2022-12-01

Processes & Toolchain For Automation Of Hybrid Direct-Write 3D Printing

Gilbert Thomas Carranza
University of Texas at El Paso

PROCESSES & TOOLCHAIN FOR AUTOMATION OF HYBRID DIRECT-WRITE 3D PRINTING

GILBERT THOMAS CARRANZA

Doctoral Program in Electrical and Computer Engineering

APPROVED:

Raymond C. Rumpf, Ph.D., Chair

Robert C. Roberts, Ph.D.

Sai Mounika Errapotu, Ph.D.

Brian E. Schuster, Ph.D.

Paul I. Deffenbaugh, Ph.D.

Stephen L. Crites, Jr., Ph.D.
Dean of the Graduate School

Dedication

To Him, to whom I commit my work.

To Jessica, who was always at my side.

To my parents, who instilled my love to learn.

PROCESSES & TOOLCHAIN FOR AUTOMATION OF HYBRID DIRECT-WRITE 3D PRINTING

by

GILBERT THOMAS CARRANZA, M.S.E.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

December 2022

Acknowledgements

I would like to acknowledge and thank my father Tommy Carranza, my mother Angie Carranza, and my brother David Carranza, for loving and supporting me throughout my academic journey. I would not be here if it wasn't for you. Thank you to my extended family as well.

I would especially like to thank my wife, Jessica. Your love, support, midday visits, late night calls, and random snack surprises were the highlights of many of my days.

I would like to thank everyone from the EM Lab. I will never forget the camaraderie and support from our time together at the lab. I wish the best to each of you in your prospective careers.

I would like to thank my advisor and mentor Dr. Raymond C. Rumpf for his direction and guidance in my training as an engineer and researcher. I have learned so much from him, and I truly believe there is no better mentor in the world.

I would like to thank everyone at the ECE department as well as my defense committee, for supporting, overseeing, and guiding me on my research.

I would like to thank Kraetonics, LLC, for the opportunity and funding to further pursue this research. Thank you Chris and Chris for putting up with my horrible code.

And finally, I would like to thank my friends, who are too many to name. Thank you for supporting me and cheering me on from afar. I'll make up for all the times I missed going out with you all.

Financial support and acknowledgments go to:

- Air Force Research Lab (award: FA8659-17-C-1011)
- Army Research Lab (award: W911Nf-16-2-0019)
- FlexTech (award: RFP-19-185) (UTEP Contract: 226351628R)
- National Center for Manufacturing Sciences (award: W911NF-19-0001)

Abstract

Hybrid 3D printing has evolved from a means to rapidly prototype devices to a fully viable means of manufacturing final products. In particular, electronic and electromagnetic devices have been a focus of hybrid 3D printing multi-material structures. However, while the hardware capabilities have been around for years, the software capabilities have only begun to catch up. A process and toolchain for hybrid 3D printing is critically needed.

This work details a process and toolchain for hybrid 3D printing metal-dielectric structures. In it, a basic slicing algorithm is shown along with off-axis printing and conformal printing for arbitrary curvatures. These methods are derived such that devices that are typically printed with a five- or six-axis machine can now be printed with a standard multi-tool three-axis machine. Moreover, these methods are generalized so they can be implemented with any printer, although in this work they are tested with Ultimaker and nScrypt printers. Multiple devices are demonstrated using these new algorithms.

Finally, a tool path planning process for hybrid printing functionally-graded materials is shown. The tool path planning needed for printing arbitrary functionally graded materials involves numerous routines including adjusted infill patterns, look-ahead routines, and priming routines. Arbitrary functionally graded devices are demonstrated.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Purpose

Additive manufacturing (3D printing) has garnered much attention in the last decade, evolving from a rapid prototyping process to a viable manufacturing process. What was once an expensive technology for large companies, the emergence of different 3D printer manufacturers has made 3D printing more accessible to small businesses and hobbyists. This accessibility has led to smaller businesses turning to 3D printing as a primary source of manufacturing since 3D printing allows for more niche products to be designed, manufactured, and sold in a relatively short amount of time [1].

Moreover, increased accessibility has driven an interest in research regarding 3D printing. In particular, hybrid 3D printing involves utilizing multiple printing processes within the same manufacturing job. This differs from multi-material printing, where multiple materials may be used with the same printing process. As opposed to multi-material printing, hybrid printing implies that materials of different types, such as thermoplastics, liquids, ceramics, pastes, inks, flexible materials, etc. can be incorporated in the same device in during the same print job that would otherwise require multiple printing processes. This comes with its own set of challenges, such as how the materials interact with each other, how different printing processes affect other processes in the same environment, how tool paths are generated with respect to each process, the gantry and tool head dimensions, and the limitations set by the printer hardware.

It is hypothesized that new hybrid printing processes will result in a high yield of devices with electro-mechanical functionality. It is currently unknown whether these devices can benefit from unconventional printing processes. Therefore, the research performed in this dissertation is the development of processes and a toolchain for the automation of hybrid direct-write 3D printing of arbitrary metal-dielectric parts and functionally graded materials. Up until recently, no groups other than the EM Lab and Kraetonics LLC. have been able to manufacture metal-dielectric and functionally graded structures using automated hybrid 3D printing. Since the capabilities were only just made possible, no standardized way of printing these devices has been established. Since no standards exist for hybrid 3D print electronic and electromagnetic devices, the impact of this research will be profound as it will allow businesses and researchers alike to develop and implement new standards. Electronic and electromagnetic devices will become more practical and easier to build and develop. Small businesses and hobbyists can use hybrid 3D printing as a means of manufacturing advanced functional electronic devices.

1.2 Background and State-of-the-Art

1.2.1 A Brief History of 3D Printing

Three-dimensional printing (3D printing) beyond microchip fabrication first began in the 1980s with the invention of stereolithography, a process by which a liquid resin is polymerized layer-by-layer using ultra-violet light [2]. Laminated object manufacturing (LOM) was invented not too long after. LOM uses a process that cuts cross-sections of an object from paper and adheres layers together by bonding them with a melted plastic in-between. [3] Shortly after, Selective Laser Sintering (SLS) came about from the University of Texas. [4] SLS involves shooting metal particles at a focal point and rapidly sintering it with a focused laser, leaving behind a solid metallic structure. Finally, in the late 1980s fused-deposition modeling (FDM) was invented and patented by C.S. Crump [5], who later founded Stratasys Inc. in 1992. FDM is arguably the most common 3D printing technology, where spooled filament of thermoplastic is melted and extruded out of a heated nozzle, leaving behind a solid plastic trace.

Even though 3D printing came about before the new millennium, much of the technology was limited to large industries who could afford it. This began to change around 2005 when the University of Bath began developing a 3D printer that could manufacture its own parts. The dubbed Rep Rap (Replicating Rapid Prototyping) project used an open-source software platform and fused filament fabrication, essentially the same as FDM [6]. In 2006, Cornell University developed their own 3D printing system with open-source software and commercially available parts as well [7]. Also in 2006, MakerBot Industries was founded and provided DIY kits for anyone to participate in 3D printing. Eventually MakerBot was bought out by Stratasys Inc. in 2013 after making their software proprietary. Today, the typical printers available for commercial use include MakerBot, Ultimaker, Prusa, and LulzBot among others. While some have proprietary software, others can be integrated with other open-source slicers.

Over time as 3D printing became more commonplace and accessible, many areas in different fields of science have benefited and made great strides as to what could be manufactured, including functional printed organs in biomedicine and terahertz waveguides in electromagnetics [8]–[13]. More recently, 3D printers have been able to construct functional electronics, food, houses, and even weapons [14]–[16]. Consequently, hybrid 3D printing was the next logical step in printing more complex devices.

As previously stated, hybrid 3D printing refers to printing with multiple printing processes in the same job. This implies that different materials that require different printing processes can be incorporated in the same printed device. Hybrid 3D printing has undergone considerable research within the last decade alone. The W. M. Keck Center for 3D Innovation at The University of Texas at El Paso was among the first to pioneer the capabilities and benefits of hybrid 3D printing with wire embedding, structural electronics, and patch antennas [17]–[23]. The EM Lab at UTEP demonstrated that automation of hybrid 3D printing processes could be achieved to create functional conductive interconnects that meander in three dimensions [24]. The EM Lab was also among the first to produce a functional 3D volumetric circuit manufactured with an automated hybrid 3D printing process [25]. The University of Delaware has also made great strides in hybrid 3D printing, including manufacturing graded dielectrics with dry powder printing and composite manufacturing [26], [27] along with dielectric and metallic structures for RF applications [28]–[31]. Hybrid printed structures for phased array applications have been demonstrated by the University of South Florida [32], [33] along with other electronic structures [34]–[36]. More recently, research in flexible hybrid electronics has garnered much attention in the area of hybrid 3D printed electronics [37]–[41]. NextFlex and SEMI-FlexTech are two consortiums dedicated to research of flexible electronics and their respective hybrid printing applications [42], [43].

As hybrid 3D printing gained traction, several companies emerged to provide products solely dedicated to hybrid 3D printing. A notable example is nScrypt, Inc. which is known for the 3Dn hybrid printers, capable of housing tools for fused deposition modeling, micro-dispensing, pick-and-place, surface scanning, milling, and laser curing [44]. Similar to nScrypt, Hyrel 3D makes multi-tool printers with many of the same capabilities as nScrypt, although at a lower resolution [45]. Voxel8 was once touted as the world’s first electronics 3D printer [46]. The hybrid printers, now discontinued, were aimed at rapid prototyping printed circuit boards using fused deposition modeling and dispensing conductive paste. Nano Dimension’s Dragonfly IV [47] is one of the latest commercially available printers capable of producing high resolution, high quality electronic devices via inkjet printing. However, while their printer works for small, high resolution devices, they are slow and cannot print large structures. Optomec printers are also capable of printing high resolution electronic traces using aerosol jet technology [48]. Among their demonstrated devices include printed antennas, 3D microstructures, RF interconnects, and printing on unpackaged/packaged dies. Much like Nano Dimension, their processes and devices are limited to small, fine featured devices and have trouble with larger volumetric devices.

1.2.2 Slicing for Hybrid 3D Printing

Slicing is the process of converting a 3D CAD model into the raw instructions that control a 3D printer to manufacture the part. Slicing algorithms are not new as they date back to the 1990s and were (and still are) used in CNC machines. Since then, a variety of methods for slicing different types of models have been studied and developed [49]–[51]. Current slicers are even capable of doing dual-FDM prints, such as with MakerBot and Ultimaker printers. Today there are several slicing software packages available to use, both proprietary and open source. Some of the popular slicing software include Slic3r [52], Prusa Slicer [53], Cura [54], Repetier [55], Simplify3D [56], and MakerBot Print [57].

However, slicing for hybrid 3D printing is in its infancy. Groups who have previously attempted hybrid 3D printing developed their own slicing software unique to their printer. In one instance, the hybrid slicer's block diagram describes the process as being based on relative coordinate systems, where each tool has its own relative coordinate system [58]. Another effort shows a virtual hybrid slicer capable of slicing many materials and displaying the tool paths. However, this was only shown to be a demonstration of a simulated printing environment and has not been shown to manufacture devices, though it could be modified to drive a hybrid 3D printer [59], [60]. UTEP's W. M. Keck Center for 3D Innovation developed their own software programmed into SolidWorks to drive hybrid manufacturing of FDM and wire-bonding technologies [21]. More recently, an attempt at hybrid printing was successfully achieved at the EM Lab, resulting in the first case of a fully automated print of volumetric arbitrary metal-dielectric structures. This made use of translating a dual-FDM printing process from Slic3r and Repetier into a hybrid FDM/dispensing process that drove an nScrypt TableTop 3Dn printer [24], [25].

However, despite these attempts, there was yet to exist a slicing software capable of configuring multiple tools with unconventional printing processes and different materials. Kraetonics, LLC. is the first company to develop and commercialize an advanced slicing software OmniSlice™, capable of utilizing a three-axis system to its fullest capabilities. OmniSlice™ is capable of traditional layer-by-layer slicing as well as conformal slicing, off-axis slicing, and slicing for functionally-graded materials.

This work presents both basic and more advanced slicing routines for hybrid 3D printing multi-material devices for electronic and electromagnetic devices that would otherwise need specialized equipment. The algorithms provide the tools necessary for developing environments for designing and slicing complex electronic and electromagnetic structures. The slicing algorithms presented are also confined to three-axis systems, reducing the need for expensive five-axis and six-axis machines that are currently the state-of-the-art for printing complex conformal structures. The novelty of this work in particular features a method for printing conformal structures on near-vertical walls with only three-axis systems. Demonstrated structures, thought at one time to be an impossibility for printing, are presented.

1.3 Outline

Chapter 1 provides the introduction to this document. The purpose, field background, and outline of the chapters the present work is given.

Chapter 2 provides the algorithm and development of a 3D circuit layout tool. The tool is vital to the design of 3D volumetric circuits, as no such tool had existed before it. The tool was made in Blender by using its Python interpreter and a third-party library of component models to design the circuit package and the interconnects of the components.

Chapter 3 describes the hybrid slicing and printing procedures developed for automated hybrid 3D printing. It begins with basic slicing algorithms before moving onto more complex algorithms such as off-axis and conformal slicing. It then provides additional routines for the printer to help increase the yield of printed devices. These routines include cleaning and priming routines as well as some additional routines for controlling extrusion. Finally, it provides and experimental setup and example on determining printing parameters for hybrid printed devices.

Chapter 4 provides the application of the algorithms described in Chapter 3. This includes examples of 3D volumetric circuits and a conformal frequency selective surface. The results are shown along with the printed devices.

Chapter 5 provides methods on printing functionally graded materials. It describes the techniques and considerations used when printing including look-ahead routines, g-code writing, and parameter configuration.

Finally, Chapter 6 gives concluding remarks as well as considerations for future work.

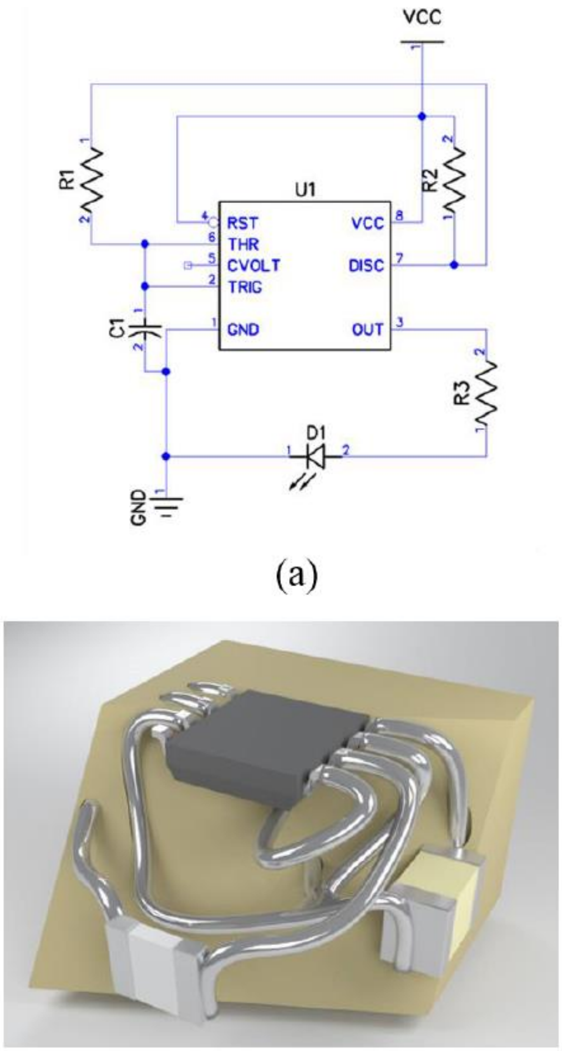
Chapter 2: 3D Circuit Layout[1]

The advancements in 3D printing over the past decade have long established that hybrid 3D printing is at the forefront of a new way of manufacturing devices. The prospects of manufacturing 3D volumetric circuits (3DVC) had consequently become more realistic. However, there had been no way of designing 3DVCs in a similar way to PCB design. Traditional PCB circuit design involves many steps, including but not limited to creating a netlist, importing and placing components, and laying out traces, footprints, vias, etc. on a PCB. Designing a 3DVC essentially would have taken an extremely limited and painstaking process of modeling specific component, package, and trace geometries. In addition, these models would need to be exported as manifold STL files suitable for slicing. For each new device, this process would have to be repeated from the very beginning. Therefore, a dedicated CAD tool to designing 3DVCs was critically needed. This chapter outlines the development of that design tool.

2.1 Tool Description

As with traditional circuit design and layout, the circuit must be designed in schematic form and possibly even simulated to ensure proper functionality. Given the schematic, a netlist is generated describing the interconnects and their corresponding connections to the circuit elements. This same information is needed for both 2D and 3D circuit layout designs. A netlist and the component geometries are imported into a 3D CAD software where the physical layout of the circuit is performed. The schematic diagram for a 3D 555 timer circuit is given in Figure 1 along with the CAD model of the designed 3D circuit.

[1] This chapter is an excerpt taken from [25]



(a)

(b)

Figure 1. (a) Circuit schematic for a blinking 555 timer circuit, and (b) corresponding model showing components, interconnects, and sliced substrate.

2.1.1 Software Description

The schematic capture tool used in this project was the freeware version of DipTrace. DipTrace was selected for its schematic capture tool, its ability to export netlists for popular layout software as well as netlists for SPICE simulations, and its extensive 3D component library. Each component comes with its own STL model. These models include common standard integrated circuit (IC) packages as well as specialized components from different manufacturers. This library of parts was a tremendous benefit to this research.

The tools for the CAD environment were programmed into the open-source graphics design software Blender as an add-on. Blender utilizes surface meshes for drawing and manipulating 3D objects and features a Python console and text editor to write scripts and add-ons. The 3D design tool was created by developing add-ons that import information from DipTrace, perform placement of parts, route interconnects, and export the 3D design.

2.2 Data Import and Layout

In order to have easy access to each component while working within Blender, an add-on was created that imports and names individual components. The add-on allows the user to search through the list of components and select those needed for the design. The add-on also allows the user to import components directly from a netlist created by DipTrace. The tool reads the netlist and recognizes the names of each component's STL model and the nets connecting them. Each STL file is then imported separately so the corresponding components can be placed and oriented individually during the layout process. After all the components have been imported, the tool automatically names them according to the netlist so that each component is a direct representation of the original schematic.

The components are then placed by the user at any position and in any orientation, analogous to layout process of an ordinary 2D PCB layout software. Blender allows easy manipulation of the position (x, y, and z) and orientation (θ and ϕ) of each component. This lets the user minimize the distance between components, minimize trace lengths, and form the circuit into any shape. In Fig. 3, the 3D arranged components of the 555 circuit can be seen.

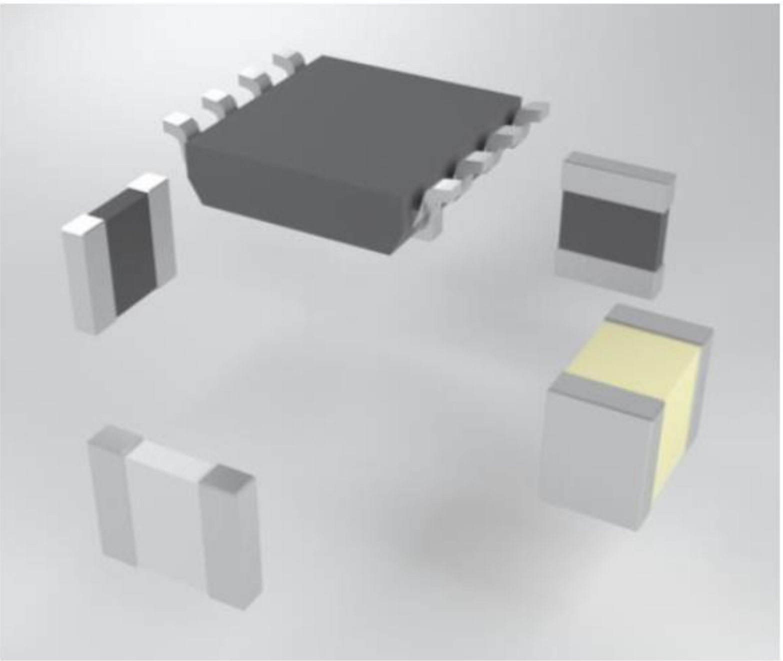


Figure 2. 3D layout of 555 timer circuit components. [25]

2.3 Interconnect Router

Another add-on was created to perform the routing process in 3D. The add-on functions in two possible ways: (1) a signal trace can be added as a free-floating connection to be routed manually, or (2) a signal trace can be placed semi-automatically between two selected components. There are also two choices of interconnects: (1) a DC line and (2) a high-frequency parallel plate transmission line (PPTL). Much of this add-on takes Blender's already-established functions and abilities and packages them into an interconnect router.

When creating an interconnect, the router generates a Nurbs curve (called a "path") via Blender's own modeler. The interconnect's body is then drawn along the path in a cylindrical shape using Blender's beveling function, where the thickness of the interconnect can be adjusted. Figure 3 shows the step-by-step process of how an interconnect is generated. The position and orientation of the interconnect itself is controllable via control points. Manipulation of these points allows the interconnect to bend and twist in any desired direction following a spline curvature. In Figure 4, an interconnect is shown drawn between two selected components along with its control points, and Figure 5 depicts a computer render of the circuit with both components and signal traces.

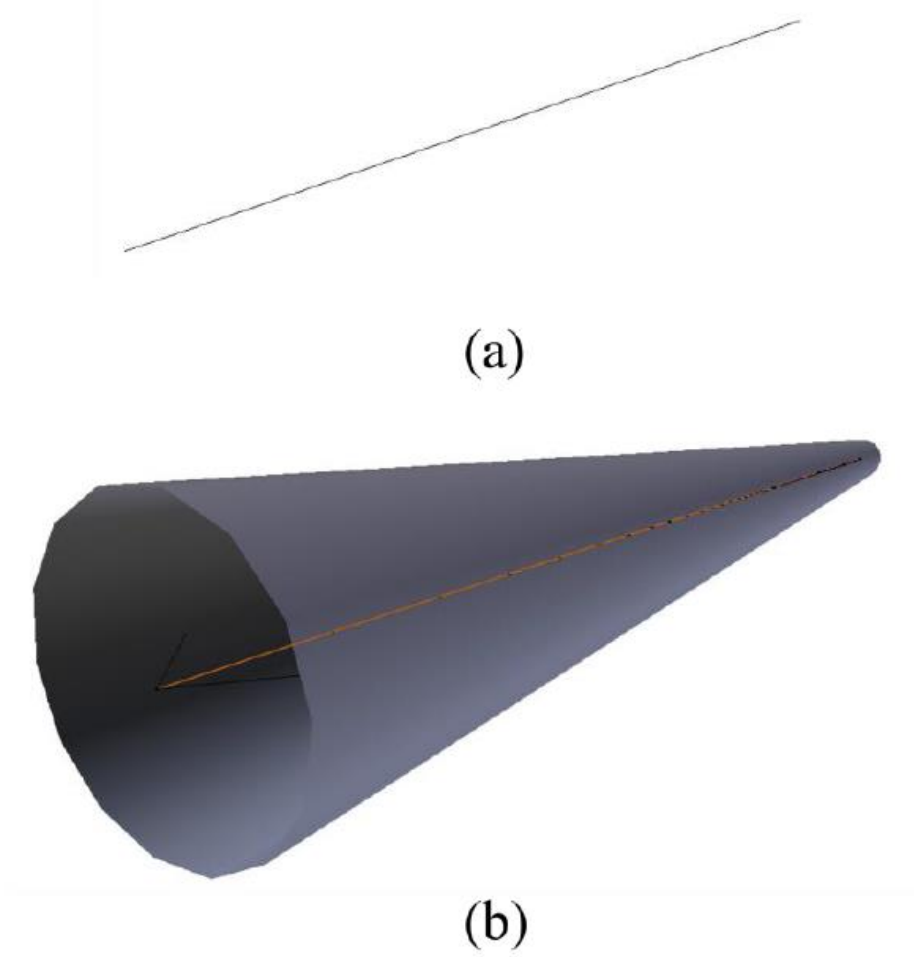


Figure 3. Generation of an interconnect. The Nurbs path is shown in (a), and the drawn body of the interconnect is shown in (b).

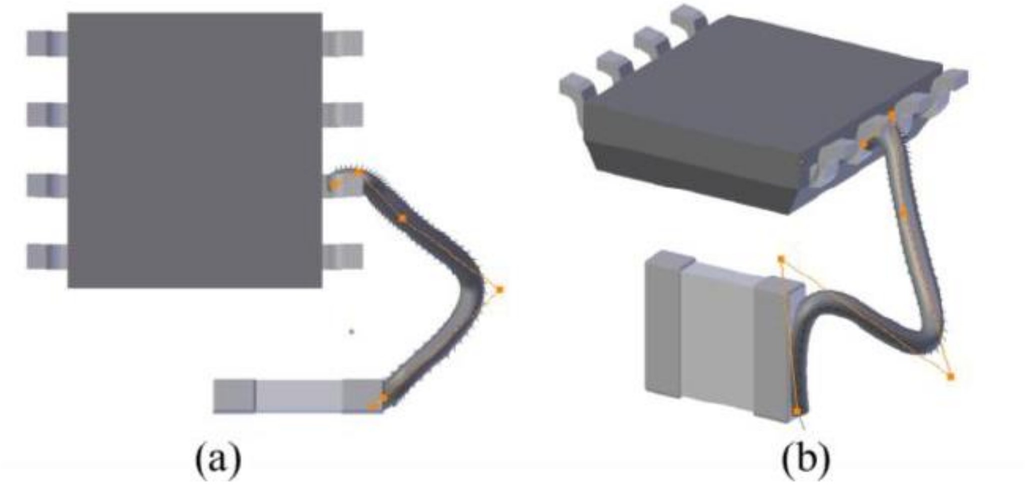


Figure 4. Top view (a) and perspective view (b) of an interconnect drawn between an IC and surface mount component. The control points are depicted as orange dots along the interconnect.

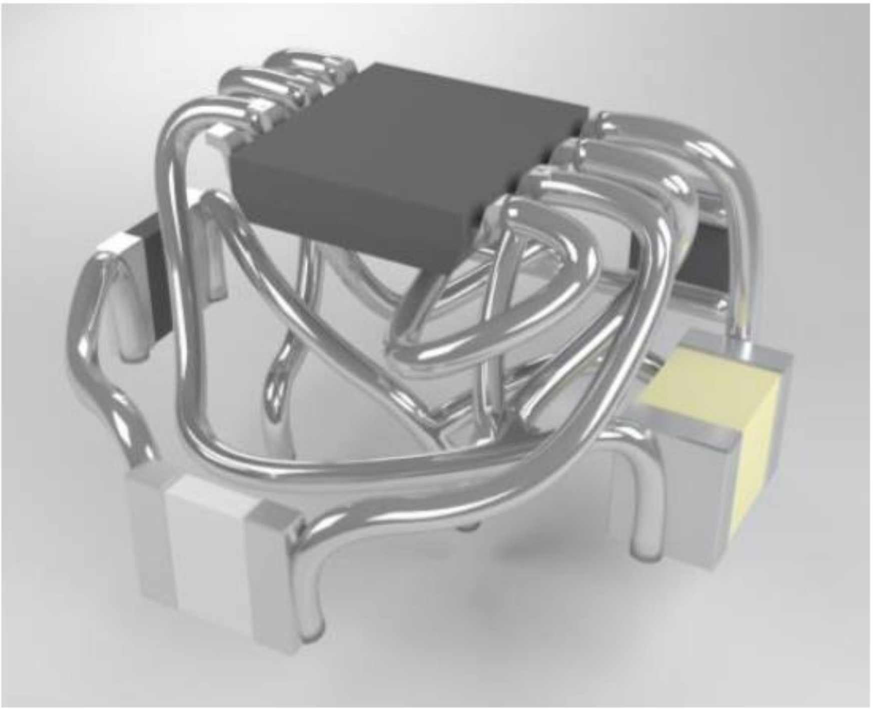


Figure 5. Final 555 circuit model with components and interconnects. Dielectric not shown.

The tool defaults to generating a free-floating interconnect when no electronic components are selected. However, the semi-automatic feature of the router allows for faster interconnect manipulation by placing an interconnect between any two desired components. To do this, the user must first manually select the mesh faces of two different component terminals. The tool then draws a straight interconnect between the two selected sets of faces. The user then only needs to move the control points to the desired curvature, eliminating the step of moving the entire interconnect to the desired location. Figure 6 shows an interconnect being drawn these two ways.

The interconnect router also has capabilities for generating a high-frequency differential transmission line. The only design implemented so far is a parallel-plate transmission line (PPTL). The PPTL was chosen due to its ability to confine fields and its apparent relative ease of manufacturing in a 3D setting compared to coaxial lines. Figure 7 shows a rendered model of a 3D PPTL. Because 3D RF circuits are still in its infancy stage, more research is needed in the future concerning this topic.

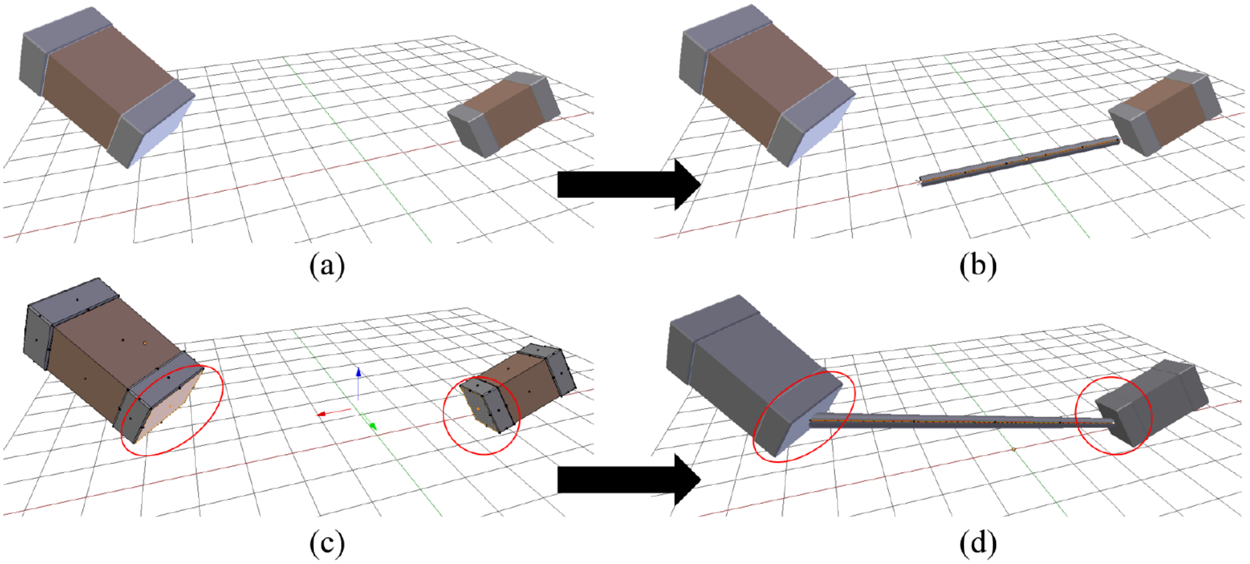


Figure 6. Routing process. (a) No components are selected. (b) The interconnect is drawn free-floating when no components are selected. (c) The faces of two different components are selected. (d) The interconnect is drawn connecting the two selected faces.

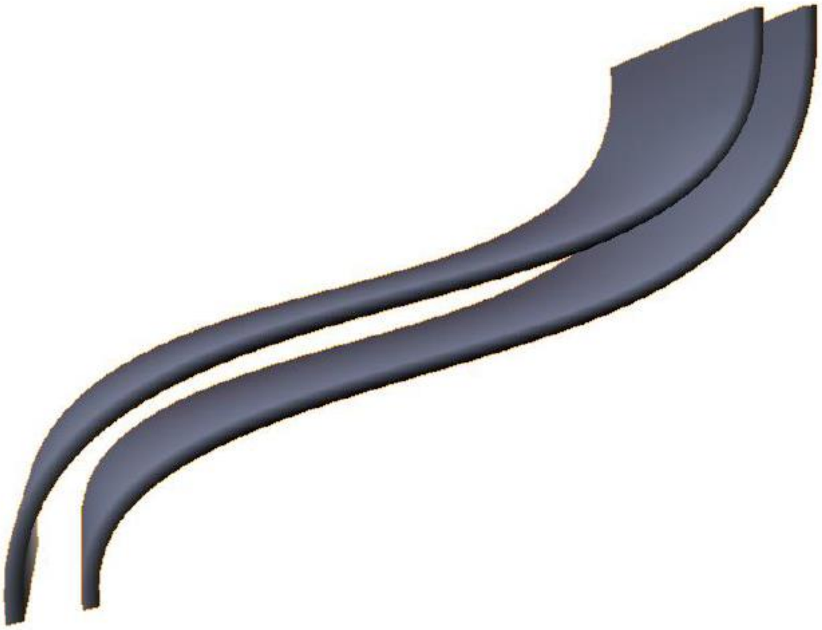


Figure 7. Rendered model of a parallel-plate transmission line.

2.4 Mesh Refinement and File Export

The hybrid 3D printing process requires separate STL files for the metal interconnects and for the dielectric. In addition, the meshes must be healthy and manifold or problems can arise in the slicing process. Blender has a built-in 3D Printing Toolbox that heals and exports an STL file for a given mesh. Simple Boolean operations were used to join interconnects into a single object as well as to subtract components and traces from the dielectric. The built-in features of the 3D Printing Toolbox were used to ensure the resulting meshes were manifold and ready to be exported.

Chapter 3: Slicing and Printing Procedures

Slicing procedures for additive manufacturing for monolithic structures have been well in place since the 1980's and 1990's [1]–[5]. The mathematical algorithms and procedures used to generate tool paths are generally very simple yet form a solid basis for different and more complex forms of slicing. The general slicing algorithm is as follows: read CAD model data, generate perimeters at a discrete height, generate tool paths of the infill and shells based on the perimeters, and repeat the previous two steps with an increasing height until the whole model has been traversed. After the tool paths have been generated, a g-code text file that describes the movements of the printer can be written. All g-code files follow a general standard (more commonly known as g-code flavor), but each printer usually only accepts a g-code file that is specific to itself. Developing algorithms that are agnostic to different printers therefore needs to take into consideration the specific g-code syntax for each printer.

The following sections describe basic slicing algorithms as well as more complex algorithms for hybrid 3D printing implemented in MATLAB. Based on the findings in [24], it can be reasonably inferred that traditionally sliced tool paths are sufficient for both fused deposition modeling (FDM) and micro-dispensing (µD) in the same print. Therefore, no special considerations are needed when designing basic slicing procedures for hybrid printing using FDM and µD. Theoretically, these procedures are also viable for other methods of combining multiple 3D printing processes within the same print.

3.1 Fused Deposition Modeling and Micro-Dispensing

The two main 3D printing technologies considered were FDM and µD. Both processes were considered because of their versatility in printing small to large devices. [24] demonstrates that printing with both FDM and µD in the same print job is entirely feasible for printing metal-dielectric structures. While other printing technologies have shown success in hybrid 3D printing, FDM and µD seemed to be an optimal choice for their versatility in printing small-to-large scale devices. Moreover, it is so far the only combination of printing processes that can be automated to a single print job.

3.1.1 Fused Deposition Modeling

FDM is arguably the most popular form of 3D printing today, being used to print small-to-large volumetric. Some examples of FDM 3D printers are Ultimakers, Lulzbots, MakerBots, and Prusa printers. FDM involves feeding spooled filament into a heated nozzle. The feeding mechanism is driven by a stepper motor connected to a teethed gear that grabs and guides the filament to the nozzle. The nozzle is heated to a temperature that melts the plastic as it is extruded from the nozzle. More stepper motors control *x*, *y*, and *z* axis movements that move the nozzle's housing (or print head). The material is extruded onto a bed. The bed is normally heated to encourage adhesion. The extruded plastic then cools to form a solid trace. This is repeated over an area to form a layer of deposited material. Another layer is extruded on top of the previous layer, and the process repeats until a full model is formed.

Ideally, when extruding tool paths that are directly adjacent to each other, there should be no imperfections between the adjacent lines. Figure 8 shows the difference between properly extruded adjacent tool paths, and improperly extruded tool paths. Properly extruded tool paths should have no "peaks" or "valleys" between the lines. Peaks are formed by extruding too much material too close between paths. Valleys are formed when extruding too little material too far apart between tool paths. By rule of thumb extrapolated from [61], ideal extrusion happens when the tool paths physical width is roughly 1.20x the nozzle diameter when the line spacing is equal to the nozzle diameter. The exact multiplier will be dependent on the material, the printer, the tool head, and the external environment.

