDM printer to allow for the print head to rotate and avoid existing geometry, but the idea was not explored as a commercially available VP printer would not have such an issue. This is due to the fact that VP utilizes a light source to cure photopolymer resin at a specific point, therefore collision between existing geometry cannot occur as the light source is able to penetrate through.

There are several AM methods which are capable of producing the resolution and accuracy required, such as: SLS, LAM, and NanoParticle Jetting, however these methods are strictly layer-based as existing shell geometry would collide with the laser head / binder jetting nozzle.

There are two different kinds of VP printers available, top-down, and bottom-up. Bottom-up printers produce parts by lifting the build platform upwards out of the resin tank and are much more popular on the market due to the ability for the machines to have smaller resin tanks. These kinds of VP printers can come in a desktop size platform with less initial material cost. Conversely, top-down VP printers have the build platform submerge into the resin tank with the laser curing towards the surface of the resin [6]. Since this research will involve printing shells upon existing shells, top-down print platforms are the only option. Otherwise, with a bottom-up VP, the existing shell will collide with the bottom of the resin tank.

Due to machine availability, the concept of this research will be proven using an FDM printer. The issues of resolution and accuracy will be resolved by increasing the scale of the crown geometry 500%. This will allow the printer to produce the complex geometry accurately.

3.2 Slicing Software

Slicing software is used to turn 3D computer models in the form of an .STL file into g-code, a language that AM printing machines use to position while printing. Commercially available slicing software used by VP machines creates g-code that follows the traditional layer-by-layer printing method. Since this research will be printed using a shell-based method, a program must be developed that can create shell-based g-code, separated by the required number of shells. This program will be discussed in the following chapter.

The slicing software to be used in this research is created by CURA for the LULZBOT series FDM printers. After uploading an .STL file, the user can change various print settings pertaining to wall thickness, support structure, infill density, and build plate adhesion. Most of these settings are unimportant for the program to gather positional coordinates, however there are some settings that must be adjusted. The most important setting to adjust is the *wall line count* setting. With this setting set to 1, the slicing software will only generate coordinates for the inner and outer most walls. Another adjustment to be made is the setting for *Infill Density*, which must be adjusted to 100% so that the final print will have no internal cavities. Other settings should be set to their recommended values. Build plate adhesion can be set to none as the g-code generated will not actually be used to print an object, although build plate adhesion must be used later as it will allow the object to easily come free from the build plate surface after printing.

After g-code is generated from the slicing software, there are many different codes within the g-code that tell the machine where to move, how fast to move, where to print, the type of feature to print, startup procedures, and finishing procedures. The most important code for the program to read is “G1” which tells the AM machine to deposit material from its position beforehand to the X and Y coordinates that follow. Examples of the two types of G1 codes are:

```
| G1 X128.969 Y163.815 E37.59749 | and | G1 F600 X157.684 Y147.823 E42.71879 |
```

Where the print speed is the number that follows F, the X coordinate and Y coordinate are the numbers that follow X and Y respectively, and the amount of material to extrude is the number that follows E. Another important code is “G0”. This code tells the machine to move to a certain position but to not extrude any material. For the purpose of this research, only X and Y values from G1 will be used, as this is the only place where material will be deposited.

Comments in g-code are created by starting a line with a semicolon and are often followed by a specific print feature. Examples of comments are: “;LAYER:0”, “;TYPE:SUPPORT”, “;TYPE:WALL-OUTER”, “;TYPE:FILL”, and “;TYPE:SKIN”. The number that follows the Layer comment is the layer that will be printed. This comment will be used by the program to determine which row in an array to put the X and Y coordinates that follow. The print feature ;TYPE:SUPPORT, is followed by G1 codes, but will not be used for finding coordinates as they are not part of the interior or exterior walls. However, this comment will be used for determining the layer at which the overhang starts in the crown geometry. If there are no more support structures being created, that layer is where the overhang starts. The X and Y coordinates of the interior and exterior walls are followed by the comment ;TYPE:WALL-OUTER. The program will scan for

this comment and start to populate arrays with the coordinates that follow. Interior and exterior walls are separated by G0 movements. The program will search for G0 values to determine when to switch between populating an interior or exterior coordinate array. Typically, the g-code will print the interior wall first because it is closer to the support feature that was printed right before. The print feature ;TYPE:FILL, uses the infill density value to print infill structures to fill in the space between the interior and exterior wall. The X and Y coordinates will not be used from this section; however, the program will scan for this line to know when there are no longer any wall coordinates since this feature will always follow that section. The print feature ;TYPE:SKIN creates infill between walls that are on, or near, the surface of the object to be printed. The infill density for this feature is typically 100% as it adds rigidity to the walls near the object's surface and improves the surface finish. ;TYPE:SKIN is considered in the same way as ;TYPE:FILL.

Each layer of g-code follows the same structure. It starts with a layer comment followed by a G0 command which tells the printer to move vertically a certain layer height distance and the starting position of that layer. Next the printer will create features in the order: support structure, outer walls, then skin/ infill (Figure 8).


```

;LAYER:X
G0
;TYPE:SUPPORT
...
...
...
;TYPE:WALL-OUTER
G0
G1 coordinates for interior wall
...
...
G0
G1 coordinates for exterior wall
...
...
;TYPE:FILL or ;TYPE:SKIN
...
...
...
;LAYER:X+1

```

Figure 8: Order of print features in g-code

3.3: Achieving Functional Grading Through Changes in Particle Density

To mimic the natural tooth structure, the crown must have a functional gradient where it becomes harder towards the outer surface. To achieve this, the intention is to change the density of leucite containing ceramic particles in the VP chamber. After each shell, more particles will be added so that the density increases. Theoretically, this should increase the hardness of each shell, however more research is required to explore if this is true.

3.4: Modifications to the crown .STL file

The given .STL file features a crown designed for a molar. The .STL file can be seen in Figure 9 - Figure 11. The molar crown is designed to fit with a chamfer type tooth preparation as seen in Figure 12 and Figure 13. During tooth preparation, most of the enamel is removed with a

burring tool, leaving behind the tooth dentin[12]. The internal geometry consists of an inwards taper and cusps. The creation of a crown designed for a prepared tooth will not be explored in this research as the focus is to create shells, therefore, the internal geometry of the crown is simplified. Future research is necessary for a crown manufactured using this method for real-world dental applications on a prepared tooth surface.

Adjustments to the .STL file were made to create a flat internal overhang and consistent base with a taper of 12 degrees as recommended[13]. The region below where the first shell would start is removed (Line A in Figure 14), and the internal structure is replaced with cylindrical geometry (Figure 15 and Figure 16). The modifications allow for the manufacture of the area where all shells are included. Future program development could allow for the printing of a support structure below this region where all shells are included for a more detailed base.

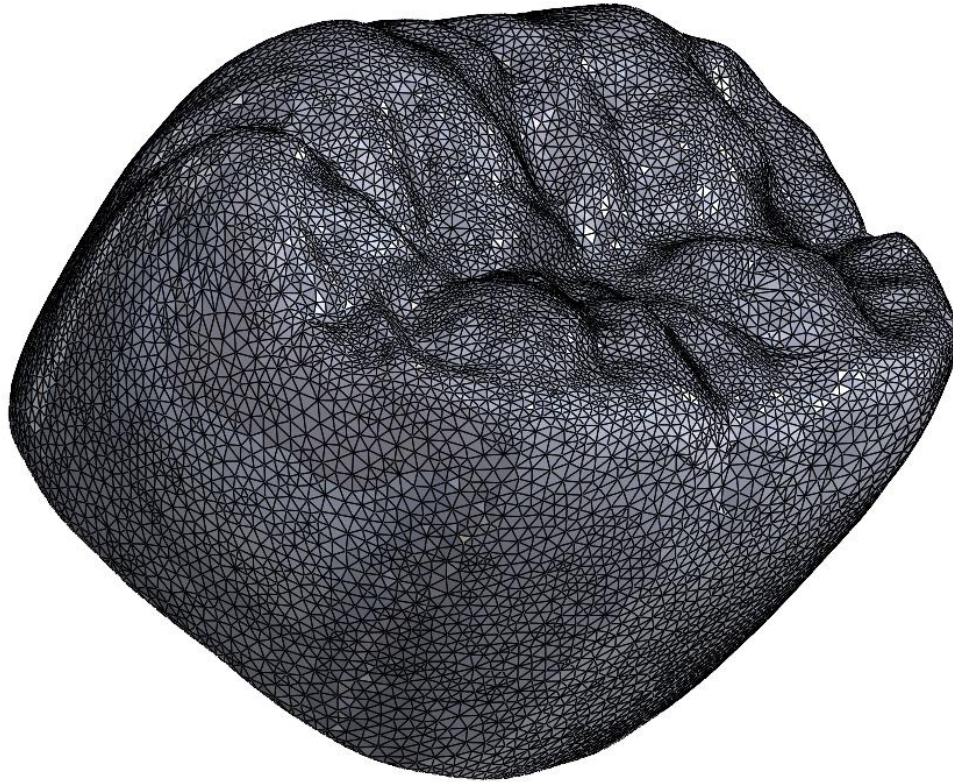


Figure 9: Isometric view of crown geometry .STL

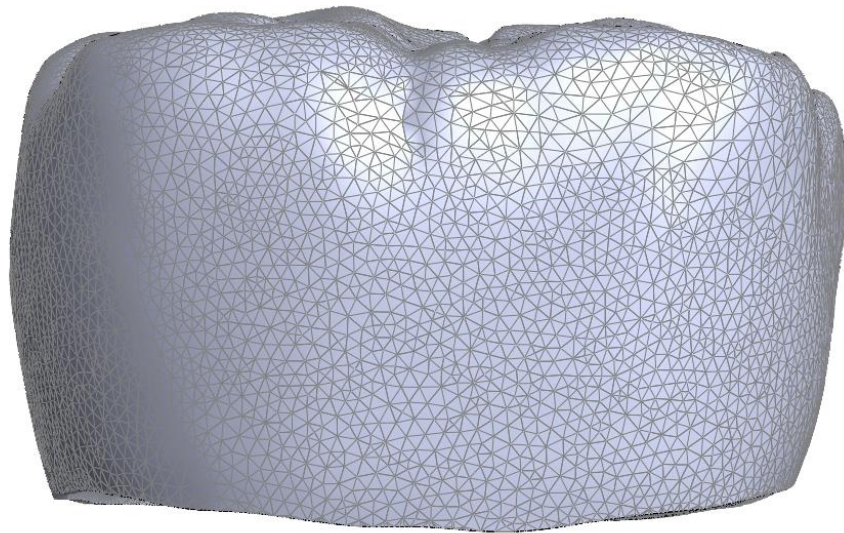


Figure 10: Front view of crown geometry .STL



Figure 11: Interior view of crown geometry .STL

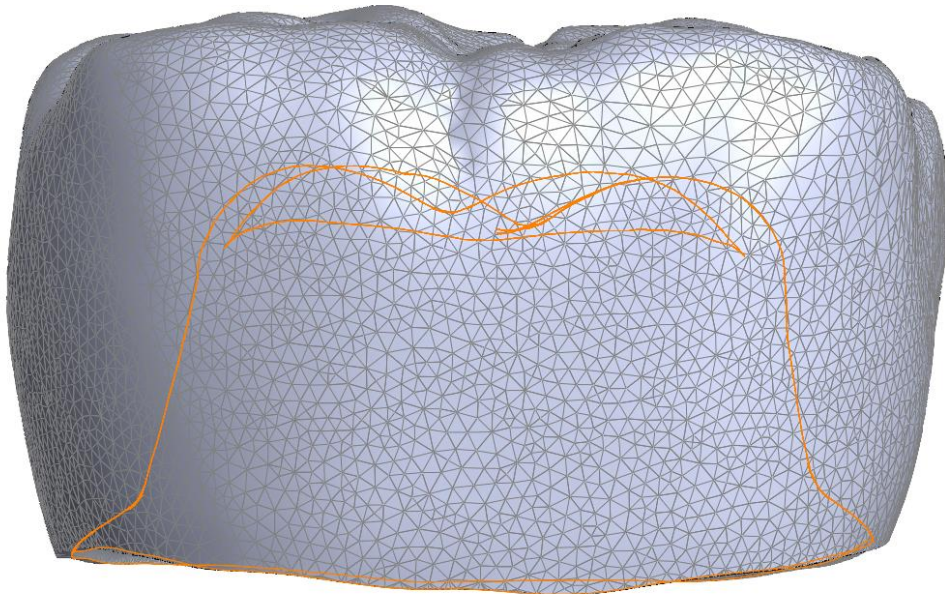


Figure 12: Tooth preparation type chamfer in orange outline

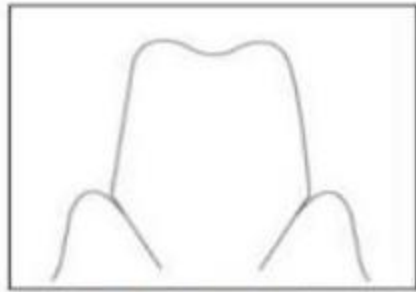


Figure 3a: Knife or feather edge

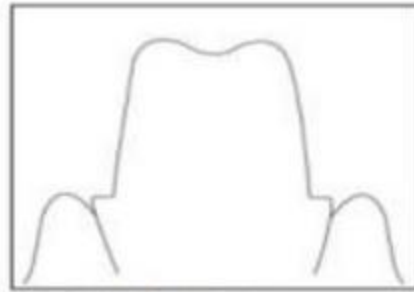


Figure 3b: Shoulder

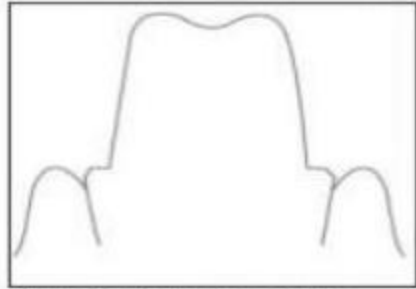


Figure 3c: Beveled shoulder

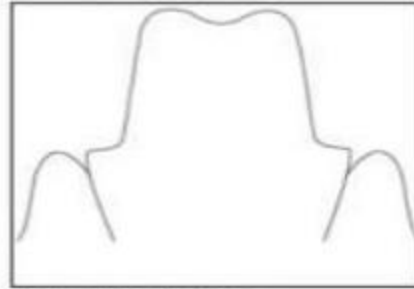


Figure 3d: Chamfer

Figure 13: Types of tooth preparation[11]