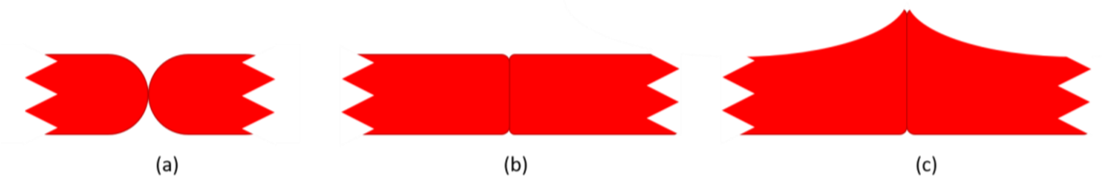


Figure 8. Extrusion examples in FDM printing. (a) Under-extrusion. (b) Proper extrusion. (c) Over-extrusion.

3.1.2 Micro-Dispensing

Micro-dispensing from nScrypt, Inc., is a printing technology that dispenses viscous material with high resolution. Micro-dispensing uses a pressurized system to push material from a syringe through a controllable valve and out a swappable ceramic pen tip. The benefit of µD is the versatility of the types of materials that can be deposited as well as versatility in printing small to large devices. Deposited materials can be as thin as a few microns with low viscosity materials or even up to a thickness of 400 µm like in typical FDM with high viscosity materials.

In this work, µD is preferred for depositing conductive traces using silver paste. Ideally, flat traces should be considerably thin, around 50 µm, to reduce the skin effect when it comes to RF applications. However, in the case of volumetric structures, thicker traces are more desirable as it provides better assurance of contract throughout the entire trace. µD is capable of doing both. Dispensing with tip sizes of around 200 µm can be easily achieved with µD as opposed to printing technologies like inkjet or aerosol jet. Micro-dispensing, able to print thin, thick, small and large structures, therefore has a good advantage in hybrid printing alongside FDM.

3.2 Configurable Parameters

Before discussing the slicing algorithms, a brief introduction to 3D printing parameters is needed. These parameters control how the print will commence. Table 1 gives a rundown of the most important and basic slicing parameters. The values of the parameters are based on basic FDM print with a nozzle diameter of 0.400 mm at 100% infill of acrylonitrile butadiene styrene (ABS) plastic. The values account for good extrusion (not over- or under-extruded), timely prints, feature size, and cleanliness. An updated table with more configurable parameters and their descriptions is given in Appendix A – Configurable Printing Parameters.

Table 1. Important FDM printing parameters.

Name	Typical Value	Units	Description
Layer height	0.200	mm	Height or thickness of each layer.
Line width	0.480	mm	Width of infill lines at 100% infill.
Line spacing	0.400	mm	Spacing between infill lines at 100% infill.
Infill angle	45	°	Angle of the infill orientation.
Print speed	20	mm/sec	How fast the tool head moves while printing.
Filament diameter	1.75	mm	Diameter of the filament.
Extrusion factor	1	N/A	Multiple of flow when extruding.
Shell count	2	#	Number of shells to be printed.
Nozzle Temperature	230	°F	Temperature of the extruder nozzle.
Bed Temperature	95	°F	Temperature of the bed.

3.3 Basic Slicing

3.3.1 Model Import

Generally, models used for 3D printing follow the standard tessellation language (STL) file format when being sliced. STL files describe models in terms of surface of triangles (facets). These triangles each have three vertices that are potentially shared with adjacent triangles. The STL file lists the facets along with their respective vertices. When importing the data from an STL file into MATLAB using the `stlread()` function, the vertices and facets are represented as arrays. The sizes of the arrays are M×3 and N×3 for facets and vertices, respectively. This can be interpreted as the model having M facets but not necessarily N vertices as some vertices may be repeated. Each row vector within the facet array contains integers that represent the indices of the rows within the vertex array. For example, a facet row vector that reads $[1 \quad 2 \quad 3]$ means that this particular facet contains the vertices stored in rows 1, 2, and 3. Figure 9 shows an example of an STL file, and Figure 10 shows a screenshot of the file's data after importing.

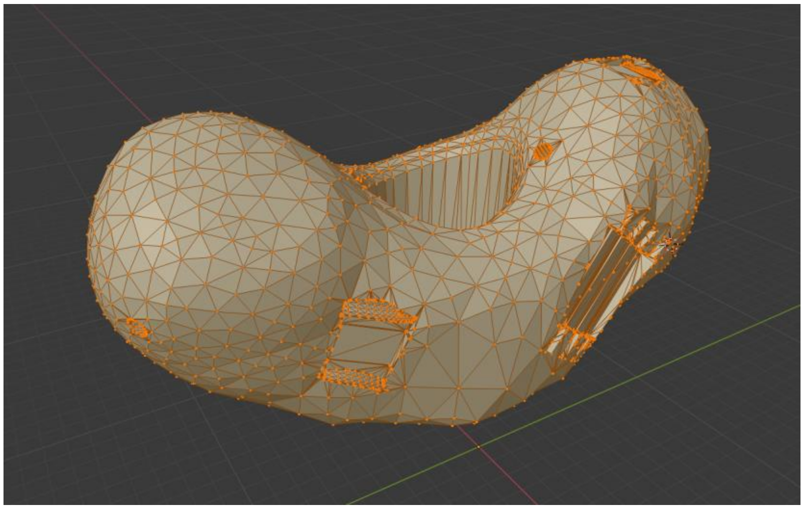
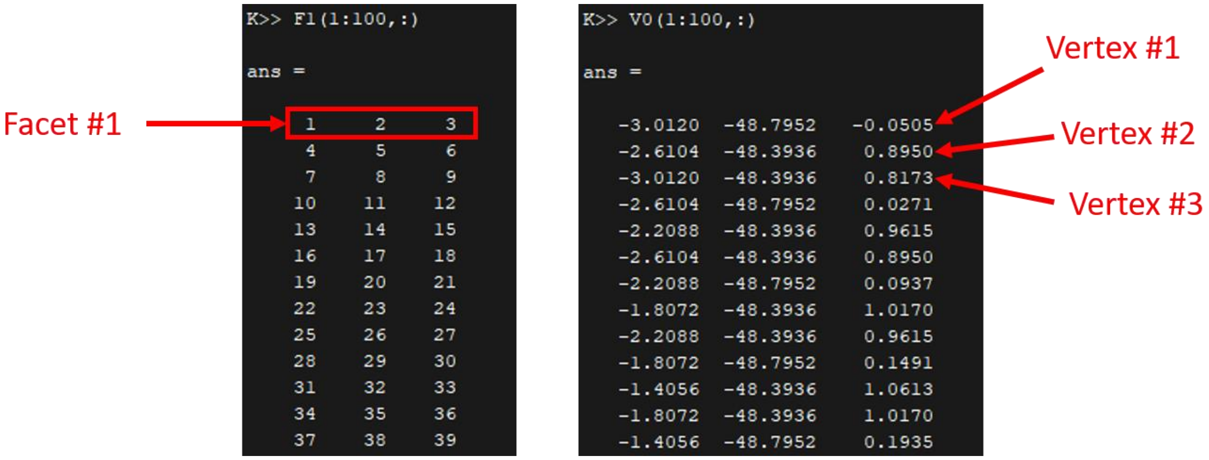


Figure 9. STL model example.



```
K>> F1(1:100,:)

ans =

     1     2     3
     4     5     6
     7     8     9
    10    11    12
    13    14    15
    16    17    18
    19    20    21
    22    23    24
    25    26    27
    28    29    30
    31    32    33
    34    35    36
    37    38    39
```

```
K>> V0(1:100,:)

ans =

   -3.0120  -48.7952   -0.0505
   -2.6104  -48.3936    0.8950
   -3.0120  -48.3936    0.8173
   -2.6104  -48.7952    0.0271
   -2.2088  -48.3936    0.9615
   -2.6104  -48.3936    0.8950
   -2.2088  -48.7952    0.0937
   -1.8072  -48.3936    1.0170
   -2.2088  -48.3936    0.9615
   -1.8072  -48.7952    0.1491
   -1.4056  -48.3936    1.0613
   -1.8072  -48.3936    1.0170
   -1.4056  -48.7952    0.1935
```

Figure 10. STL data in MATLAB. Facet #1 (Left) contains vertices #1, #2, and #3 (Right).

3.3.2 Perimeters and Shells Calculation

A perimeter in this work is defined as a closed loop of connected line segments that is used as a mathematical boundary to separate regions of infill from empty regions. These perimeters are initially generated when taking the intersection of the slicing plane $z = z_{\text{height}}$ and the STL triangle facets. Realistically, the perimeter needs to be marched inward to generate shells, which are the physical tool paths that enclose the infill. These shells are necessary for housing conductive material in embedded traces as well as creating a non-jagged surface to print conformal paths.

To generate the perimeter, the triangle facets that intersect the slicing plane first need to be identified. This is done by finding the intersecting line between the plane of the facet with the slicing plane. This intersecting line is tested whether it crosses through two edges of the triangle. The two edge crossings result in two vertices that serve as endpoints for a line segment. If the line does cross through the two triangle edges, the corresponding line segment intersection is taken from it. In certain cases where the line crosses through a point of a triangle, the point is then recorded instead of the line segment. In other cases, the slicing plane's side profile (a line) may be collinear with the triangle's edge. In this situation, the line segment describing the triangle edge is used. In total, there are five general cases of intersections: no intersection, an intersection of two edges, an intersection of a point and an edge, an intersection of just a point, and a line segment collinear with the plane. The slicing/facet plane intersection and the different cases are shown in Figure 11. Discussions on tolerances of vertex and line crossings can be found in Appendix B – Calculation Tolerances. The algorithms to determine the intersection of planes and the line segment formation are defined in Appendix C – Useful Geometric Formulas.

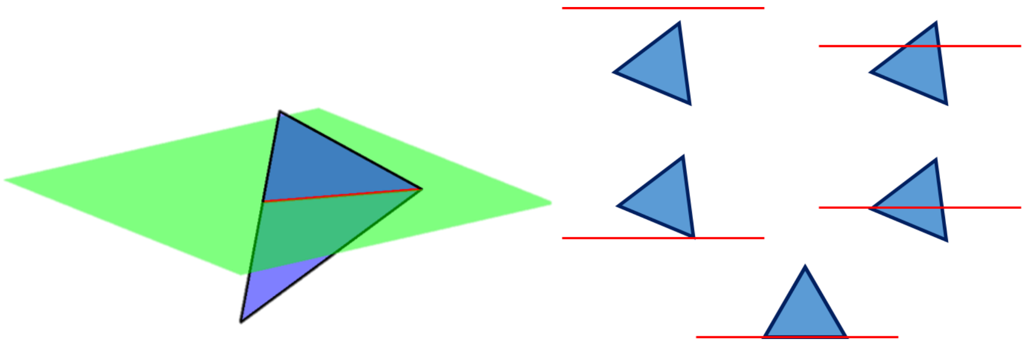


Figure 11. (Left) Plane slicing a facet. (Right) General cases of a plane crossing a facet.

Once all the intersecting line segments and points are found, the line segments and points are usually out of order if the triangle facets were not particularly ordered themselves. The line segments and points therefore need to be ordered to form a closed loop of vertices. This is done by first locating two line-segments that share a common endpoint and joining the two segments by that common endpoint. Figure 12 shows this process with colored vertices. This is done until all the line segments and vertices are checked. (Note: For a non-manifold STL model, this will likely fail as there will inevitably be a line segment without a connecting line segment. More complex algorithms involving interpolation and/or mesh fixing are needed in this case.)

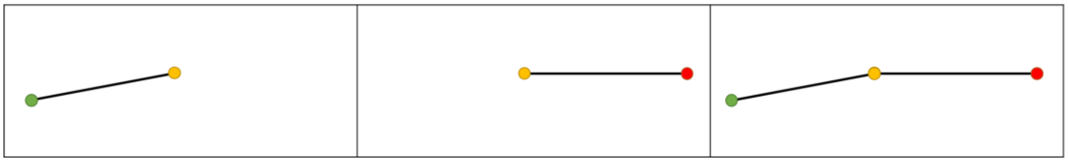


Figure 12. (Left) Individual left line segment. (Middle) Individual right line segment. (Right) Connected line segments with a common vertex.

After the initial perimeter is defined (or multiple perimeters in the case of many objects in one file), the shells need to be calculated. In order to calculate the shells with relative ease, the fast marching method [62] was used to generate a distance map of the distance to the initial perimeter. The shells are located at integer multiples of the line width on this distance map. Taking the contours of the distance function using MATLAB's `contourc()` function at these levels results in the desired shells. Figure 13 shows this. The resultant shells are then appended to a big tool path list that will be used when generating the g-code file. Each element in the list is an array that contains the vertices of a single tool path.

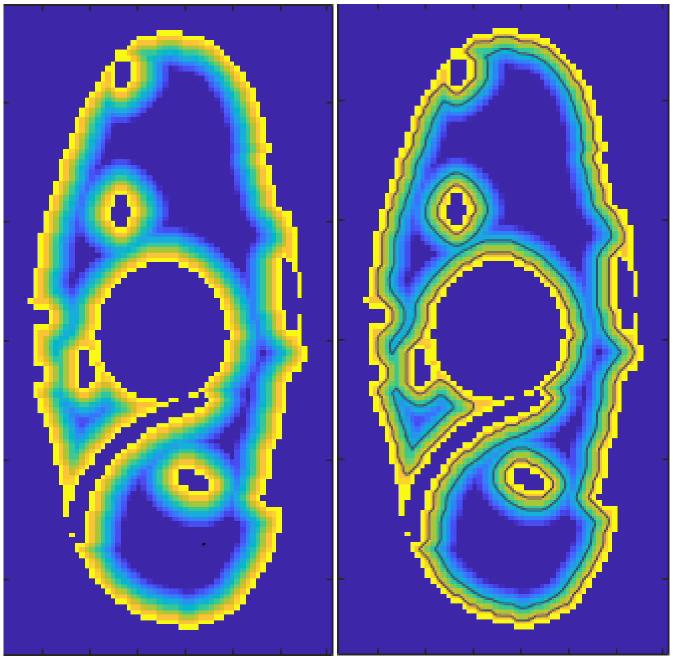


Figure 13. (Left) Distance map of perimeters generated by the fast marching method. (Right) Resultant shells taken from contours.

Before the infill generation, an updated perimeter needs to be calculated to form a boundary for the infill. This is done by taking one more contour one-half line width away from the previous shell. When the infill meets the perimeter and turns direction, one-half a line width is deposited during the turn. Therefore, one-half line width is used to ensure a completely filled layer.

In some cases, resultant perimeters can be small and cannot be printed accurately. These perimeters encompass an area that is negligibly small. In the case of these small perimeters, a tolerance can be given to filter them out. This tolerance can be based on the length of the perimeter and/or the area of the enclosed space. If the perimeter length is less than the tolerance that was pre-defined, or if the area of the perimeter is less than the cross-sectional area of the printed material, the small perimeter can be discarded.

3.3.3 Infill Calculation

Infill is the material that makes up the interior of the print. There are several types of infill patterns, such as rectilinear, hexagonal, rectangular, triangular, and Hilbert fill to name a few [63]. Infill is necessary for structural support when printing mechanical items as well as dielectric homogeneity when printing electronic and EM materials. Passive EM devices such as photonic crystals, diffraction gratings, and dielectric waveguides rely on material homogeneity to maintain proper wave propagation. Inhomogeneity can lead to unwanted scattering and anisotropy that can affect the integrity of the wave. Alternatively, anisotropy and scattering may be desired by introducing periodic artifacts in the material or alternating high and low dielectric material. Either way, the deposited material needs to be free of cavities. Therefore, for EM applications, an infill of 100% is highly desired. The simplest way to generate tool paths for 100% infill is creating rectilinear line segments and connecting them at the perimeters. Figure 14 shows the infill generation process step-by-step.

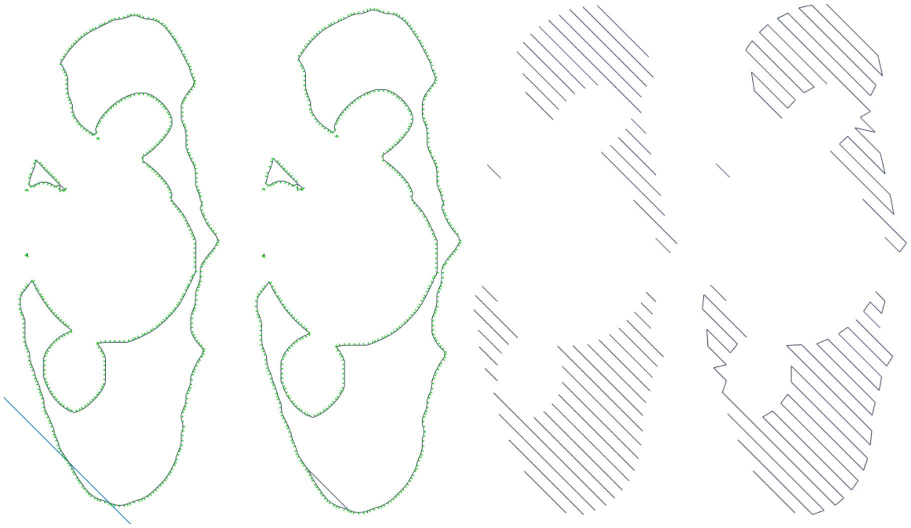


Figure 14. (Left to Right) Steps of generating infill. 1. Draw a line through the perimeter. 2. Clip the line at the perimeter. 3. Repeat until the slice has been covered. (Perimeter not shown.) 4. Stitch the lines together via the closest line segment.

To begin the infill generation, a line segment is constructed bounded by the maxima and minima of the perimeters. By popular convention, this line segment is perpendicular to the defined infill angle from the slicing parameters. Once constructed, intersections of the line segment with the perimeters are searched for. This is results in four scenarios: (1) no intersections are found, (2) one intersection is found, (3) two intersections are found, and (4) more than two intersections are found. If no intersections are found, the line segment can be discarded. If one intersection is found, the line segment is crossing a perimeter tangentially and does not lie within the perimeter. Therefore, the line segment is discarded. If two intersections are found, the line segment is clipped at the perimeter intersections, resulting in a segment that lies within the perimeter. If more than two intersections are found, the segment is divided into multiple segments with the intersections as endpoints. Ideally, an even number of intersections will be found and result in neatly clipped segments. Although rare, it is possible that an odd number of segments is found, indicating that the line segment crosses into, out of, and tangentially to a perimeter. In this situation, the tangential intersection is removed, leaving only the interior segment.

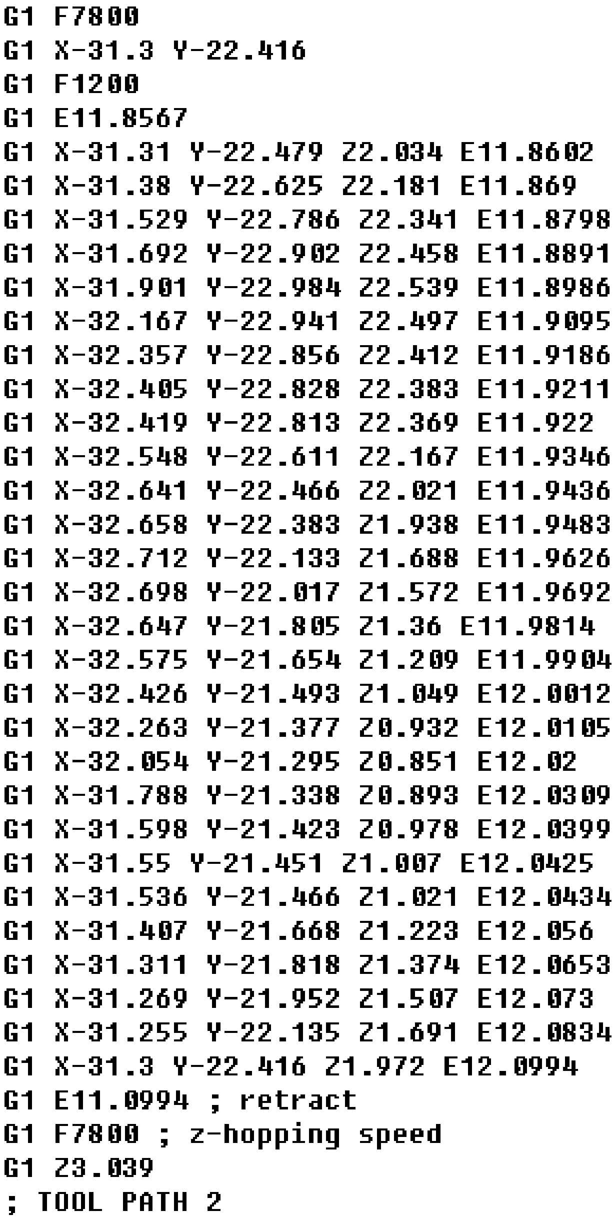
Once the line segment is clipped, a second line segment is drawn parallel to the original line segment but is positioned one line width away. The same clipping procedure is applied to this next segment. This is repeated with more line segments until the entire slice has been traversed and filled. Once the clipped line segments are obtained, they are attached together via the closest endpoint of the adjacent line segment within a tolerance. This continues to form an infill tool path that zig-zags across the interior of the model. If no segments are found within the vicinity of the tool path end points, the tool path is finished, and a new line segment is chosen as the start of a new tool path. Once the line segments have been exhausted, all the tool paths have been generated and get appended to the main tool path list.

3.3.4 Looping across the Model

Once the tool paths for a single layer have been generated, the same algorithms are applied to the next layer. The next layer is vertically above the previous layer. This process continues on until the entire model has been traversed. The tool paths that are generated are all appended to the main tool path list. Once the entire model has been traversed, the resulting tool path list contains all the tool paths that the printer will use to construct the model.

3.3.5 G-code Generation

G-code generation is the final step in slicing and preparing for printing. A g-code file is a text file that contains a list of simple commands that drive the printer. The g-code generation procedure involves taking the data from the main tool path list and converting it into lines of text that describe the printer’s movements, operating states, temperatures, file syntax, motion preferences, and printer specific initialization commands. The g-code generation is a three-step process: (1) write the header, (2) write the body, and (3) write the footer. The header contains all the information needed to initialize the printer and its tools for the print. The body contains all the movement and extrusion commands that drive the tools. The footer contains the commands necessary to stop the tools and bring the printer to a proper stop. The header and footer usually contain commands that are specific to the printer while the body contains the information needed for extrusion, speed, and position of the printer's tools. Figure 15 gives a screenshot of a typical g-code file.



```
G1 F7800
G1 X-31.3 Y-22.416
G1 F1200
G1 E11.8567
G1 X-31.31 Y-22.479 Z2.034 E11.8602
G1 X-31.38 Y-22.625 Z2.181 E11.869
G1 X-31.529 Y-22.786 Z2.341 E11.8798
G1 X-31.692 Y-22.902 Z2.458 E11.8891
G1 X-31.901 Y-22.984 Z2.539 E11.8986
G1 X-32.167 Y-22.941 Z2.497 E11.9095
G1 X-32.357 Y-22.856 Z2.412 E11.9186
G1 X-32.405 Y-22.828 Z2.383 E11.9211
G1 X-32.419 Y-22.813 Z2.369 E11.922
G1 X-32.548 Y-22.611 Z2.167 E11.9346
G1 X-32.641 Y-22.466 Z2.021 E11.9436
G1 X-32.658 Y-22.383 Z1.938 E11.9483
G1 X-32.712 Y-22.133 Z1.688 E11.9626
G1 X-32.698 Y-22.017 Z1.572 E11.9692
G1 X-32.647 Y-21.805 Z1.36 E11.9814
G1 X-32.575 Y-21.654 Z1.209 E11.9904
G1 X-32.426 Y-21.493 Z1.049 E12.0012
G1 X-32.263 Y-21.377 Z0.932 E12.0105
G1 X-32.054 Y-21.295 Z0.851 E12.02
G1 X-31.788 Y-21.338 Z0.893 E12.0309
G1 X-31.598 Y-21.423 Z0.978 E12.0399
G1 X-31.55 Y-21.451 Z1.007 E12.0425
G1 X-31.536 Y-21.466 Z1.021 E12.0434
G1 X-31.407 Y-21.668 Z1.223 E12.056
G1 X-31.311 Y-21.818 Z1.374 E12.0653
G1 X-31.269 Y-21.952 Z1.507 E12.073
G1 X-31.255 Y-22.135 Z1.691 E12.0834
G1 X-31.3 Y-22.416 Z1.972 E12.0994
G1 E11.0994 ; retract
G1 F7800 ; z-hopping speed
G1 Z3.039
; TOOL PATH 2
```

Figure 15. G-code screenshot.

Each command follows a syntax of "G# [arguments]" or "M# [arguments]". "G" commands typically indicate some kind of movement of the stepper motors while "M" commands directly affect firmware or other non-movement commands. The "G" and "M" flags are followed by a number that indicates what command to execute. The arguments that follow the "G#" give the proper inputs for the commands themselves. A comprehensive list of g-code commands can be found in [64].

Writing the g-code file is a straightforward process. The configurable parameters to initialize the printer are written first in the header. These commands usually consist of setting the nozzle temperature, setting the bed temperature, enabling absolute or relative extrusion commands, and resetting the axes.

Next, the body is written based on the main tool path list consisting of the shell and infill tool paths. The most common command in this part of the file is the "G1" command followed by the arguments "X#", "Y#", "Z#", "E#", or "F#". The "X", "Y", and "Z" values represent the point the tool head will move to. The "E" value represents the length the heated thermoplastic material needed to be extruded. The "F" command defines the feed rate, or speed, of the movement. The "X", "Y", and "Z" values of the g-code commands are taken directly from the vertex information within the main tool path list. The "E" command defines the length of material that needs to be extruded. The value is taken from Equation (1) where $f_{\text{extrusion}}$ is the extrusion factor, l is the length of the tool path line segment, d_{filament} is the filament diameter, h is the layer height, and w is the line width. (The extrusion calculation is described in Appendix D – Extrusion Math [61].)

$$l_{\mathrm{E}} = f_{\text{extrusion}} \frac{4l}{\pi d_{\text{filament}}^2}\left(h(w-h) + \pi\left(\frac{h}{2}\right)^2\right) \tag{1}$$

Finally, the stopping commands are written in the footer. These commands including cooling down the nozzle and bed and moving the tools to safe positions. A commented list of the parameters used is usually added to the end of the footer.

When writing the g-code file, it is advisable to limit the number of calls to a specific command. Consider the example in Figure 15. A "G1" flag can be followed by different arguments for extrusion. In this case of printing infill where concurrent line segments are printed at the same speed, it is not recommended to include an “F” flag defining the same speed for each command. This not only makes the g-code file unnecessarily larger, it also sets the speed each time in the firmware, which can lead to the firmware crashing.

3.4 OFF-AXIS SLICING

While the progression of layers is vertical for typical 3D printing, off-axis printing involves dispensing material layer-by-layer at an angled, non-vertical progression of layers. A typical 3D printing process only dispenses in the x and the y direction while moving up layer-by-layer in the z direction. Off-axis printing can print in all three x, y, and z directions in a plane that is not parallel to $z = 0$. The motivation for pursuing off-axis printing is the possible advantages offered by it. Different structural mechanics can be designed around differing printing directions [65], [66]. Conductive paste can be dispensed on a smoother surface as opposed to staircasing along a slope, reducing the effects of surface roughness [13]. Finally, material usage can be reduced when printing overhanging parts by eliminating the need for supporting structures [67]–[69]. These processes however were made on single FDM printing processes. The algorithm presented in this section builds on that idea by applying it to a dual FDM process.

3.4.1 Off-Axis Algorithm

The off-axis algorithm is based on the fact that off-axis tool paths are very similar to those generated with the basic slicing routines. The same slicing engine can be used to slice in an off-axis manner if the STL model is rotated before slicing. Therefore, right after the STL data is loaded, the model is rotated to the desired orientation of slicing. The basic slicing engine can then be used to slice the rotated model. The resultant tool paths are then rotated back to the model’s original orientation, resulting in off-axis tool paths.

3.4.1.1 Rotation Matrices

A rotation matrix is a type of matrix that performs rotation of elements, such as a vector or vertex, when multiplying that element. In 3D space, these matrices take the form of

$$\mathrm{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{2}$$

$$\mathrm{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{3}$$

$$\mathrm{R}_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where the angle θ is the angle of rotation. Each matrix rotates the element around the corresponding *x*-, *y*-, or *z*-axis with the R_x, R_y, and R_z matrices respectively. If a vertex takes the form of a column vector

$$\mathrm{V} = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \tag{5}$$

then the rotated vertex can be calculated as

$$\mathrm{V_r} = \mathrm{RV} \tag{6}$$

Often times, a vertex needs to be rotated in more than one axis. This is easily achieved by cascading multiplying rotation matrices. The order of multiplication of rotation matrices determines the order of rotation. The resultant matrix will perform the rotations based on which matrix the vertex is multiplied by first. For example, if a vertex is desired to be rotated about the *x*-axis followed by the *z*-axis, the composite rotation matrix can be calculated as

$$\mathrm{R}' = \mathrm{R}_z\mathrm{R}_x \tag{7}$$

3.4.1.2 Model Rotation

The desired off-axis direction of slicing can be achieved using the azimuthal angle ϕ and the elevation angle θ based on spherical coordinates. These angles make up the direction of a three-dimensional vector v which points in the direction of off-axis slicing. To properly rotate the model to slice in the desired direction, v needs to be rotated by some θ_z and θ_y such that v now points in the positive *z* direction. Figure 16 shows this idea. This can be done by first rotating the model azimuthally to align v with *xz*-plane by some θ_z followed by a rotation in elevation by θ_y. The angles θ_z and θ_y are therefore used as input angles for rotation matrices that will rotate the model. These angles can be calculated from the spherical coordinate angles by the relations

$$\theta_y = \frac{\pi}{2} - \theta \tag{8}$$

$$\theta_z = -\phi \tag{9}$$

The order of rotation must be about the *z*-axis first before the *y*-axis. The final rotation matrix will therefore be

$$\mathrm{R} = \mathrm{R}_y(\theta_y)\mathrm{R}_z(\theta_z) \tag{10}$$

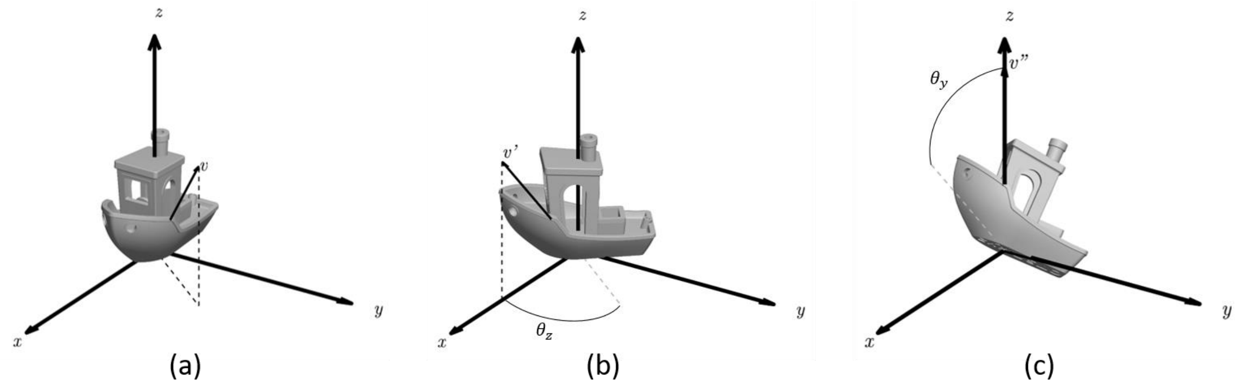


Figure 16. Off-axis rotation. (a) Unrotated model. (b) Azimuthally rotated model. (c) Elevated rotated model.

The model vertex data imported from the STL file is stored as an N×3 matrix. The model can therefore be rotated by multiplying the transverse vertex matrix by the final rotation matrix. (Before multiplying, the lowest *z* value of the STL vertices should be saved for a future step.) Occasionally, some of the newly rotated vertices will cross below the *xy*-plane and have negative *z* values. This is undesirable, as the slicer will ignore these regions of the model. In this cases of the model crossing below the *xy*-plane, the absolute value of the lowest resultant *z* value is added to all the *z*-values of the vertices. This value is then saved for a later step. Figure 17 shows these steps of rotation and offset. Once the new vertex matrix is calculated, the normal slicing routines can begin generating tool paths.

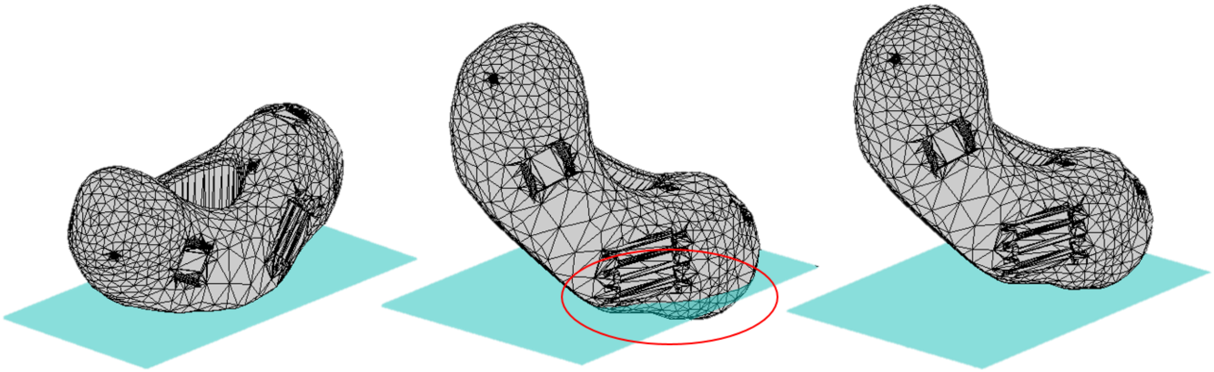


Figure 17. (Left) Imported model. (Middle) Model rotated with a portion crossing the *xy*-plane. (Right) Rotated model offset in *z* so it sits on the *xy*-plane

Once the tool paths are calculated, the *z* value saved from before is applied by subtracting it from the tool path *z* values. Finally, new rotation matrices must be generated that rotate the tool path vertices in the reverse order. This is easily done by creating a new rotation matrix with negative angles multiplied in reverse order

$$\mathrm{R}_{\mathrm{reverse}} = \mathrm{R}_z(-\theta_z)\mathrm{R}_y\left(-\theta_y\right) \tag{11}$$

and multiplying it by the transverse matrix holding the tool path vertices.

3.4.2 Off-Axis Implementation and Testing

The algorithm described in the previous section was incorporated and tested on a dual-extrusion Ultimaker3. Due to the firmware movements of the Ultimaker3, points needed to be interpolated when moving in all *x*, *y*, and *z* directions. If the points are too far apart, the Ulimaker3's firmware will not allow it to travel in a straight line. Therefore, more points are needed when moving the tool head. Figure 18 gives an example of this. The newly interpolated points do not interfere with the printing and can be incorporated into other printers if similar issues arise.

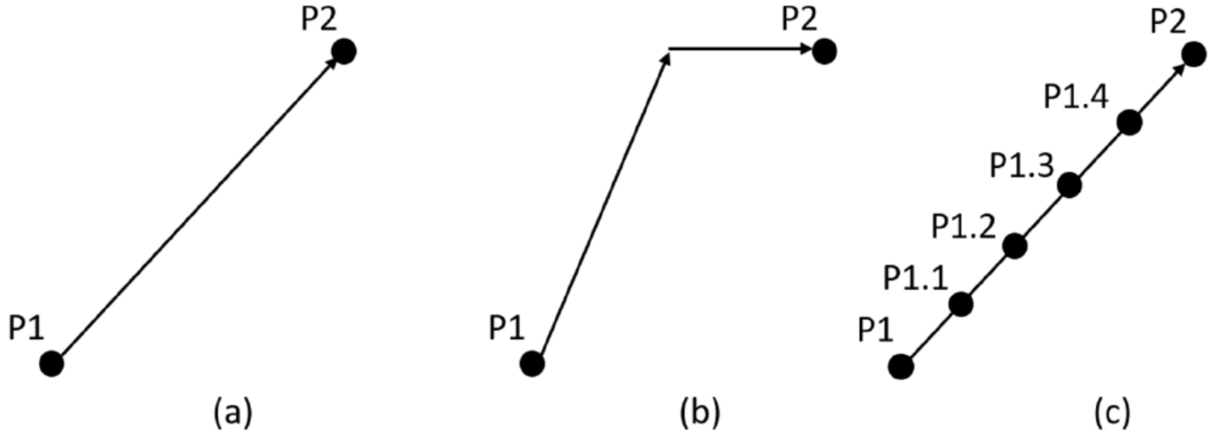


Figure 18. (a) Typical linear movement between two points, (b) Ultimaker3 movement between two points, (c) Corrected Ultimaker3 movement between two points with additional collinear points.

Unlike in [69] where only one material extruder was tested, these off-axis capabilities were tested by printing two different dual-material structures on an Ultimaker3. Both structures each include two separate areas that were sliced, printed in transparent PLA plastic and blue PLA plastic. The goal was to demonstrate that off-axis printing can be incorporating within the same printing process as normal planar 3D printing in a dual-extrusion setting. The different areas of plastic each represent the different areas that were sliced, both normally and with off-axis.

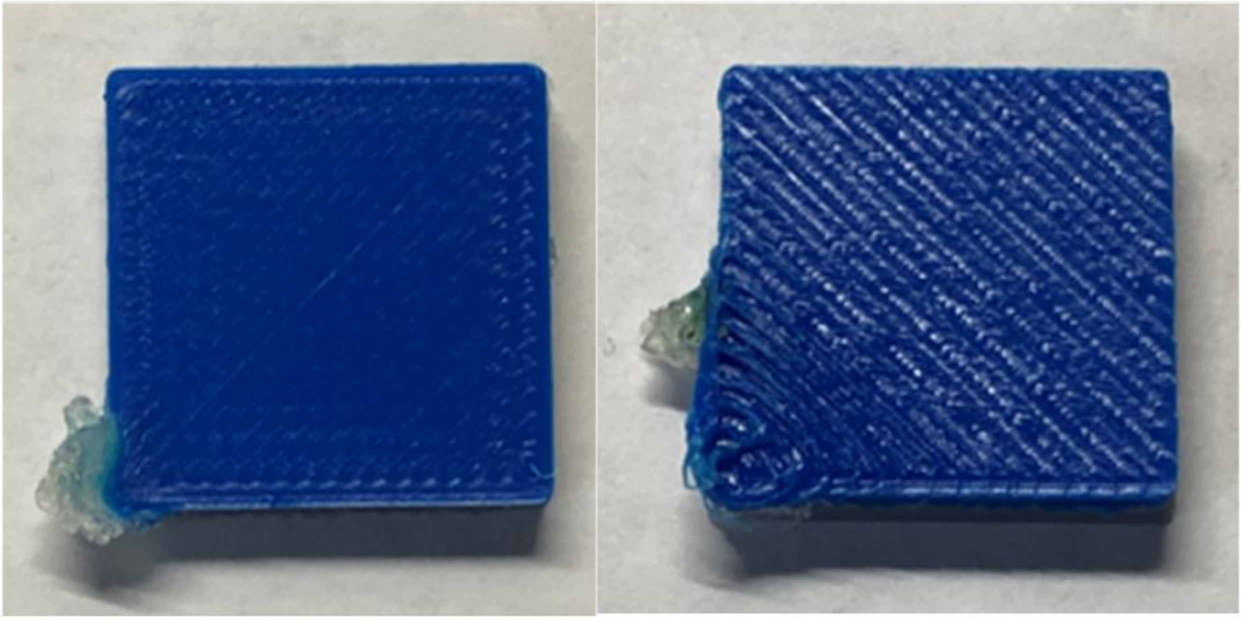


Figure 19. (Left) Dual off-axis print, transparent off-axis and blue normal slicing. (Right) Dual off-axis print, transparent normal slicing and blue off-axis print.

The first printed structures are shown in Figure 19 and Figure 20, which show two slabs of PLA plastic printed on top of each other. The left images have transparent PLA printed off-axis as the bottom slab at a 15° angle and blue PLA printed normally on the top slab. The right images have the reverse case. The bottom transparent PLA is printed normally, and the blue PLA is printed off-axis at a 15° angle. Both of these prints demonstrate that the off-axis capabilities can be used in any part of the print and not limited to a single area.

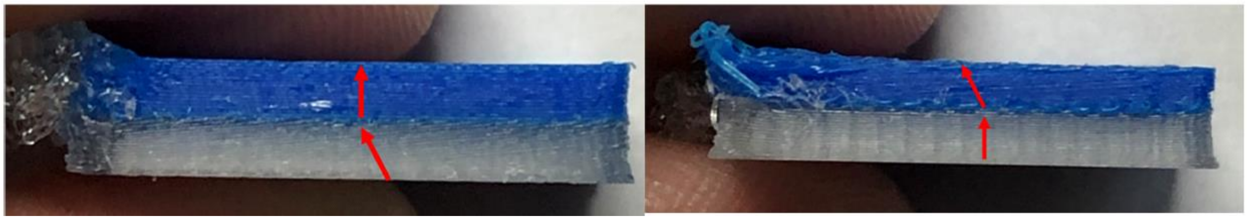


Figure 20. Off-axis print side view. (Left) Dual off-axis print, transparent off-axis, and blue normal slicing. (Right) Dual off-axis print, transparent normal slicing, and blue off-axis print.

The next device printed was a model of two different materials with a 90° overhanging structure shown in Figure 21 and Figure 22. It was decided to recreate a model given in [69] to test the boundaries of the off-axis printing capabilities with two extruders. This demonstration shows the advantage of printing off-axis in the sense of reducing the amount of material needed. Normally, a device with an overhanging structure needs support material to be printed underneath. Printing off-axis allows for no material to be printed as supporting structures. This demonstration does not exhibit optimized parameters, as the print itself looks messy. However, the result is sufficient to demonstrate the capabilities and advantages that off-axis printing offers with multiple materials.

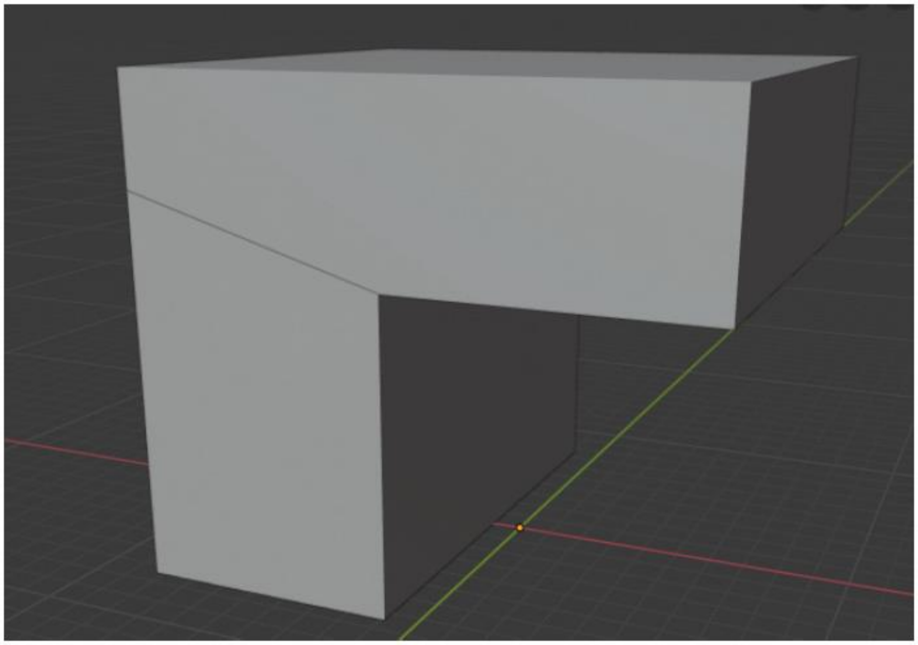


Figure 21. 3D model with long overhanging structure.

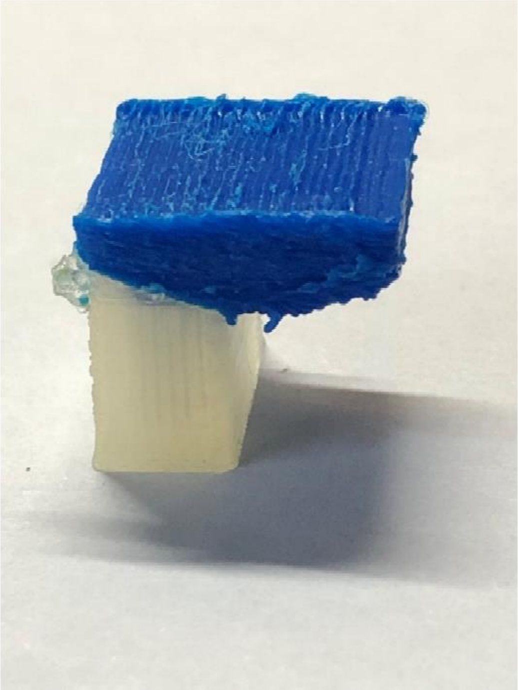


Figure 22. Off-axis dual printed model of long overhanging structure.

Off-axis printing does pose some considerations that need to be taken during the parameter decision process. The most apparent is the angle of printing with regards to the size of the model. The angle of printing off-axis depends solely on the profile of the printer's print head along with the housing. This angle is limited by the dimensions of the model relative to the dimensions of the print head. This means that smaller areas to be printed off-axis can tolerate a steeper angle than a larger area. Moreover, the position of the second print nozzle can also hinder where the printing takes place and even cause unwanted material to be extruded. This case is shown in Figure 19 where the excess transparent PLA was deposited onto the top slab after coming into contact with the printed structure. More on this and other issues and how to mitigate them can be found in [69].

3.5 CONFORMAL SLICING

Conformal slicing is the process of slicing a model where the resultant tool paths conform to a curve or a surface and therefore do not have the same *z* value [70]–[73]. Typical conformal path planning has been achieved with the use of a *z*-scanner, which maps out the surface desired to be printed on into a rasterized array of height values. Tool paths are generated separate from creating the rasterized array and are adjusted post-slicing within the printer driver software. This process is usually expensive as the *z*-scanner quality may vary depending on the model, algorithms, and the surface itself when generating the rasterized array. Moreover, very few printers aside from nScrypt printers [74] and specialized machines have the hardware capabilities for this approach.

Conformal slicing also has ties to non-planar tool path planning such as in [73]. Non-planar tool path planning involves designing tool paths that transition from planar tool paths to non-planar tool paths. By the end of the print, the tool paths that are generated completely conform to the surface of the model. [73] also demonstrates that the resultant tool paths give the finished model a smoother finish. This is to reduce the number of discretized layers that appear at the slopes of the curvature. It is apparent that tool pathing from a single model can forgo the need for a *z*-scanner and expensive printer hardware and print immediately within the same print job. [72] demonstrates this capability very well.

A common problem when it comes to conformal printing is the limitation of printable regions due to the print head's dimensions, as shown in Figure 23 and Figure 24. This means that conformal printing is limited to non-steep areas determined by the dimensions of the print head. This is typically resolved by printing different parts of a device one-at-a-time and assembling it post-process. Another way is by limiting the design to the printable region. Finally, five- and six-axis machines are used to be able to print on steep curvatures. While these are viable solutions, they are either time consuming, design limiting, or expensive in contrast to fully printing a device with a basic three-axis system. This section, therefore, describes a novel solution for conformal printing that is not limited to surface curvature or the print head's dimensions and can be run on a typical three-axis machine.

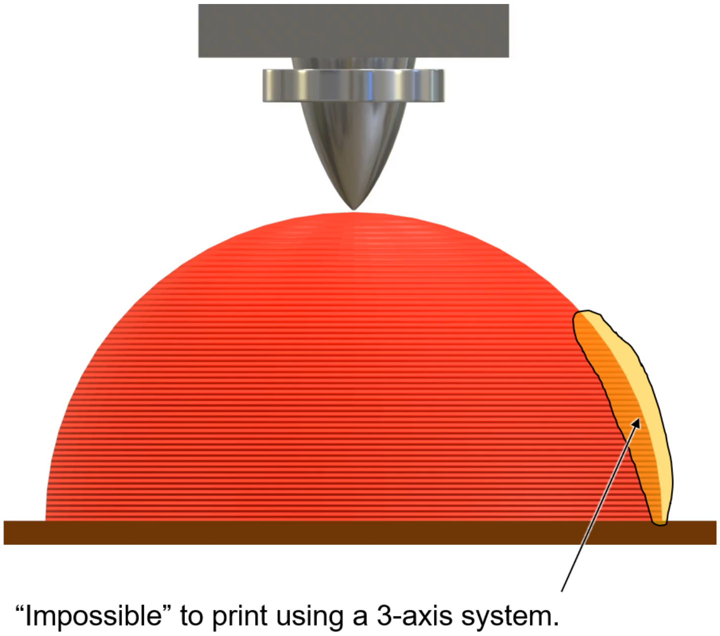


Figure 23. "Impossible" printing region when conformal printing with a three-axis system.

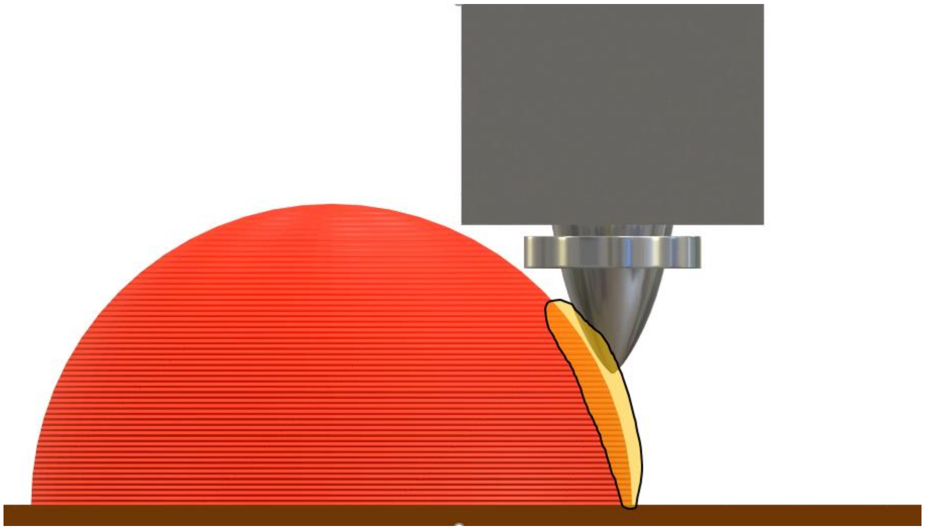


Figure 24. Tool head trying to print in the impossible region.

3.5.1 Model Preparation

Conformal slicing firstly works best when slicing an STL-defined surface rather than a volume as shown in Figure 25. This is because it is simple to take the outer edges of the STL and use them as perimeters. Otherwise, the perimeters would need to be calculated in three dimensions as opposed to a plane. With this said, the only consideration needed for the STL surface to be suitable for conformal slicing is that it must have no stray vertices or edges. Ideally, the STL model should not be volumetric and should conform completely to the target surface.

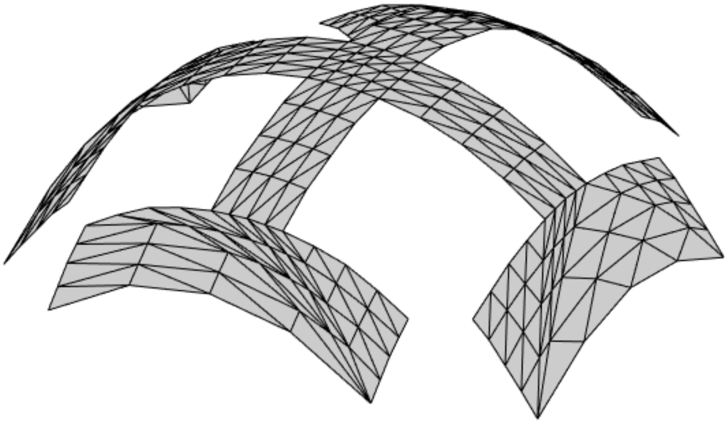


Figure 25. Curved STL model.

Properly conforming the model to a target surface can be easily done in Blender with the "Knife Project" tool. The tool projects the model onto the target surface by creating an imprint of the model within the facets of the target surface. Figure 26 shows this action. The advantage of the "Knife Project" tool is that the resulting facet arrangement exactly matches the arrangement of the surface. When the shells are generated from this arrangement, the paths will follow the slopes of the individual facets, making a truly conformal tool path to that surface.

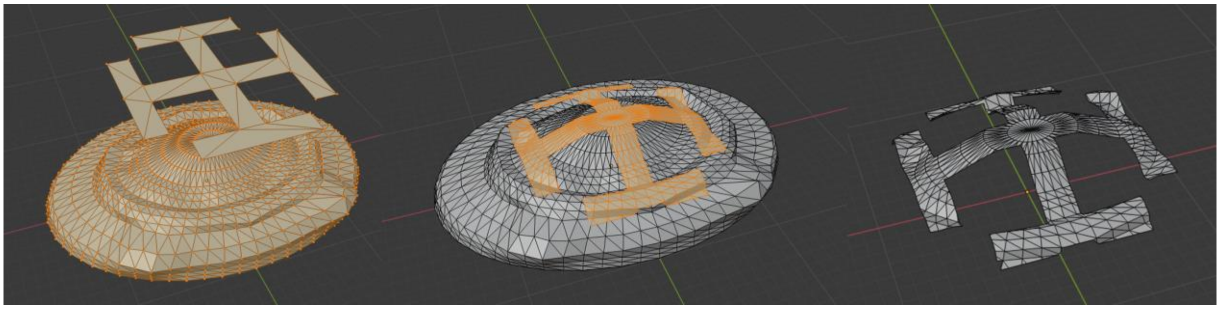


Figure 26. Conforming a model to a surface using Knife Project in Blender. (Left) Model above target surface. (Middle) Knife Project used on the model. (Right) Extracted conformal surface.

3.5.2 Conformal Algorithm

Once a suitable model has been generated, the conformal slicing algorithm can begin. This algorithm utilizes much of the basic slicing engine. First, the vertex coordinate values of the STL surface need to be saved separately from the normal vertex array. Once they are saved, the vertices can then be projected down to the *xy*-plane preserving the *x* and *y* coordinate values. This is shown in Figure 27. Following this, the perimeters need to be calculated. This is easily done by assigning the boundary edges of the model as the perimeters. Once these perimeters are identified, the shell and infill generation can commence.

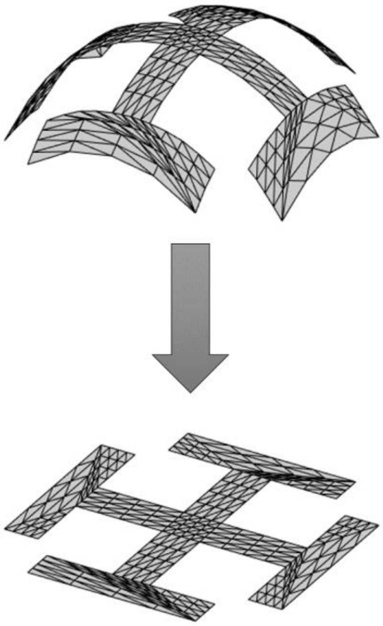


Figure 27. Projection of curved surface onto the *xy*-plane.

Once the shells and infill are generated, the vertices of the tool paths need to be interpolated according to the intersections between the tool path line segments and the edges of the facets. If no interpolation occurs, there will be tool paths that cross into the surface, potentially causing a crash while printing. The interpolation simply involves finding the intersections of the tool paths and the facet edges of the surface and adding those vertices to the tool paths in their corresponding order. Figure 28 shows how the interpolation occurs.

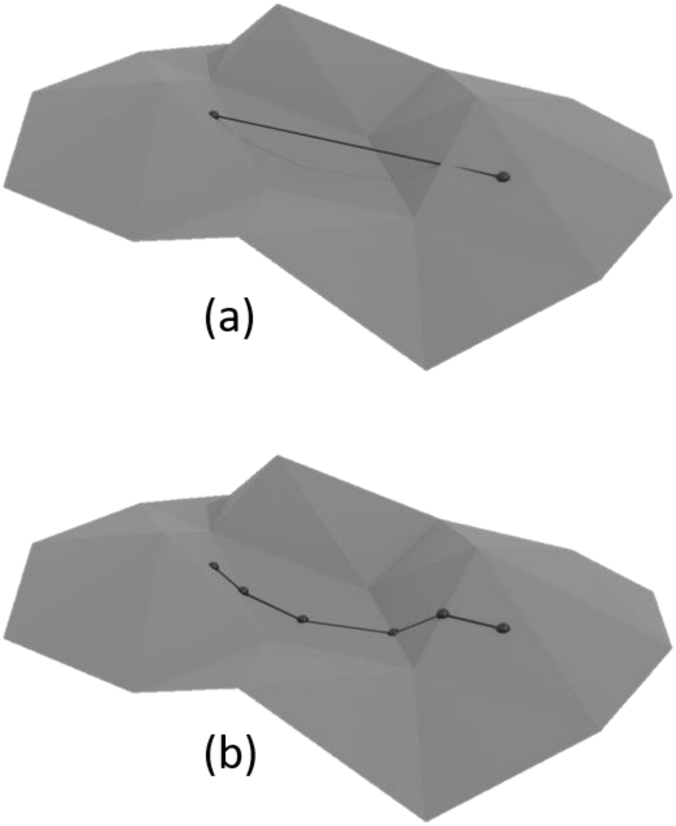


Figure 28. Tool path interpolation for conformal printing. (a) Tool path across facets. (b) Interpolated tool path across facets.

The shells and infill are still planar at this point. To return them back to the correct *z* values, the vertices are projected back up to the facets of the original conformal model. This is done by taking each vertex and projecting it back up to their corresponding facets of the conformal surface. Because each shell lies within the original surface perimeter, there should be no tool path vertex that does not cross a facet.

After the new *z* values are assigned, an extra offset (the layer height equivalent) must be applied to the newly raised vertex, as shown in Figure 29. This is due to the dimensions of the tool head that will be following the tool path. Without the offset, the tool path will position the tool head center at the surface, causing a part of the tool head to crash into the surface.

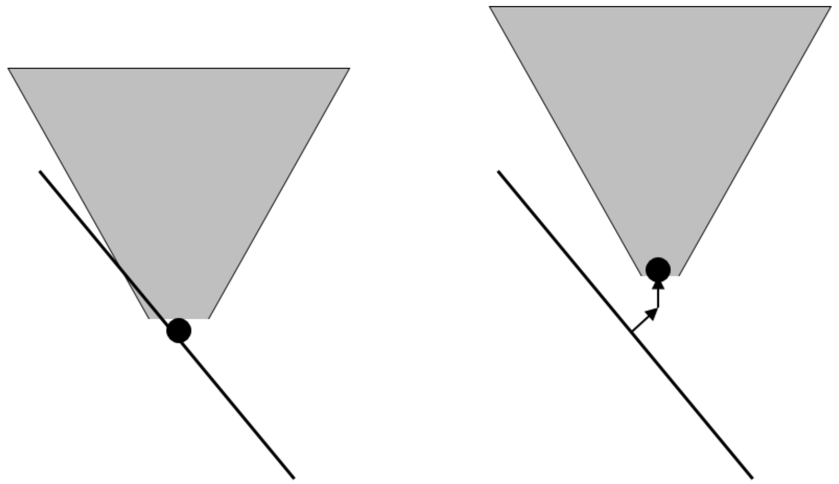


Figure 29. (Left) conformal tool path vertex with no offset, causing the pen tip to intersect with the surface. (Right) Offset applied using the normal of the facet. The normal component and vertical component are both shown.

The direction of the offset has a component in the vertical *z* direction and one component in the direction normal to the facet. The vertical component serves to compensate for low viscosity materials that could smear easily. The normal component serves as the primary component to move the tool head away from the surface. Realistically, the values of the components will vary depending on the slope of the surface, the extrusion rate of the material, and the expansion of the target surface material. If the printing material is viscous enough (or solid like a thermoplastic), subsequent layers can be expanded by this offset. Finally, Figure 30 shows different sets of tool paths that conform to a curved surface.

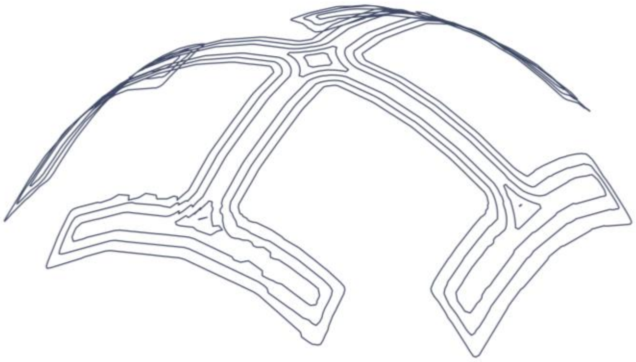


Figure 30. Conformal tool paths.

3.5.3 Conformal Algorithm for Steep Angles

One of the biggest issues in conformal printing with three-axis machines is the inability to print on steep angles. This is due to the problem shown in Figure 23. For very steep angles, it is physically impossible for the tool head of a three-axis machine to print conformally with the previous algorithm. This is due to the angle of the surface being smaller than the angle of the tool head. If the angle of the surface is less than the angle of the too head, the tool head cannot physically print the tool paths. Typically, this is overcome by using a five- or six-axis machine or redesigning the device entirely. A possible solution that uses a three-axis machine involves slicing a surface much like the standard slicing algorithm. Instead of using projections to generate tool paths, the surfaces are sliced much in the same way as a typical STL model. The slicing plane is intersected with the model facets, and the intersecting line segments are used as tool paths. Each intersection is joined to its adjacent neighbor, and the sets of the tool paths are then assigned to the big tool path list. Figure 31 shows this.

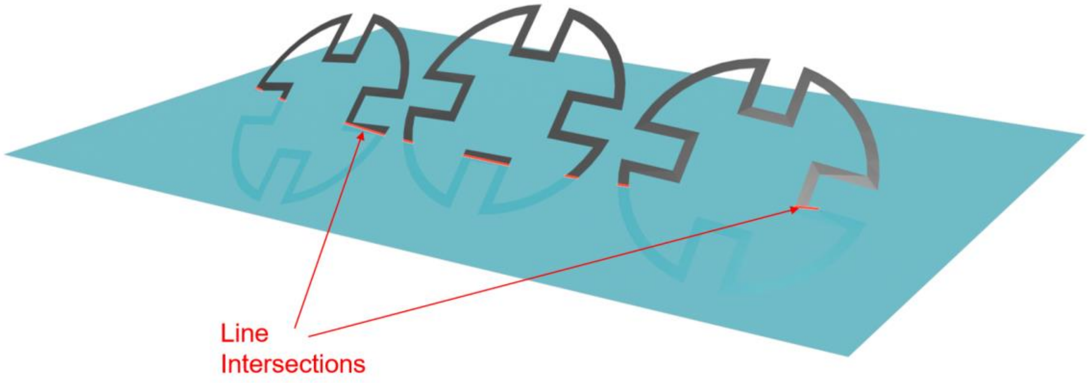


Figure 31. Slicing plane intersecting with the surface STL. Intersections are highlighted in red.

Once tool paths are generated, a dedicated offset is applied to these tool paths as well. This offset points in the direction normal to the shells of the surface model. This effectively moves the tool head out normal to the surface. The value of the offset should be configured so that the tool head prints just beyond the edge of the surface. Ideally, one-half to two-thirds of the tool head should lay beyond the surface, and one-third to one-half should still stay over the surface. This is to ensure that the deposited material adheres properly to the surface. Realistically, this offset will vary depending on the material and surface resolution.

3.6 Conformal Printing Implementation and Testing

To test the conformal printing algorithms, three types of examples were printed. The first example shows a randomly generated hill that a Jerusalem cross was printed on. The hill was designed to have random peaks and valleys to test the robustness of the normal conformal printing algorithm. The hill was printed on an nScrypt TableTop 3Dn hybrid printer with ABS plastic and DuPont CB028 conductive paste. The resultant device is shown in Figure 32. Some inconsistencies in printing can be seen that was attributed to the age of the ink. The primary purpose of this test was to ensure that the conformal slicing algorithm was viable across a randomly generated curved surface. The second surface in Figure 32 shows a small array of ring loop elements. The fidelity of the elements are much better when newer paste was used.

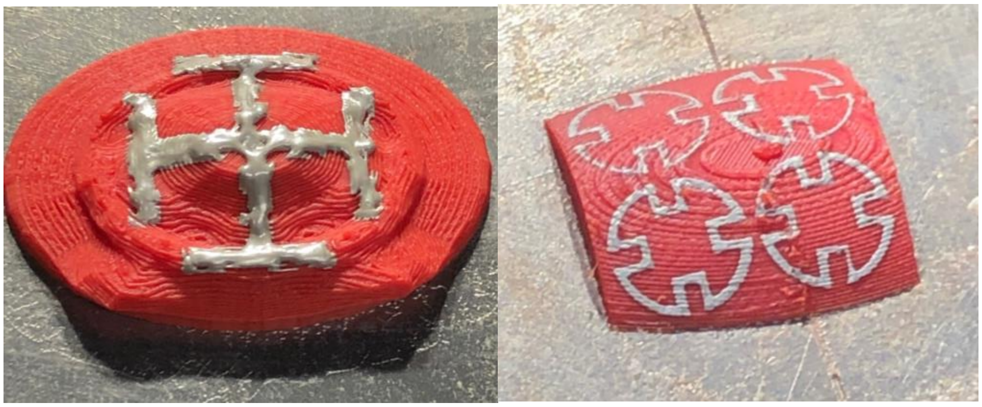


Figure 32. (Left) Jerusalem cross conformally printed on a random hill. (Right) Small array of ring loops printed on a curved surface.

The third device that was printed was a Jerusalem cross on a steep wall. Four of the same devices were printed 90° from each other, reminiscent of Stonehenge. This was to test that the calibration and offset of the SmartPump™ tool were both configured correctly. Figure 33 shows the printed devices. Small defects can be seen in the left image where there appears to be some staircasing and extra width. This was attributed to not using an offset for the end vertices of the individual tool paths. Once the offset was implemented, the dimensions of the crosses resolved, shown in the right image.

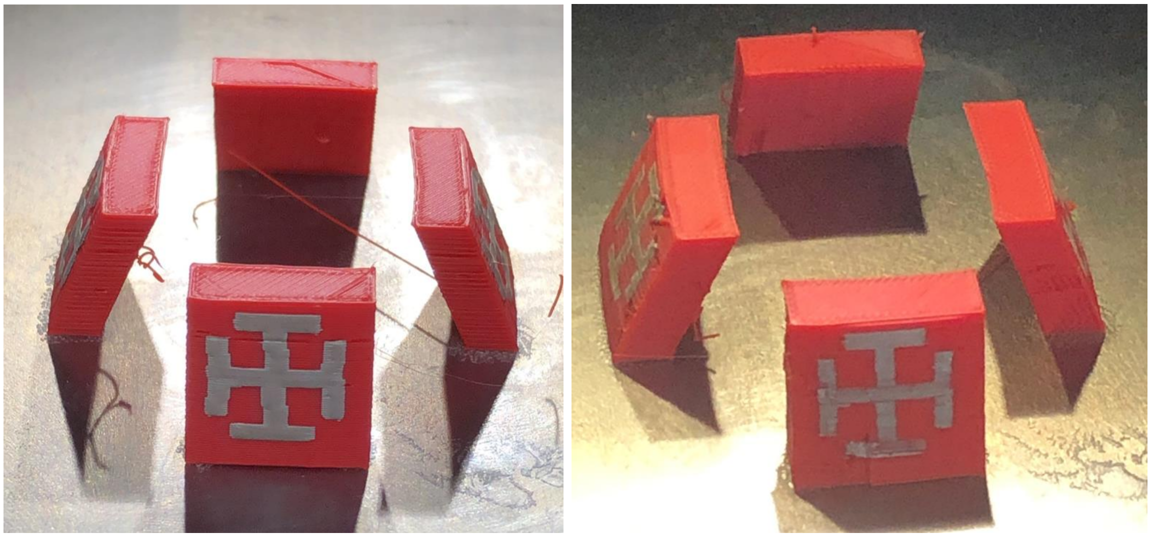


Figure 33. Stonehenge print. (Left) First attempt where crosses are slightly too thick. (Right) Mended cross after implementing vertex endpoint offset.

Further quantification of the Stonehenge prints are shown in Figure 34 and Figure 35. Figure 34 shows noticeable defects in the cross along its edges. The defects appear as the cross shifting positions as the printing continues. This was mitigated by printing the tool path again over the deposited silver paste in the reverse direction. The result of this mitigation is shown Figure 35 where the shifting has noticeably subsided.

One thing to note is in Figure 35 where the measured distance of the uppermost arm is 1.00 mm rounded. The original dimension was modeled to be 0.90 mm. The discrepancy is caused by a number of discrete layers at a printed layer height of 0.125 mm. The seven layers that were printed restrict the dimensions in the *z* direction to 0.125*n* mm, where *n* is the number of printed layers. One should keep this dimension in mind when designing structures to be printed with the steep conformal algorithm.

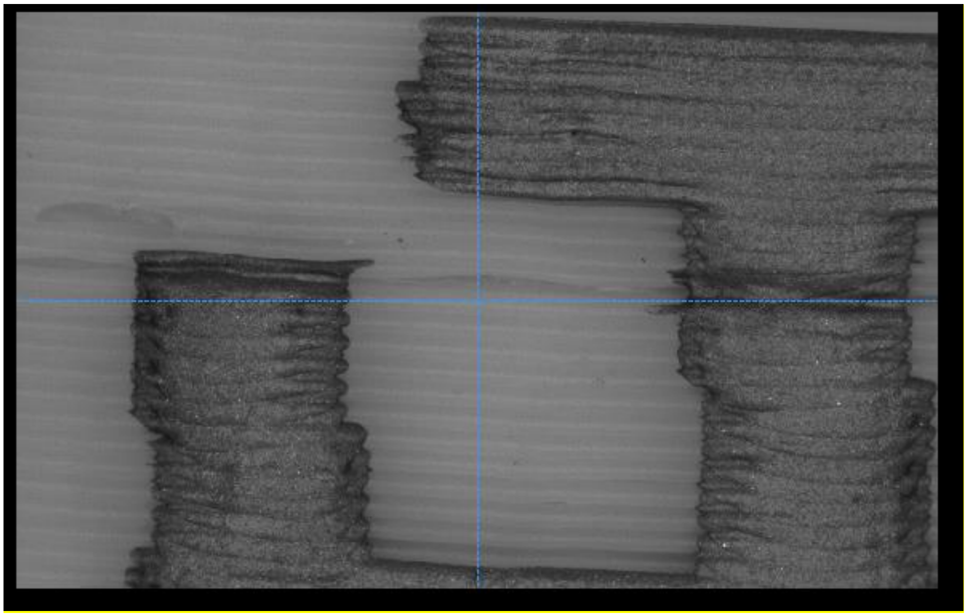


Figure 34. Close-up of steep conformal Stonehenge print. Noticeable shifting is observed.

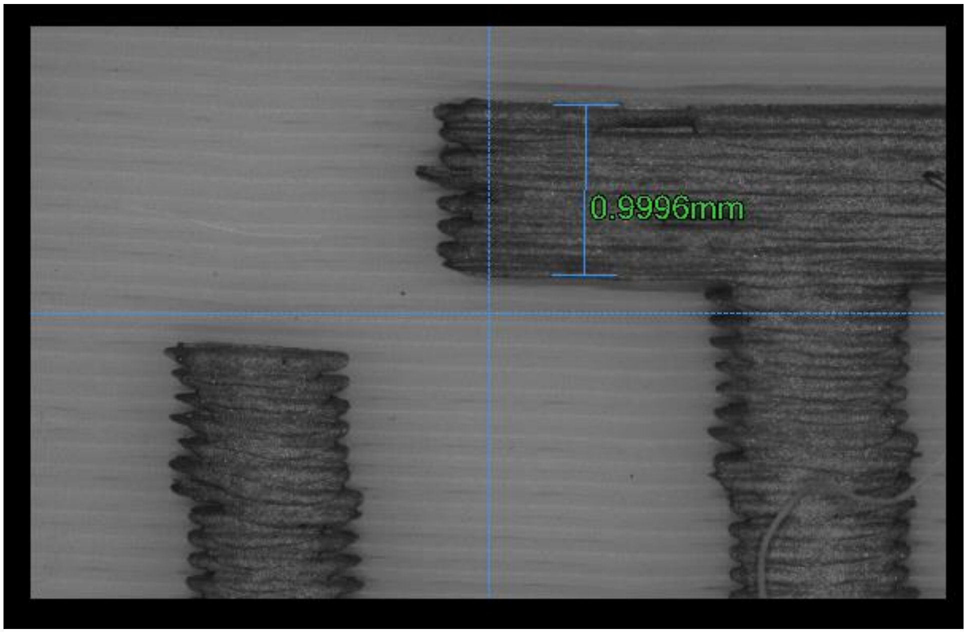


Figure 35. Close-up of second Stonehenge print. Shifting has subsided after adjusting for an additional pass.

3.7 TOOL PATH ORDERING

The previously described algorithms can be applied to both FDM and µD (and presumably other forms of 3D printing) when generating tool paths. Although the g-code commands might appear different for both tools, the tool paths they follow often do not differentiate from each other. The more important aspect is the order in which the tool paths are placed according to their respective tools. Aspects such as the layer height of the dispensed thermoplastic versus that of the conductive ink, the perimeter offset of the interface between the materials, and the types of slicing involved all need to be taken into consideration. For this work, three orderings were tested when combining the tool paths: normal alternating, sequential, and section alternating.

3.7.1 Normal Alternating Tool Path Ordering

Normal tool path ordering refers to the ordering of alternating tool paths within a single slice while prioritizing one tool over the other. This can be interpreted as printing the tool paths of one tool during a layer followed by printing the tool paths of the next tool. For example, if the conductive ink needs to be housed in some cavity to prevent runoff and short circuits, normal ordering involves dispensing layer-by-layer prioritizing FDM over micro-dispensing. On the other hand, the conductive ink may want to be dispensed first to ensure it is cured within the layer. The conductive ink tool paths will first be printed followed by the tool paths for FDM. Figure 36 shows the normal ordering of tool paths within a slice.