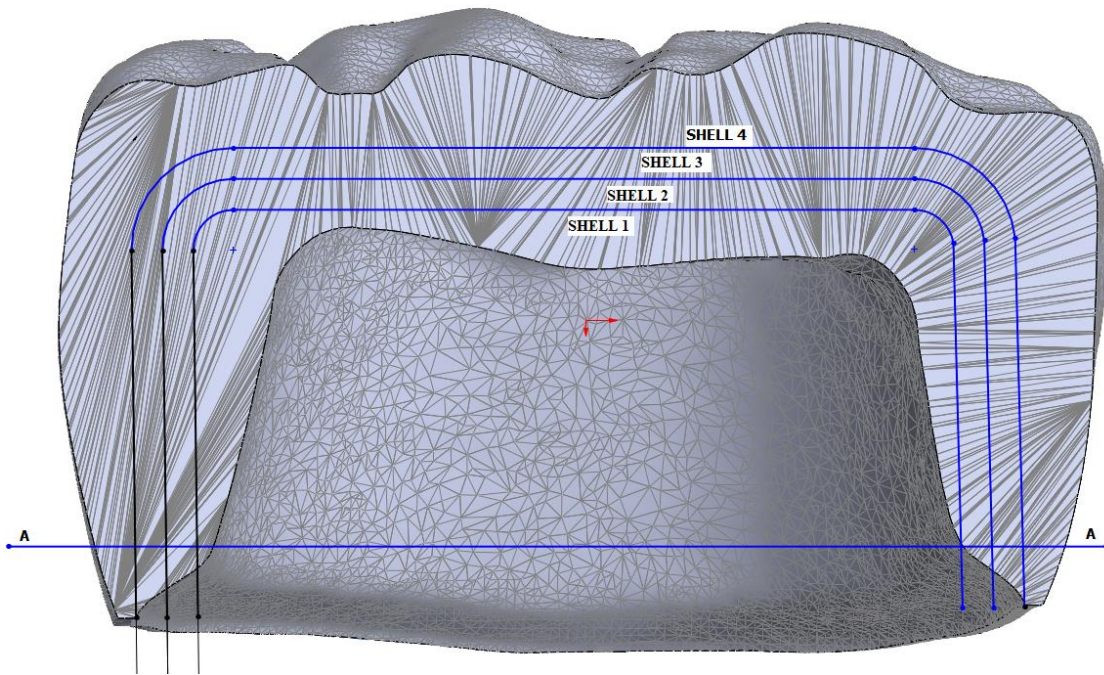


Figure 14: Shells of .STL crown geometry

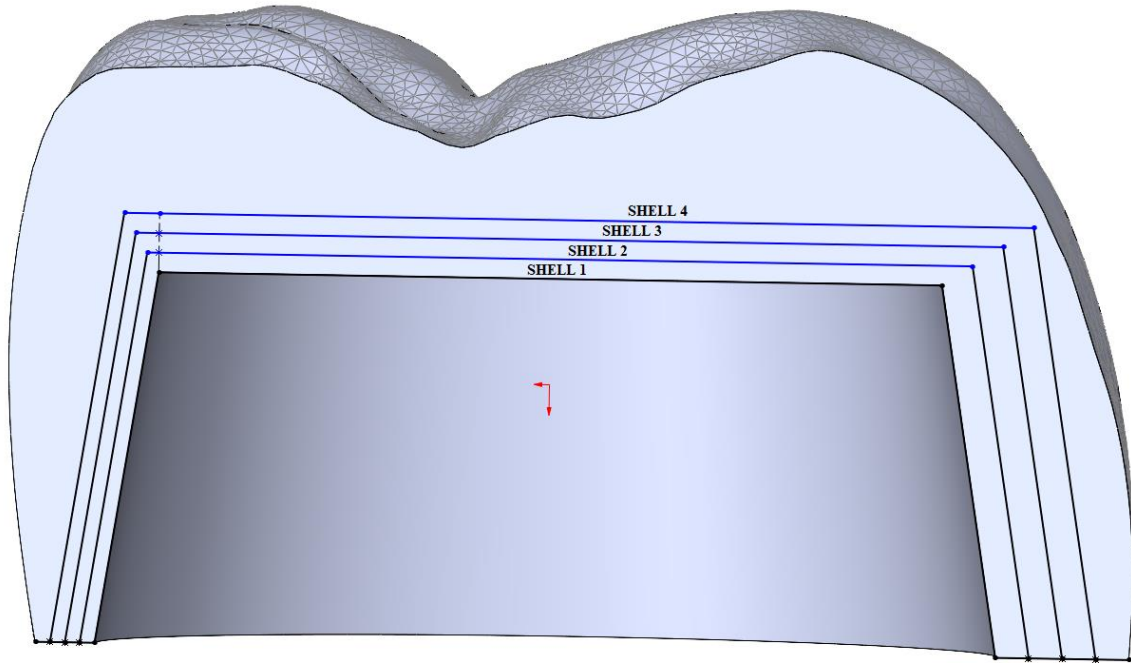


Figure 15: Section view of crown geometry after modification

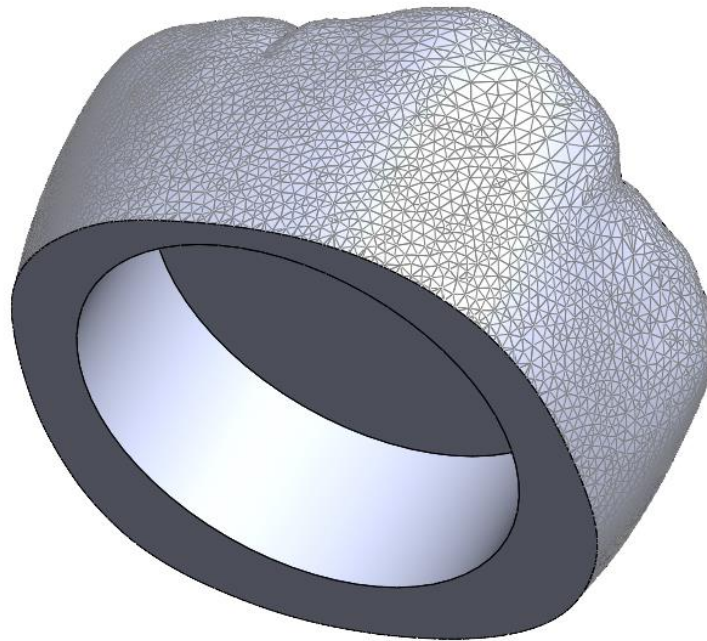


Figure 16: Isometric view of crown geometry after modification

3.5: Block Diagram of Manufacturing Process

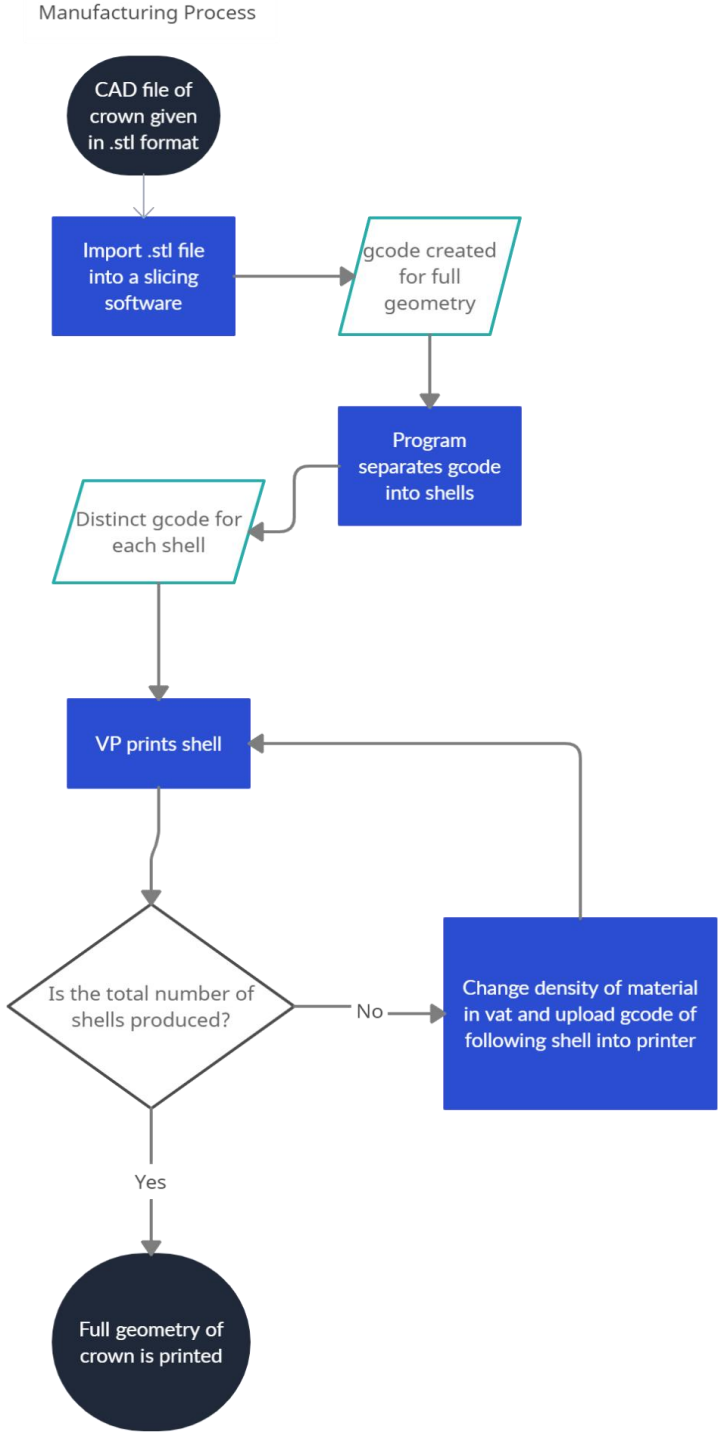


Figure 17: Block Diagram of Manufacturing Process

3.6: Adjustments made to process for printing with FDM

Due to the loss of resolution and accuracy when using an FDM printer versus a VP, the crown geometry will be increased in size by 500%. Another necessary adjustment to be made involves the elimination of exterior wall overhang for interior shells. This is due to the fact that each shell will be printed individually and need to fit inside other shells after. Interior shells must be vertically straight, or have an inwards taper, to ensure that there will not be any collision with the shells they are designed to fit into. Interior shell overhang will not be a problem for a VP process since the photocuring light can be directed anywhere inside of the vat.

CHAPTER 4 PROGRAM DEVELOPMENT

The program to be developed will utilize the g-code created by the slicing software CURA for the entirety of the crown geometry. The program will extrapolate X and Y coordinates from each layer and create new coordinates for each shell based on the extrapolated data. The following pseudocode and block diagrams show this process.

4.1: Using imported g-code to extrapolate relevant X and Y coordinate data

The first part of the program is designed to extrapolate relevant X and Y coordinates from the full g-code given by the slicing software. The program reads each line and determines if the values within are part of the WALL-OUTER feature. The program then appends two different arrays with those given values if it is determined to be part of the interior or exterior wall. When the program sees a layer change, the program will begin appending a different part of the overarching arrays. A variable, `walls_reading`, is used to determine if the line is coordinate data for the interior and exterior walls.

Due to the complexity of geometry towards the top of the crown, and the fact that it will be part of the last shell, it is not necessary for the program to create shells for this region. Therefore, the g-code of the topmost layers will be saved as is, and appended to the g-code of the last shell. The g-code from this section is saved in a separate file with the same file name but with `_ceiling` at the end. A visual representation of this area is shown in Figure 19.

1. Read line of g-code. If g-code line is ;END, quit. Else continue
2. Determine If the line contains relevant information.