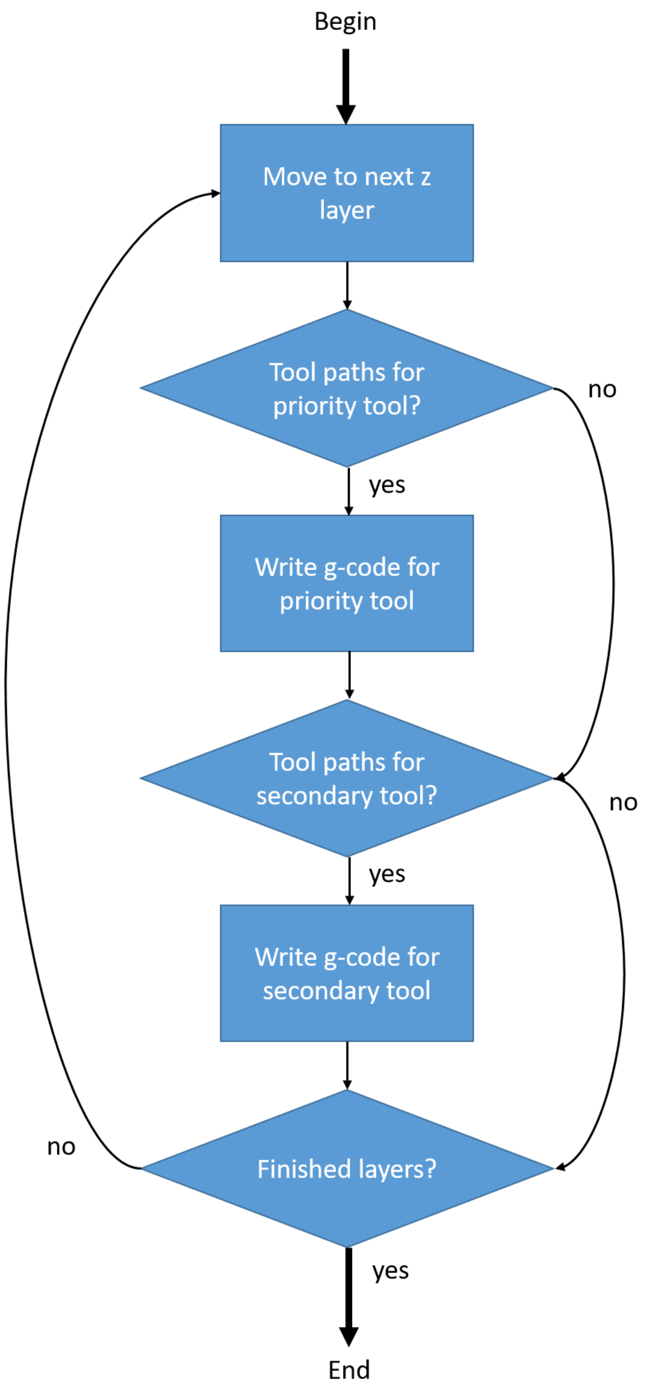

Begin
Move to next z layer
Tool paths for priority tool?
no
yes
Write g-code for priority tool
Tool paths for secondary tool?
no
yes
Write g-code for secondary tool
no
Finished layers?
yes
End

Figure 36. Normal tool path ordering flow-chart.

This approach to tool path ordering is suitable for printing 3D volumetric circuits [25] and other structures that involve meandering embedded conductive traces [24]. This is the most straightforward and familiar way of dispensing multiple materials within a single print since it closely follows typical multi-tool 3D printing processes already.

Considerations when printing in this order include the amount of extrusion, the number of walls encasing other material regions, and regions unbounded by encasing material. Extrusion/dispensing needs to be tuned appropriately for both printing processes, as too much deposited material can hinder the deposition of the other tool. For example, overly-dispensed conductive ink can cause overfilling of regions, causing the FDM tool to smear the ink along the layer. The number of plastic shells plays an important part as they help prevent the conductive ink from seeping in between the gaps of tool paths and layers if FDM extrusion is not tuned properly. Finally, in regions where conductive ink is not encased, such as on the exposed exterior of the device, the tool paths and the extrusion need to be tuned correctly so ink does not smear or flow into unwanted places. Many of the issues discussed here can be mitigated by the cleaning routines described in Section 3.8 assuming that they persist after proper parameter configuration.

3.7.2 Sequential Tool Path Ordering

Sequential ordering involves printing all of the material (including of different heights) completely before printing all of the second material completely. No considerations for layer height or process prioritization are needed. This method is useful for printing devices that require printed materials that are completely separated from the other. Examples include circuit boards or substrates with traces printed on top, devices with conductive arrays printed on them, conformally printed devices, off-axis printed devices, or even different devices of different materials altogether. Figure 37 shows the ordering for this.

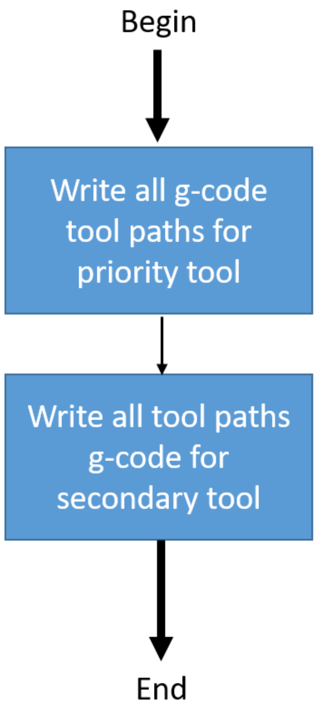

Begin
Write all g-code tool paths for priority tool
Write all tool paths g-code for secondary tool
End

Figure 37. Sequential tool path ordering flow-chart.

There are fewer considerations for this style of order than for normal ordering. The major consideration for this order is the regions of printing where the tools will be physically located. If one tool needs to print somewhere where the tool gantry or head cannot physically move to because of another printed part, then this device cannot be printed or else the tool head will crash into the print. To prevent this, the location of the models needs to be placed with consideration to one another. If the location cannot be moved, then section alternating (discussed in the following section) ordering needs to be done.

3.7.3 Section Alternating Tool Path Ordering

Section alternating ordering involves any combination of the past two ordering methods along with breaking up a full device into "sections" that will be printed. Each section will have its own combination of normal or sequential ordering. As an example, if a tall device needs to be printed with conductive ink along its surface, section alternating ordering will print the base of the structure first along with the conductive traces using the steep conformal approach. The tool paths for this section will have normal ordering. Once this region is finished, the next part of the section will be printed, using the sequential ordering by printing the base then the conductive paths immediately after.

This is the most general way of ordering tool paths. The criteria of where the tools need to transition will depend on the device itself. To simplify the criteria, the sections can be partitioned by height. Each region can be given an ordering preference that will cumulate into one final process defined by one g-code file at the end. Figure 38 gives the flow-chart for this type of ordering.

It should be noted that this way of ordering tool paths determines which slicing algorithms to use for each section. Because different sets of tool paths need to be generated according to the regions, the slicer needs to know ahead of time which slicing process to use in which regions. This does not affect the slicing processes themselves, and the tool paths will still be ordered correctly.

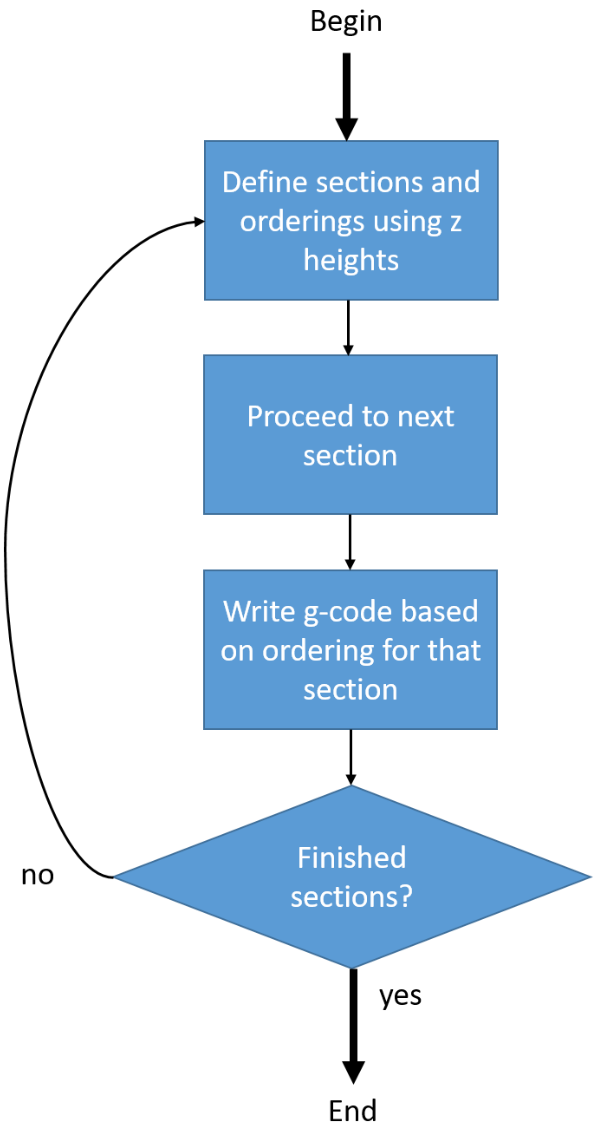

Begin
Define sections and orderings using z heights
Proceed to next section
Write g-code based on ordering for that section
Finished sections?
no
yes
End

Figure 38. Section alternating tool path flow-chart.

3.8 Cleaning and Priming Routines

For prints to be successful, the tools need to maintain a constant and reliable extrusion and dispense rate. For particular testing on the nScrypt TableTop 3Dn printer, it was found that parameters for both nFD™ and SmartPump™ tools tended to drift over a period of time, particularly the SmartPump™. This meant that, despite printing at a constant pressure and valve position, conductive paste would stop flowing either for a period of time or for the remainder of the print. Ultimately, this was attributed to the position of SmartPump™ relative to the nFD™ tool on the printer itself. Figure 39 shows these positions. A necessary process that was incorporated was the introduction of priming and cleaning routines during the printing process. (It should be noted that not all machines require this process. They should typically be used if environmental factors cause parameters to drift repeatedly.)

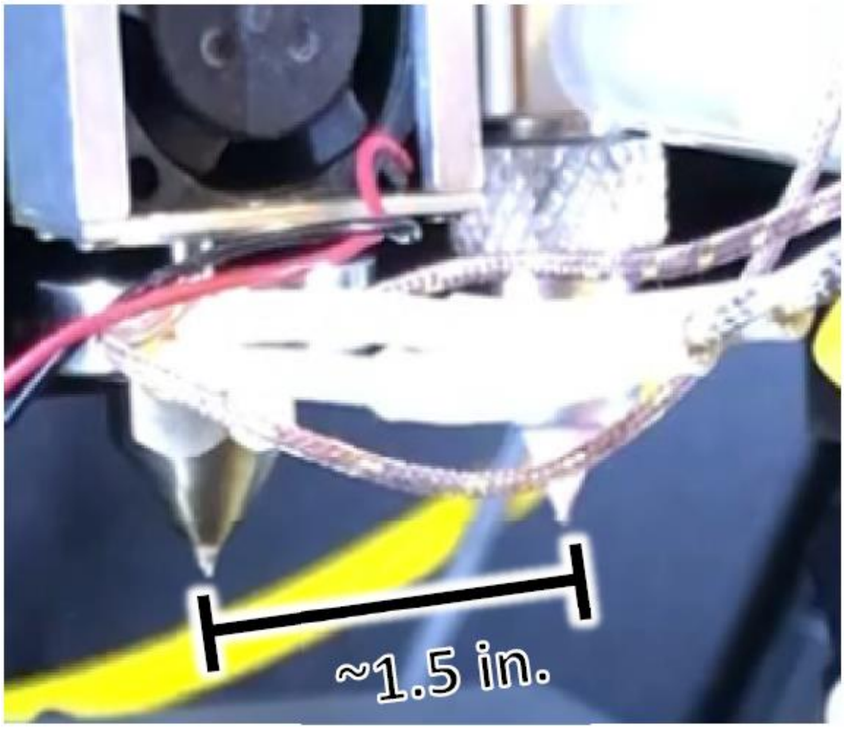


Figure 39. Distance between nFD™ (Left) and the SmartPump™ (Right).

The nFD™'s temperature can increase to around 230°C and up to 400°C. Due to the SmartPump™ being so close to the nFD™, it was believed that the heat was causing the ink in the pen tip to dry before being dispensed. To mitigate this issue, a priming routine for the SmartPump™ was incorporated into the printing process. This routine dispenses ink at the edge of the print bed at a larger quantity than when printing traces, clearing the pen tip of any dried ink that might have coagulated. The routine can be set to execute at any desired time. It was found the best results occur when the SmartPump™ is primed when tools switch, and every five seconds when the SmartPump™ is dispensing paste. Figure 40 shows the priming routine in action.

Similarly, a priming routine for the nFD™ was also implemented. It was observed that at some points, excess cured paste would be picked up by the heated nFD™ head, causing issues in the print. In the past, this was fixed by manually clearing off the pen tip with steel wool during the print. The process was automated by implementing a routine that cleans the nFD™ before each tool change. This was done by placing a piece of scrubbing wool on the opposite side of the bedplate and directing the nFD™ to scrub itself during the print. Figure 41 shows this process.

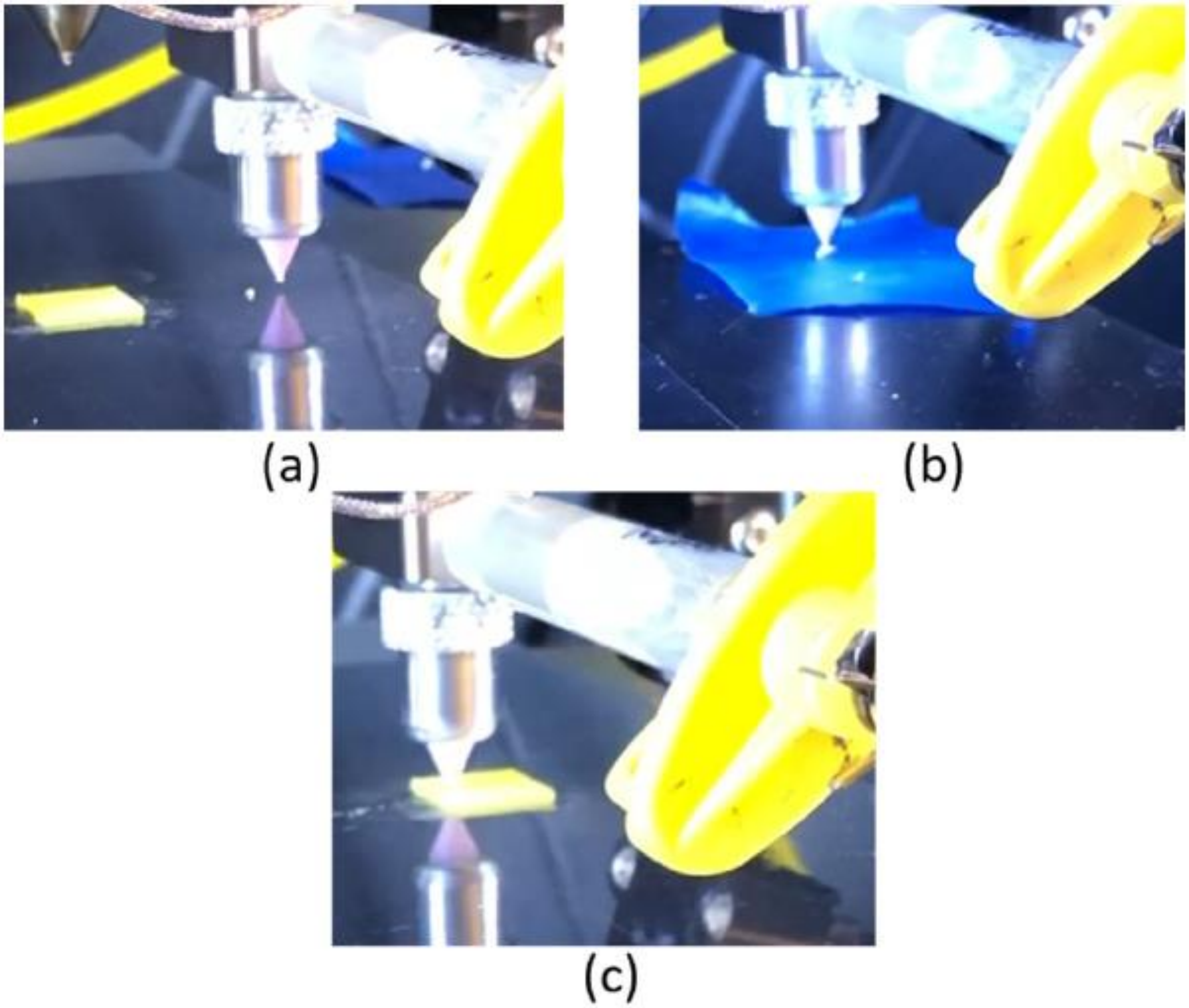


Figure 40. SmartPump™ priming process. (a) SmartPump™ tool change, (b) SmartPump™ dispensing on the side of the bed, and (c) SmartPump™ moving back to the print.

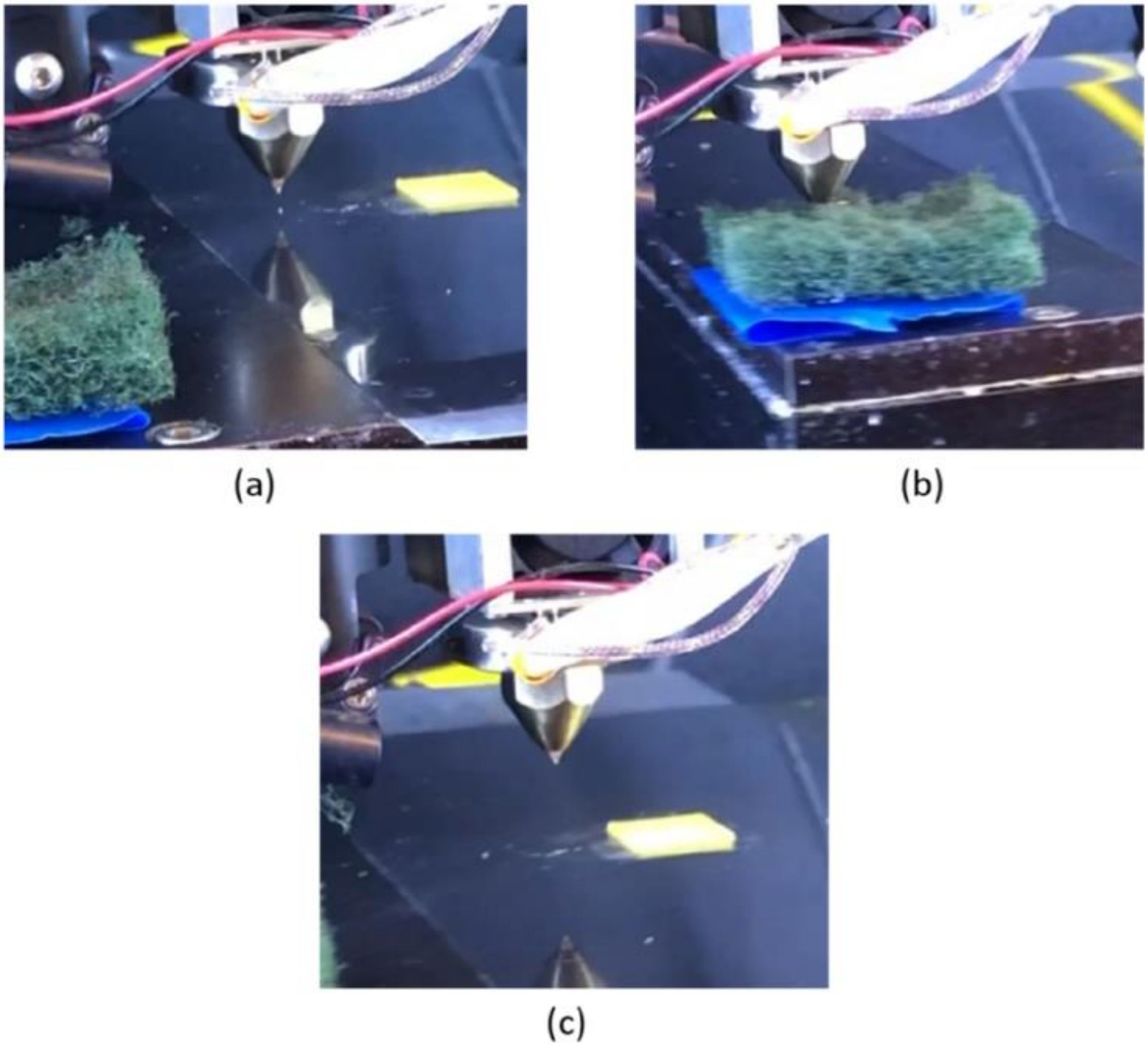


Figure 41. nFD™ priming procedure. (a) nFD™ switching, (b) nFD™ cleaning, and (c) nFD™ returning to printed device.

The feasibility of these priming moves was tested on a simple structure, a dielectric block with a cylindrical hole in it. Figure 42 shows the results of printing both with and without priming moves. The four devices on the left did not use the priming moves, while the two on the right did. The device on the far left shows an overflow of silver paste when the purging routine was not included. While there is no indication of breaks in the flow of silver ink, this result is undesirable.

If the print required the nFD™ to print over the paste, the paste would have been smeared due to the overflow of paste in the cavity. Naturally, the immediate fix would be to reduce the flow of the paste. However, reducing the flow of paste eventually leads to premature curing within the pen tip. This can be seen in the next three attempts in Figure 42 where the paste eventually stops flowing during the print. The print attempt with the priming routine involved the SmartPump™ periodically priming itself every 10 seconds. The result of this is shown second from the far right. Since there was still premature curing, the priming time was reduced to 5 seconds, with its result shown on the far right. This result was pleasing, as it shows no overflow of ink while still filling in the majority of the cavity. Adjusting the dispensing parameters further can smooth out the fill of the paste more evenly.

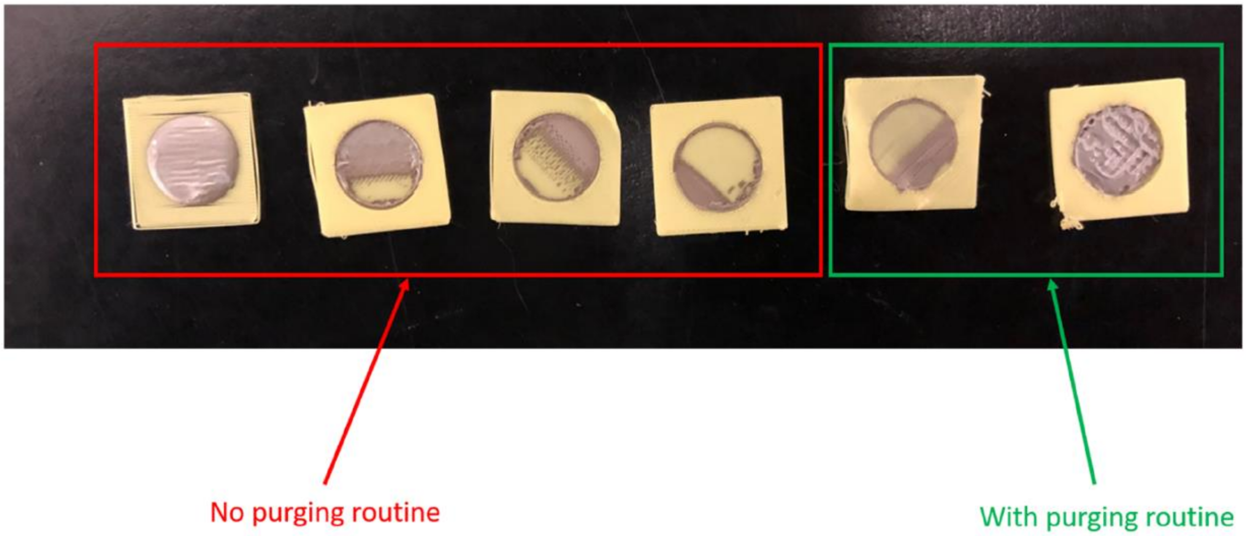


Figure 42. SmartPump™ purging routine test.

3.9 Additional Routines

In certain cases, some printed objects experience small deformations at the beginnings and endings of the printed paths. At some points, the beginnings of a printed tool paths lack material and the ends of tool paths have excess material that strings out. Both of these occurrences are undesirable for hybrid printing as they can obstruct and hinder the dispensing head as it deposits conductive paste. To ensure there is always a smooth surface after printing a layer, two additional routines were developed. The precision line start and precision line end routines combine both head movement, retraction, and extrusion to attempt avoid the deformations in the objects.

3.9.1 Precision Line Start Routine

Occasionally, if the retraction of the thermoplastic is set too high, the start of new tool paths may be under-extruded due to filament having to travel down inside the pen tip. To compensate for this, the precision line start routine was implemented. This involves controlling the amount of extrusion just before beginning a tool path. The routine works by gradually extruding thermoplastic as the tool travels to its next destination. For example, if a length of retraction is set to five millimeters, normally the filament would retract five millimeters at the end of a tool path and extrude five millimeters immediately at the starting point of the next tool path. If the precision line start routine is enabled, the filament will retract five millimeters at the end of a tool path, and then extrude five millimeters gradually while traveling to the start of the next tool path. This is illustrated in Figure 43.

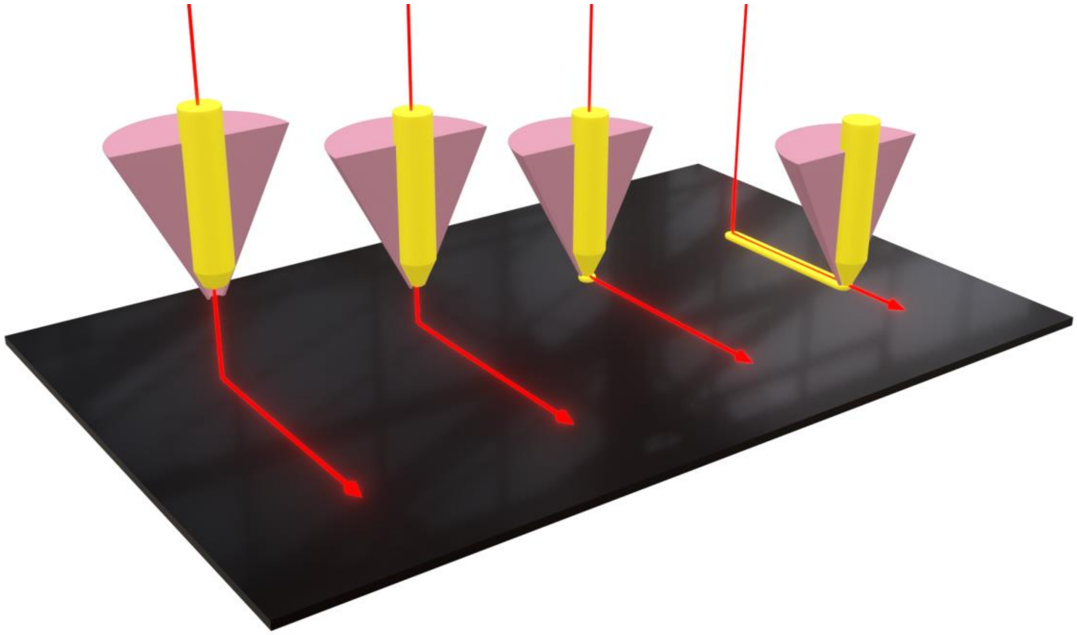


Figure 43. Precision line start routine. (Left to Right) As the pen tip nears the floor, the filament begins extruding.

3.9.2 Precision Line End Routine

The precision line end routine does the opposite of the previous routine. This routine gradually retracts the filament to avoid small hairs of thermoplastic at the ends of tool paths. This is controlled by the defining the number of tool path vertices of when the filament should begin retracting. For example, if the number of points is three and the retraction is 6 mm, the filament will begin retracting the 6 mm of filament three vertices before the reaching the last vertex. Once the last vertex is reached, the entire 6 mm of filament should be completely retracted before lifting from the layer. Figure 44 shows this process.

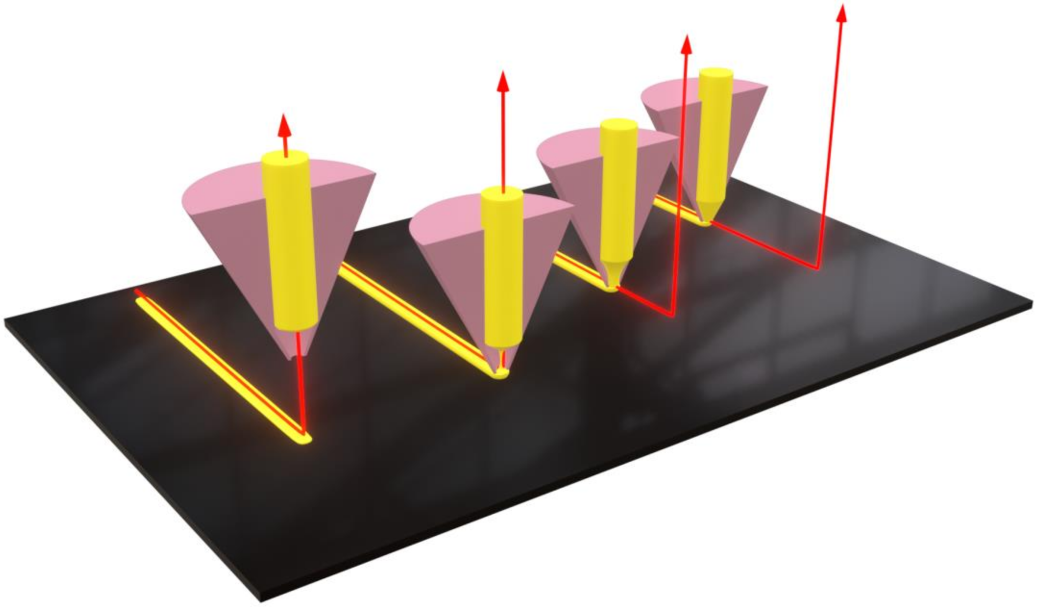


Figure 44. (Right to Left) Precision line end routine retracting before lifting.

3.10 HYBRID PRINTING PARAMETER CONFIGURATION

Before full devices can be printed, proper parameter configuration must first take place. Basic parameter configuration is necessary to ensure that the substrate packaging is homogenous, lacking deformations in the plastic as well as the conductive ink creating the proper dimensions that are desired. While the device that needs to be printed will need proper parameter configuration custom to itself, basic parameters such as line width, layer height, print speed, and the extrusion factor are necessary to dial in before attempting more complex prints. The following special printing processes were developed to configure these parameters in a logical order. Each of these tests were used when configuring ULTEM™ [75] for printing hybrid printing. Parameters for other thermoplastics can also be configured with these tests.

3.10.1 Dot Matrix

Figure 45 shows the printing process for the first test. The dot matrix printing process is designed to test the extrusion and retraction settings and the oozing between dots. This simulates if there is enough extrusion and retraction to ensure proper adhesion to the bed as well as ensuring there is a clean transition between dots. This test involves printing dots in a grid where the columns and rows each vary a certain parameter. The parameters to be varied are extrusion/retraction speed, the extrusion/retraction distance, and the tool head travel speed.

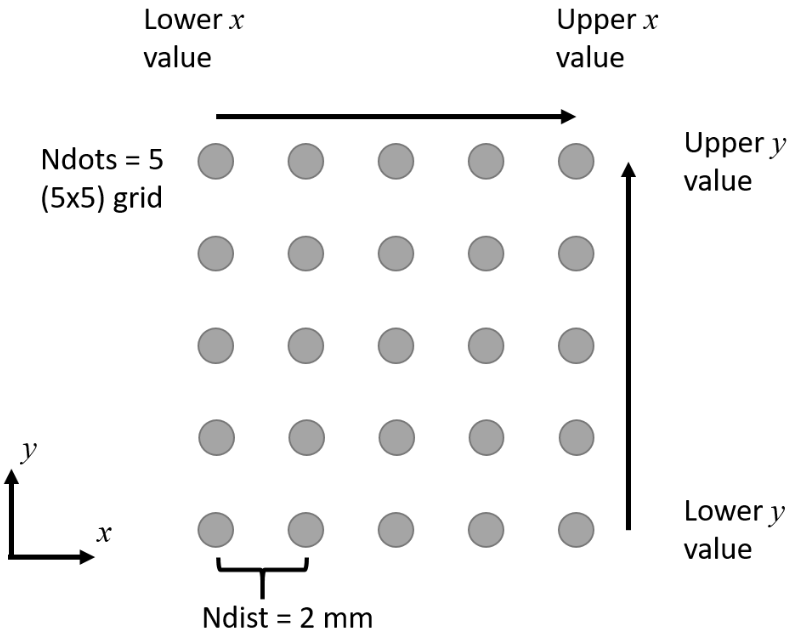


Figure 45. Dot matrix test.

The dot matrix test prints a grid of n-by-n dots each a distance of N from each other in the horizontal and vertical directions at a certain travel speed. Once a set of dots is printed, they are visually inspected to see which dots adhered best and did not connect to other dots via stringing. If there is too much oozing between all the dots, the travel speed can be increased to tear the stringing away from the dots. Figure 46 shows an example of using the dot matrix test.

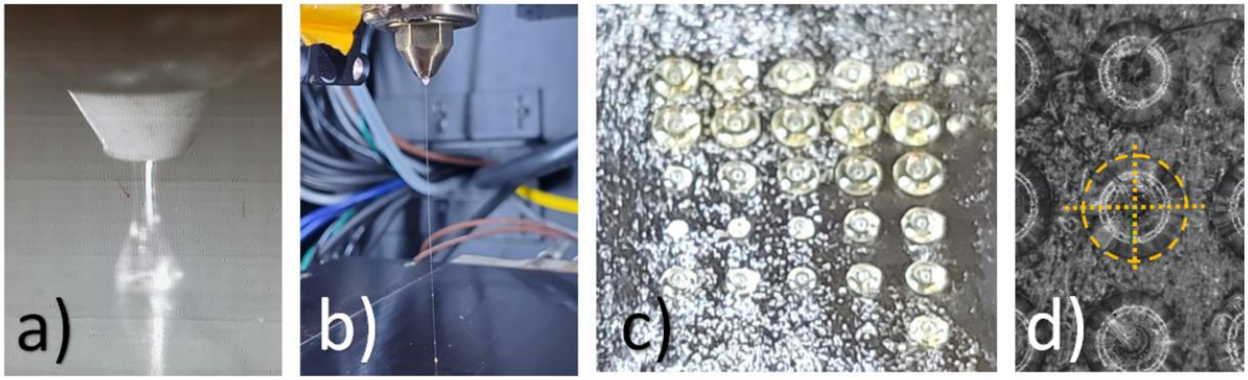


Figure 46. Dot matrix test results using ULTEM. a) Oozing after printing a single dot. b) Further oozing when moving the head away. c) Printed dot matrix. d) Measurement of a single dot.

3.10.2 Line Arrays

The next printing test is a series of line arrays. This test is to configure the line width, the extrusion rate, and the print speed. The line array consists of printing several lines in a series. Two types of arrays can be printed, separate and connected lines. The test is simulate jumping between tool paths as well as printing a single layer. Under-extrusion and over-extrusion both need to be examined when printing this test. The parameters that were found using the dot matrix experiment are carried over for this one. Figure 47 shows the layout for the line arrays. An example of using a line test is shown in Figure 48.

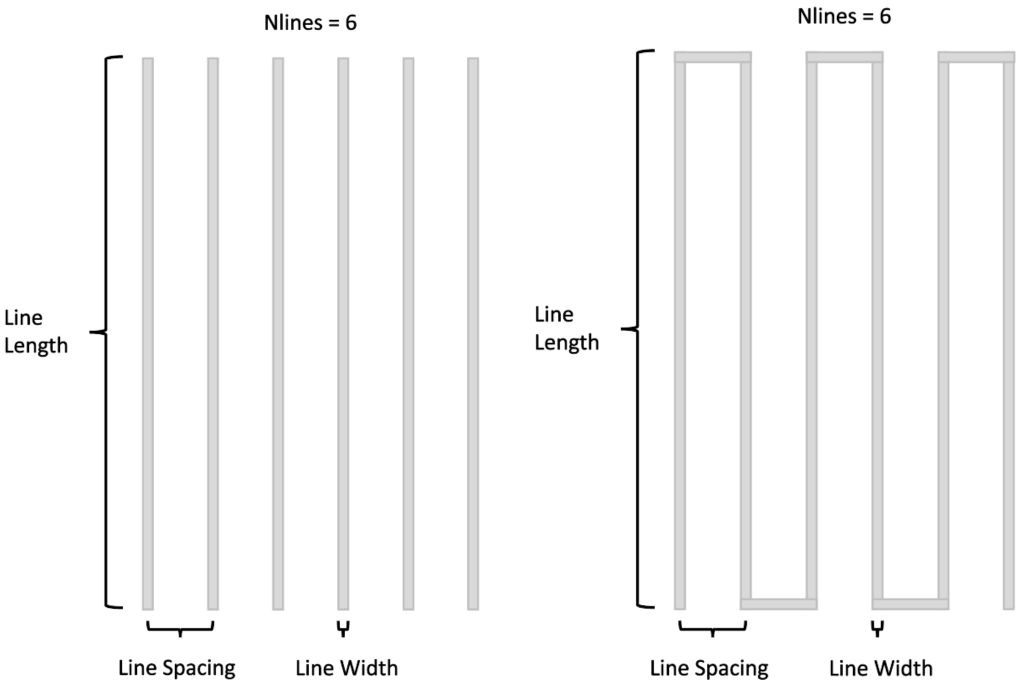


Figure 47. Line test.

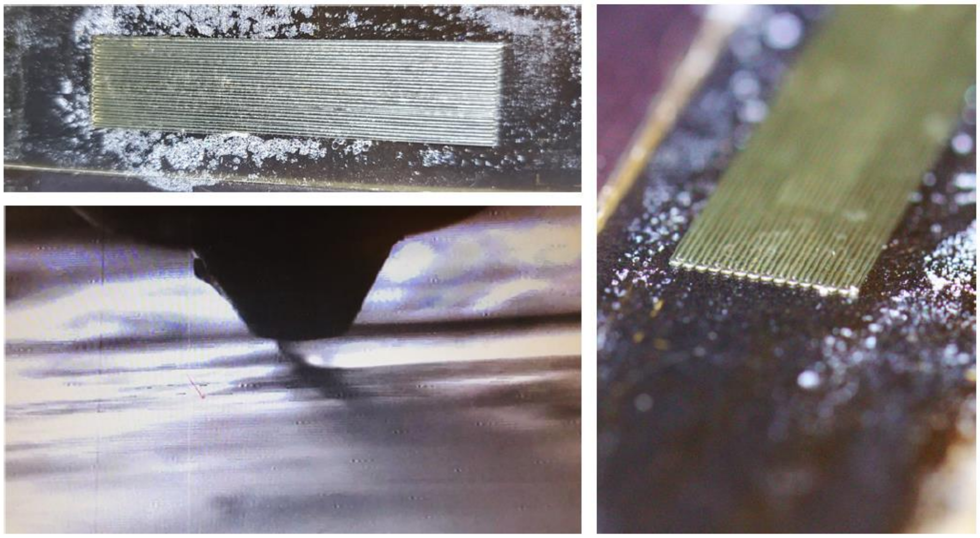


Figure 48. Line test results using ULTEM™.

3.10.3 Multi-Layer Prints

Once the parameters from the dot matrix and line array tests are configured, a simple model can be printed to test printing multiple layers. This model can be printed multiple times within the same print job to fine-tune any parameters that may still need configuration. The additional cleaning and extrusion routines described in Section 3.8 can be used during this step if the parameters changed at this point. Priming structures such as priming lines can be included too. Other deformations or artifacts typical of volumetric prints, such as shrinkage, bowing, etc., can be examined for as well. Figure 49 shows the layout for this test.

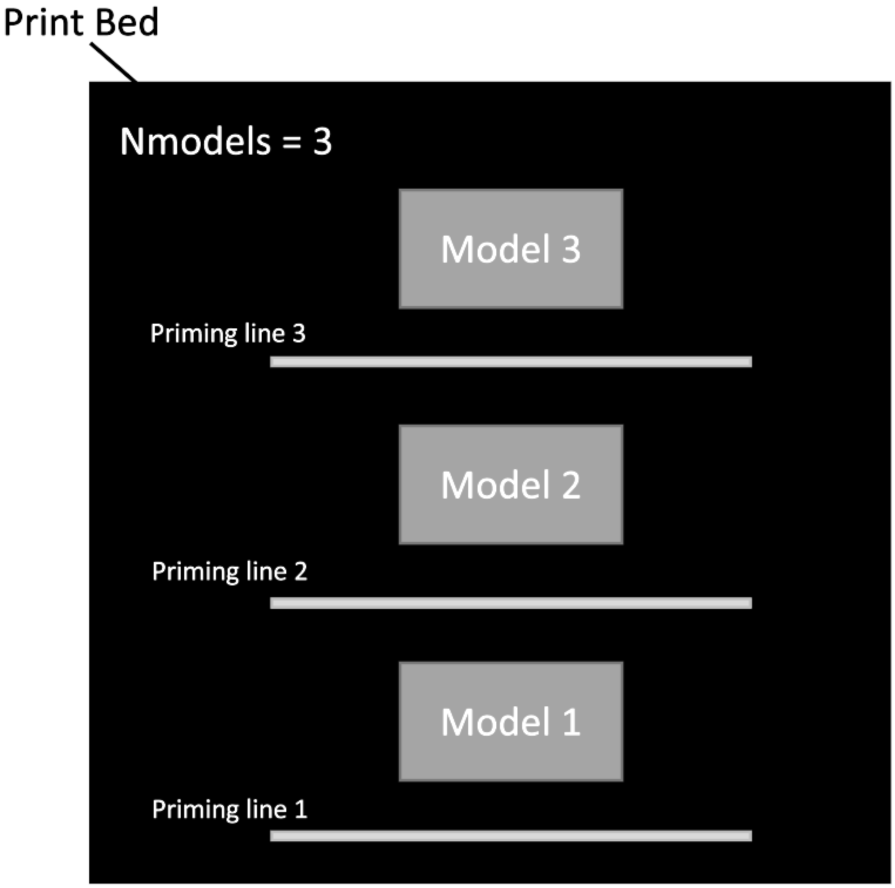


Figure 49. Multiple model test.

Chapter 4: Hybrid Printed Devices

4.1 Hardware and Materials

The following devices were printed on an nScrypt TableTop 3Dn hybrid 3D printer. The nScrypt TableTop was chosen for its versatility in hybrid 3D printing by mixing different tool configurations as well as its precision and reliability in retaining its printing parameters and configurations. Acrylonitrile butadiene-styrene (ABS) plastic was used for the housing, and DuPont CB028 conductive ink was used as the conductor. ABS plastic is a common thermoplastic found in many items such as containers, toys, and figurines that ideally softens and becomes extrudable around 230 °C. For our tests, 245 °C was used as the higher temperature aided in the ABS adhering to itself. CB028 conductive ink is a loaded material containing nanoflakes of silver. The flakes are housed in a mixing agent that dries and leaves behind solid traces of silver. CB028 was used for its reliability and high conductivity.

4.2 3D Volumetric Circuits and Packaging

4.2.1 Cubic 555 Timer Circuit[2]

The first demonstration of printing 3DVCs was done in [25] using normal tool path ordering and the slicing algorithm in [24]. The results of this print are significant as it was the first time a 3D circuit package had been hybrid 3D printed in an automated setting. Figure 50 shows the internal design of the interconnects, and Figure 51 shows the model and side-by-side resultant device. The nFD™ and SmartPump™ line widths were both set to 125 µm, and the layer heights were both set to 50 µm each. The components were placed by hand and held in place with EPO-TEK H20E conductive epoxy. Figure 52 shows the modeled tool paths and the printed tool paths

[2] This sub-section was taken from [25]

side-by-side. Finally, Figure 53 shows the device successfully working when powered with a 9V power source.

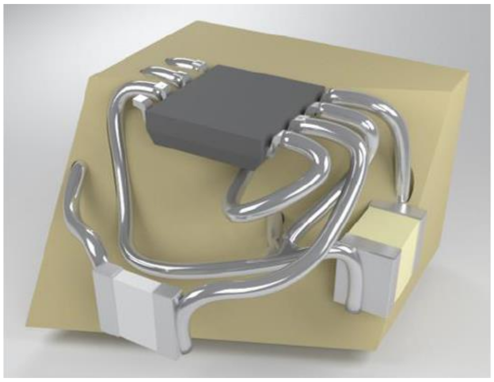
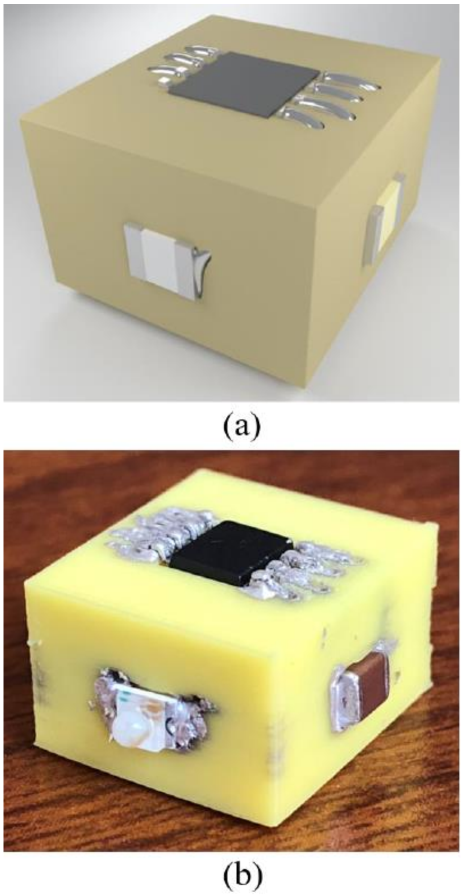


Figure 50. 555 timer circuit model. Package, components, and interconnects are shown. [25]



(a)

(b)

Figure 51. (a) Rendered model of 555 timer circuit. (b) Printed 555 timer circuit. [25]

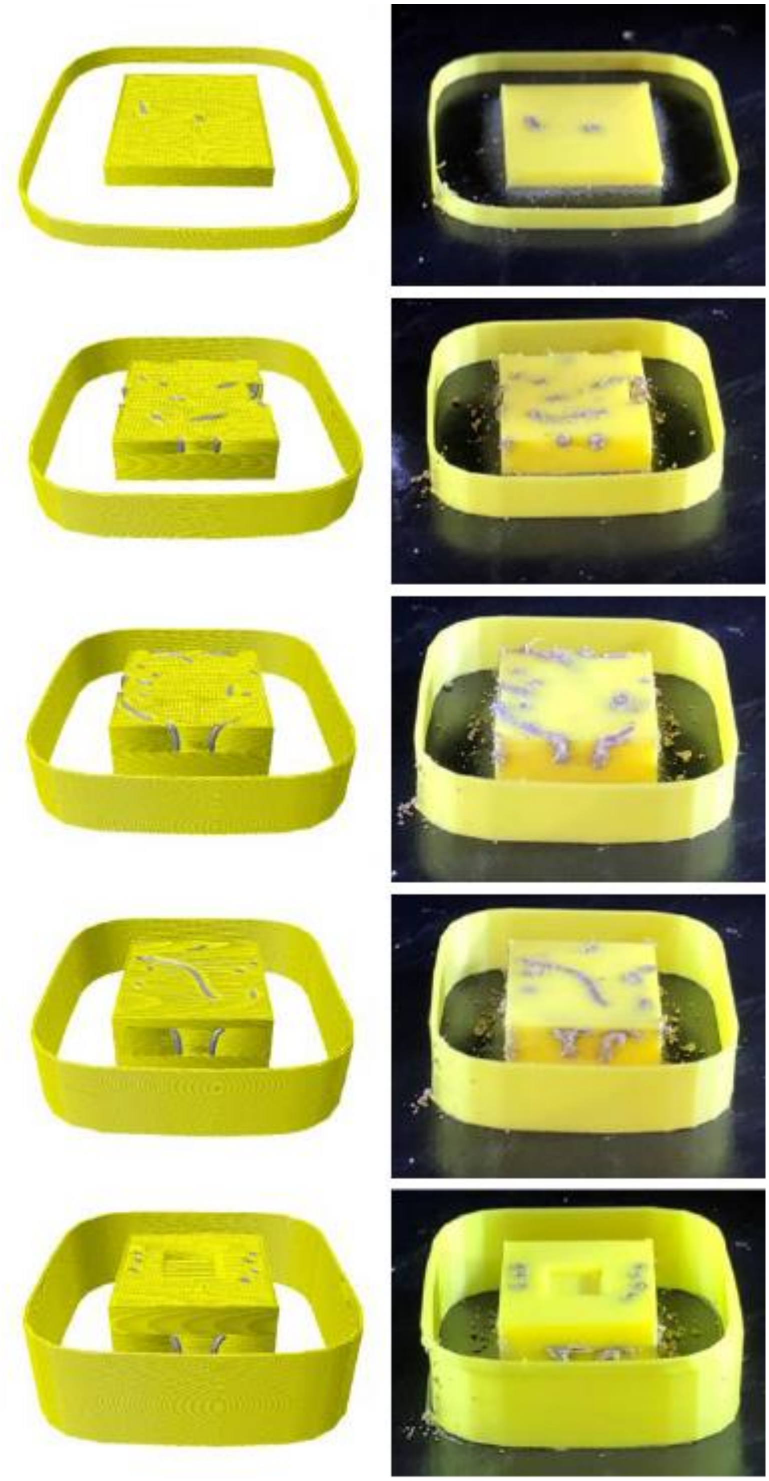


Figure 52. (Left) Modeled tool paths. (Right) Printed tool paths. [25]

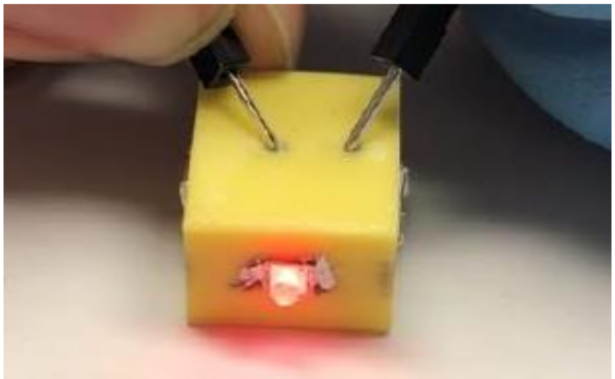
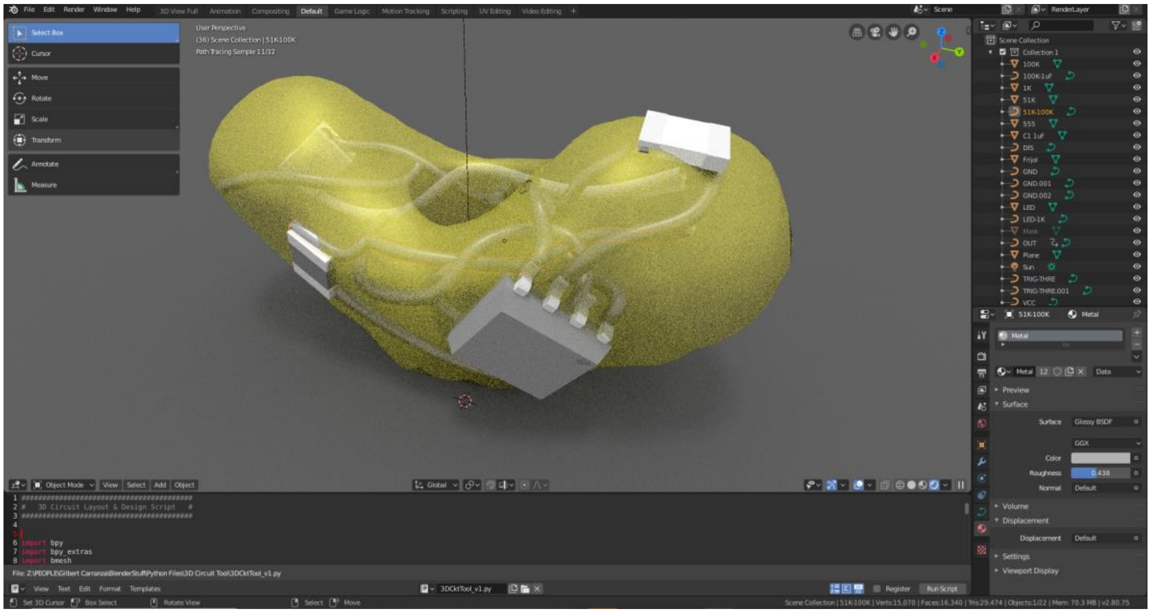


Figure 53. Working 555 timer demonstration. [25]

The completion of this device resulted in several design rules that have proved beneficial for the following printed devices. First, the distance between adjacent components needed to be at least four times the inner diameter of the SmartPump™ tip, and the trace widths needed to be at least two times the inner diameter of the tip. This allows at least two passes of the SmartPump™ tip for one trace and at least one shell of ABS on either sides of the traces to come between two components. Second, it was beneficial to print the shells of ABS first as it provides a clean cavity for conductive ink to be filled in. Otherwise, short circuits became more frequent. Finally, a printed skirt using the nFD™ was found to reduce the amount of debris left behind by the printing process. When the nFD™ would pick up debris from the conductive ink or plastic, the printed skirt would help clear the debris off. This inspired the idea to include cleaning and priming routines for the printing tools.

4.2.2 Holey Frijole

To further prove the robustness of printing 3DVCs (and to appease Dr. Rumpf after he said the first one was too square and looked too much like a conventional but thick circuit), a second version of the 555-timer circuit was printed in a shape that left no doubts of the meandering traces and arbitrary form factor. Dubbed the "holey frijole," the new iteration featured a bean-like package that contains meandering traces and the components in random locations across the surface of the package. The same printing parameters that were used for the first iteration were used for this one as well. Figure 54 shows the design of the device, Figure 55 shows the printed device, and Figure 56 shows the device being powered.



```
##########################################
#   3D Circuit Layout & Design Script   #
##########################################


import bpy
import bpy_extras
import bmesh
```

File: Z:\PEOPLE\Gilbert Carranza\BlenderStuff\Python Files\3D Circuit Tool\3DCktTool_v1.py

Figure 54. Holey frijole design using the 3D circuit tool.

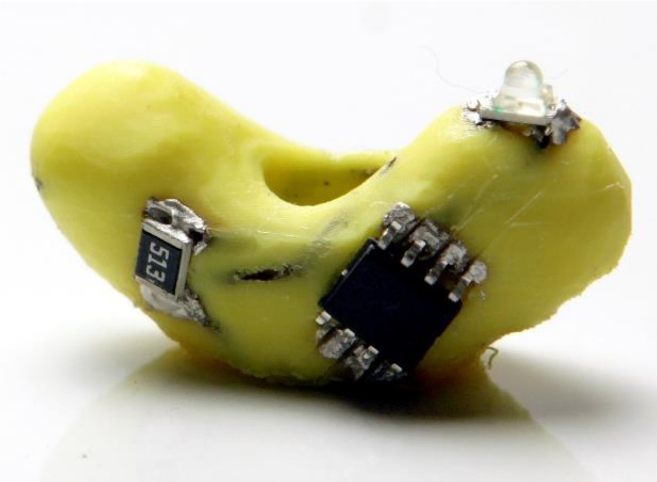


Figure 55. Holey Frijole.

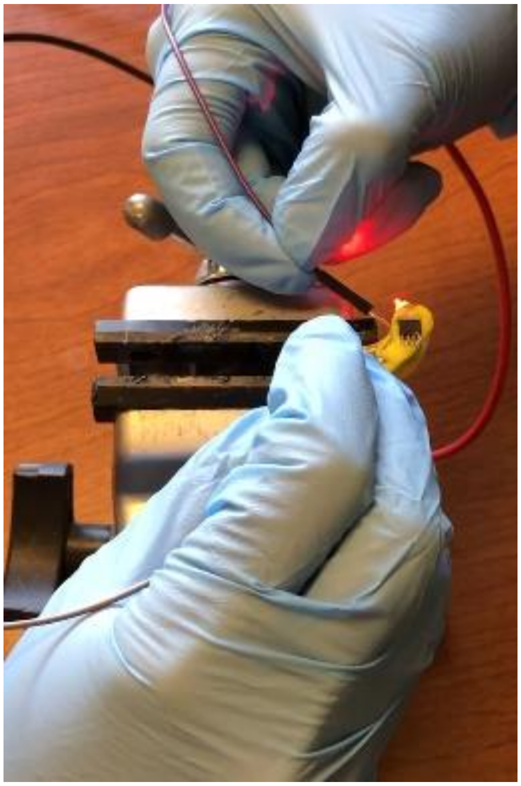


Figure 56. Powered Holey Frijole.

4.2.3 Improved Printing with the 555 Timer

During the initial printing of the 555 timer circuit, there were many failed attempts before successfully printing one. Many of these attempts were attributed to the issues outlined in Section 3.8. Because of this, the initial circuit package was printed again with the inclusion of the cleaning routines. The resultant package is shown in Figure 57. This print was significant because it was first time a 3DVC had been successfully printed on the first attempt.

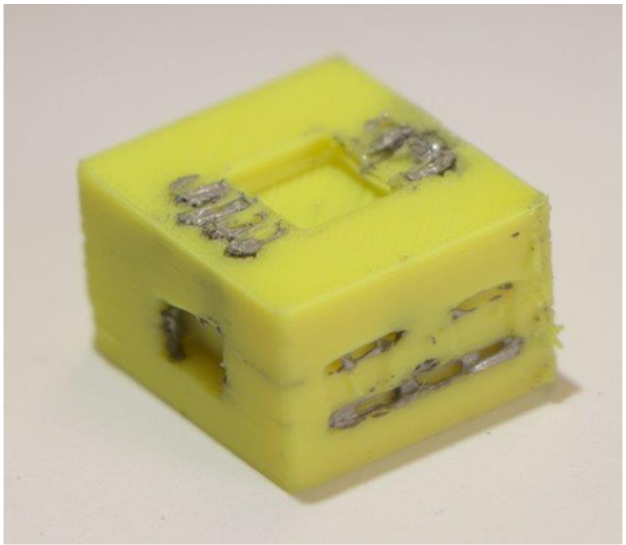


Figure 57. First attempt success at printing a 3DVC circuit package.

4.3 Conformal Frequency Selective Surface

Frequency selective surfaces (FSSs) are of great interest in the fields of antennas and electromagnetics. A FSS is an array of conductive elements that create a filtering response for electromagnetic waves when illuminated. The EM response comes from the periodicity of a single element placed in some symmetrical form. The frequency of the response can also be tuned according to the element's size and feature dimensions. This response can be tuned accordingly and can be made to exhibit band-pass or band-reject filters. Typical uses of FSSs include electromagnetic shielding [76], [77] and stealth [78]–[80] and among many other things.

While FSSs are not new, the idea of conforming a FSS to a curved surface is novel in its own regard. There have been multiple attempts in recent years to place a FSS across a curved surface [44]–[55], but most are either projected onto a surface or only exhibit a surface that is curved in one direction. The EM Lab developed a tool that can place elements periodically across an arbitrarily curved surface using the spatially-variant lattice (SVL) algorithm defined in [93]. To demonstrate the effectiveness of the algorithm, a FSS of Jerusalem cross copper elements was manually placed along a 3D printed parabolic dome of non-equal axes. Two other devices, a flat FSS and a projected FSS, were also made and measured for comparison. The infinite array model is shown in Figure 58, the devices are shown in Figure 59, and their responses are shown in Figure 60. It can be seen that electromagnetic properties of a corresponding flat FSS are preserved when the algorithm is used.

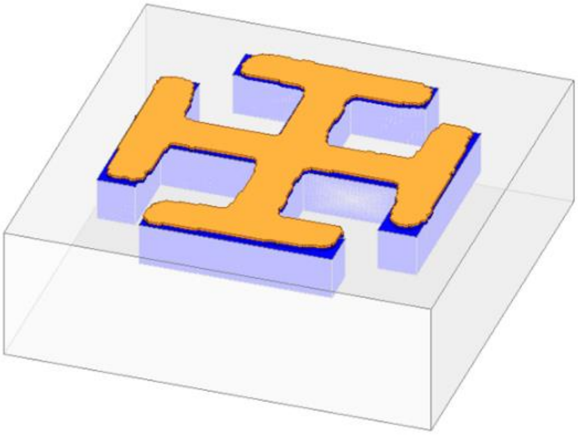


Figure 58. Infinite array model [94].

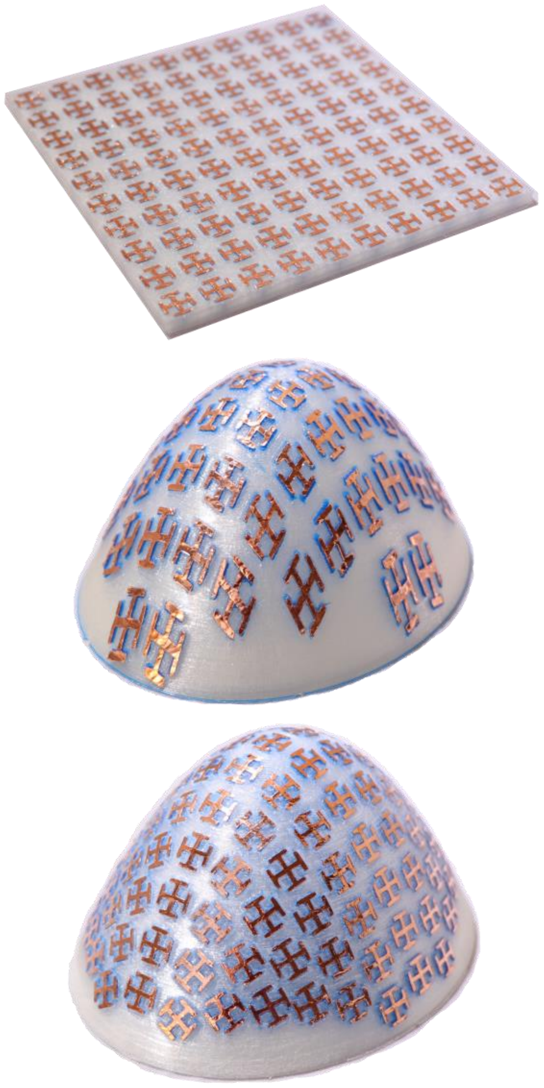


Figure 59. Manufactured FSS with Jerusalem crosses. (Top) Standard flat FSS. (Middle) Projected FSS. (Bottom) SVL tool FSS. [94]

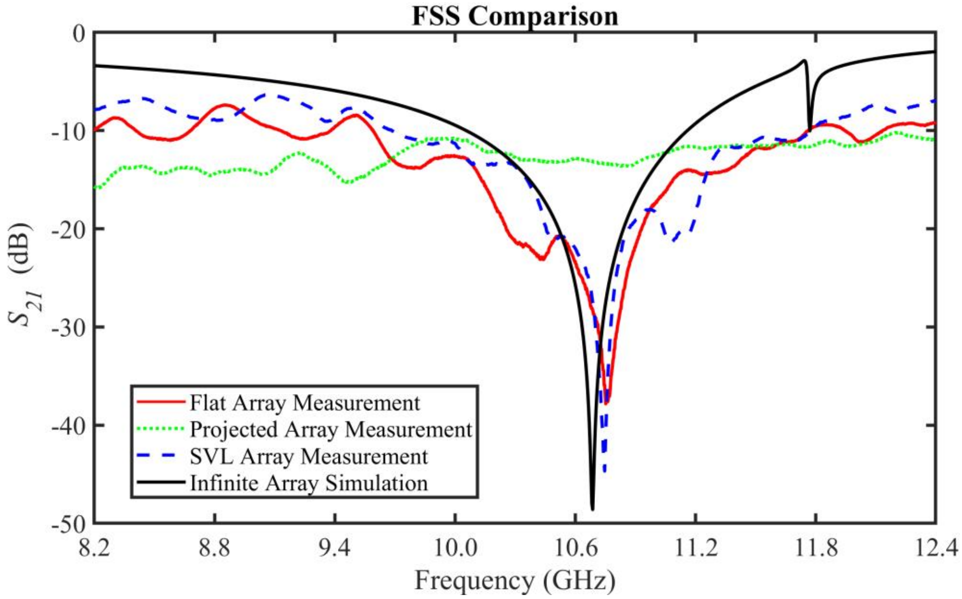


Figure 60. Simulated infinite array results along with measured results of the three manufactured FSS [94].

While the algorithm successfully demonstrated that a FSS can be placed on a curved surface, it was apparent that manufacturing the device proved to be a challenge due to each cross needing to be manually placed. To 3D print the FSS in a single process, a 5- or 6-axis machine needs to be used to ensure each cross is printed perpendicularly to the surface. These machines are usually expensive and inaccessible except to a select number of groups. It would be beneficial for a three-axis machine with hybrid capabilities to have the ability to rapidly prototype, and in the future, fully manufacture devices of this caliber. As the complexity of these devices grow, so do the capabilities of 3D printers. In the next sections, a similar FSS is shown being printed using the hybrid printing techniques discussed in this work.

4.3.1 Manufacturing Approach

The manufacturing approach to print the FSS was to print the substrate in sections while printing the elements conformally after each printed section. The motivation for this is that a typical layer-by-layer approach would cause unnecessary issues during the print such as creating very thin volumes in the array STL file, switching tools between every layer creating possibility for inconsistent dispensing, increased slicing time, and increased printing time. To prevent these issues, the approach taken was to print the substrate layer-by-layer to a certain height, print sections of elements using the regular conformal or steep conformal slicing algorithm, and repeat the process until the dome is complete. This would ensure that each element has less possibilities of having discontinuities and prove that three-axis machines are fully capable of printing devices intended for multiple-axis machines. Figure 61 shows the device to be printed.

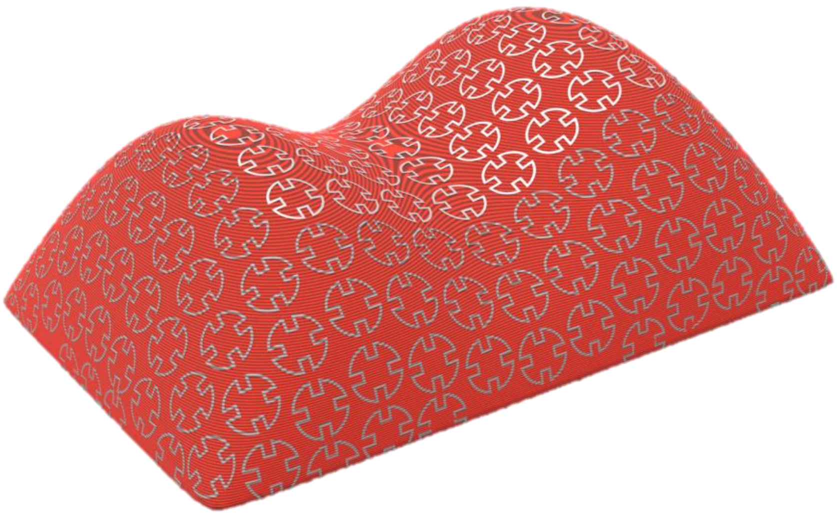


Figure 61. Model of doubly curved FSS.

4.3.2 CAD Model

The FSS model shown in Figure 61 was created using Blender and the FSS SVL tool [94]. The substrate is modeled as a thick shell with a thickness of 2 mm. The FSS element is a ring loop element with some indentations on each side. The period of the element is 7.5 mm. The array is modeled as flat STL surfaces with no volume, suitable for conformal printing. The device features both conformal and steep conformal printing.

4.3.3 Regions of Slicing

The method of slicing and tool path ordering used was the section alternating tool path ordering. Different parts of the FSS were sliced using normal, conformal, and steep conformal slicing. The main concern when performing the slicing was to identify regions of conformal printing and steep conformal printing the elements. Figure 62 shows the regions of tool path ordering based on the slope of the dome and the physical dimensions of the tool gantry. The yellow regions show the space where normal conformal slicing is to be performed while the blue shows the region for steep conformal slicing.

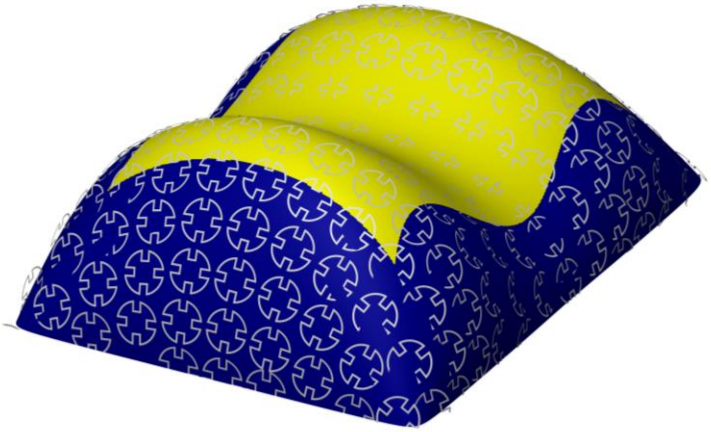


Figure 62. Regions of conformal slicing in yellow vs. steep conformal slicing in blue.

The sections were derived based on the angle of the SmartPump™ pen tip dimensions as well as the layer height resolution of the final print. Based on Figure 63, the instantaneous angle of the surface θ_s can be calculated by

$$\theta_{s,x} = \arctan\left(\frac{dz}{dx}\right), \qquad \theta_{s,y} = \arctan\left(\frac{dz}{dy}\right) \tag{12}$$

These angles are the slopes of the surface in both the *x* and *y* directions respectively.

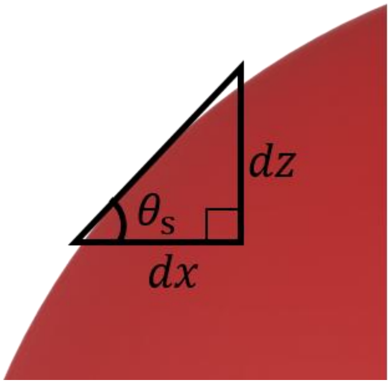


Figure 63. Angle of inclination on a surface.

The regions defining conformal and steep conformal printing depend on two factors: (1) the angle of the pen tip, and (2) the printing distance when performing steep conformal printing. The angle of the pen tip is simple to calculate. The measured dimensions of the pen tip while seated in the SmartPump™ are shown in Figure 64, where the calculated angle θ_p is roughly 19°. This means that regions where the angle of inclination greater than 71° cannot be conformally printed on and need steep conformal printing. Figure 65 shows these regions in blue for the curved FSS.

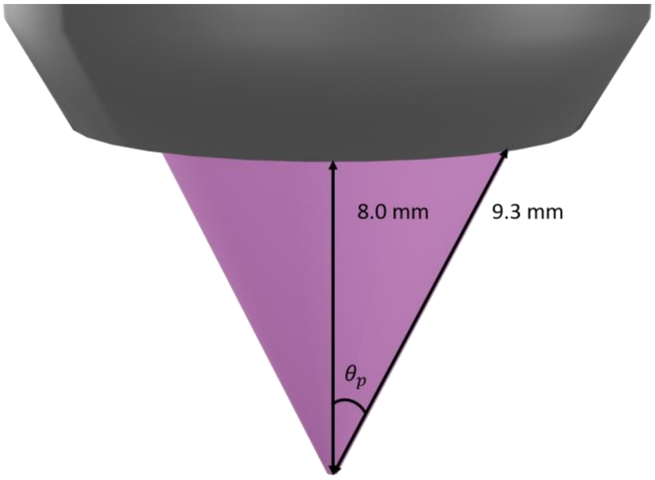


Figure 64. Measured pen tip dimensions.

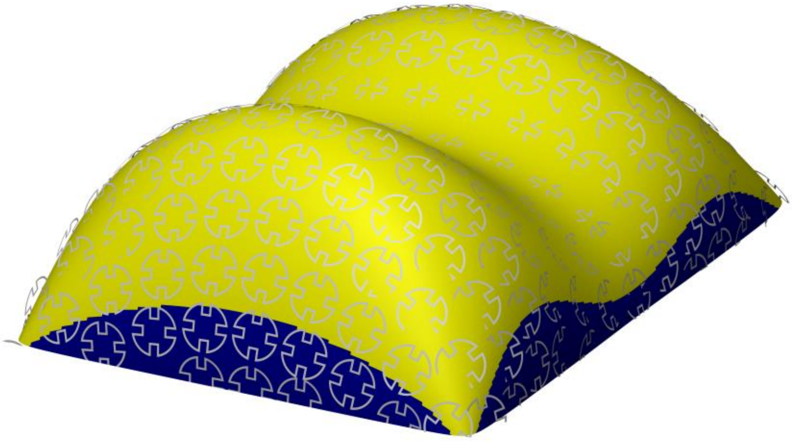


Figure 65. Regions based on 19° angle of inclination.

The other regions that need steep conformal printing are calculated from examining the printed discrete layers. Figure 66 shows the idea of how conductive material is to be dispensed. Steep conformal printing requires conductive paste to be dispensed at the edge of a printed layer such that the paste runs off the edge and forms connection to the previous layer. There also needs to be enough paste on top of the layer at the edge so the next printed layer can make a connection.