- a. If the line contains a layer comment, switch to the next row in overarching interior and exterior arrays.
 - b. If the layer value is greater than the roof layer. Store line as is in g-code file_ceiling.
Go back to step 1
 - c. If the line contains a TYPE:WALL-OUTER comment, change variable walls_reading to True. Continue
 - d. If line contains TYPE:SUPPORT feature, increment ceiling variable and change variable walls_reading to false. Continue
 - e. Else If line contains any other feature comment, change variable walls_reading to False. Continue
3. If walls_reading = True, begin extracting X and Y coordinate data. Else, continue.
 - a. Store values that follow X and Y in the overarching interior or exterior array. Ignore other text for speed and extrusion values.
 - b. If G0 values are given, switch array to populate
 4. Go back to step 1.

At this point the overarching arrays have been populated with every significant X and Y coordinate in the g-code. The arrays are quite large as every row consists of each X and Y coordinate in that layer. A block diagram for this section of populating overarching arrays is shown below in Figure 18.

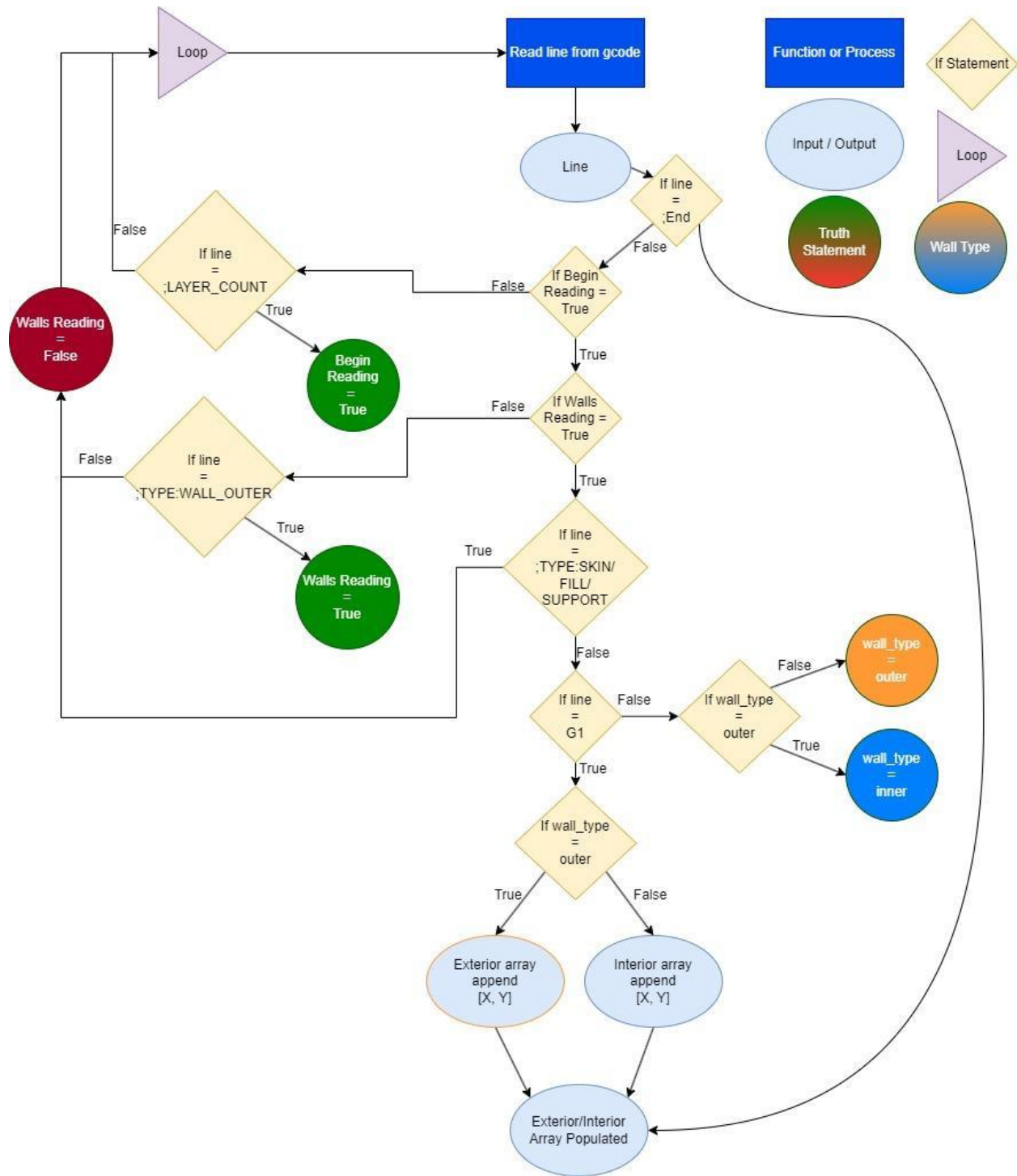


Figure 18: Block diagram for extracting X and Y coordinate values

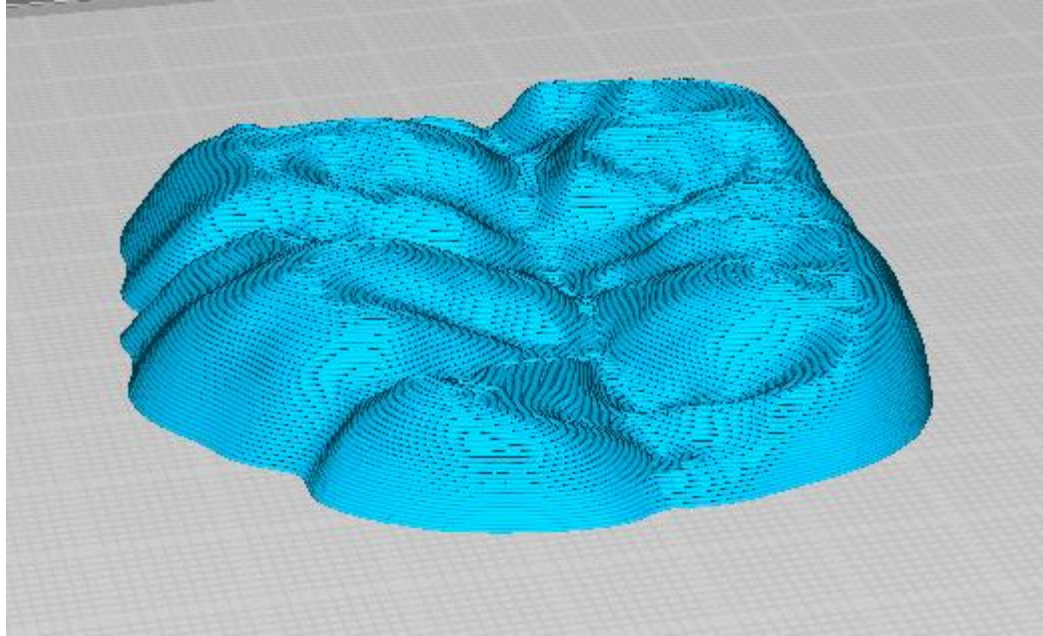


Figure 19: g-code of file_ceilng saved as is

4.2: Using ceiling and roof layers to find the start of overhang for each shell

The ceiling layer found in the previous section determines the layer in which there are no longer any interior walls. To ensure that the following shells fit into existing ones, the program uses the initial interior wall coordinates to find X and Y coordinates for each shell. To determine which layer the overhang starts on each shell, the program takes the ceiling layer and the roof layer, and divides this region dubbed “the attic” by the number of shells. The layers in which to start the overhang geometry in each shell is stored in an array.

The roof layer (Figure 22) is where the final shell will copy and paste existing g-code into the final shell g-code. This layer can be determined by using the slicing software to show a visual representation of g-code. The roof layer is a user input, and it is the layer in which complex crown geometry can be seen (Figure 20 and Figure 21).

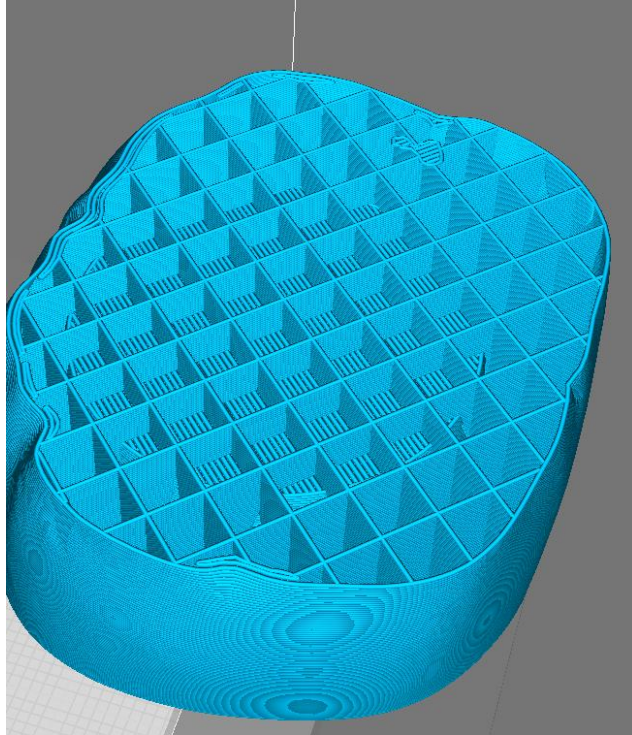


Figure 20: Complex geometry is starting to print on layer 89

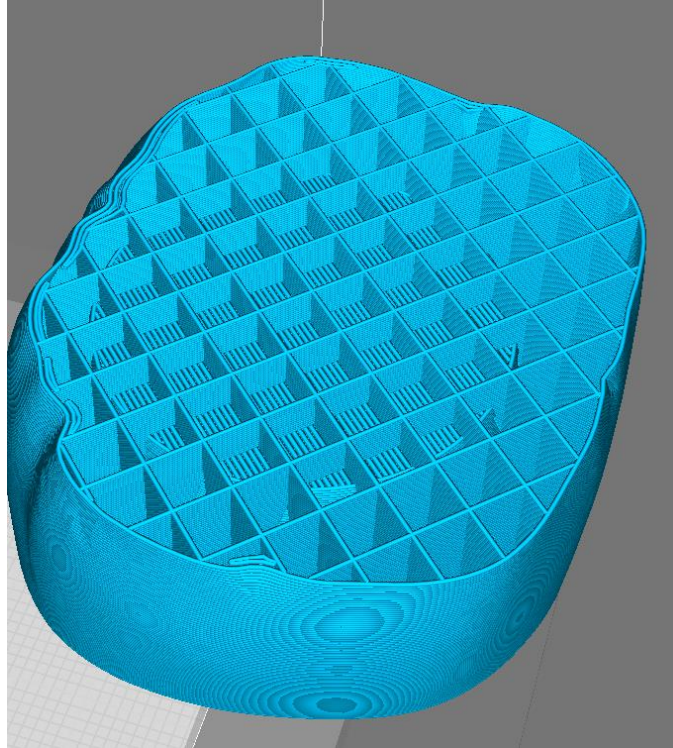


Figure 21: No complex geometry found on layer 88

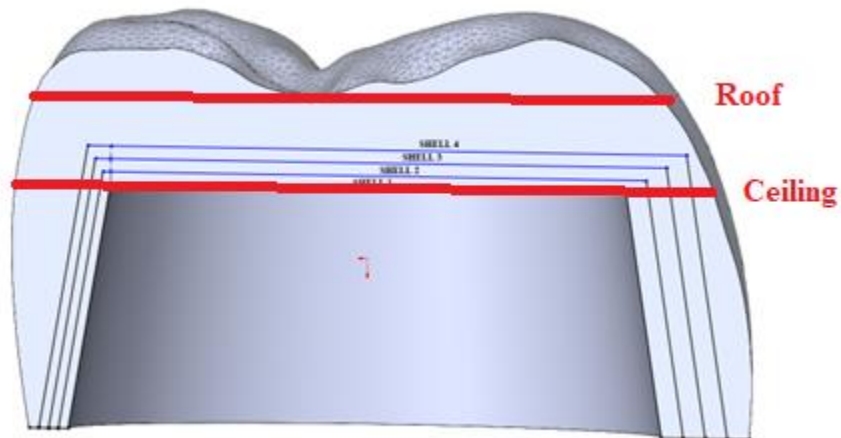


Figure 22: Roof and ceiling layers

4.3: Manipulating X and Y coordinates to create new interior shell coordinates

Due to fitting limitations with FDM, the interior shells must not have exterior overhang. Therefore, the interior shell coordinates will be created using the X and Y wall coordinates for the first layer. This section creates a new shell coordinate at a certain distance between two corresponding interior and exterior wall coordinates. This distance is found by taking the total distance between the two points and dividing that distance by the number of shells. Interior wall coordinates for future layers are calculated by using the 12 degree taper and the layer height.

4.3.1: Corresponding exterior points to interior points

The first part of this section involves using a function that corresponds an interior point to every exterior point based on the slope of both points to the centroid of the exterior wall. Essentially, a line is drawn from the centroid of the exterior wall to each point of the exterior. A line is also drawn from the same centroid to each interior point in the same region. Whichever interior point line most closely matches the slope of the exterior point line will be corresponded to that exterior point. The result of this function is the creation of an array filled with interior points the same size as the exterior array, and array indices that correspond. The pseudocode and block diagram for this function are shown below in text and in Figure 24 respectively.

1. Find centroid of exterior wall points
2. For each exterior wall point do the following:
 - a. Calculate the slope of a line that connects the point to the centroid.
 - b. Find which quadrant the point is in.
3. For each interior wall point do the following:
 - a. Find which quadrant the point is in.

- b. If the quadrant is in the same region* as the exterior. Calculate the slope of a line that connects the point to the centroid. Else go to step 3.
 - c. Compare the slope of the interior point line to the exterior point line.
 - d. Test to see If the slope matches closer than any elements tested so far. If it is, remember it. If it is not, forget it. Go to step 3
4. Take the best interior element found above and save it in an array with the same index as the exterior element. Go to step 2

* To reduce computing time, a point is only considered for finding its slope if it is in the same general region as the exterior point in respect to the centroid. For example, if an exterior point is near the x axis, with a slope less than one, only interior points in quadrant I and IV will be considered. Quadrant identification follows the standard quadrant naming convention as seen in Figure 23.

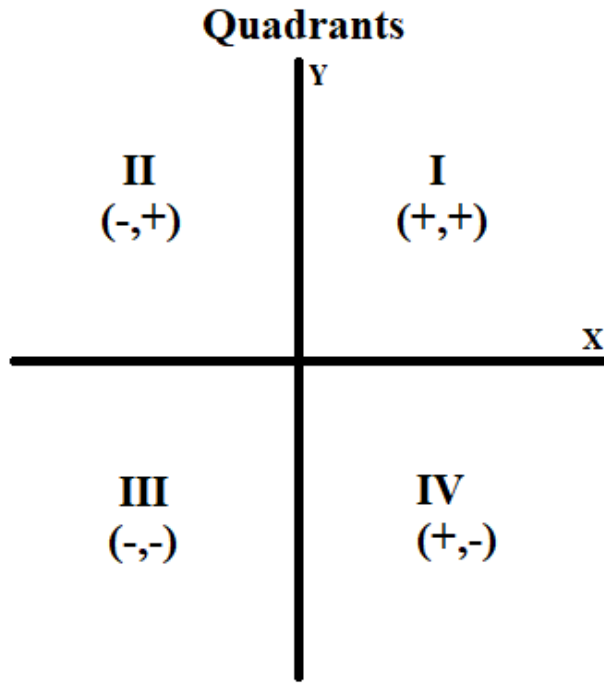


Figure 23: Quadrant naming convention

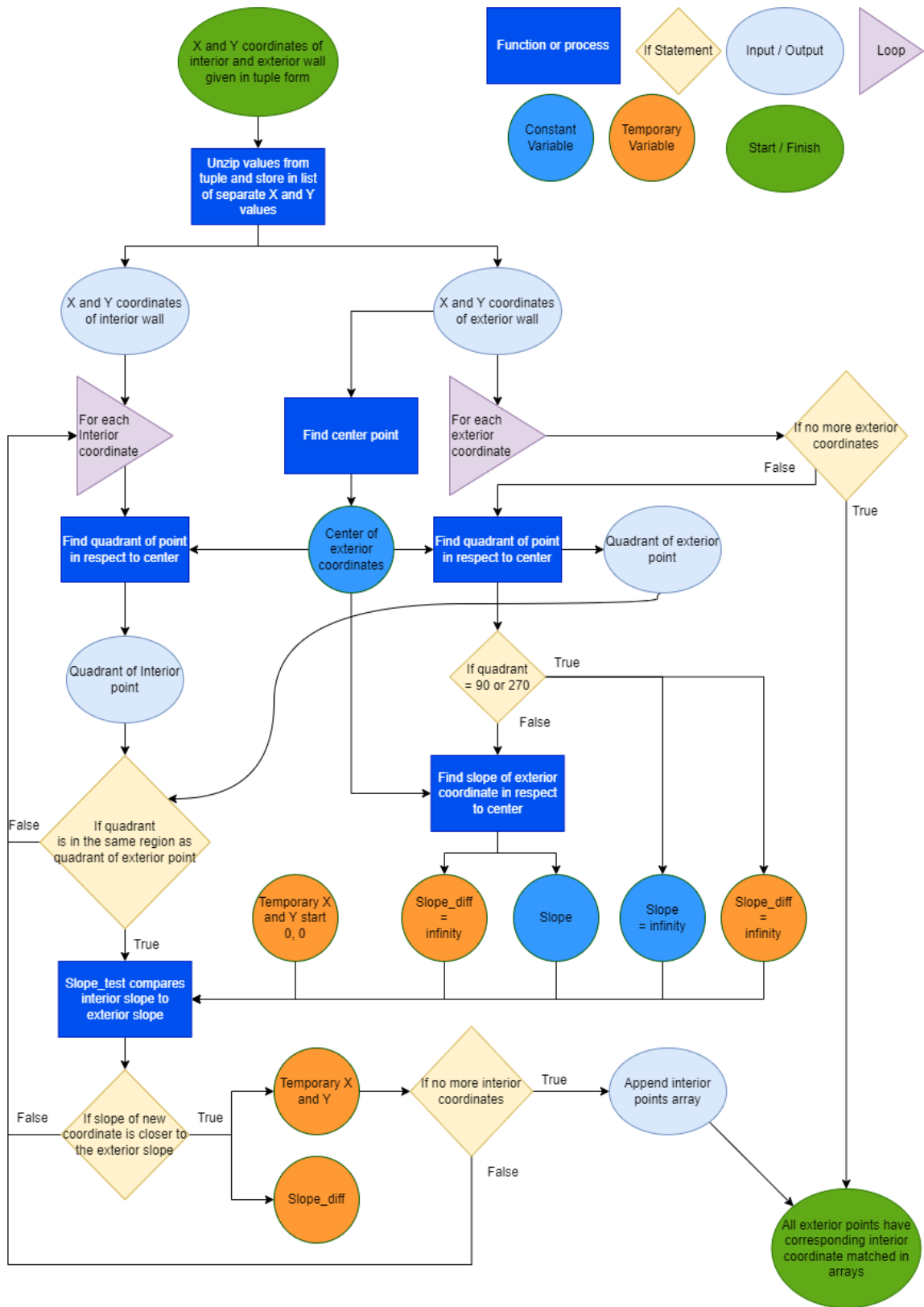


Figure 24: Block diagram of interior points function

4.3.2: Creating new shell points

With corresponding interior and exterior points, the program takes the distance between each point and divides it by the total number of shells. A function then creates a new shell point between the two, at each distance division.

This function, `quad_change`, takes the centroid of the coordinates you want to change, the specific point you want to change, the coordinates of the point you want it to move in relation to, the distance that it will move, and the direction to move, either towards or away from the relation point. Using trigonometric equations, the program identifies how far the X and Y coordinate must be adjusted, and using the point's quadrant location either adds or subtracts that distance from the original value. After the creation of new shell points, new arrays for X and Y coordinates of each shell are appended with these values. The pseudocode of this function is shown below.

1. Determine which quadrant the point is located in, in relation to the centroid
 - a. If the first point X value is equal to the second point X value or the centroid X value. The point is at 90 or 270 degrees
 - i. Add or subtract the distance from the Y value
 - b. If the first point Y value is equal to the second point Y value or the centroid Y value. The point is at 0 or 180 degrees
 - i. Add or subtract the distance from the X value
 - c. Else Continue
2. Calculate the difference in X values between the first and second point.
3. Calculate the difference in Y values between the first and second point.
4. Calculate the angle of a right triangle given the X and Y difference using inverse tan.

5. Using the total distance to move as the hypotenuse of a right triangle. Calculate the distance to change the X and Y value cosine and sin respectively.
6. Add or subtract the distance to change X and Y values depending on the quadrant.
7. Return the newly created point

4.4: Creating new g-code files for each shell

The shell coordinates found in the previous section will be used to fill a g-code file for each shell. The created g-code will follow the same structure as the original, where first the support structure is created, then the interior wall, then the exterior wall, then the infill. Build plate adhesion will also be added for the first layer. Fit arrays will also be created that utilize the `quadrant_change` function to reduce the exterior wall of interior shells by the nozzle diameter. This will allow interior shells to fit inside of each other.

4.4.1: Initializing g-code

Copying the g-code from the slicing software, new g-code files are created for each of the shells. This portion of g-code includes information for the printer to use for startup procedures like the nozzle and bed temperature, bed leveling sequences, and positioning modes. The only thing that is adjusted from this section compared to the slicing software is the `LAYER_COUNT` comment, which will be set to the top layer value of each shell.

4.4.2: Keeping track of extrusion and height values

The extrusion, E, and height, Z, values which were ignored from the slicing software g-code must be utilized in the creation of new g-code. These values tell the FDM machine how much

filament to extrude for any given distance to move, therefore, variables for E and Z must be adjusted throughout the shell g-code. The ratio of extrusion to movement was calculated using values from the original g-code (Figure 25). This ratio is fairly consistent and is different depending on the speed at which the printer moves. A variable for each shell is adjusted during the g-code and a function is used to add to this value during printing. This function, find_e, takes the distance between two points and divides that value by the speed dependent ratio.

F = 600	x	y	E	distance	dif in E	ratio	
first	48.276	67.534	7.42457				
second	51.503	66.906	7.52586	3.287539	0.10129	32.4567	
third	54.622	65.87	7.62711	3.286557	0.10125	32.45982	
fourth	57.583	64.444	7.72836	3.286487	0.10125	32.45913	
fifth	60.338	62.652	7.82961	3.286531	0.10125	32.45957	
six	62.84	60.521	7.93086	3.286513	0.10125	32.45938	
						32.45892	ave
F = 2700	x	y	E	distance	dif in E	ratio	
first	151.194	153.506	186.9719				
second	152.428	152.243	187.0042	1.765765	0.0323	54.66764	
third	153.534	150.865	187.0365	1.766952	0.03232	54.67055	
fourth	154.503	149.388	187.0688	1.766491	0.03231	54.67319	
fifth	155.324	147.823	187.1011	1.767276	0.03233	54.66367	
six	155.99	146.188	187.1334	1.765441	0.03229	54.67454	
						54.66992	ave

Figure 25: ratio of extrusion to movement

4.4.3: Writing g-code from X and Y coordinates

X and Y coordinates must be changed back into g-code for the printer to read. G-code lines for movement and extrusion must include: a G0 or G1 at the start of the line, a speed value, if it is changed, starting with the letter F, X coordinate starting with the letter X, Y coordinate starting

with the letter Y, and an extrusion value starting with the letter E. The function, `gcode_write`, takes the two arrays of X and Y coordinates and writes a line of g-code into a given shell file. The first coordinates become a G0 movement with no extrusion. The following coordinates are G1 extruding movements where the function `find_e` is utilized to adjust the extrusion value. For most circumstances, the intention is to print a complete wall. In this case, an additional G1 line is written to the first coordinate of the arrays.

4.4.4: Writing g-code for build plate adhesion

Build plate adhesion is necessary for an FDM printer because it allows the part to be easily removed from the bed after printing. It also gives the printer time to extrude filament before printing the actual object. There are several types of build plate adhesion, but the skirt type will be used for this research. This kind of build plate adhesion involves printing several one-layer thick walls around the perimeter of the object. These walls follow the contour of the exterior wall but are a set distance away from the object.

For the first layer of each shell, the program uses a function to write the g-code for build plate adhesion. The function takes the X and Y coordinates of the exterior wall and utilizes the `quadrant_change` function to create new coordinates 5 units away from the original. The value of 5 is a user input that is based on the original g-code when build plate adhesion is selected. After the build coordinates are created, the `gcode_write` function opens the shell file and writes G0 and G1 movement. The build plate adhesion can be seen in the three perimeter walls of Figure 26.