In order for this to reliably happen, the uncovered region of a layer has to be less than half the pen tip's outer diameter. The limit that the uncovered region can span gives a limit to the regions of the surface that can only be steep conformally printed. Figure 67 shows the angle of inclination that can be derived when considering discrete layers of a printed surface. Using the same formula as above, the angle of inclination can be derived as

$$\tilde{\theta}_{\mathrm{s},x} = \arctan\left(\frac{h}{\Delta x}\right), \qquad \tilde{\theta}_{\mathrm{s},y} = \arctan\left(\frac{h}{\Delta y}\right) \tag{13}$$

where h is the layer height of printed model. The conditions for steep conformal printing are dictated by

$$\Delta x < \frac{1}{2} d_{\mathrm{out}}, \qquad \Delta y < \frac{1}{2} d_{\mathrm{out}} \tag{14}$$

Bringing equations (12) and (13) together, the angles of inclination for steep conformal printing are calculated by

$$\tilde{\theta}_{\mathrm{s},x} > \arctan\left(\frac{2h}{d_{\mathrm{out}}}\right), \qquad \tilde{\theta}_{s,y} > \arctan\left(\frac{2h}{d_{\mathrm{out}}}\right) \tag{15}$$

For a layer height of 0.2 mm and a line width of 0.25 mm, the maximum angle of inclination for steep conformal printing is 53° rounded to the nearest degree. The region of steep conformal printing for this condition is shown in Figure 62. Because the regions calculated with a 19° angle of inclination lie within these regions, the regions shown in Figure 62 are therefore the combined results of the angle of the pen tip and the distance over the uncovered regions of discrete layers.

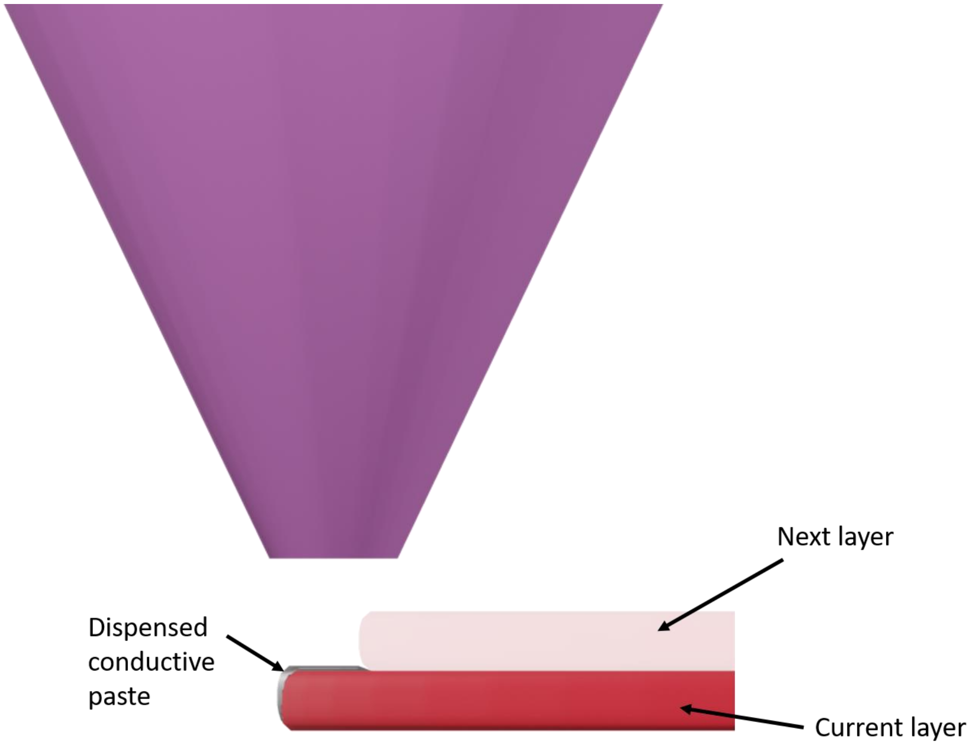


Figure 66. Close-up of steep conformal printing.

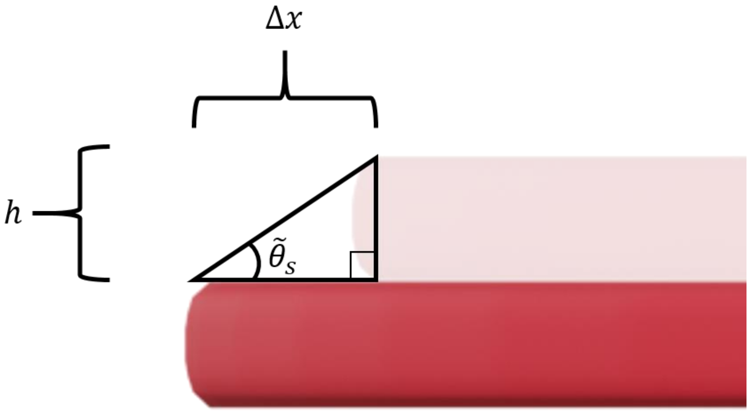


Figure 67. Angle of incidence on two printed layers.

For the regions where normal conformal printing was performed, each section was printed in thicknesses of 5 mm to allow enough room for the SmartPump™ pen tip to move. The vertical length from the pen tip to the pen tip carrier is about 6 mm, so 5 mm is enough to ensure that the tool gantry did not crash into the print.

4.3.4 Printed Model and Measurements

The printed model is shown in Figure 68. To measure the response, two horn antennas connected to a vector network signal analyzer were used. The corresponding S-parameters were taken from the VNA. Figure 69 shows the measured results in comparison to an infinite array simulation. The results show that printing a full FSS using complex hybrid 3D printing methods is not only feasible, but also viable for typical three-axis machines. The suppression in dB can be attributed to the imperfections in the printed elements.

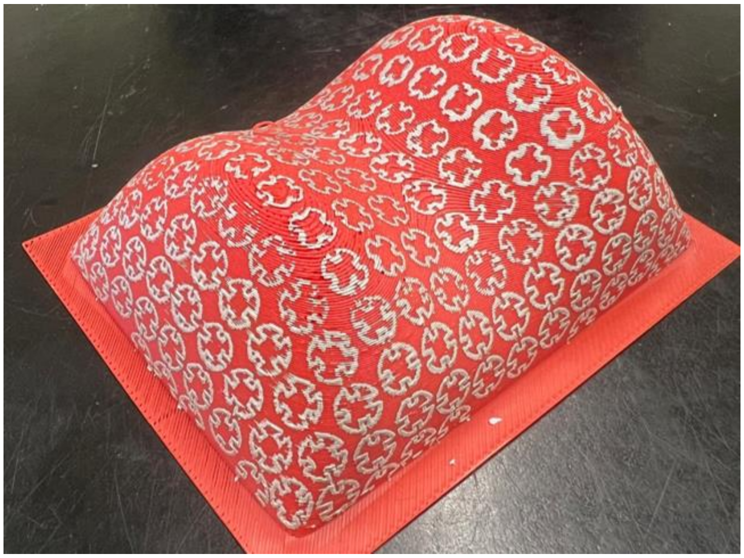


Figure 68. Final printed FSS.

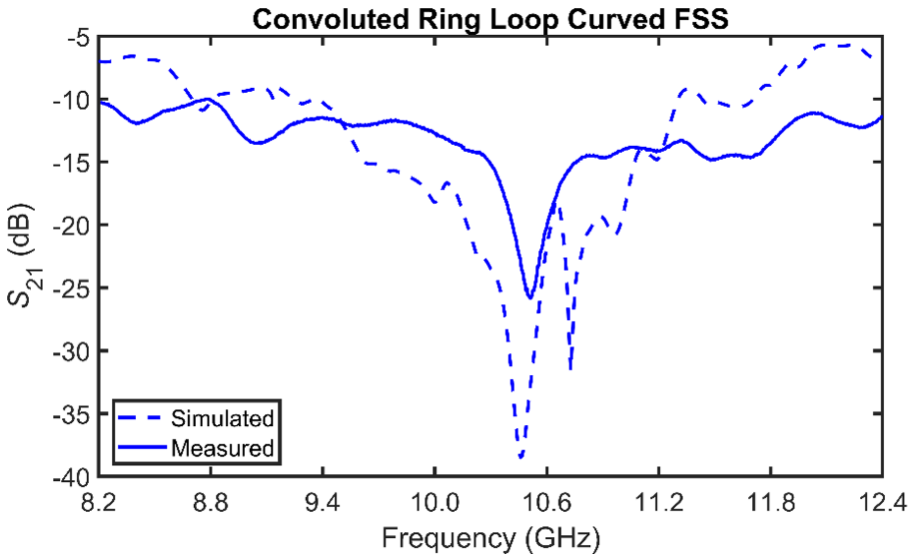


Figure 69. Free space measurement results of fully printed FSS.

Chapter 5: Printing Functionally Graded Materials

Functionally graded materials (FGMs) are materials that slowly transition between two or more different compositions as a function of position. Traditionally, manufacturing these devices was limited to gaseous methods of deposition using chemical vapors, centrifuging liquids, or aligning metallic powders and sintering them. However, these techniques led to problems such as high-energy consumption, hazardous byproducts, limitations in the resultant gradients, and instability in the material itself. With the rise in popularity and usage of additive manufacturing, research and interest in designing and making FGMs has increased. These recent advances in additive manufacturing have enabled the rapid prototyping and development of FGMs while reducing the negative aspects of traditionally manufacturing FGMs [95]–[98].

This chapter describes the ability to print FGMs by deposition of two mixed viscous materials. There have been multiple attempts to provide algorithms and software to design and plan out processes for depositing mixed materials. One such attempt involves tool paths being generated based on motion planning in robotics [99], [100]. Another is given in [101] where the authors generate paths based on contours of the model. Finally, [102] provides a thorough investigation of paths that could possibly be implemented for different types of FGMs. It is noted in their study that predictive measures may be required to realize properly graded paths and devices. The rest of this chapter aims to fulfill that need by providing a possible predictive solution as well as some resultant devices using these methods.

5.1 FGM Definition

The FGM material definition consists of the data used to describe the material gradient throughout a volume of space. The definition consists of a set of three-dimensional parametric equations that define a spline in space. The spline is used to set the material gradient information at fixed points to be distributed along the volume of the model. This method was chosen for its flexibility in altering the position of the gradient with the spline as well as its flexibility in altering the gradient transition.

The algorithm for generating the gradient went as follows: (1) define a spline for the material gradient to follow, (2) define a function for the profile of the material gradient along the spline, and (3) assign a mix ratio to each point in a part based on the gradient at the closest point on the spline. The spline was created by defining $x(s)$, $y(s)$, and $z(s)$ as the variable s progresses from 0 to 1 from the start to the end of the spline, respectively. The functions are defined to control the path of the spline and were to be within the boundaries of the part being printed. Figure 70 shows an example of a spline defining the direction of a gradient.

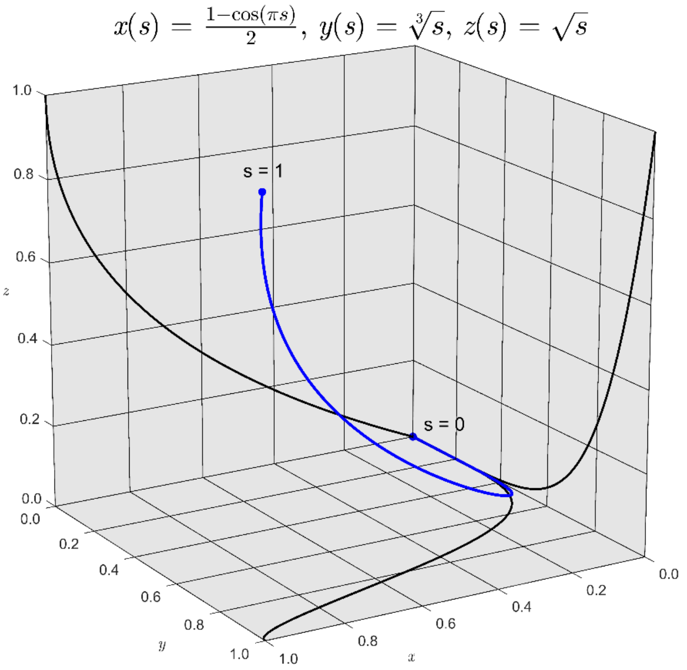


Figure 70. Example of a spline that defines the direction of the gradient.

Next, the mixing ratio along the length of the spline was defined by the function $m(s)$. The function ranges from 0 to 1 corresponding to a 0% to 100% mixing ratio, respectively, to realize the gradient. The function $m(s)$ was defined over the domain $0 \leq s \leq 1$ to cover the entire length of the spline through the part. Figure 71 shows different material profiles that can be generated through the function $m(s)$. The material gradient is then assigned to each position along the spline.

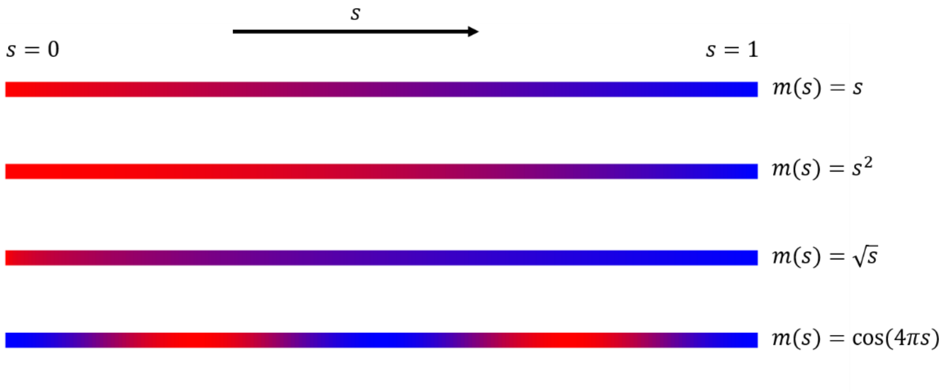


Figure 71. Examples of material profiles.

Last, the material mix ratios throughout the entire volume of the part are calculated. The mix ratio at any position inside the part is set equal to the mix ratio that is assigned to the nearest point on the spline. Figure 72 shows various gradients across a cube model where the spline is a straight line connecting one corner of the cube to the opposite corner. It should be noted that the spline does not necessarily have to be a straight line. In the present work, however, only a straight spline and a linear gradient were implemented.

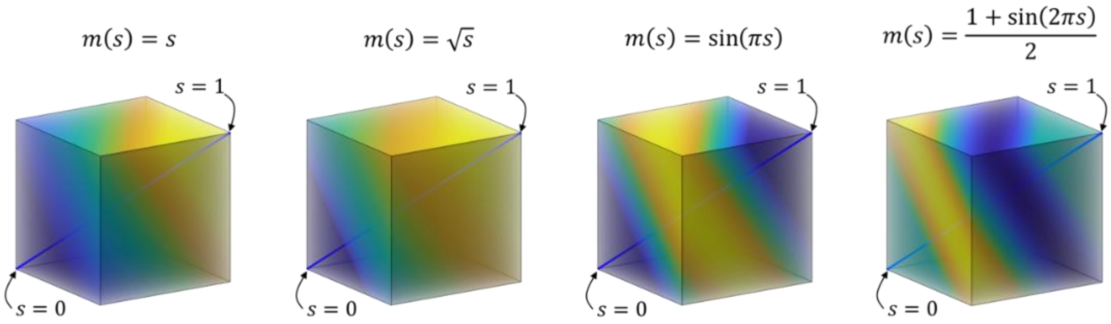


Figure 72. Cube models with varying gradient profiles along a straight line.

5.1.1 Parameters for Optimal FGM Algorithm Functionality

Throughout developing the algorithm, different tests were performed to ensure its viability before printing. First, tests were done to see if discretizing the spline profile would cause severe discretization of the overall gradient in the model. Figure 73 shows three attempts using different amounts of discrete points. The differences are slight, but it can be seen that for 50 discrete points within the spline, the gradient throughout the entire model is discretized. For 200 points, there is no visible change within the gradient. Ultimately, discretizing the spline to 100 points seemed to be the threshold where no visible improvement is seen by increasing the point count.

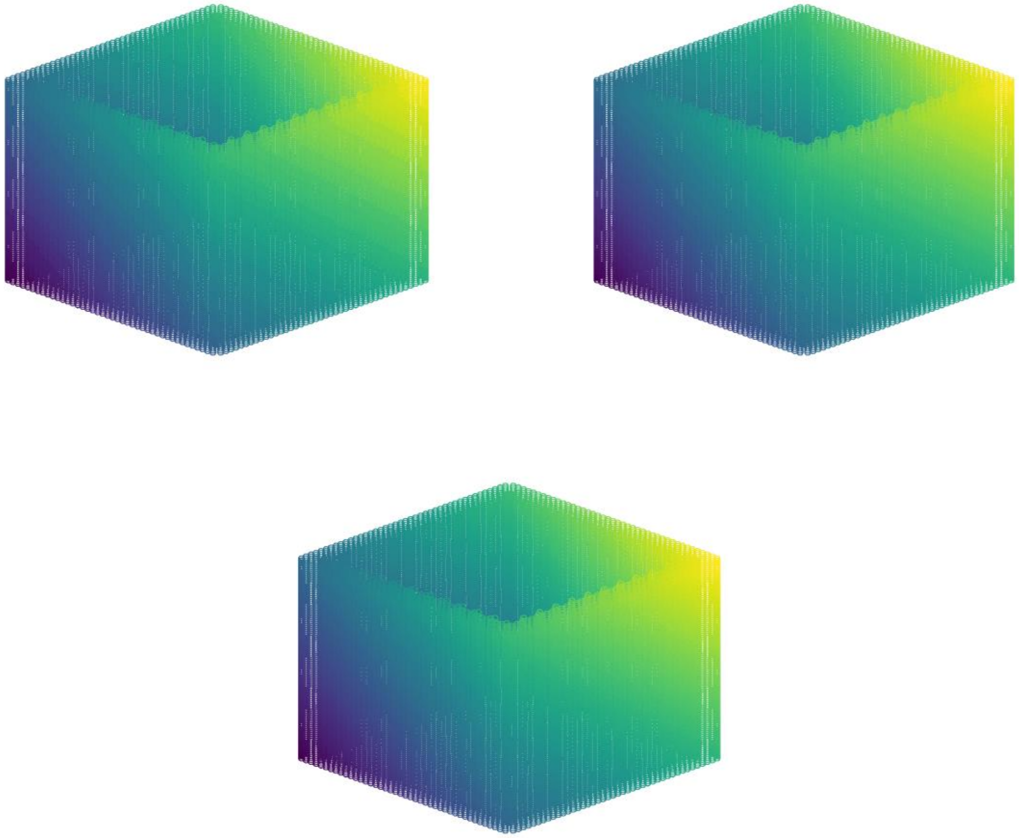


Figure 73. Results of number of points used in the gradient spline profile. (Top left) 50 points used. (Top right) 100 points used. (Bottom) 200 points used.

Secondly, a test was performed to see if the curvature of the gradient spline would affect the gradient of the model. It was found that curving the spline too much would result in sharp changes at different certain regions in the model, as seen in Figure 74. For certain regions, the mixing ratios of the opposite endpoints of the spline profile could be close together. Given these results, it is recommended to design the model and gradient in such a way that avoids these drastic changes. It is also recommended to keep the spline profile as close to the center of the model as possible. Models that have more complicated structures should have a spline that follows the overall curvature of the model to avoid the drastic changes.

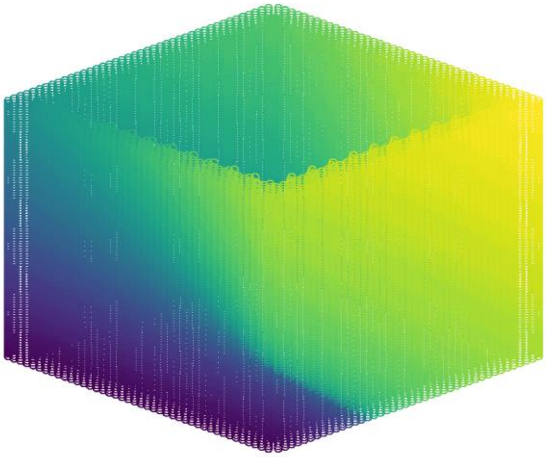


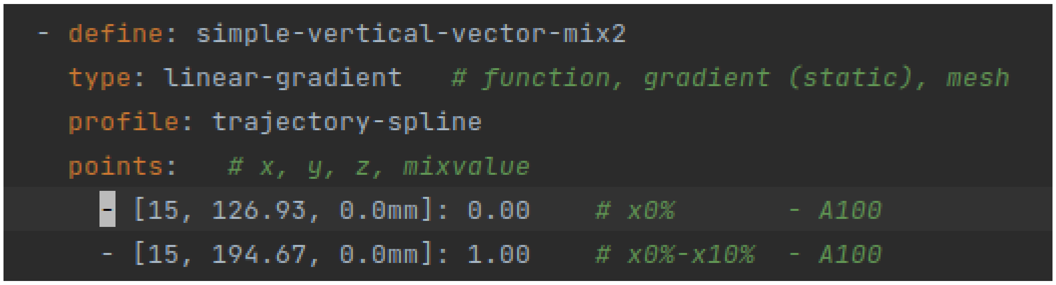
Figure 74. Curved spline and its corresponding material gradient. Drastic gradient changes can be seen towards the front of the cube.

5.2 FGM Implementation

There are three main components of the FGM implementation: the configuration, the calculation, and the g-code writing. The algorithm was implemented while considering waste material, print speed, and mixing time. It was assumed that there would be some delay in time when the printer would read the mixing command and when the printer would actually extrude the correctly mixing material, therefore needing a look-ahead routine. For testing purposes, only a linear spline profile was implemented.

5.2.1 FGM Configuration

The FGM configuration begins with defining two data points that describe the gradient spline definition and the mixing ratio definition. This is given as a set of two points with *x*, *y*, and *z* coordinates to describe the beginning and end points of the spline profile. An extra value at the following each point describes the mix ratio values at the endpoints. An example of the data inputs are shown in Figure 75. From these values, necessary information can be taken and used in both the slicing routines and g-code writing routines.



```
- define: simple-vertical-vector-mix2
  type: linear-gradient   # function, gradient (static), mesh
  profile: trajectory-spline
  points:   # x, y, z, mixvalue
    - [15, 126.93, 0.0mm]: 0.00    # x0%       - A100
    - [15, 194.67, 0.0mm]: 1.00    # x0%-x10%  - A100
```

Figure 75. Example of FGM input data.

5.2.2 FGM Calculation

Simply assigning the gradient information to the model was not sufficient because this did not consider anything about the printing hardware. In general, there will be a lag between when the printer begins mixing and when the mixed material is dispensed. The lag comes from the volume of space in the mixing auger's housing that the material travels through before reaching the end of the dispenser. Therefore, a path with a desired material mixture or a fast change in gradient cannot be immediately printed. Furthermore, not anticipating a mix ratio ahead of time can cause excess material waste when transitioning mix ratios. Algorithms were developed to take these issues into account and optimize the mix ratio dispensing. Optimized gradient control for a print was achieved by combining three routines: (1) a dynamic infill routine, (2) a purge routine, and (3) a look-ahead routine.

5.2.2.1 Infill Orientation

A straightforward gradient assignment to the points of the tool paths results in paths where the gradient changes too rapidly to print. Part (a) of Figure 76 shows this issue. The infill direction is 45° from the *x* axis while the gradient change is vertical along the *y* axis. Several individual line segments show a rapid change from one material to the next and back again. The volume needed to print each line is less than the lag volume needed to change a mix ratio. To address this issue, the infill orientation was aligned to the isocontours of the gradient for each slice. Part (b) of Figure 76 shows this adjustment where both the infill orientation and the isocontours of the gradient are aligned along the y axis.

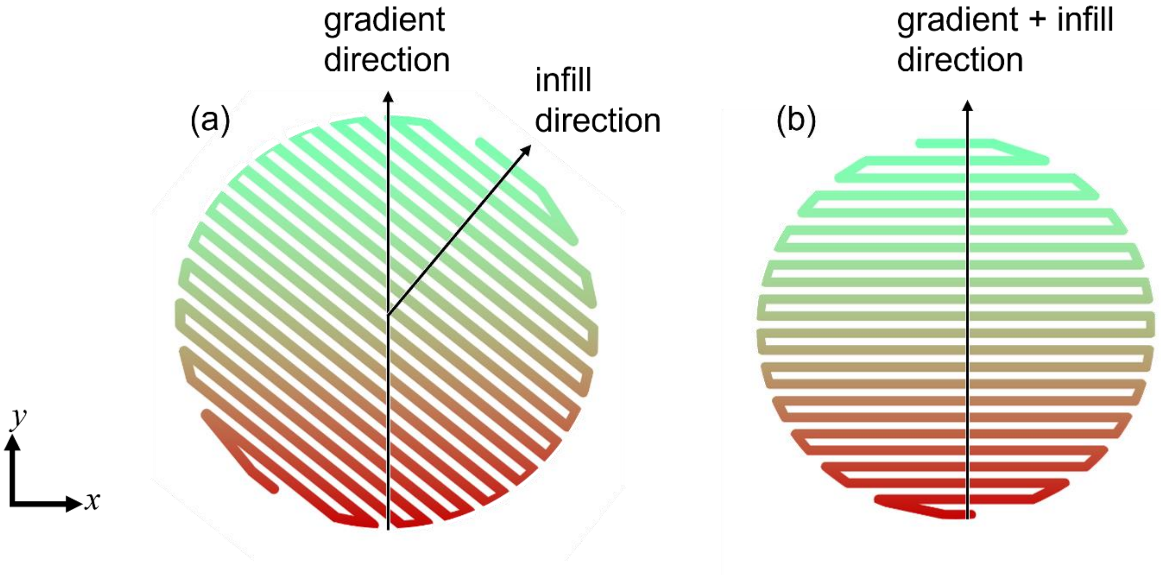


Figure 76. (a) Arbitrary gradient applied to arbitrary tool paths. (b) Tool paths aligned with gradient direction.

After the mix ratio profile and the spline profile are calculated, the mix ratio values are assigned accordingly to the generated tool paths. Each vertex within a tool path is assigned a single gradient matrix that describes the gradient of the following line segment. This gradient matrix defines the possibility of a gradient transition that might occur within a single tool path line segment, such as the tool paths on the left in Figure 76. The size of the matrix depends on two criteria: the difference in value between the two mixing ratios and the physical distance between the two endpoints. The difference in value between two mix points accounts for a sudden change in mix ratio data that needs to be graded over that line segment. The physical distance accounts for the possibility that a gradient might transition to one value and then back to the original value of the starting endpoint. In the end, the matrix will be of size M x 4 where there are M intermediate gradient points along a line segment. The four columns account for the x, y, and z position with the mix ratio appended.

5.2.2.2 Purge routine

The purge routine is a simple maneuver that consists of dispensing material in a waste tin away from the part in anticipation of a new mix ratio. The role of the purge is to bring the dispensed material to a certain mix ratio whenever the mix ratio rapidly changes, such as between layers. A user-defined tolerance was incorporated that initiates the routine every time the mixing ratio changes more than the defined tolerance. To execute the purge maneuver, the auger head is moved to the location of a waste tin and purges the necessary volume needed to make the transition in mix ratio.

The routine is essential to a functional print due to the lag volume previously described. According to [103], the lag volume needed to transition from 100% material A to 100% material B is 3.2 cm^3 for the printed used in the present work. This volume was assumed to be sufficient to change between any two mixing ratios since it purges the entire volume between the mixer and dispenser.

5.2.2.3 FGM Look-ahead

In conjunction with the purge routine and the infill reorientation, the look-ahead routine is essential for optimized gradient control. The look-ahead routine considers the volume lag of when the printer begins driving the pistons at a certain mix ratio to when that mix ratio is dispensed at the end of the nozzle. Without the look-ahead routine, the transition between materials would not be as continuous and accurate as desired. There are three phases during that encompass the look-ahead routine: (1) the priming phase, (2) the mid-slice phase, and (3) the end phase. Figure 77 shows the predictive mixing across these three phases.

During the priming phase, the correct starting mix ratio needs to be loaded into the chamber and be ready at the nozzle to begin dispensing. There are many possibilities of the chamber's state before starting to print a new layer. Therefore, it is necessary to purge enough material and fill the chamber with material of the starting mix ratio. This amount of material is the priming volume, and it should be set to at least the value of the lag volume. Once the starting mix ratio is reached, the algorithm looks ahead for any change in mix ratio according to the predefined tolerance previously described. The chamber is then filled with predictive mixing ratios that are to be printed during the layer. Figure 77(a) shows this priming step. The initial material is all red that slowly grades to blue. The chamber is therefore primed with a lag volume amount of material with predictive mixtures of materials.

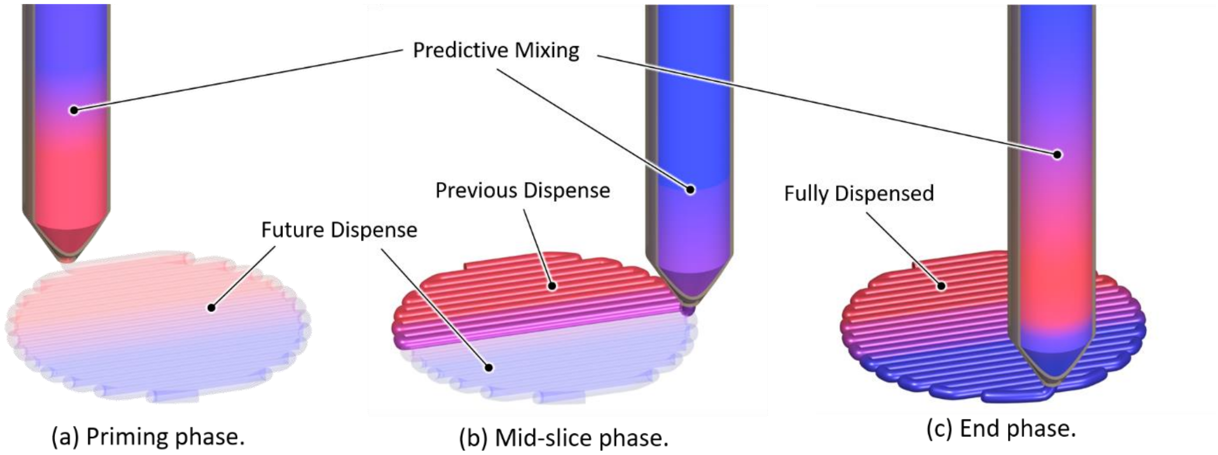


Figure 77. Look-ahead routine across a layer.

During the mid-slice phase, the layer is actively being printed while the material with future mix ratios of the layer is loaded into the chamber. Figure 77(b) shows this step. The current material being printed mid layer is purple, while the material being loaded into the chamber is blue. Once the end of the layer has been reached, the printer will begin preparation for the next slice.

During the end phase, when the layer is finished printing, the beginning material mixture of the beginning of the next layer is primed into the mixing auger. The mix ratio is set by purging a lag volume between the last mix ratio of the previous layer and the starting mix ratio of the next layer. To account for varieties that may have emerged when priming the material, a small volume is added to the purge volume after priming the starting mixture. Once the layer is complete, the printer initiates the purge. Once the purge is complete the entire look-ahead process repeats beginning with the priming phase. If there are no other tool paths to be printed, there is no change to final mix ratio. Figure 77(c) shows the final step in the look-ahead routine where the beginning red material of the next slice is prepared ahead of the current blue material. In the case of the fast transition as depicted in the figure, a fast transition may or may not happen depending on the material composition and the time when the next layer beings priming. In either case, the priming phase should be initiated with enough time to ensure the material with the starting mix ratio is ready to be printed.

5.2.3 FGM G-Code Writing

Figure 78 shows the flow chart for how the g-code is written with FGMs included. Figure 79 shows a portion of the g-code with the priming and look-ahead routines. The sets of commands for each routine are inserted in their corresponding positions as the g-code is being written.

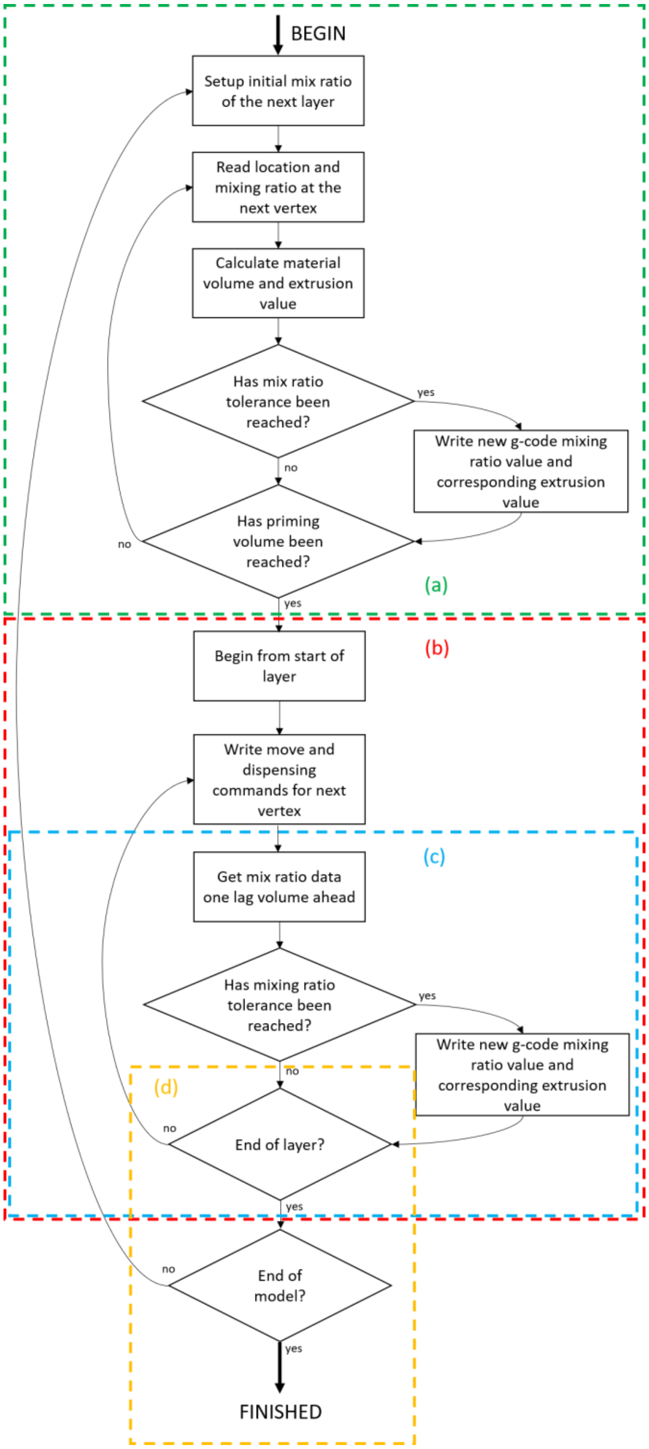
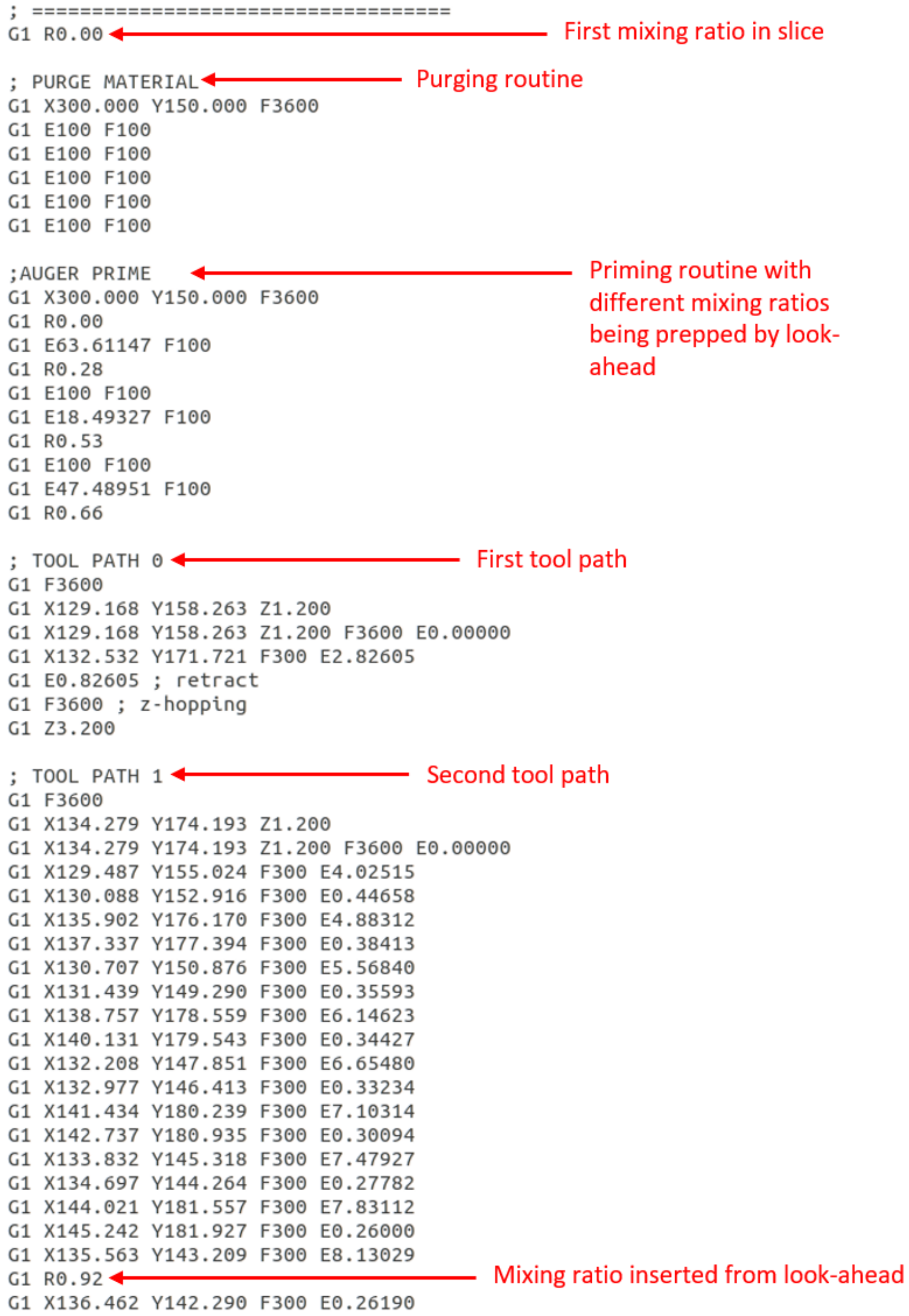

BEGIN
Setup initial mix ratio of the next layer
Read location and mixing ratio at the next vertex
Calculate material volume and extrusion value
Has mix ratio tolerance been reached?
yes
Write new g-code mixing ratio value and corresponding extrusion value
no
Has priming volume been reached?
no
yes
(a)
(b)
Begin from start of layer
Write move and dispensing commands for next vertex
(c)
Get mix ratio data one lag volume ahead
Has mixing ratio tolerance been reached?
yes
Write new g-code mixing ratio value and corresponding extrusion value
no
(d)
no
End of layer?
yes
no
End of model?
yes
FINISHED

Figure 78. G-code generation flowchart. (a) Priming phase. (b) Mid-slice phase. (c) Look-ahead routine nestled within the mid-slice phase. (d) End phase.



```
; ===================================
G1 R0.00

; PURGE MATERIAL
G1 X300.000 Y150.000 F3600
G1 E100 F100
G1 E100 F100
G1 E100 F100
G1 E100 F100
G1 E100 F100

;AUGER PRIME
G1 X300.000 Y150.000 F3600
G1 R0.00
G1 E63.61147 F100
G1 R0.28
G1 E100 F100
G1 E18.49327 F100
G1 R0.53
G1 E100 F100
G1 E47.48951 F100
G1 R0.66

; TOOL PATH 0
G1 F3600
G1 X129.168 Y158.263 Z1.200
G1 X129.168 Y158.263 Z1.200 F3600 E0.00000
G1 X132.532 Y171.721 F300 E2.82605
G1 E0.82605 ; retract
G1 F3600 ; z-hopping
G1 Z3.200

; TOOL PATH 1
G1 F3600
G1 X134.279 Y174.193 Z1.200
G1 X134.279 Y174.193 Z1.200 F3600 E0.00000
G1 X129.487 Y155.024 F300 E4.02515
G1 X130.088 Y152.916 F300 E0.44658
G1 X135.902 Y176.170 F300 E4.88312
G1 X137.337 Y177.394 F300 E0.38413
G1 X130.707 Y150.876 F300 E5.56840
G1 X131.439 Y149.290 F300 E0.35593
G1 X138.757 Y178.559 F300 E6.14623
G1 X140.131 Y179.543 F300 E0.34427
G1 X132.208 Y147.851 F300 E6.65480
G1 X132.977 Y146.413 F300 E0.33234
G1 X141.434 Y180.239 F300 E7.10314
G1 X142.737 Y180.935 F300 E0.30094
G1 X133.832 Y145.318 F300 E7.47927
G1 X134.697 Y144.264 F300 E0.27782
G1 X144.021 Y181.557 F300 E7.83112
G1 X145.242 Y181.927 F300 E0.26000
G1 X135.563 Y143.209 F300 E8.13029
G1 R0.92
G1 X136.462 Y142.290 F300 E0.26190
```

First mixing ratio in slice

Purging routine

Priming routine with different mixing ratios being prepped by look-ahead

First tool path

Second tool path

Mixing ratio inserted from look-ahead

Figure 79. Sample of g-code showing the results of the implemented routines.