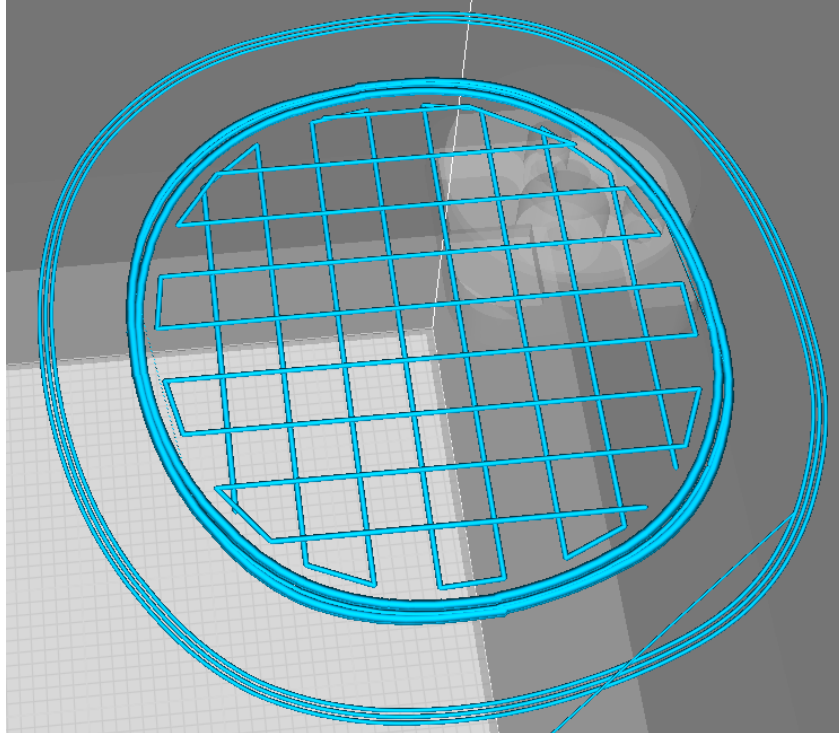


Figure 26: Build plate adhesion of shell3

4.4.5: Writing g-code for support structure

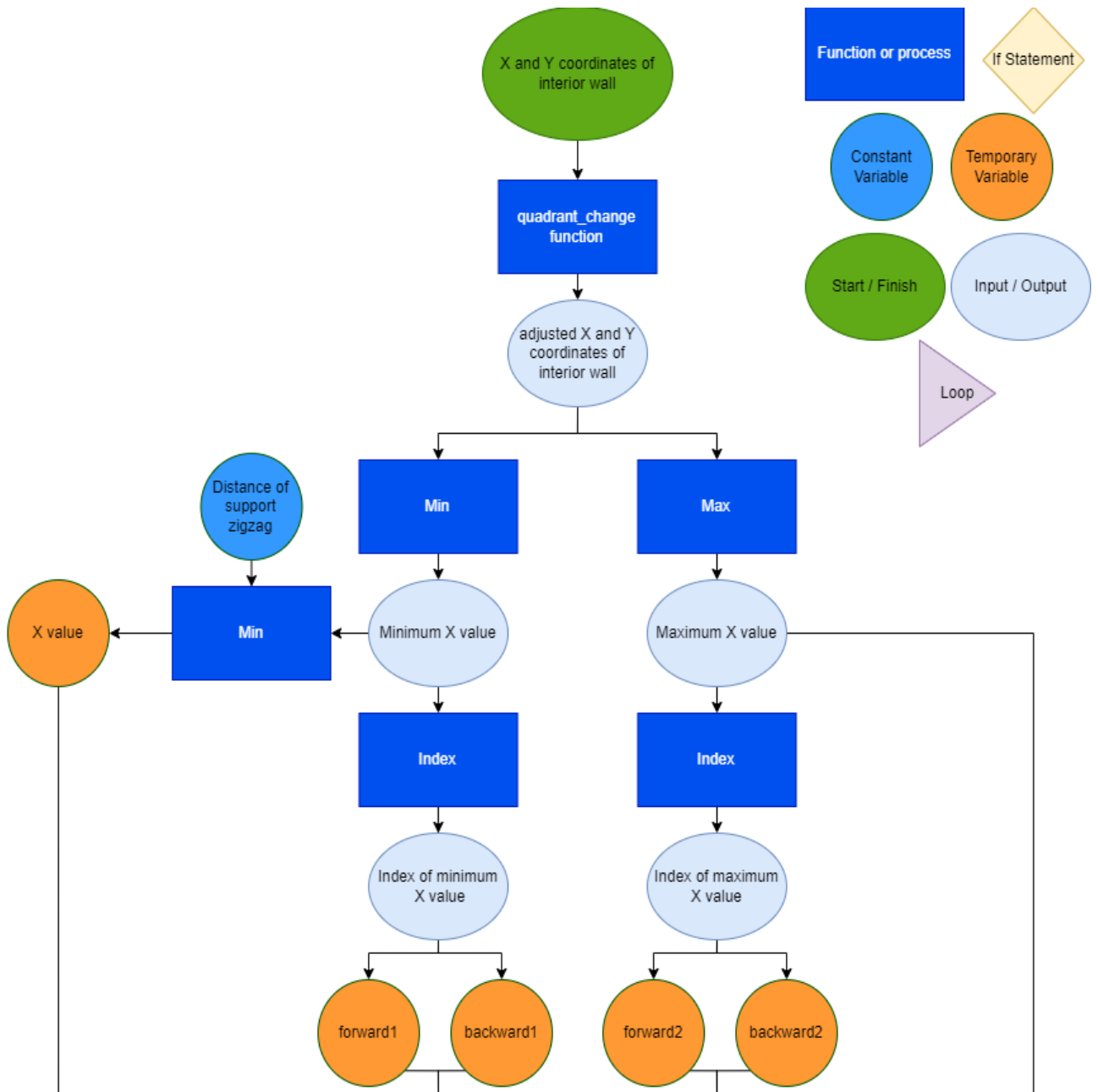
A support structure must be used in FDM when printing objects with an overhang. For each shell, the support structure starts at layer one and ends when the ceiling layer for that shell starts. There are many types of support structures, but the simplest and most widely used kind is called zigzag. The function `infill_zig` creates the points for printing the support structure, and `gcode_write` turns those coordinates into lines of g-code.

The function `infill_zig` takes X and Y coordinate data from the interior wall and creates a zigzag pattern that changes orientation based on the layer (also seen in Figure 26). Since the supporting structure must be easily removed, it should not be directly connected to the object. The function utilizes the `quadrant_change` function to create new points away from the interior wall.

Depending on the orientation, the lines it creates will be horizontal or vertical, connected with lines that are determined by where the horizontal or vertical lines intercept the newly created interior wall array. The pseudocode and block diagram for this function are shown below in text and in Figure 27 respectively.

1. Perform quadrant_change function to create new array of interior wall coordinates
2. Determine which orientation of zig zag to create
 - a. If the layer is an even number. Horizontal lines
 - b. If the layer is an odd number. Vertical lines
3. If horizontal lines do the following: Else Continue
 - a. Calculate the minimum value of array
 - b. Calculate the maximum value of array
 - c. Find the index of maximum value in the array. Save as forward1 and backward1
 - d. Find the index of minimum value in the array. Save as forward2 and backward2
 - e. Set X value equal to minimum value plus support distance
 - f. For each value in array do the following:
 - i. Compare the value of array at forward1 index to X value. If it is greater than X. decrement forward1. Go to 3e. Else Continue
 - ii. Compare the value of array at maximum index to X value. If it is less than X. increment forward2. Go to 3e. Else Continue
 - g. Create new point based on the X value and the equation of a line between forward1 and forward2
 - h. For each value in array do the following:

- i. Compare the value of array at forward1 index to X value. If it is greater than X. decrement backward1. Go to 3e. Else Continue
 - ii. Compare the value of array at maximum index to X value. If it is less than X. increment backward2. Go to 3e. Else Continue
 - i. Create new point based on the X value and the equation of a line between backward1 and backward2
 - j. If X value plus support distance is greater than maximum value of array. Perform function write_g-code. Else add support distance to X value. Reset forward1,2 and backward1,2. Go to 3f
4. If vertical lines do the following:
 - a. Same as 3 except with Y values instead of X



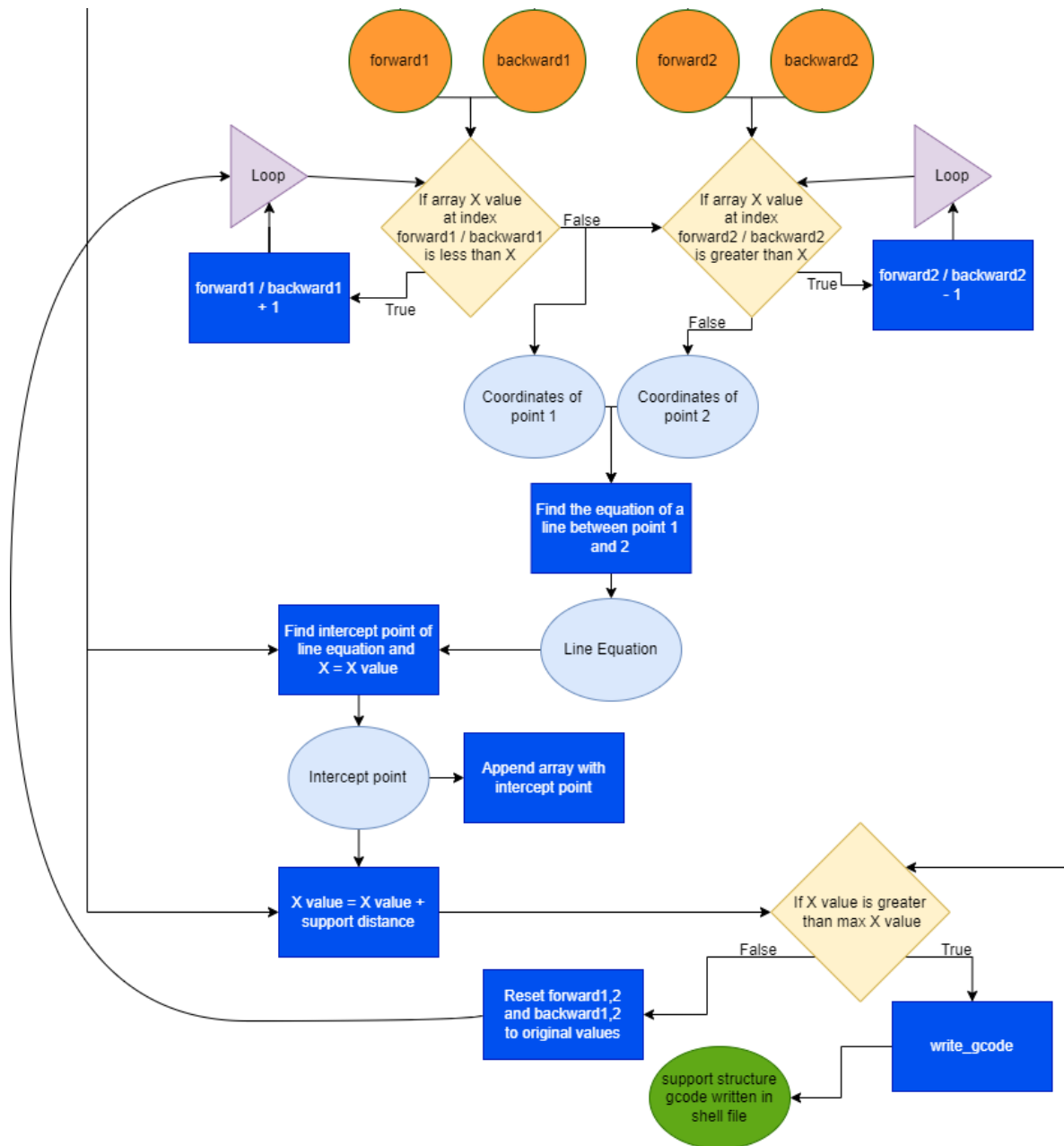


Figure 27: Block diagram of infill_zig for even layers

4.4.6: Writing g-code for interior and exterior walls

The function `gcode_shell` uses X and Y coordinates from the shell and fit arrays to write the g-code for interior shells. For the last shell, the function will use the unchanged exterior wall instead of a fit array. The function simply calls on the `gcode_write` function to write the given

coordinates to the shell's g-code file. It will also increase the Z height for layers where there is no support structure.

4.4.7: Writing g-code for infill

The infill structure is used to fill the space between interior and exterior walls. There are many types of infill structures but the kind that will be used in this research is called concentric. This type of infill involves filling the space with concentric circles starting from the interior wall and going outward. This method will be used because the object to be printed is circular in nature, would work well with VP, and because it will require the use of functions that have already been created.

The function, `gcode_infill`, takes the interior and exterior X and Y coordinate arrays as inputs. It calculates the greatest distance between the two walls and divides this value by the nozzle diameter to find the amount of concentric walls needed to fill the space. It performs a `quadrant_change` based on the interior wall for each of the concentric walls needed and removes any of the points that are created past the exterior wall. The `gcode_write` function writes these points into the shell file as they are being created. The pseudocode and block diagram for this function are shown below in text and in Figure 28 respectively.

1. For each point in the arrays do the following:
 - a. Calculate the distance between the interior and exterior point.
 - b. If the distance is greater than any of the elements tested so far. Remember the distance. Else forget it

2. Take the greatest distance found above and divide the value by the nozzle diameter. This value corresponds to the number of concentric interior walls that are necessary to fill the space between the interior and exterior walls
3. For each instance of necessary concentric walls do the following:
 - a. For each point in the interior wall do the following:
 - i. Perform the `quadrant_change` function to find a point between the interior and exterior wall based on the centroid at a distance of the nozzle diameter
 - ii. Calculate the distance between the centroid and the new point
 - iii. Calculate the distance between the centroid and the exterior wall point
 - iv. If the distance for the exterior wall point is greater than the new point. Append a temporary array with the X and Y coordinates Go to 3a. Else If the temporary array has values, perform `gcode_write` function and clear the temporary array of all values Go to 3a. Else go to 3a
 - b. Perform the `gcode_write` function to write any remaining temporary values into the shell g-code. Go to 3a with an increased nozzle diameter distance

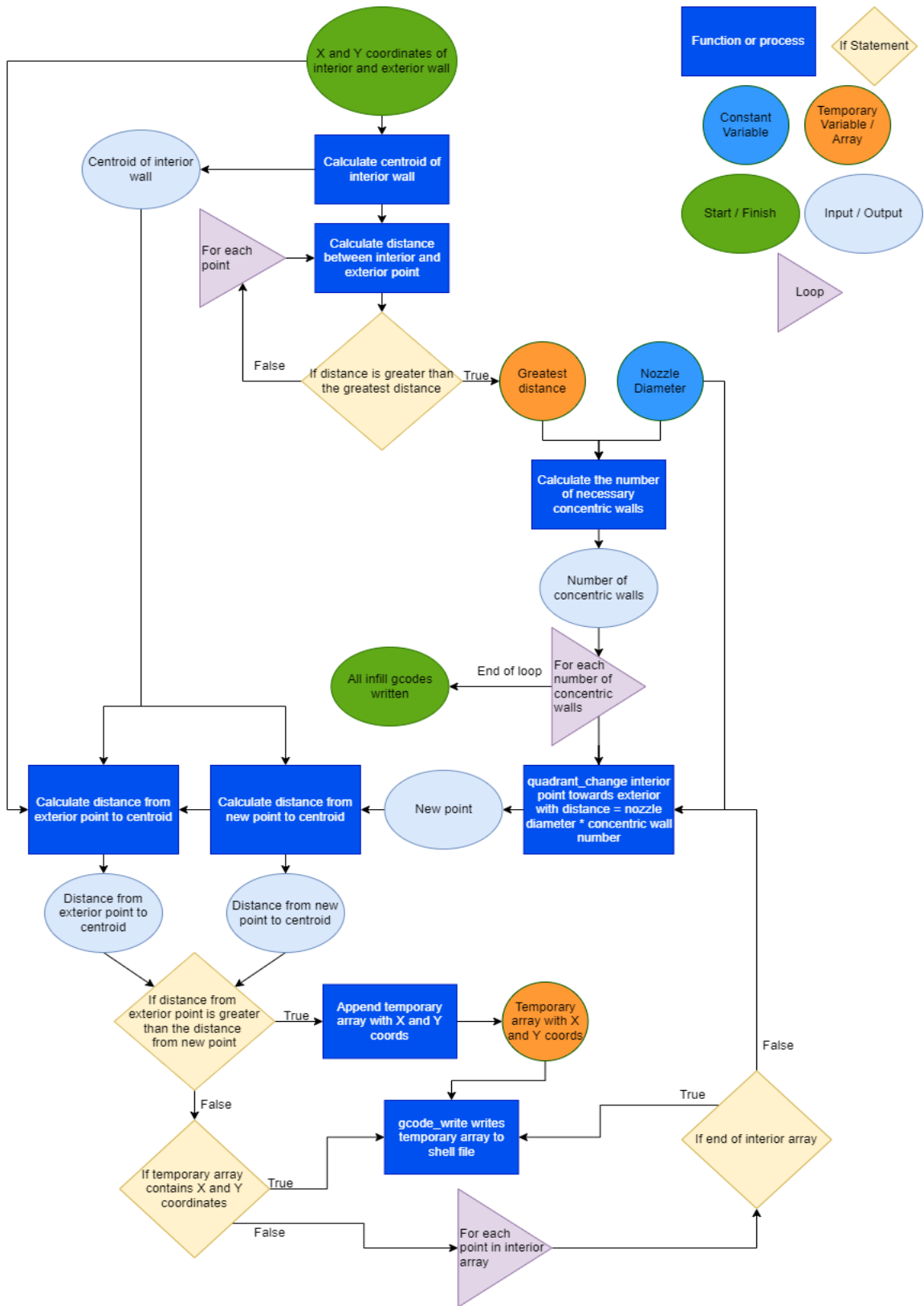


Figure 28: Block diagram of gcode_infill

4.4.8: Writing g-code in region above shell ceiling

The region above the ceiling for each shell must be infilled, however the type of infill in this region will be different from the concentric infill used earlier. From this layer onwards, the type of infill will be a zigzag infill as top layers will be more resistant to compressive forces applied from above, as well as having a better surface finish. To write the g-code for this region, the function `gcode_shell` is utilized to print the outer wall, and the `infill_zig` function is used to fill in the space between. In this case, the distance to infill will be the nozzle diameter. The printed gcode is visually represented in Figure 29.

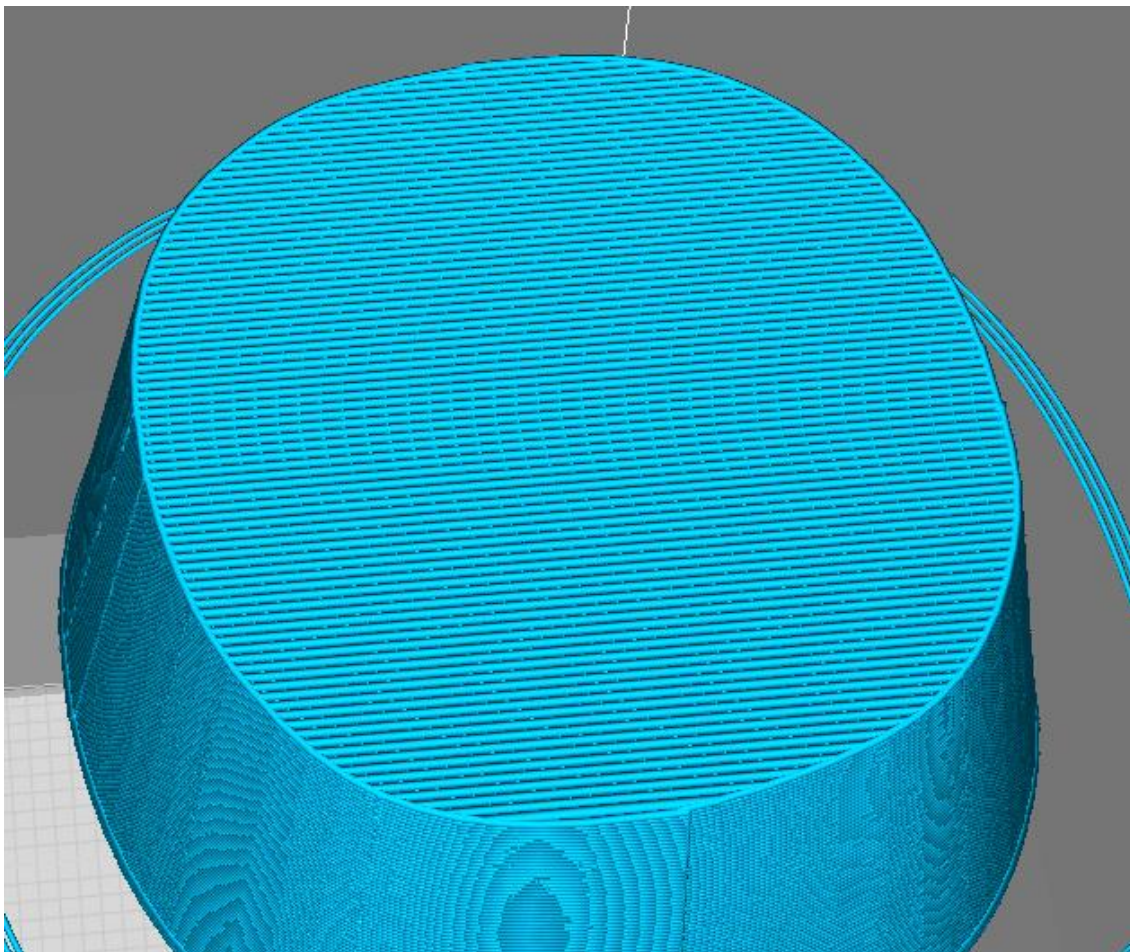


Figure 29: Result of `gcode_shell` and `infill_zig` in regions above the shell ceiling

4.4.9: Finalizing g-code

At the end of g-code there are several commands that tell the machine to cool down, release nozzle pressure, and move the bed to allow for easy removal of the object. These final g-code commands are appended to all but the last shell g-code, copied from the original slicing software. The last shell will be appended with the final layers of complex geometry g-code which includes these final commands.

When the final shell reaches the layer at which to append the `_ceiling` g-code, the function `gcode_roof` is performed. This function is necessary because the extrusion values from the original slicing software are incorrect for the g-code that was created. This function reads each line of the `_ceiling` g-code file and replaces the E value of G1 movements with the correct value from previous layers of g-code.

4.6: Block diagram of full program

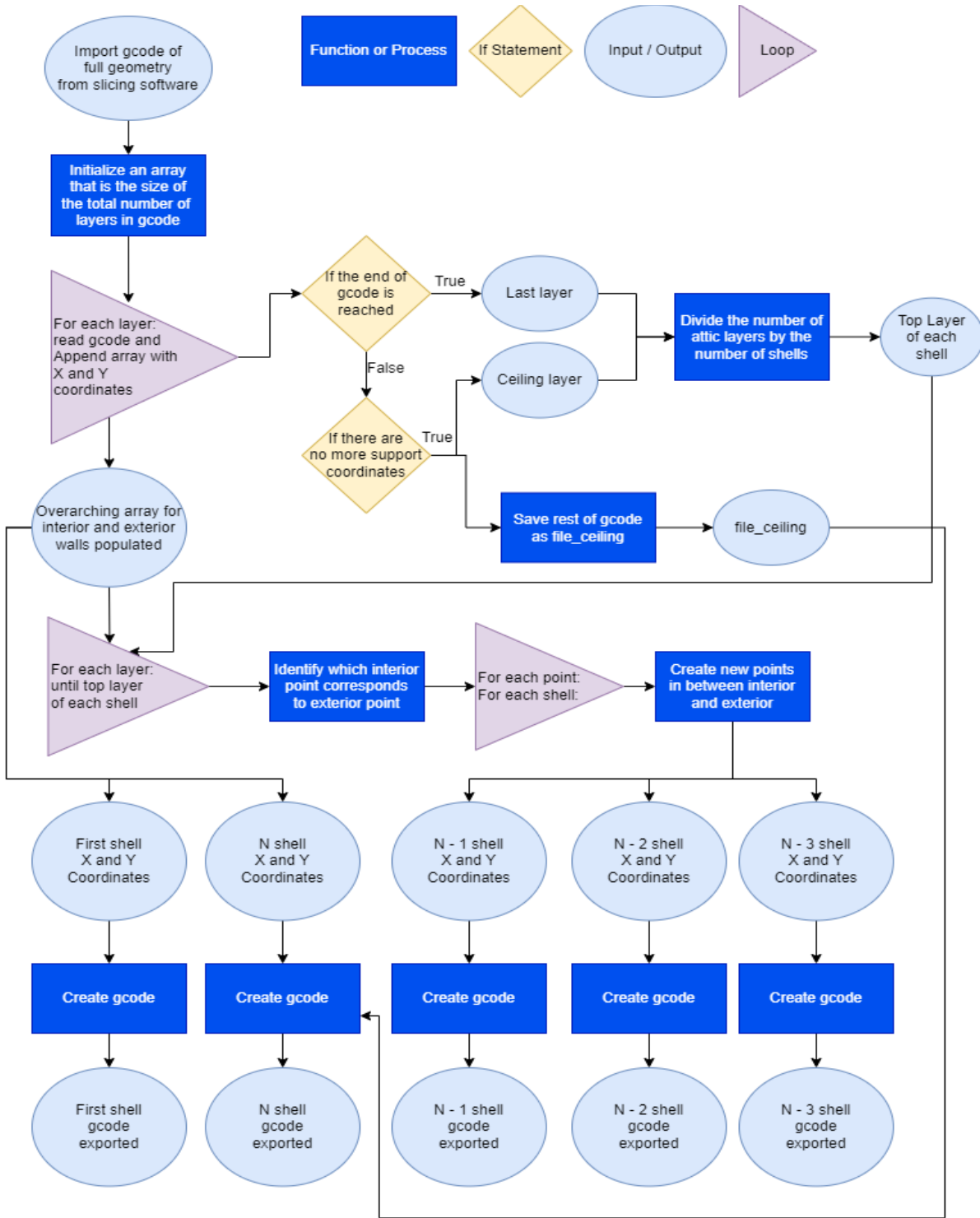


Figure 30: Block diagram of full program

CHAPTER 5 PRINTING PROCEDURE

5.1: FDM printer details

The concept of this research will be proven using an FDM printer. The FDM printer to be used is a LulzBot TAZ 6 using CHROMA STRAND ABS filament. The nozzle diameter of this machine is 0.5 mm. Figure 31 shows the printer in the lab setting.



Figure 31: LULZBOT TAZ 6

5.2: Slicing software settings

Slicing is performed using CURA Lulzbot Edition version 3.6.25. The .STL file of the crown after modification is opened and the geometry is scaled 500%. The standard layer height is 0.25 mm. Other settings used are shown in Appendix A.

5.3: Modifications to manufacturing process to accommodate FDM

Due to the lower resolution and accuracy of this machine, the crown geometry was scaled 500% to allow for small distance and complex geometry to be shown. To eliminate interior shell exterior wall overhang, which would cause fit problems after printing, The X and Y coordinate data for interior shells was kept consistent, resulting in a constant taper for interior shells. The print speed was lowered to 600 for the first three layers of the print. This was done to ensure that the first walls would be printed accurately, as experimental print tests showed that the layers contacting the build platform became warped and uneven if not printed at a slower speed.