In Figure 78(a), the priming phase is executed. The command to dispense initial mix ratio is written followed by the priming volume purge routine. This is shown as "; PURGE

MATERIAL” in Figure 79. The vertices of the following tool paths are read and used to calculate the accumulating volume to be dispensed. The mix ratios assigned to each vertex are also read. Two conditions are then tested: (1) if the mix ratio tolerance has been reached and (2) if the priming volume has been reached. New mix ratio commands are written every time the mix ratio goes beyond the tolerance. It continues to do this until the accumulated volume equals the priming volume. Once this happens, the priming routine is now complete and begins to print the layer. This portion of g-code is shown under “;AUGER PRIME” in Figure 79. At this point, none of the tool paths have been written to the g-code.

In Figure 78(b), the main loop for the mid-slice phase is shown. This part of the routine is straightforward as the vertices of the tool paths are read and written to the g-code. The look-ahead routine is executed as outlined in Figure 78(c). The vertex that is one lag volume ahead of the current vertex is read, and the corresponding mix ratio of the future vertex is compared to the mix ratio of the current vertex. A new mix ratio “R” value is written just after the current vertex if the difference is beyond the mix ratio tolerance. If a new layer is detected, the “R” values for the starting mix ratios are written as well. Once the layer is finished, the priming routine initiates and begins the next layer, shown in Figure 78(d). Once the last tool path is reached, the look-ahead routine no longer needs to execute and the mix ratio stops being updated.

5.3 MATERIALS AND HARDWARE

5.3.1 Modified Printer for FGM

The printers used to demonstrate the predictive algorithm were modified Lulzbot Taz 6 printers shown in Figure 80. This is the same printer setup used in [103]. The printers were fitted with three stepper motors that control two pistons and a mixing auger. Each piston houses the materials to be printed. The pistons feed viscous material via plastic tubes that connect to the auger's housing. All of the housings and the auger were printed with polyethylene terephthalate glycol (PETG) filament to allow for simple and low-cost swapping of parts.

An Arduino controller, shown in Figure 81 connected via I2C to the Lulzbot's control board provides the commands for the external stepper motors. The Arduino intercepts the input g-code file and parses its text to control the stepper motors analogous to a typical FDM printer. It does this by accepting a single "R" flag in the g-code file followed by a value between 0 and 1 that determines the mix ratio to be dispensed. A desired mix-ratio of 25% material A and 75% material B would be represented as "R0.25" in the g-code. The stepper motors for the pistons and the auger are driven by completing a certain number of turns according to the mix ratio. Following the previous example, for a mix ratio of 25% material A and 75% material B, for every turn of piston A, piston B will turn three times.

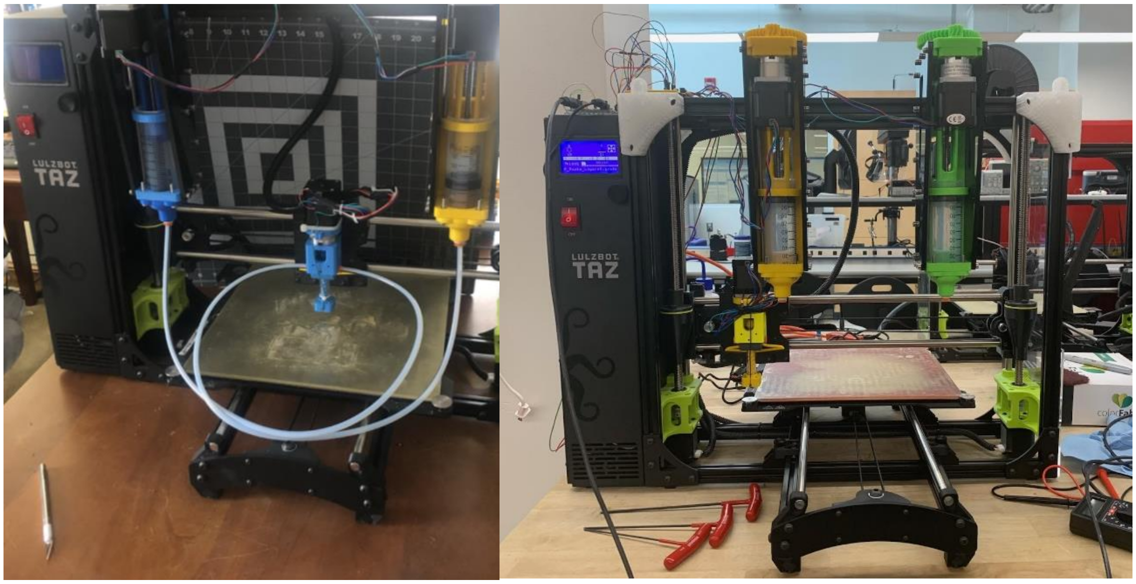


Figure 80. FGM modifications to two Lulzbot Taz 6 printers.

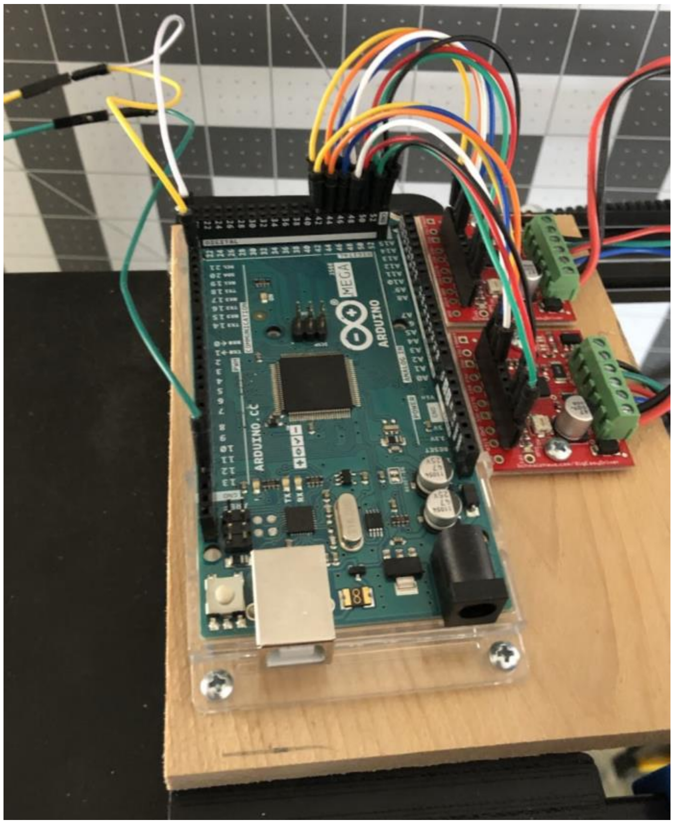


Figure 81. Arduino attached to the printer.

5.3.2 Materials Preparation

For the raw materials, food coloring was mixed into Colgate Optic White toothpaste to make two batches of two different colors. Toothpaste was used since it has a similar viscosity to certain ceramic pastes that would be used to build an FGM part, is non-toxic, inexpensive, easy to work with, and easy to clean. Food coloring was used to visually inspect the composition of the mixed materials. Red and blue colors were chosen due to their contrasting hues and simple inspection after mixing the two.

To prepare the materials, the toothpaste was mixed with the food coloring in separate containers and loaded into two syringes. Care was taken to ensure each loaded syringe did not have any air bubbles suspended in the toothpaste. Couplings to plastic tubes were attached to the ends of the syringes. After loading the syringes, the Arduino's serial monitor was used to move the pistons to a raised position. The syringes were then placed into the holders and screwed tight to the printer. To load the materials into the chamber, the serial monitor was used to push the pistons down and load enough material into the tubes until it reached the mixing chamber. Figure 82 shows the toothpaste mixture, and Figure 83 shows the loaded syringes attached to the printer.

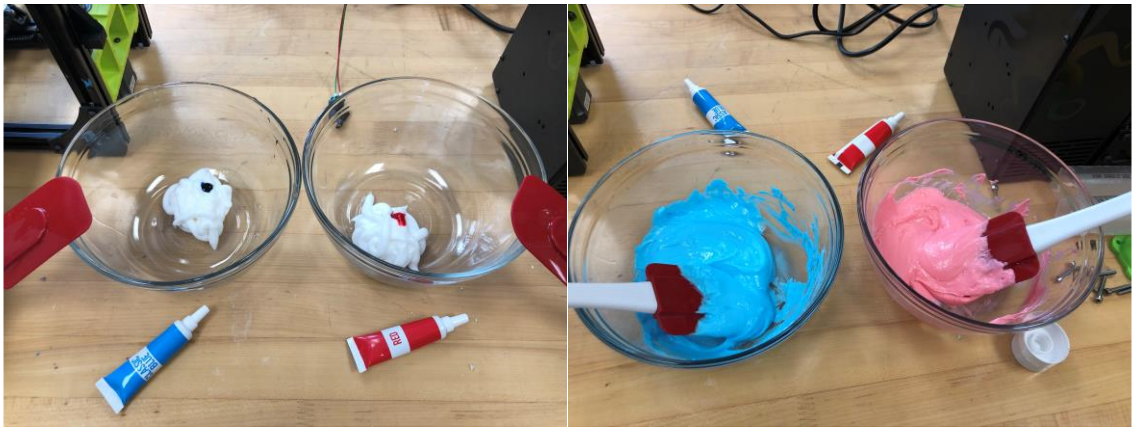


Figure 82. (Left) Toothpaste and food coloring before mixing. (Right) Toothpaste and food coloring after mixing.

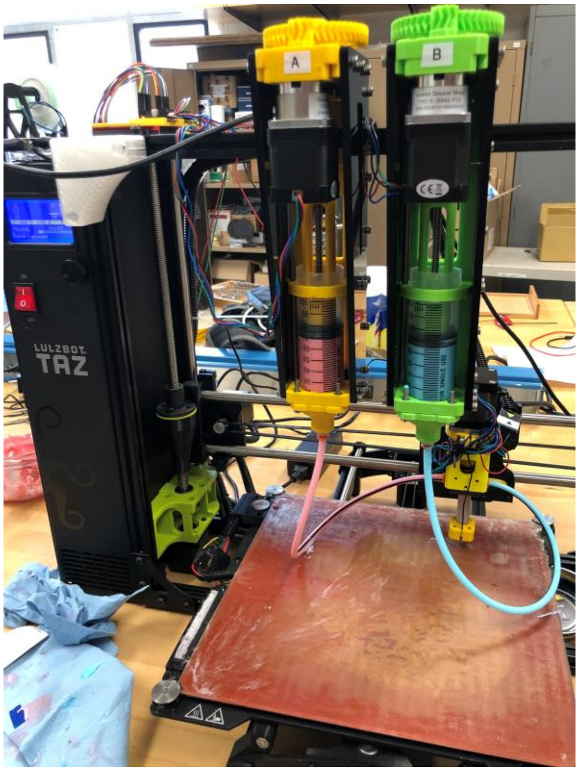


Figure 83. Syringes loaded onto the FGM printer.

5.4 Printed Prototype Devices

Several printed prototype components of both vertical and non-vertical gradient profiles were printed. The printed devices included: conical structures with a vertical gradient, one-layer disks with a horizontal gradient, and a conical device with a gradient running diagonally across normal to the side of the cone. Each print highlights the feasibility of the FGM implementation and suggests some recommendations for future printing FGMs. These prints were used as tests to continuously modify the FGM implementation and therefore at times do not exhibit complete successes of design.

5.4.1 Vertical Gradient Conical Device

Initial testing of the FGM algorithm involved printing small conical devices with a vertical gradient. This meant that there was no change in mix ratio within a single slice, and that purging would only happen between slices. The main objective of this print was to see how much material needed to be purged for a full print. Purging during this test was carried out by using the baseline 3.2 cm^3 chamber volume stated in [103] for each transition between mix ratios. The mix ratio tolerance was set to 2%. The results of this test led to the conclusion that the mix ratio tolerance directly affects the amount of waste material, particularly that the tighter the tolerance, the more material wasted. Figure 84 shows the printed device next to the wasted material. Figure 85 shows the vertical conical devices.

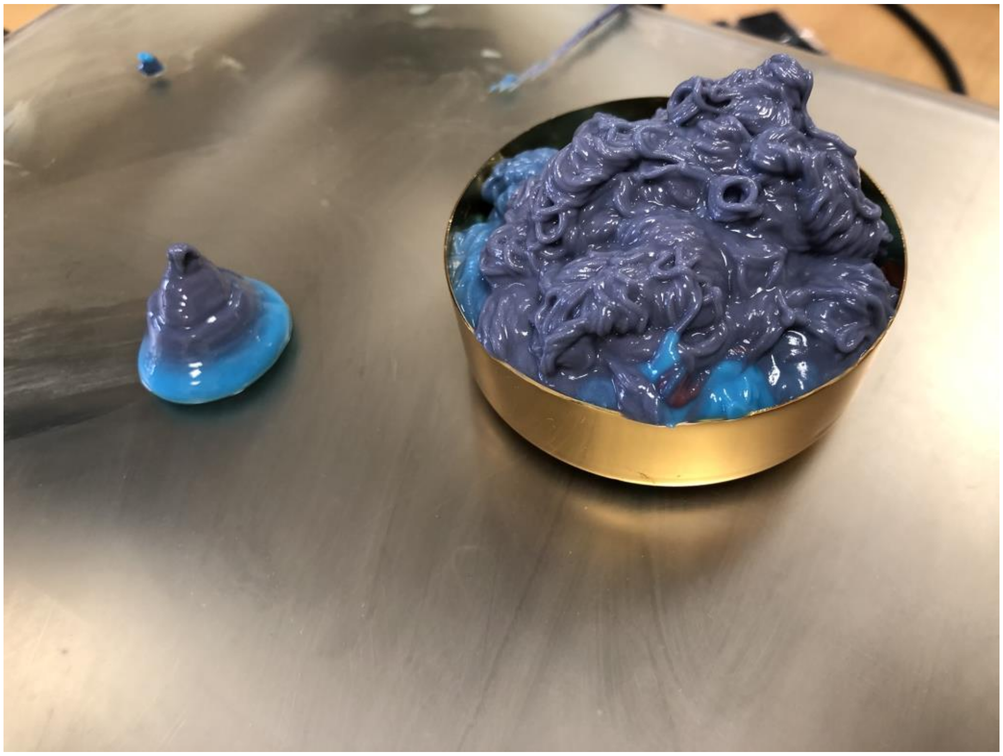


Figure 84. Printed conical device with vertical linear gradient net to the waste material required to print it.

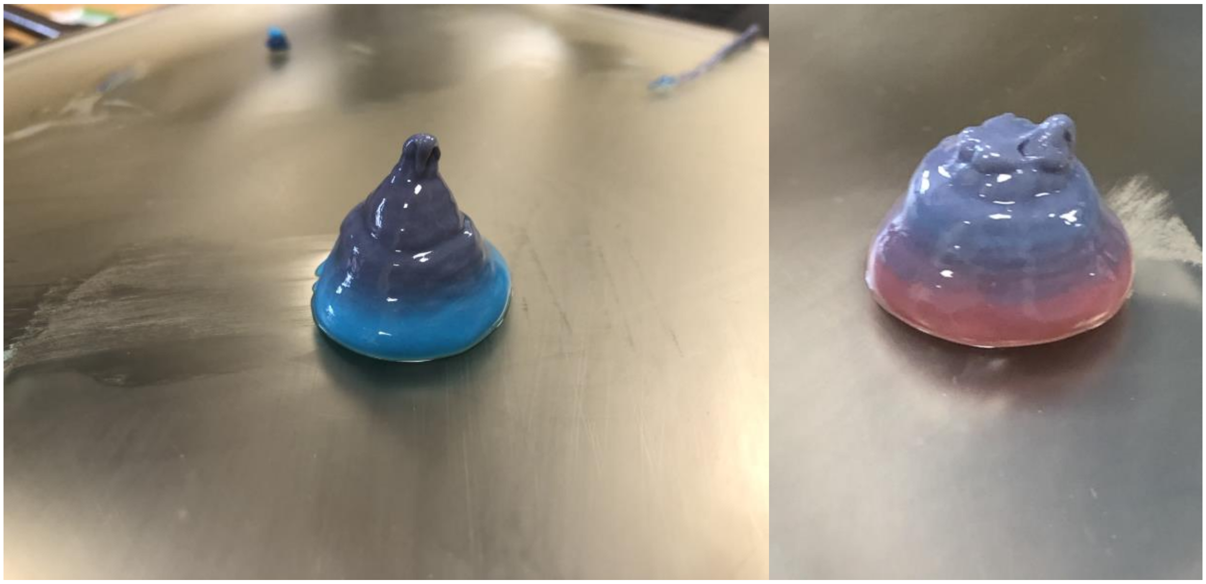


Figure 85. Conical devices with a vertical gradient.

Figure 84 depicts a large amount of waste material in comparison to the printed cone. It is evident that the amount of waste material for a 2% gradient tolerance is too much, and that higher tolerances would be more efficient at material usage.

5.4.2 Single-Layer Disk

Single-layer disks with a horizontal gradient were printed to demonstrate the look-ahead and transition capabilities from one mix ratio to another within a single slice. Figure 86 shows the successful attempts at printing a single-layer disk with a gradient. The gradient was set to change from 100% red-colored toothpaste to 100% blue-colored toothpaste for both attempts. The mix ratio tolerance for both prints were set at 25%. Part (a) depicts a disk 1 in. in diameter. Part (b) depicts a disk 3 in. in diameter. Each print had a different set of toothpaste and food coloring prepared and consequently show slight changes in hue.

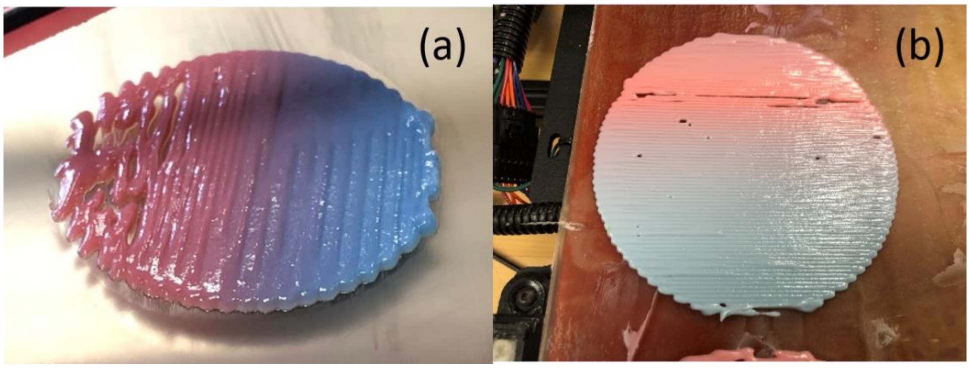


Figure 86. Gradient within a single-layer disk. (a) 1 in. diameter disk. (b) 3 in. diameter disk.

5.4.3 Conical Device with Diagonal Gradient

Finally, a conical device with a diagonal gradient running normal to the side of the cone was printed. Figure 87 shows the printed device. The cone was designed to grade from one material extreme at the side of the cone to the other at the end of the diagonal. This device was printed before the configurable parameters were implemented. The full cone top was not printed due to material running out from purging waste material between slices.

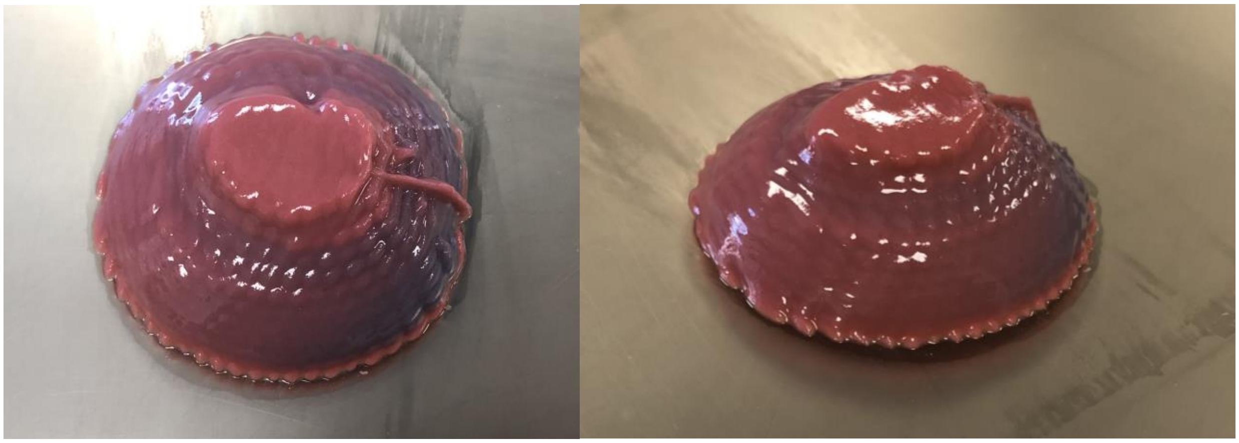


Figure 87. Conical device with a diagonal gradient.

Despite not creating the desired gradient change that was modeled, the results of the print provided enough assurance that the FGM algorithm was capable of printing full 3D models with a gradient change provided proper parameter configuration. This print along with the one-layer disks printed gave valuable insight into how the printer responds leading to implement configurable parameters. Because of the time allotted, a full 3D print with configurable parameters was unable to be tested.

5.5 DISCUSSION OF PRINTED PROTOTYPES

5.5.1 Purged Material

A recurring drawback that was encountered when testing the look-ahead was the need to purge a lot of material. Based on the findings in [103], a volume of 3.2 cm^3 was used as the amount of material to ensure a change from one gradient to another. Using this amount of material to change between mixing ratios resulted in a large amount of material needing to be purged before the desired mixing ratio was reached. To reduce the amount of material needed to be purged before the slice begins, it is recommended to begin the next slice at a point where the mixing ratio is the most similar to the previous slice.

Another recommendation for limiting the amount of material needed to be purged is to do a comprehensive study of the volume needed to change between two different mixing ratios. The value given in [103] was measured while changing from 100% material A to 100% material B. It is very likely that this is only the maximum volume needed to purge between routines, and changing between mix ratios that are less different may require less volume. Determining the range of volumes needed to purge between two arbitrary mixing ratios may further reduce the amount of waste material during a full print.

5.5.2 Feed Rate Limitations

During testing, it was observed that there was a limit as to how fast the printer's print speed can be set. The printer's speed is normally determined by an "F" value in the g-code, often called a feed rate. The feed rate sets the speed at which the stepper motors turn. The stepper motors for the *x*, *y*, and *z* axes are automatically adjusted so the overall displacement speed of the head is equal to the "F" value. This adjustment extends to the mixing auger driven by the Arduino.

On the other hand, the Arduino driven pistons operate on a fixed-speed step-based system rather than a variable pressure-based system. Their feed rates are not affected by a change in the printing speed. The static rate of the pistons gives limited range for the overall feed rate of the auger, both when it is printing material on the bed and when it is purging material on the side. If the auger feed rate is too fast, the material will dispense faster than the pistons are feeding the auger. This empties the auger too quickly. If the auger feed rate is too slow, the auger will back-flow with material, resulting in an unknown mix ratio. Due to these limitations, the printing speed that was found to work the best was around 5 mm/s.

5.5.3 Parameter Configuration

It was found that the most important parameters to configure were the mix ratio tolerance and the lag volume. Improper configuration of either of these values caused the gradient to be skewed, as depicted in Figure 88.

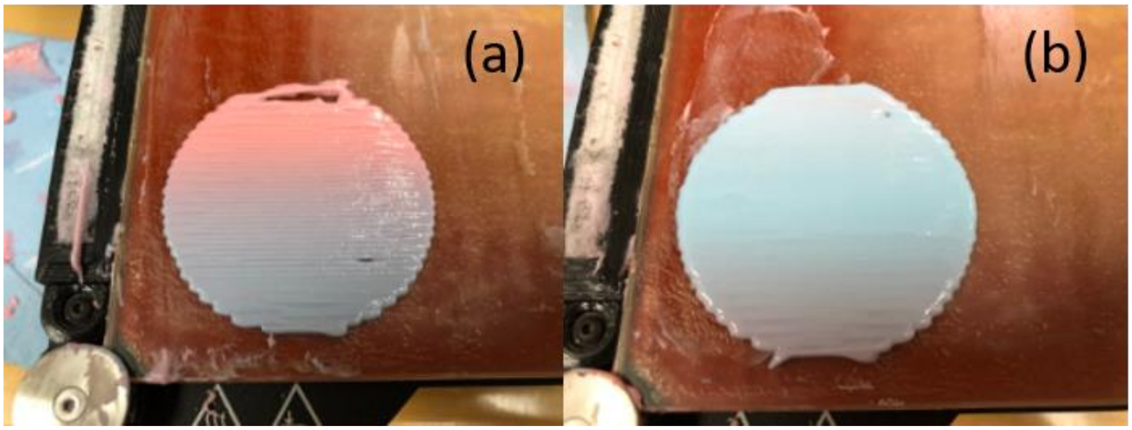


Figure 88. FGM prints with incorrectly set parameters. (a) Improper mix ratio tolerance. (b) Improper lag volume.

Considering the mix ratio tolerance, too small of a tolerance called the pistons to begin dispensing new mix ratios very frequently. This caused concurrent mix ratios in the prepared material to become indistinguishable from each other. This was due to the static feed rate of both pistons described in Section 5.5.3 and the number of turns needed to create a mix ratio described in Section 5.3.1. The mix ratio of the dispensed material is ultimately dependent on the amount of turns the stepper motors complete. Since the feed rates for both stepper motors are the same, both materials are fed into the auger at the same rate for a period of time. If the mix ratio tolerance is too small and the print speed is too high, the pistons and the auger will not be able to complete a proper mix ratio before performing the next look-ahead, resulting in an incomplete transition. Figure 88(a) shows an example of red toothpaste never completing the transition to blue toothpaste. Considering this fact, the mix ratio tolerance that seemed to work well with a print speed of 5 mm/s and yield successful prints was 25%.

The lag volume is another parameter that can result in an incorrect gradient change if not configured properly. Too small of a lag volume produces a similar issue to a small mix ratio tolerance. The new mix ratios will become too frequent and cause indistinguishable gradients. Too large of a lag volume will cause the gradient to not change quickly enough. This is shown in Figure 88(b) where the blue toothpaste seems to dominate the majority of the disk's area while never reaching the red section. The value that proved successful was the given 3.2 cm^3.

Chapter 6: Conclusion

In summary, this work provides insight into designing, slicing, and printing electronic and electromagnetic devices using a combination of new hybrid 3D printing processes. Previous state-of-the-art technology involved the use of five- or six- axis machines to print conformally, a *z* scanner to scan the profile of a surface to alter the machine instructions to print conformally, post-processing techniques for backfilling or depositing conductive paste for volumetric multi-material devices, and simple tool path planning for simple gradients in functionally-graded materials.

It was shown in this work that three-axis machines are more than capable of producing complex multi-material devices in a single print. Some of the new processes that were tested included off-axis printing of two materials, conformal printing onto extreme doubly-curved surfaces, and tool path ordering so tool switching becomes more optimized. To increase the reliability of prints, priming and cleaning routines were introduced in hybrid printing of thermoplastic and conductive paste. The processes used together produced several functional devices, including volumetric 3D circuits and conformal frequency selective surfaces. The successful demonstration of these devices shows that hybrid 3D printing is indeed a viable manufacturing method for complex devices without the need of complex or expensive machines.

Furthermore, functionally-graded materials are also capable of being printed within a single print using new printing processes as well. The algorithms presented for defining and printing functionally-graded materials include an arbitrary gradient definition across a volume, a look-ahead routine to predictively mix material to a desired mix ratio, and dynamically change the infill direction to reduce the amount of waste material needed to be purged. The algorithms were demonstrated by printing several structures with arbitrary gradients across them. These devices included single-layer disks and conical structures with gradients being oriented in different directions.

The accomplishments presented in this work hold significance in regards to the ability to print devices with complex features using simple 3D printers. Simply using software algorithms is sufficient to produce functional devices for electronic and electromagnetic applications. It can also be assumed that the presented algorithms can work on other deposition based 3D printing technologies. With this in mind, future considerations include incorporating other advanced tool pathing algorithms to improve the efficiency of printing. This may include incorporating automatic recognition of printable regions, automatic signal routing, in situ measurements of printed material, single tool path printing of a model, and multiple material mixing for FGMs.

A specific area of research to consider would be to improve the devices' functionalities to match those of simulations. At higher frequencies, the feature size of the elements and the device attributes become critically important. Deposition based 3D printers often times have trouble resolving small feature sizes. Furthermore, deposition based 3D printers often produce errors when printing if proper care is not taken when setting up parameters. While the most straightforward idea would be to use higher resolution tools or printing technologies, it is not far-fetched to assume that software algorithms could provide a possible solution to the issue. The success of this research would further enable more entities who own economic 3D printers to produce their own fully functional devices or perform their own research.

References

[1] T. Rayna and L. Striukova, “From rapid prototyping to home fabrication: How 3D printing is changing business model innovation,” *Technol. Forecast. Soc. Change*, 2016, doi: 10.1016/j.techfore.2015.07.023.

[2] C. W. Hull, “Apparatus for production of three-dimensional objects by stereolithography. U.S. Patent No 4,575,300. Washington DC: Patent and Trademak Office,” 1986.

[3] M. Feygin, “Apparatus and method for forming an integral object from laminations,” 1988.

[4] C. R. . Deckard, J. J. . Beaman, and J. F. Darrah, “Method and Apparatus for Producing Parts by Selective Sintering,” *U.S. Pat. 4,863,538*, 1992.

[5] S. S. Crump, “Apparatus and method for creating three-dimensional objects,” 1992.

[6] C. Biever, “3D Printer to churn out copies of itself,” *New Scientist Magazine*, 2005.

[7] E. Malone and H. Lipson, “Fab@Home: The personal desktop fabricator kit,” in *17th Solid Freeform Fabrication Symposium, SFF 2006*, 2006.

[8] Y. L. Kong, M. K. Gupta, B. N. Johnson, and M. C. McAlpine, “3D printed bionic nanodevices,” *Nano Today*. 2016, doi: 10.1016/j.nantod.2016.04.007.

[9] R. P. Visconti, V. Kasyanov, C. Gentile, J. Zhang, R. R. Markwald, and V. Mironov, “Towards organ printing: Engineering an intra-organ branched vascular tree,” *Expert Opinion on Biological Therapy*. 2010, doi: 10.1517/14712590903563352.

[10] D. Radenkovic, A. Solouk, and A. Seifalian, “Personalized development of human organs using 3D printing technology,” *Med. Hypotheses*, 2016, doi: 10.1016/j.mehy.2015.12.017.

[11] M. D’Auria *et al.*, “3-D Printed Metal-Pipe Rectangular Waveguides,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*. 2015, doi: 10.1109/TCPMT.2015.2462130.

[12] J. Yang *et al.*, “3D printed low-loss THz waveguide based on Kagome photonic crystal structure,” *Opt. Express*, 2016, doi: 10.1364/oe.24.022454.

[13] B. Zhang, Y.-X. Guo, H. Zirath, and Y. P. Zhang, “Investigation on 3-D-Printing Technologies for Millimeter- Wave and Terahertz Applications,” *Proc. IEEE*, 2017, doi: 10.1109/jproc.2016.2639520.

[14] A. Savini and G. G. Savini, “A short history of 3D printing, a technological revolution just started,” in *Proceedings of the 2015 ICOHTEC/IEEE International History of High-Technologies and their Socio-Cultural Contexts Conference, HISTELCON 2015: The 4th IEEE Region 8 Conference on the History of Electrotechnologies*, 2015, doi: 10.1109/HISTELCON.2015.7307314.

[15] M. Vaezi, H. Seitz, and S. Yang, “A review on 3D micro-additive manufacturing technologies,” *International Journal of Advanced Manufacturing Technology*. 2013, doi: 10.1007/s00170-012-4605-2.

[16] K. V. Wong and A. Hernandez, “A Review of Additive Manufacturing,” *ISRN Mech.*

Eng., 2012, doi: 10.5402/2012/208760.

[17] E. MacDonald and R. Wicker, “Multiprocess 3D printing for increasing component functionality,” *Science*. 2016, doi: 10.1126/science.aaf2093.

[18] S. Ambriz *et al.*, “Material handling and registration for an additive manufacturing-based hybrid system,” *J. Manuf. Syst.*, vol. 45, pp. 17–27, Oct. 2017, doi: 10.1016/J.JMSY.2017.07.003.

[19] C. Kim *et al.*, “Cooperative tool path planning for wire embedding on additively manufactured curved surfaces using robot kinematics,” *J. Mech. Robot.*, 2015, doi: 10.1115/1.4029473.

[20] E. MacDonald *et al.*, “Fabricating patch antennas within complex dielectric structures through multi-process 3D printing,” *J. Manuf. Process.*, vol. 34, pp. 197–203, Aug. 2018, doi: 10.1016/J.JMAPRO.2018.05.013.

[21] C. Bailey *et al.*, “Augmenting computer-aided design software with multi-functional capabilities to automate multi-process additive manufacturing,” *IEEE Access*, 2017, doi: 10.1109/ACCESS.2017.2781249.