5.4: Post processing

Removal of support was performed with a small screwdriver (Figure 32). Figure 33 shows the part after support removal. Sanding was performed with a DREMEL 3000 rotary tool with the DREMEL sanding band accessory, product number 432 (120 grit) on speed setting 8, and finish sanding with 240 grit sandpaper. The interior ceiling support was grinded away with a DREMEL aluminum oxide grinding stone, product number 953 on speed setting 8. Figure 35 shows the interior ceiling after grinding away the support structure. Figure 34 shows the DREMEL rotary tool and accessories. Sanding was first performed vertically along the slope of the part until layers were no longer visible as shown in Figure 36. Afterwards, sanding was performed along the width of the part with respect to the taper to remove the build up of filament towards the bottom of the

part where the print speed was lowered to ensure print wall accuracy. Figure 37 shows this sanding orientation and Figure 38 shows the buildup of filament due to a lower print speed. Figure 39 shows before and after post processing for two parts printed using the same g-code.

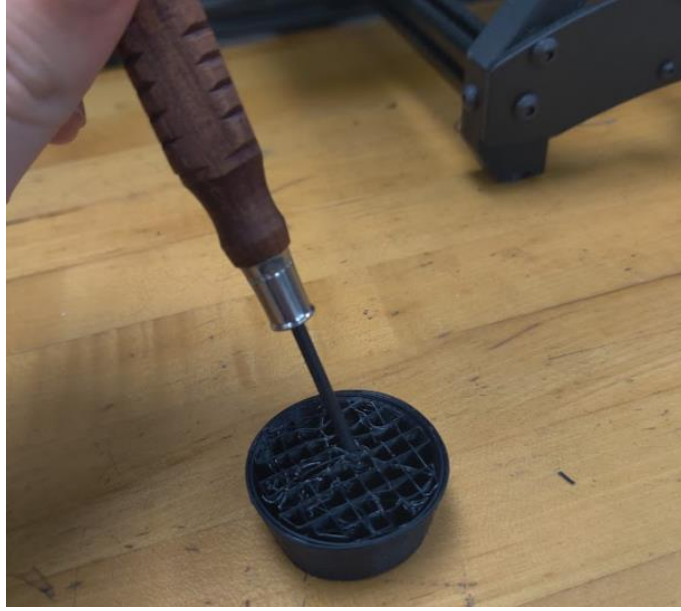


Figure 32: Support removal with small screwdriver

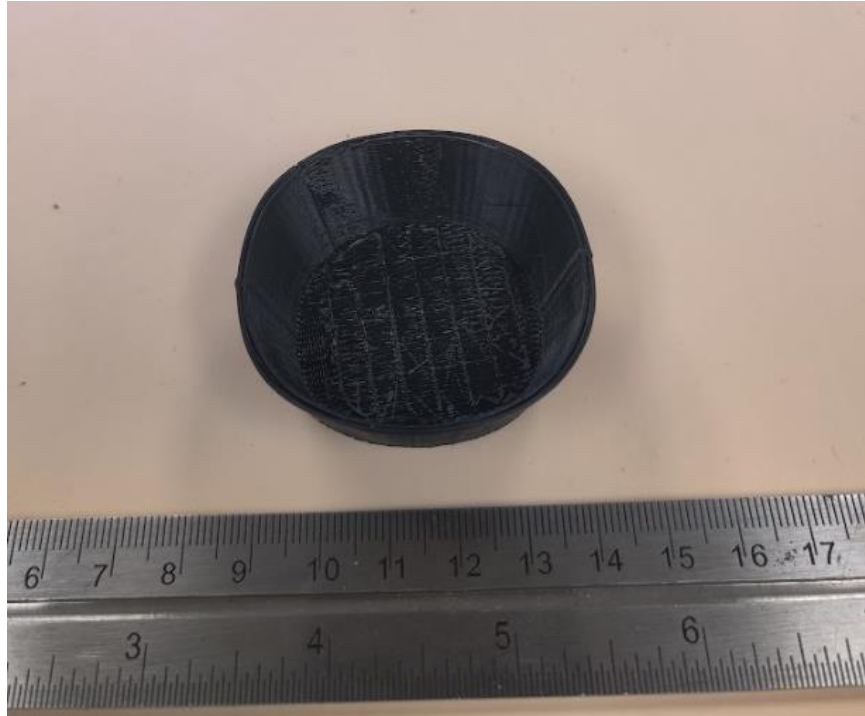


Figure 33: Printed part after support removal



Figure 34: DREMEL with sanding and polishing accessories



Figure 35: Interior ceiling after grinding away support structure



Figure 36: Sanding vertically along the taper of the part until layers are no longer visible

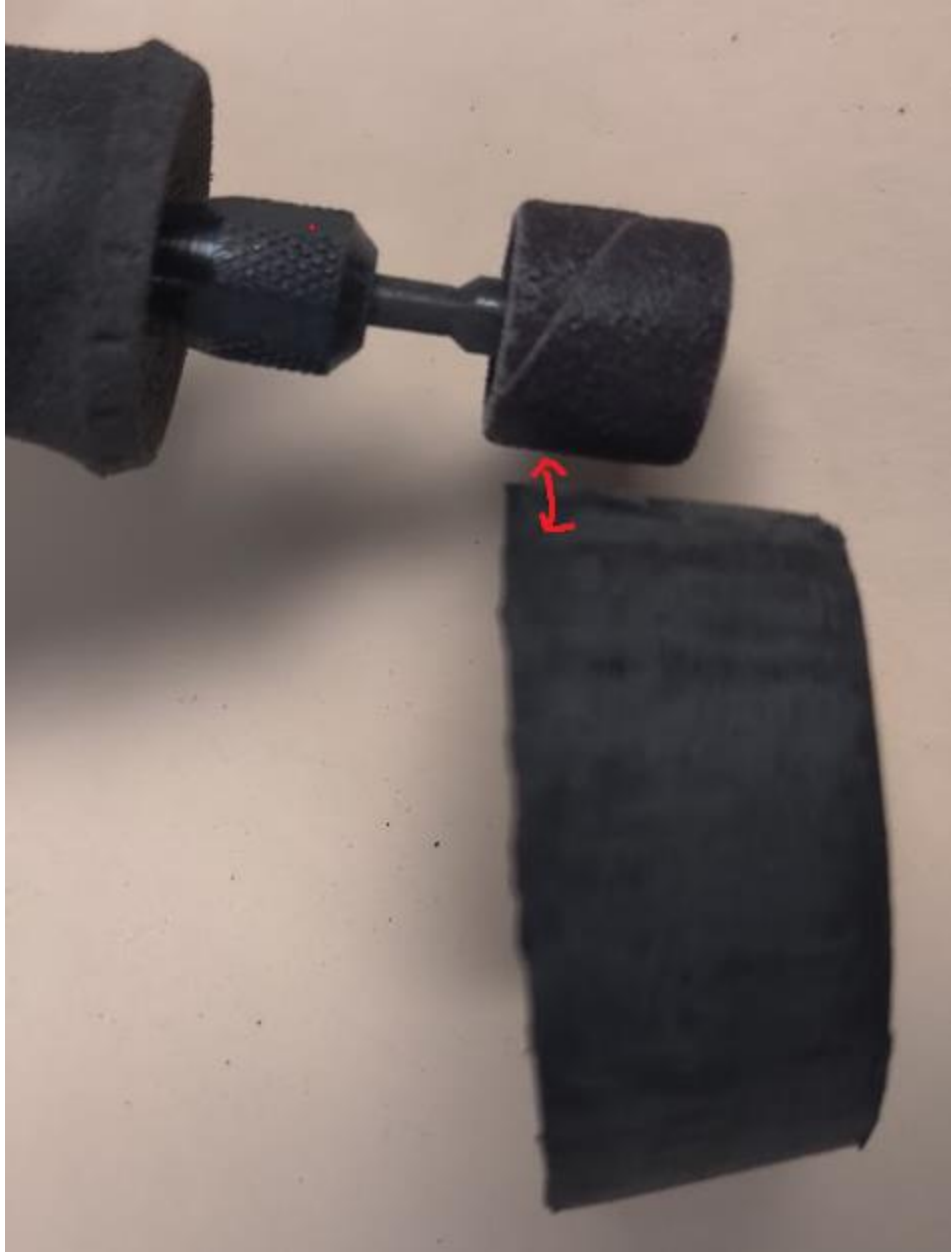


Figure 37: Sanding along the width of the part with respect to the taper



Figure 38: Buildup of filament near the build platform due to reduced print speed



Figure 39: Before and after post processing

Section 5.5: Results

Figure 40 and Figure 41 show the results of printing the four separate g-code. Figure 42 and Figure 43 show the results after the four shells are fit into each other.



Figure 40: Isometric view of printed shells



Figure 41: Side view of printed shells



Figure 42: Side view of combined shells



Figure 43: Underside view of shells combined shells

CHAPTER 6 CONCLUSIONS AND RECOMMENDATIONS

This thesis presents a method of manufacturing dental crowns using FGAM. The method proposes using a VP process for manufacturing. To achieve functional grading, a unique method of shell-based manufacturing was devised, where the density of leucite containing ceramic particles will be adjusted between shells. To alter the traditional layer-based additive manufacturing g-code to shell-based, a program was developed to convert the g-code given by a slicing software to unique g-code for each shell. The concept was proven using an FDM AM machine. The results showed that a shell-based method of additive manufacturing is possible.

Recommendations for Future Work

The concept of this research was proven using FDM. The concept remains untested with the original intention of using a VP process. Follow up research is required to test if FGAM using a VP process can produce a dental crown with functional grading.

The theory that adjusting particle density in a ceramic suspension for use in VP remains untested. Ceramic suspensions should have a high filler density to avoid internal cracks and pores, therefore intentionally reducing the density will invite these issues to occur. Using this method will reduce the hardness of the material to achieve a functional grade, however there may be unintentional reductions in other material properties as a result.

This research did not take into account the prep geometry of a tooth for real world dental applications. The interior geometry of a crown contains areas where the tooth cusp fits, and the crowns to be manufactured using the proposed method have a flat interior ceiling. More development into the program to accommodate for this region is required. The bottom region of a crown that contacts the gum line is also not explored in this research since only the outermost

shells would need to be printed for this area. More development into the program is necessary to account for this portion.

The debinding and sintering process for green parts after a ceramic VP process remains an issue for future research as the process often results in clinically unacceptable parts after manufacturing. Efforts have been made to improve this aspect of manufacturing but is largely material dependent. Creating an adequate debinding and sintering process for leucite containing ceramic must be studied in future research to produce a clinically acceptable dental composite made with this material.

REFERENCES

- [1] Turssi, Cecilia Pedroso, et al. “Wear of Dental Resin Composites: Insights into Underlying Processes and Assessment Methods—A Review.” *Journal of Biomedical Materials Research*, vol. 65B, no. 2, 2003, pp. 280–285., <https://doi.org/10.1002/jbm.b.10563>.
- [2] Gutiérrez-Salazar, Maria del, and Jorge Reyes-Gasga. “Microhardness and Chemical Composition of Human Tooth.” *Materials Research*, vol. 6, no. 3, 2003, pp. 367–373., <https://doi.org/10.1590/s1516-14392003000300011>.
- [3] Lakhdar, Y., et al. “Additive Manufacturing of Advanced Ceramic Materials.” *Progress in Materials Science*, vol. 116, 2021, p. 100736., <https://doi.org/10.1016/j.pmatsci.2020.100736>.
- [4] Atai M, Yassini E, Amini M, Watts DC. The effect of a leucite-containing ceramic filler on the abrasive wear of dental composites. *Dent Mater*. 2007 Sep;23(9):1181-7. doi: 10.1016/j.dental.2007.03.006. Epub 2007 May 15. PMID: 17507087.
- [5] Revilla-León M, Methani MM, Morton D, Zandinejad A. Internal and marginal discrepancies associated with stereolithography (SLA) additively manufactured zirconia crowns. *J Prosthet Dent*. 2020 Dec;124(6):730-737. doi: 10.1016/j.prosdent.2019.09.018. Epub 2020 Jan 22. PMID: 31980204.
- [6] O. Santoliquido, P. Colombo, A. Ortona, Additive Manufacturing of ceramic components by Digital Light Processing: a comparison between the “bottom-up” and the “top-down” approaches, *J. Eur. Ceram. Soc.* 39 (2019) 2140–2148, <https://doi.org/10.1016/j.jeurceramsoc.2019.01.044>.
- [7] Camargo, I. L., Morais, M. M., Fortulan, C. A., & Branciforti, M. C. (2021). A review on the rheological behavior and formulations of ceramic suspensions for VAT

- photopolymerization. *Ceramics International*, 47(9), 11906–11921.
<https://doi.org/10.1016/j.ceramint.2021.01.031>
- [8] M. Schwentenwein, J. Homa, Additive manufacturing of dense alumina ceramics, *Int. J. Appl. Ceram. Technol.* 12 (2015) 1–7, <https://doi.org/10.1111/ijac.12319>.
- [9] Q. Lian, F. Yang, H. Xin, D. Li, Oxygen-controlled bottom-up mask-projection stereolithography for ceramic 3D printing, *Ceram. Int.* 43 (2017) 14956–14961, <https://doi.org/10.1016/j.ceramint.2017.08.014>.
- [10] Song, X., Chen, Z., Lei, L., Shung, K., Zhou, Q., & Chen, Y. (2017). Piezoelectric component fabrication using projection-based stereolithography of barium titanate ceramic suspensions. *Rapid Prototyping Journal*, 23(1), 44–53. <https://doi.org/10.1108/rpj-11-2015-0162>
- [11] +, & Article by Varun Pandula I am Varun. (2019, September 2). *Finish lines in tooth preparation – indications, advantages and disadvantages*. JuniorDentist.com - Oral Care Tips by a Dentist. Retrieved October 14, 2022, from <https://www.juniordentist.com/finish-lines-in-tooth-preparation.html>
- [12] Borelli, B., Sorrentino, R., Goracci, C., Zarone, F., & Ferrari, M. (2013). In vitro analysis of residual tooth structure of maxillary anterior teeth after different prosthetic finish line preparations for full-coverage single crowns. *Journal of Oral Science*, 55(1), 79–84. <https://doi.org/10.2334/josnusd.55.79>
- [13] BEUER, F., EDELHOFF, D., GERNET, W., & NAUMANN, M. (2008). Effect of preparation angles on the precision of zirconia crown copings fabricated by CAD/CAM system. *Dental Materials Journal*, 27(6), 814–820. <https://doi.org/10.4012/dmj.27.814>

- [14] Wu, D., Shi, J., Niu, F., Ma, G., Zhou, C., & Zhang, B. (2022). Direct additive manufacturing of melt growth $\text{Al}_2\text{O}_3\text{-ZrO}_2$ functionally graded ceramics by laser directed energy deposition. *Journal of the European Ceramic Society*, 42(6), 2957–2973. <https://doi.org/10.1016/j.jeurceramsoc.2022.01.034>
- [15] Li, W., Armani, A., Martin, A., Kroehler, B., Henderson, A., Huang, T., Watts, J., Hilmas, G., & Leu, M. (2021). Extrusion-based additive manufacturing of functionally graded ceramics. *Journal of the European Ceramic Society*, 41(3), 2049–2057. <https://doi.org/10.1016/j.jeurceramsoc.2020.10.029>
- [16] Lv, M., Chen, W., & Liu, C. (2014). Fabrication and mechanical properties of $\text{TiB}_2/\text{ZrO}_2$ functionally graded ceramics. *International Journal of Refractory Metals and Hard Materials*, 46, 1–5. <https://doi.org/10.1016/j.ijrmhm.2014.04.019>
- [17] Ghazanfari, Amir, et al. “Additive Manufacturing and Mechanical Characterization of High Density Fully Stabilized Zirconia.” *Ceramics International*, vol. 43, no. 8, 2017, pp. 6082–6088., <https://doi.org/10.1016/j.ceramint.2017.01.154>.
- [18] Yu, Tianyu, et al. “Extrusion-Based Additive Manufacturing of Yttria-Partially-Stabilized Zirconia Ceramics.” *Ceramics International*, vol. 46, no. 4, 2020, pp. 5020–5027., <https://doi.org/10.1016/j.ceramint.2019.10.245>.
- [19] Pfeiffer, Stefan, et al. “Direct Laser Additive Manufacturing of High Performance Oxide Ceramics: A State-of-the-Art Review.” *Journal of the European Ceramic Society*, vol. 41, no. 13, 2021, pp. 6087–6114., <https://doi.org/10.1016/j.jeurceramsoc.2021.05.035>.
- [20] Shahzad, Khuram, et al. “Additive Manufacturing of Zirconia Parts by Indirect Selective Laser Sintering.” *Journal of the European Ceramic Society*, vol. 34, no. 1, 2014, pp. 81–89., <https://doi.org/10.1016/j.jeurceramsoc.2013.07.023>.

- [21] Zhong, S., Shi, Q., Deng, Y., Sun, Y., Politis, C., & Yang, S. (2022). High-performance zirconia ceramic additively manufactured via nanoparticle jetting. *Ceramics International*, 48(22), 33485–33498. <https://doi.org/10.1016/j.ceramint.2022.07.294>
- [22] Wu, Haidong, et al. “Fabrication of Dense Zirconia-Toughened Alumina Ceramics through a Stereolithography-Based Additive Manufacturing.” *Ceramics International*, vol. 43, no. 1, 2017, pp. 968–972., <https://doi.org/10.1016/j.ceramint.2016.10.027>.
- [23] Xing, Hongyu, et al. “Study on Surface Quality, Precision and Mechanical Properties of 3D Printed ZRO₂ Ceramic Components by Laser Scanning Stereolithography.” *Ceramics International*, vol. 43, no. 18, 2017, pp. 16340–16347., <https://doi.org/10.1016/j.ceramint.2017.09.007>.
- [24] Lian, Qin, et al. “Additive Manufacturing of Zro₂ Ceramic Dental Bridges by Stereolithography.” *Rapid Prototyping Journal*, vol. 24, no. 1, 2018, pp. 114–119., <https://doi.org/10.1108/rpj-09-2016-0144>.
- [25] Fu, Xiangsong, et al. “Effect of Printing Strategies on Forming Accuracy and Mechanical Properties of zro₂ Parts Fabricated by SLA Technology.” *Ceramics International*, vol. 45, no. 14, 2019, pp. 17630–17637., <https://doi.org/10.1016/j.ceramint.2019.05.328>.

APPENDIX ACURA SLICING SOFTWARE SETTINGS

Quality			
Layer Height		0.25	mm
Initial Layer Height		0.4	mm
Line Width		0.5	mm
Wall Line Width		0.5	mm
Outer Wall Line Width		0.5	mm
Inner Wall(s) Line Width		0.5	mm
Top/Bottom Line Width		0.5	mm
Infill Line Width		0.5	mm
Support Line Width		0.5	mm

Figure A.1: CURA settings for Quality

Shell			
Wall Thickness		1.0	mm
<i>Wall Line Count</i>		1	
Outer Wall Wipe Distance		0.25	mm
Top/Bottom Thickness		0.8	mm
Top Thickness		1.1	mm
Top Layers		5	
Bottom Thickness		1.1	mm
<i>Bottom Layers</i>		5	
Top/Bottom Pattern		Lines	
Bottom Pattern Initial Layer		Lines	
Top/Bottom Line Directions		[]	
Outer Wall Inset		0	mm
Outer Before Inner Walls		<input type="checkbox"/>	
Alternate Extra Wall		<input type="checkbox"/>	
<i>Compensate Wall Overlaps</i>		<input type="checkbox"/>	
Compensate Outer Wall Overlaps		<input type="checkbox"/>	
Compensate Inner Wall Overlaps		<input type="checkbox"/>	
Fill Gaps Between Walls		Everywhere	
Horizontal Expansion		0	mm
Z Seam Alignment		Sharpest Corner	
Ignore Small Z Gaps		<input checked="" type="checkbox"/>	

Figure A.2: CURA settings for shell

Infill	
Infill Density	100 %
Infill Line Distance	0.5 mm
Infill Pattern	Lines
Infill Line Directions	[]
Infill Overlap Percentage	0 %
Infill Overlap	0 mm
Skin Overlap Percentage	5 %
Skin Overlap	0.025 mm
Infill Wipe Distance	0.125 mm
Infill Layer Thickness	0.25 mm
Gradual Infill Steps	0
Infill Before Walls	<input type="checkbox"/>
Minimum Infill Area	0 mm ²
Skin Expand Distance	0.5 mm
Maximum Skin Angle for Expansion	90 °
Minimum Skin Width for Expansion	0.0 mm

Figure A.3: CURA settings for infill

Material		
Default Printing Temperature	240	°C
Printing Temperature	240	°C
Probe Temperature	170	°C
Soften Temperature	170	°C
Wipe Temperature	170	°C
Printing Temperature Initial Layer	245	°C
Initial Printing Temperature	230	°C
Final Printing Temperature	225	°C
Build Plate Temperature	105	°C
Part Removal Temperature	60	°C
Keep Heating	<input checked="" type="checkbox"/>	
Build Plate Temperature Initial Layer	105	°C
Diameter	2.875	mm
Flow	95	%
Initial Layer Flow Rate	100	%
Enable Retraction	<input checked="" type="checkbox"/>	
Retract at Layer Change	<input type="checkbox"/>	
Retraction Distance	1	mm
Retraction Speed	10	mm/s
Retraction Retract Speed	10	mm/s
Retraction Prime Speed	10	mm/s
Retraction Extra Prime Amount	0	mm ³
Retraction Minimum Travel	1.0	mm
Maximum Retraction Count	90	
Minimum Extrusion Distance Window	1	mm
Nozzle Switch Retraction Distance	16	mm
Nozzle Switch Retraction Speed	20	mm/s
Nozzle Switch Retract Speed	20	mm/s
Nozzle Switch Prime Speed	20	mm/s

Figure A.4: CURA settings for material

Speed		
Print Speed		60 mm/s
Infill Speed		60 mm/s
Wall Speed		30.0 mm/s
Outer Wall Speed		45 mm/s
Inner Wall Speed		60.0 mm/s
Top/Bottom Speed		60 mm/s
Support Speed		60 mm/s
Support Infill Speed		60 mm/s
Travel Speed		175 mm/s
Initial Layer Speed		10 mm/s
Initial Layer Print Speed		10 mm/s
Initial Layer Travel Speed		29.1667 mm/s
Maximum Z Speed		0 mm/s
Number of Slower Layers		2
Equalize Filament Flow		<input type="checkbox"/>
Enable Acceleration Control		<input type="checkbox"/>
Enable Jerk Control		<input type="checkbox"/>

Figure A.5: CURA settings for speed

Travel		
Combing Mode		All
Retract Before Outer Wall		<input type="checkbox"/>
Avoid Printed Parts When Traveling		<input checked="" type="checkbox"/>
Travel Avoid Distance		0.625 mm
Layer Start X		0.0 mm
Layer Start Y		0.0 mm
Z Hop When Retracted		<input type="checkbox"/>

Figure A.6: CURA settings for travel



Cooling		
Enable Print Cooling	<input checked="" type="checkbox"/>	
Fan Speed	100.0	%
Regular Fan Speed	0	%
Maximum Fan Speed	30	%
Regular/Maximum Fan Speed Threshold	120	s
Initial Fan Speed	0	%
Regular Fan Speed at Height	6	mm
Regular Fan Speed at Layer	3	
Minimum Layer Time	30	s
Minimum Speed	5	mm/s
Lift Head	<input type="checkbox"/>	

Figure A.7: CURA settings for cooling

Support		
Generate Support	<input checked="" type="checkbox"/>	
Support Placement	<input type="checkbox"/>	Everywhere
Support Overhang Angle	<input type="checkbox"/>	61 °
Support Pattern	<input type="checkbox"/>	Zig Zag
Connect Support ZigZags	<input checked="" type="checkbox"/>	
Support Density	<input type="checkbox"/>	30 %
Support Line Distance	<input type="checkbox"/>	1.6667 mm
Support Z Distance	<input type="checkbox"/>	0.2 mm
Support Top Distance	<input type="checkbox"/>	0.2 mm
Support Bottom Distance	<input type="checkbox"/>	0.2 mm
Support X/Y Distance	<input type="checkbox"/>	0.4 mm
Support Distance Priority	<input type="checkbox"/>	X/Y overrides Z
Support Stair Step Height	<input type="checkbox"/>	0.3 mm
Support Stair Step Maximum Width	<input type="checkbox"/>	5.0 mm
Support Join Distance	<input type="checkbox"/>	2.0 mm
Support Horizontal Expansion	<input type="checkbox"/>	0.2 mm
Enable Support Interface	<input type="checkbox"/>	
Enable Support Roof	<input type="checkbox"/>	
Enable Support Floor	<input type="checkbox"/>	

Use Towers		<input checked="" type="checkbox"/>
Tower Diameter		3.0 mm
Minimum Diameter		3.0 mm
Tower Roof Angle		65 °

Figure A.8: CURA settings for support

 **Build Plate Adhesion** 




Build Plate Adhesion Type   None 

Figure A.9: CURA settings for build plate adhesion

 **Mesh Fixes** 

Union Overlapping Volumes

Remove All Holes

Extensive Stitching

Keep Disconnected Faces

Merged Meshes Overlap 0.15 mm







Remove Mesh Intersection 

Figure A.10: CURA settings for mesh fixes

 **Special Modes** 

Print Sequence  All at Once 

Mold

Surface Mode Normal 


Spiralize Outer Contour 

Figure A.11: CURA settings for special modes

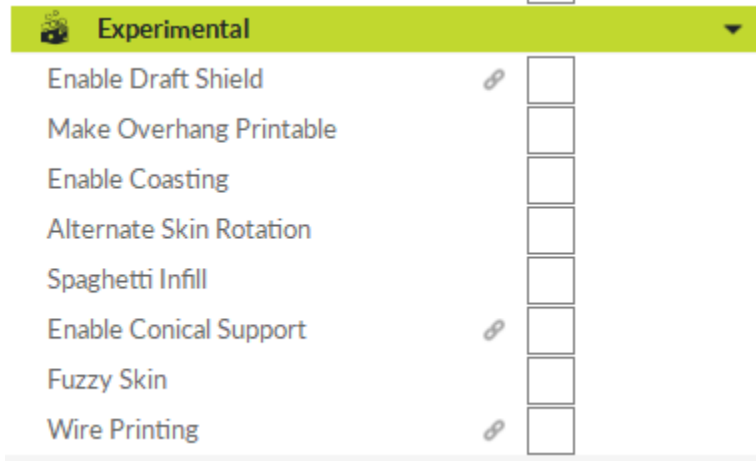


Figure A.12: CURA settings for experimental