[22] D. Espalin, D. W. Muse, E. MacDonald, and R. B. Wicker, “3D Printing multifunctionality: Structures with electronics,” *Int. J. Adv. Manuf. Technol.*, 2014, doi: 10.1007/s00170-014-5717-7.

[23] A. J. Lopes, E. MacDonald, and R. B. Wicker, “Integrating stereolithography and direct print technologies for 3D structural electronics fabrication,” *Rapid Prototyp. J.*, 2012, doi: 10.1108/13552541211212113.

[24] U. Robles, A. Kudzal, and R. C. Rumpf, “Automated Hybrid 3-D Printing of 3-D Meandering Interconnects,” *IEEE Trans. Components, Packag. Manuf. Technol.*, 2019, doi: 10.1109/TCPMT.2019.2909979.

[25] G. T. Carranza, U. Robles, C. L. Valle, J. J. Gutierrez, and R. C. Rumpf, “Design and Hybrid Additive Manufacturing of 3-D/Volumetric Electrical Circuits,” *IEEE Trans. Components, Packag. Manuf. Technol.*, 2019, doi: 10.1109/TCPMT.2019.2892389.

[26] A. J. Good, D. Roper, B. Good, S. Yarlagadda, and M. S. Mirotznik, “Multifunctional graded dielectrics fabricated using dry powder printing *,” 2017, doi: 10.1088/1361-665X/aa782b.

[27] P. Pa, Z. Larimore, P. Parsons, and M. Mirotznik, “Multi-material additive manufacturing of embedded low-profile antennas,” 2015, doi: 10.1049/el.2015.2186.

[28] K. McParland *et al.*, “Additive Manufacture of Custom Radiofrequency Connectors,” *IEEE Trans. COMPONENTS, Packag. Manuf. Technol.*, vol. 12, no. 1, 2022, doi: 10.1109/TCPMT.2021.3134603.

[29] K. McParland, Z. Larimore, P. Parsons, A. Good, and M. Mirotznik, “Additively Manufactured Conformal Feeds for Passive Beamforming,” *2021 Int. Conf. Electromagn. Adv. Appl.*, 2021, doi: 10.1109/ICEAA52647.2021.9539630.

[30] Z. J. Larimore, P. E. Parsons, A. Good, K. McParland, and M. Mirotznik, “Materials for Use in the Additive Manufacture of RF Components and Devices,” *2021 Int. Conf.*

Electromagn. Adv. Appl., 2021, doi: 10.1109/ICEAA52647.2021.9539699.

[31] Z. J. Larimore, “MULTI-MATERIAL ADDITIVE MANUFACTURE OF RADIOFREQUENCY DEVICES AND SYSTEMS,” 2019.

[32] T. P. Ketterl *et al.*, “A 2.45 GHz Phased Array Antenna Unit Cell Fabricated Using 3-D Multi-Layer Direct Digital Manufacturing; A 2.45 GHz Phased Array Antenna Unit Cell Fabricated Using 3-D Multi-Layer Direct Digital Manufacturing,” *IEEE Trans. Microw. Theory Tech.*, vol. 63, no. 12, 2015, doi: 10.1109/TMTT.2015.2496180.

[33] M. Kacar *et al.*, “Phased Array Antenna Element with Embedded Cavity and MMIC using Direct Digital Manufacturing.”

[34] K. H. Church *et al.*, “Multi-Material and Multi-Layer Direct Digital Manufacturing of 3D Structural Microwave Electronics,” doi: 10.1109/JPROC.2017.2653178.

[35] I. T. Nassar, T. M. Weller, and H. Tsang, “3-D Printed Antenna Arrays for Harmonic Radar Applications,” *WAMICON 2014*, 2014, doi: 10.1109/WAMICON.2014.6857741.

[36] P. B. Nesbitt, H. Tsang, T. P. Ketterl, K. Church, and T. M. Weller, “4 GHz 3D-printed balun-fed bowtie antenna with finite ground plane for gain and impedance matching enhancement; 4 GHz 3D-printed balun-fed bowtie antenna with finite ground plane for gain and impedance matching enhancement,” 2016, doi: 10.1109/WAMICON.2016.7483854.

[37] Y. Khan, A. Thielens, S. Muin, J. Ting, C. Baumbauer, and A. C. Arias, “905279 (1 of 29) A New Frontier of Printed Electronics: Flexible Hybrid Electronics,” 2019, doi: 10.1002/adma.201905279.

[38] J. G. Hester *et al.*, “Additively manufactured nanotechnology and origami-enabled flexible microwave electronics,” *Proc. IEEE*, 2015, doi: 10.1109/JPROC.2015.2405545.

[39] A. D. Valentine *et al.*, “Hybrid 3D Printing of Soft Electronics,” *Adv. Mater.*, 2017, doi: 10.1002/adma.201703817.

[40] K. Church *et al.*, “Printed electronic processes for flexible hybrid circuits and antennas,” in *2009 Flexible Electronics and Displays Conference and Exhibition, FLEX 2009*, 2009, doi: 10.1109/FEDC.2009.5069282.

[41] D. Corzo, G. Tostado-Blázquez, and D. Baran, “Flexible Electronics: Status, Challenges and Opportunities,” 1984, doi: 10.3389/felec.2020.594003.

[42] “FlexTech | SEMI.” [Online]. Available: https://www.semi.org/en/communities/flextech.

[43] “Flexible Hybrid Electronics Manufacturing Institute | NextFlex.” [Online]. Available: https://www.nextflex.us/.

[44] “Products – nScrypt.” [Online]. Available: https://www.nscrypt.com/products/.

[45] “Highly Quality 3D Printers Made in USA |Prinhead Wiki|Best Manufacturer Hyrel3D.” [Online]. Available: https://www.hyrel3d.com/.

[46] “Voxel8: 1st Electronics 3D Printer - 3D Printing Industry.” [Online]. Available: https://3dprintingindustry.com/news/voxel8-unleashes-electronics-3d-printer-ces-world-39060/.

[47] "DragonFly IV 3D Printer | Nano Dimension." [Online]. Available: https://www.nano-di.com/dragonfly-iv.

[48] "Optomec - 3D PRINTING SOLUTIONS FOR AN ADDITIVE MANUFACTURING WORLD." [Online]. Available: https://optomec.com/.

[49] A. Dolenc and I. Mäkelä, "Slicing procedures for layered manufacturing techniques," *Comput. Des.*, 1994, doi: 10.1016/0010-4485(94)90032-9.

[50] P. Mohan Pandey, N. Venkata Reddy, and S. G. Dhande, "Slicing procedures in layered manufacturing: A review," *Rapid Prototyp. J.*, 2003, doi: 10.1108/13552540310502185.

[51] M. Livesu, S. Ellero, J. Martínez, S. Lefebvre, and M. Attene, "From 3D models to 3D prints: an overview of the processing pipeline," *Comput. Graph. Forum*, 2017, doi: 10.1111/cgf.13147.

[52] "Slic3r." [Online]. Available: https://slic3r.org/.

[53] "Prusa Slicer." [Online]. Available: https://www.prusa3d.com/prusaslicer/.

[54] "Ultimaker Cura." [Online]. Available: https://ultimaker.com/software/ultimaker-cura.

[55] "Repetier." [Online]. Available: https://www.repetier.com/.

[56] "Professional 3D Printing Software | Simplify3D." [Online]. Available: https://www.simplify3d.com/.

[57] "MakerBot Print." [Online]. Available: https://www.makerbot.com/3d-printers/apps/makerbot-print/.

[58] D. J. Roach, C. M. Hamel, C. K. Dunn, M. V. Johnson, X. Kuang, and H. J. Qi, "The m4 3D printer: A multi-material multi-method additive manufacturing platform for future 3D printed structures," *Addit. Manuf.*, 2019, doi: 10.1016/j.addma.2019.100819.

[59] S. H. Choi and H. H. Cheung, "A multi-material virtual prototyping system," *CAD Comput. Aided Des.*, 2005, doi: 10.1016/j.cad.2004.06.002.

[60] S. H. Choi and A. M. M. Chan, "A virtual prototyping system for rapid product development," *CAD Comput. Aided Des.*, 2004, doi: 10.1016/S0010-4485(03)00110-6.

[61] "Slic3r Manual – Flow Math." [Online]. Available: https://manual.slic3r.org/advanced/flow-math. [Accessed: 04-Sep-2022].

[62] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 93, no. 4, pp. 1591–1595, 1996, doi: 10.1073/pnas.93.4.1591.

[63] D. Akhouri, D. Karmakar, D. Banerjee, and S. Mishra, "Various Infill Patterns and their Effect in 3D Printable Materials A Review Report on the Effect of Different Infill Patterns in 3D Printable Materials," *Int. J. Innov. Sci. Res. Technol.*, vol. 6, no. 9, 2021.

[64] "G-code - RepRap." [Online]. Available: https://reprap.org/wiki/G-code. [Accessed: 04-Sep-2022].

[65] B. Ge, A. N. Ghule, and D. C. Ludois, "Three-Dimensional Printed Fluid-Filled Electrostatic Rotating Machine Designed with Conformal Mapping Methods," *IEEE*

Trans. Ind. Appl., vol. 53, no. 5, doi: 10.1109/TIA.2017.2702585.

[66] N. S. Hmeidat, B. Brown, X. Jia, N. Vermaak, and B. Compton, “Effects of infill patterns on the strength and stiffness of 3D printed topologically optimized geometries,” doi: 10.1108/RPJ-11-2019-0290.

[67] Y. Gao, L. Wu, D. M. Yan, and L. Nan, “Near support-free multi-directional 3D printing via global-optimal decomposition,” *Graph. Models*, vol. 104, Jul. 2019, doi: 10.1016/J.GMOD.2019.101034.

[68] K. Xu, L. Chen, K. Tang, L. Chen, and K. Tang, “Support-Free Layered Process Planning Toward 3 + 2-Axis Additive Manufacturing,” *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, 2019, doi: 10.1109/TASE.2018.2867230.

[69] H. ming Zhao, Y. He, J. zhong Fu, and J. jiang Qiu, “Inclined layer printing for fused deposition modeling without assisted supporting structure,” *Robot. Comput. Integr. Manuf.*, 2018, doi: 10.1016/j.rcim.2017.11.011.

[70] U. F. D. E. Sergipe, “1 9% ¡« 75% ,˘‰,” no. 1, pp. 1–2, 2006.

[71] P. Nicholas, G. Rossi, E. Williams, M. Bennett, and T. Schork, “Integrating real-time multi-resolution scanning and machine learning for Conformal Robotic 3D Printing in Architecture,” *Int. J. Archit. Comput.*, vol. 18, no. 4, pp. 371–384, 2020, doi: 10.1177/1478077120948203.

[72] F. Alkadi, K. C. Lee, A. H. Bashiri, and J. W. Choi, “Conformal additive manufacturing using a direct-print process,” *Addit. Manuf.*, 2020, doi: 10.1016/j.addma.2019.100975.

[73] N. Bausch, D. P. Dawkins, R. Frei, and S. Klein, “3D Printing onto Unknown Uneven Surfaces,” *IFAC-PapersOnLine*, 2016, doi: 10.1016/j.ifacol.2016.10.664.

[74] “nScrypt Conformal Printing.” [Online]. Available: https://3dprintingindustry.com/news/nscrypt-patents-high-speed-3d-printing-method-for-patterning-conformal-surfaces-141064/.

[75] “ULTEM.” [Online]. Available: https://www.stratasys.com/materials/search/ultem9085.

[76] J. Zhang, L. Yan, R. X. K. Gao, C. Wang, and X. Zhao, “A Novel 3D Ultra-wide Stopband Frequency Selective Surface for 5G Electromagnetic Shielding,” *Proc. 2020 Int. Symp. Electromagn. Compat. - EMC Eur. EMC Eur. 2020*, pp. 2–5, 2020, doi: 10.1109/EMCEUROPE48519.2020.9245740.

[77] P. Gurrala, S. Oren, P. Liu, J. Song, and L. Dong, “Fully conformal square-patch frequency-selective surface toward wearable electromagnetic shielding,” *IEEE Antennas Wirel. Propag. Lett.*, 2017, doi: 10.1109/LAWP.2017.2735196.

[78] V. K. Varadan, K. J. Vinoy, K. A. Jose, and V. V. Varadan, “<title>Conformal fractal antenna and FSS for low-RCS applications</title>,” *Smart Struct. Mater. 2000 Smart Electron. MEMS*, vol. 3990, no. June 2000, pp. 138–145, 2000, doi: 10.1117/12.388892.

[79] C. Sudhendra, A. R. Madhu, A. C. R. Pillai, G. Shekar, T. S. Rukmini, and K. Rao, “Design and implementation of a novel rasorber for aircraft stealth applications,” *2014 1st Int. Conf. Comput. Syst. Commun. ICCSC 2014*, no. December, pp. 176–180, 2003, doi: 10.1109/COMPSC.2014.7032643.

[80] C. Xu, S. Qu, Y. Pang, J. Wang, M. Yan, and H. Ma, “A novel dual-stop-band FSS for infrared stealth application,” *2017 Int. Appl. Comput. Electromagn. Soc. Symp. China, ACES-China 2017*, pp. 13–14, 2017.

[81] L. Zhang, T. Ding, S. Zhang, and Y. Wang, “Active Frequency Selective Surface Based L-band Noise Rejection Conformal Radome,” *2018 Cross Strait Quad-Regional Radio Sci. Wirel. Technol. Conf. CSQRWC 2018*, pp. 30–31, 2018, doi: 10.1109/CSQRWC.2018.8455460.

[82] A. Chatterjee and S. K. Parui, “Beamwidth Control of Omnidirectional Antenna Using Conformal Frequency Selective Surface of Different Curvatures,” *IEEE Trans. Antennas Propag.*, vol. 66, no. 6, pp. 3225–3230, 2018, doi: 10.1109/TAP.2018.2819899.

[83] Y. Zouaoui, L. Talbi, K. Hettak, and N. K. Darimireddy, “Synthesis of Planar and Conformal Single-Layered Double-Sided Parallel-Cross Dipole FSS Based on Closed-Form Expression,” *IEEE Access*, vol. 9. pp. 104051–104058, 2021, doi: 10.1109/ACCESS.2021.3096419.

[84] A. A. Omar and Z. Shen, “Thin 3-D Bandpass Frequency-Selective Structure Based on Folded Substrate for Conformal Radome Applications,” *IEEE Trans. Antennas Propag.*, vol. 67, no. 1, pp. 282–290, 2019, doi: 10.1109/TAP.2018.2876706.

[85] Y. Zhang, T. Dong, D. Sun, Y. Wang, and S. Lu, “Electromagnetic Analysis of a Jigsaw-Shaped FSS for Conformal Application,” *2020 50th Eur. Microw. Conf. EuMC 2020*, no. January, pp. 768–771, 2021, doi: 10.23919/EuMC48046.2021.9338094.

[86] V. K. Kanth and S. Raghavan, “EM Design and Analysis of Frequency Selective Surface Based on Substrate-Integrated Waveguide Technology for Airborne Radome Application,” *IEEE Trans. Microw. Theory Tech.*, vol. 67, no. 5, pp. 1727–1739, 2019, doi: 10.1109/TMTT.2019.2905196.

[87] A. Chatterjee and S. K. Parui, “Frequency-Dependent Directive Radiation of Monopole-Dielectric Resonator Antenna Using a Conformal Frequency Selective Surface,” *IEEE Trans. Antennas Propag.*, 2017, doi: 10.1109/TAP.2017.2677914.

[88] P. Gurrala, S. Oren, P. Liu, J. Song, and L. Dong, “Fully conformal square-patch frequency-selective surface toward wearable electromagnetic shielding,” *IEEE Antennas Wirel. Propag. Lett.*, vol. 16, pp. 2602–2605, 2017, doi: 10.1109/LAWP.2017.2735196.

[89] V. Chaudhary and R. Panwar, “Hybrid ECM Blended Whale Optimization Derived Frequency-Selective Conformal EMI Shielding Structure Using Ferrite Substrate,” *IEEE Transactions on Magnetics*, vol. 57, no. 8. 2021, doi: 10.1109/TMAG.2021.3086724.

[90] “Microwave_Polarizer_Based_on_Complementary_Split_Ring_Resonators_Frequency-Selective_Surface_for_Conformal_Application.pdf.” .

[91] M. Zhao, J. Bi, J. Xu, and J. Zhao, “Millimeter Wave Fuze Radome Design Based Bandpass Frequency Selective Surface,” *Appl. Comput. Electromagn. Soc. J.*, vol. 36, no. 4, pp. 411–418, 2021, doi: 10.47037/2020.ACES.J.360406.

[92] M. Bilal, R. Saleem, Q. H. Abbasi, B. Kasi, and M. F. Shafique, “Miniaturized and Flexible FSS-Based em Shields for Conformal Applications,” *IEEE Trans. Electromagn. Compat.*, vol. 62, no. 5, pp. 1703–1710, 2020, doi: 10.1109/TEMC.2019.2961891.

[93] R. C. Rumpf and J. Pazos, “Synthesis of spatially variant lattices,” *Opt. Express*, vol. 20, no. 14, p. 15263, 2012, doi: 10.1364/oe.20.015263.

[94] C. L. Valle, G. T. Carranza, and R. C. Rumpf, “Conformal Frequency Selective Surfaces for Arbitrary Curvature,” *IEEE Antennas Propag. Soc. AP-S Int. Symp.*, 2021.

[95] Y. Li *et al.*, “A Review on Functionally Graded Materials and Structures via Additive Manufacturing: From Multi-Scale Design to Versatile Functional Properties,” *Adv. Mater. Technol.*, vol. 5, no. 6, 2020, doi: 10.1002/admt.201900981.

[96] R. M. Mahamood, E. T. Akinlabi, M. Shukla, and S. Pityana, “Functionally graded material: An overview,” *Lect. Notes Eng. Comput. Sci.*, vol. 3, pp. 1593–1597, 2012.

[97] B. Zhang, P. Jaiswal, R. Rai, and S. Nelaturi, “Additive Manufacturing of Functionally Graded Material Objects: A Review,” *J. Comput. Inf. Sci. Eng.*, vol. 18, no. 4, 2018, doi: 10.1115/1.4039683.

[98] C. Zhang *et al.*, “Additive manufacturing of functionally graded materials: A review,” *Mater. Sci. Eng. A*, vol. 764, no. May, p. 138209, 2019, doi: 10.1016/j.msea.2019.138209.

[99] T. Kirk, E. Galvan, R. Malak, and R. Arroyave, “Computational design of gradient paths in additively manufactured functionally graded materials,” *J. Mech. Des. Trans. ASME*, vol. 140, no. 11, pp. 1–9, 2018, doi: 10.1115/1.4040816.

[100] O. V. Eliseeva *et al.*, “Functionally Graded Materials through robotics-inspired path planning,” *Mater. Des.*, vol. 182, p. 107975, 2019, doi: 10.1016/j.matdes.2019.107975.

[101] R. C. Luo, Y. L. Pan, C. J. Wang, and Z. H. Huang, “Path planning and control of functionally graded materials for rapid tooling,” *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2006, no. May, pp. 883–888, 2006, doi: 10.1109/ROBOT.2006.1641821.

[102] P. Muller, J. Y. Hascoet, and P. Mognol, “Toolpaths for additive manufacturing of functionally graded materials (FGM) parts,” *Rapid Prototyp. J.*, vol. 20, no. 6, pp. 511–522, 2014, doi: 10.1108/RPJ-01-2013-0011.

[103] J. S. Pelz, N. Ku, W. T. Shoulders, M. A. Meyers, and L. R. Vargas-Gonzalez, “Multi-material additive manufacturing of functionally graded carbide ceramics via active, in-line mixing,” *Addit. Manuf.*, vol. 37, no. June 2020, p. 101647, 2021, doi: 10.1016/j.addma.2020.101647.

Glossary

3DVC – three-dimensional volumetric circuit

ABS – acrylonitrile butadiene styrene

EM – electromagnetic

FDM – fused deposition modeling

FGM – functionally graded materials

FMM – fast marching method

FSS – frequency selective surface

PLA – polylactic acid

μD – micro dispensing

Appendix

APPENDIX A – CONFIGURABLE PRINTING PARAMETERS

Table 2. Complete FDM printing parameters used.

Name	Units	Description
Layer height	mm	Height or thickness of each layer
Line width	mm	Width of infill lines at 100% infill
Line spacing	mm	Spacing between infill lines at 100% infill
Infill angle	°	Angle of the infill orientation
Print speed	mm/sec	How fast the tool head moves while printing
Filament diameter	mm	Diameter of the filament
Extrusion factor	N/A	Multiple of flow when extruding
Shell count	#	Number of shells to be printed
Nozzle Temperature	°F	Temperature of the extruder nozzle
Bed Temperature	°F	Temperature of the bed
Jump speed	mm/sec	How fast the tool moves when not printing
Tip size	mm	Diameter of the pen tip (nozzle)
Seam	bool	Same starting position for each printed layer
Hopping distance	mm	How far the tool moves up in *z* after finishing a path
Hopping speed	mm/sec	How fast the tool moves in z while hopping
Infill percentage	%	How much infill is printed in a layer
Retraction distance	mm	How far the filament retracts
Retraction speed	mm/sec	How fast the filament retracts
Unretraction distance	mm	How far the filament extrudes at the start of a path
Unretraction speed	mm/sec	How fast the filament extrudes at the start of a path

Table 3. Complete SmartPump™ printing parameters used.

Name	Units	Description
Layer height	mm	Height or thickness of each dispensed
Line width	mm	Width of infill lines at 100% infill
Infill angle	°	Angle of the infill orientation
Print speed	mm/sec	How fast the tool head moves while printing
Extrusion factor	N/A	Multiple of flow when extruding
Shell count	#	Number of shells to be printed
Bed Temperature	°F	Temperature of the bed
Jump speed	mm/sec	How fast the tool moves when not printing
Tip size	mm	Diameter of the pen tip (nozzle)
Seam	bool	Same starting position for each printed layer
Hopping distance	mm	How far the tool moves up in *z* after finishing a path
Hopping speed	mm/sec	How fast the tool moves in z while hopping
Infill percentage	%	How much infill is printed in a layer
Conformal print speed	mm/sec	Speed of the tool when conformal printing
Steep conformal speed	mm/sec	Speed of the tool when steep conformal printing
Conformal offset	mm	Physical offset when conformal printing
Steep conformal offset	mm	Physical offset when steep conformal printing
Pressure	PSI	Pressure of the SmartPump™ for dispensing
Valve speed	mm/sec	Speed of the SmartPump™ valve

APPENDIX B – CALCULATION TOLERANCES

The floating-point tolerances when calculating the perimeters and tool paths of a layer are generally unique to the resolution of the tools being used. For FDM processes in particular, these generally are limited to the nozzle diameter and the smallest possible movement that can be achieved. For the nScrypt nFD™, the tool can resolve a movement as small as 0.001 mm and can contain a pen tip as small as 12.5 µm. While it may seem safe to set the tolerance to the smallest possible value of 0.001 mm, it is actually better to set the limit based on the pen tip diameter. This is because generated line segments within the tool paths can be smaller than the pen tip diameter. While these small segments can technically be printed, the results are usually of poor quality and lead to defects in the printed device. The printed trace in this case is severely under-extruded and cannot adhere properly to the surface. The result is a blob of material that easily detaches from the device. Additionally, the inclusion of small segments can increase the file size of the g-code. This can cause issues when loading and executing the file in the printer. Therefore, it is advisable to set the tolerance to the pen tip size to remove small segments within the tool paths.

APPENDIX C – USEFUL GEOMETRIC FORMULAS

Plane/Line-segment Intersection

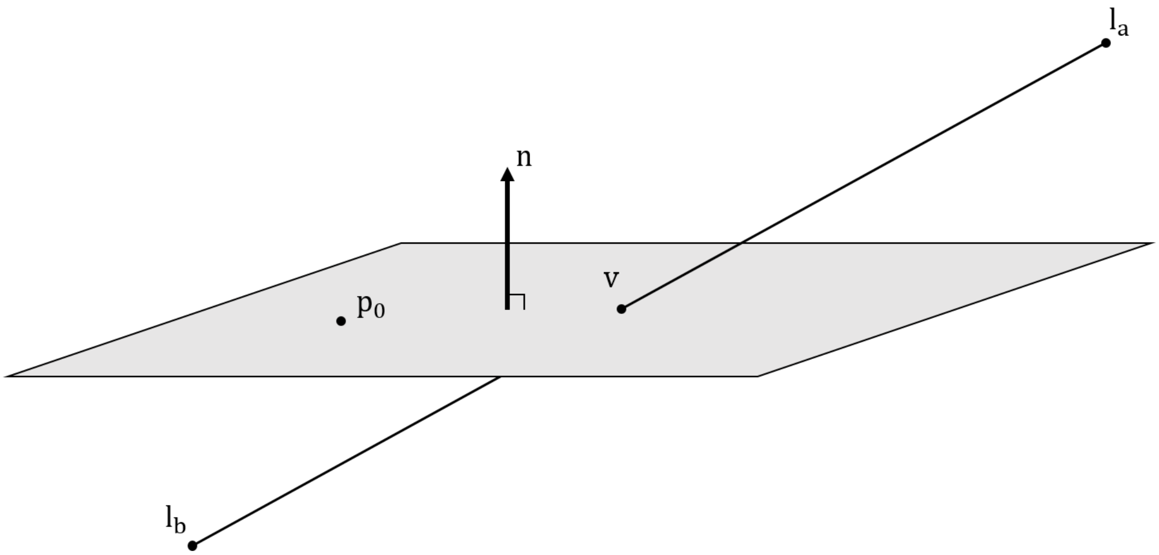


Figure 89. Plane intersecting with a line segment.

For a line segment with endpoints l_a and l_b, a point on the plane p_0, a normal vector n to the plane, and the vector of the line segment $l = l_a - l_b$, the vertex v that intersects the plane is defined by the equation

$$v = \frac{(p_0 - l_0) \cdot n}{l \cdot n} \tag{16}$$

where l_0 is any point on the line segment.

Line-segment/Line-segment Intersection

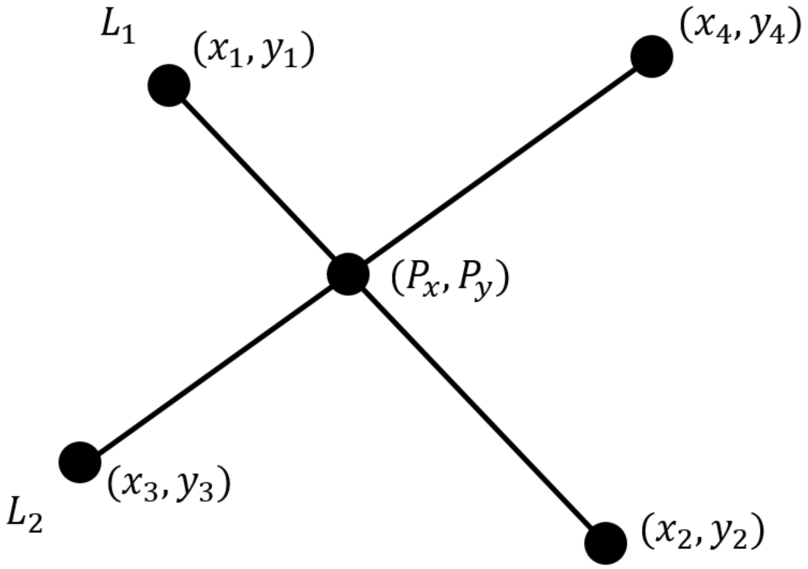


Figure 90. Intersection of two line segments.

Given

$$L_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + t \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} \tag{17}$$

$$L_2 = \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} + u \begin{bmatrix} x_4 - x_3 \\ y_4 - y_3 \end{bmatrix} \tag{18}$$

Define *t* and *u* as

$$t = \frac{\begin{vmatrix} x_1 - x_3 & x_3 - x_4 \\ y_1 - y_3 & y_3 - y_4 \end{vmatrix}}{\begin{vmatrix} x_1 - x_2 & x_3 - x_4 \\ y_1 - y_2 & y_3 - y_4 \end{vmatrix}} \tag{19}$$

$$u = \frac{\begin{vmatrix} x_1 - x_3 & x_1 - x_2 \\ y_1 - y_3 & y_1 - y_2 \end{vmatrix}}{\begin{vmatrix} x_1 - x_2 & x_3 - x_4 \\ y_1 - y_2 & y_3 - y_4 \end{vmatrix}} \tag{20}$$

Then

$$\left(P_x, P_y\right) = (x_1 + t(x_2 - x_1), y_1 + t(y_2 - y_1)) \tag{21}$$

or

$$\left(P_x, P_y\right) = (x_3 + u(x_4 - x_3), y_3 + t(y_4 - y_3)) \tag{22}$$

if

$$0 \leq \mathrm{t} \leq 1 \quad \text{and} \quad 0 \leq u \leq 1 \tag{23}$$

Area of a Polygon

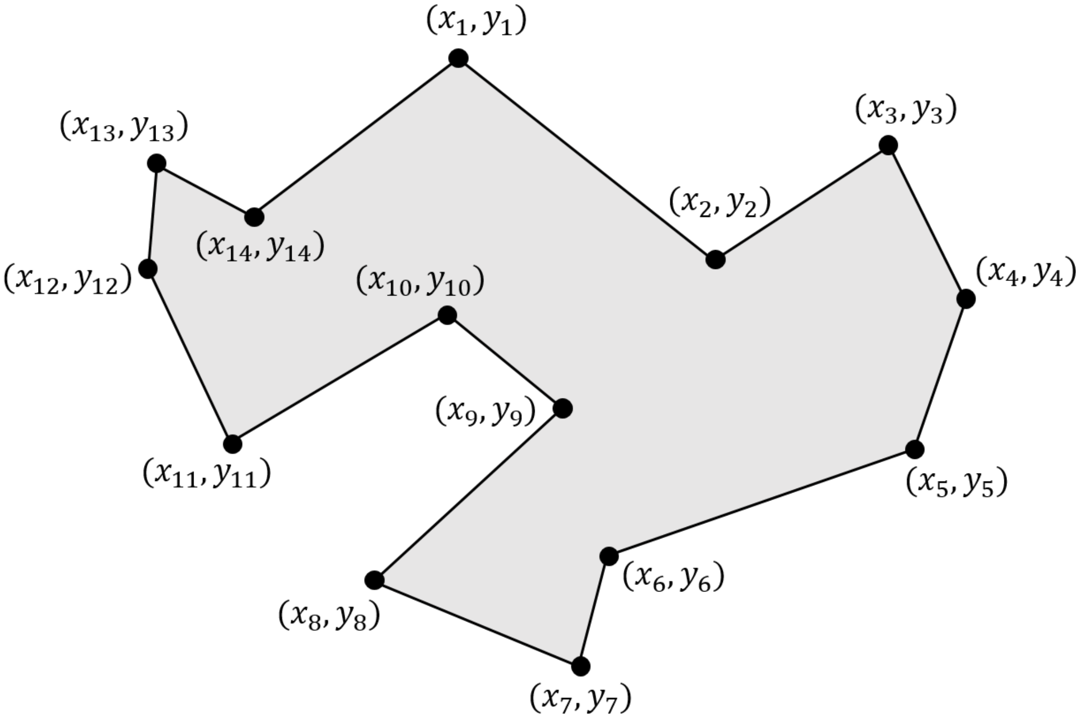


Figure 91. Random polygon with many vertices.

For a list of N vertices that trace the perimeter of the polygon (that is non-intersecting), the area is defined as

$$\text{Area} = \left|\frac{(x_1y_2 - y_1x_2) + (x_2y_3 - y_2x_3) + \cdots + (x_Ny_1 - y_Nx_1)}{2}\right| \tag{24}$$

It should be noted that the order of the vertices does not matter as an absolute value is taken to obtain the area. The order of the vertices can be determined in the following section.

Direction of Polygon Vertices

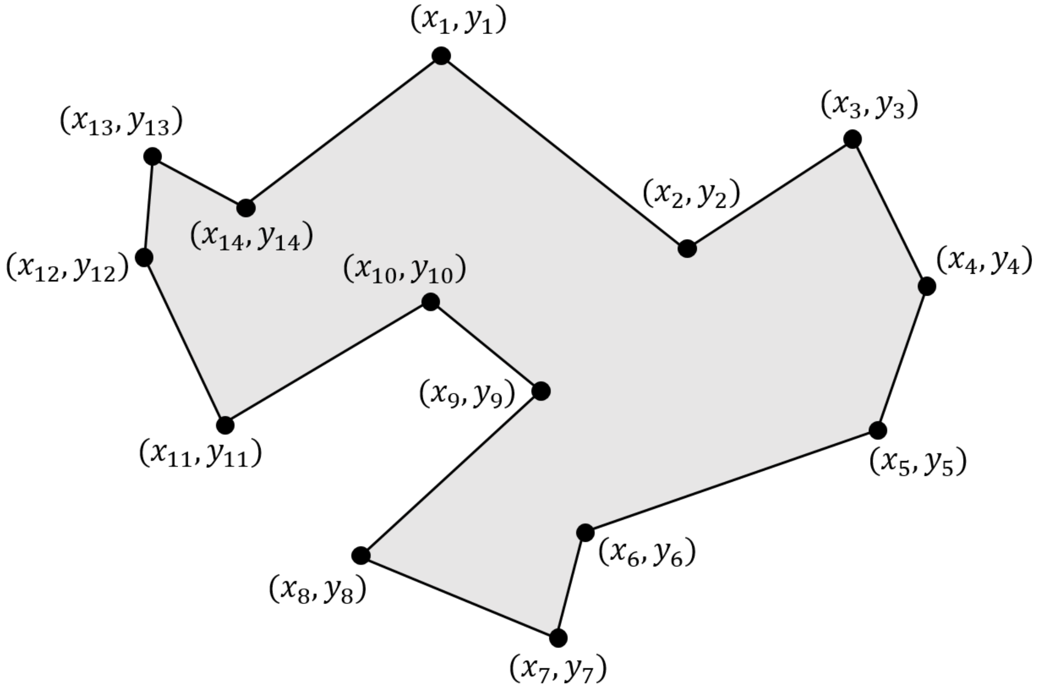


Figure 92. Random polygon with many vertices.

For a list of N vertices that trace the perimeter of the polygon (that is non-intersecting), the direction of the vertices is determined as

$$\text{CW if sign}\left(\frac{(x_1y_2 - y_1x_2) + (x_2y_3 - y_2x_3) + \cdots + (x_Ny_1 - y_Nx_1)}{2}\right) > 0 \tag{25}$$

$$\text{CCW if sign}\left(\frac{(x_1y_2 - y_1x_2) + (x_2y_3 - y_2x_3) + \cdots + (x_Ny_1 - y_Nx_1)}{2}\right) < 0 \tag{26}$$

APPENDIX D – EXTRUSION MATH

The value used for the “E” flag in a “G1” command represents the amount of material to be extruded in a typical FDM process. This number represents the distance the axis needs to turn when feeding the filament into the heated nozzle. To find this value, we start with a volume equalization

$$V_{\text{in}} = V_{\text{out}} \tag{27}$$

where V_{in} is the volume of material feeding into the nozzle and V_{out} is the melted filament coming out of the nozzle. Assuming the filament feeding into the nozzle can be modeled as a cylinder, Equation (27) can be expanded to the following

$$\pi \left(\frac{d_{\text{filament}}}{2}\right)^2 l_E = V_{\text{out}} \tag{28}$$

$$l_{\text{E}} = V_{\text{out}} \frac{4}{\pi d_{\text{filament}}^2} \tag{29}$$

where l_{E} is the value used for the “E” flag and d_{filament} is the diameter of the filament.

The value V_{out} can be determined by modeling a printed trace. Different models exist, but the model in Figure 93 is a good estimate of the cross section of a printed trace.

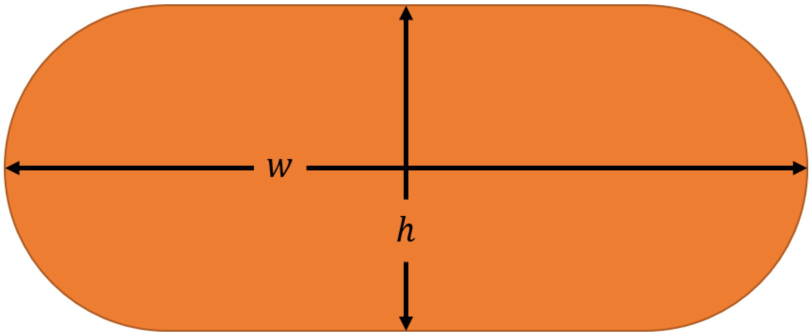


Figure 93. Printed trace cross section.

The model consists of a rectangle with two semicircles on both sides. The layer height is h, the total line width is *w*, giving the cross sectional area as

$$A = h(w - h) + \pi \left(\frac{h}{2}\right)^2 \tag{30}$$

Therefore, the volume of a printed trace printed over some distance *l* is

$$V_{\mathrm{out}} = Al \tag{31}$$

$$V_{\mathrm{out}} = l\left(h(w - h) + \pi \left(\frac{h}{2}\right)^2\right) \tag{32}$$

which gives the axis extrusion length value as

$$l_{\mathrm{E}} = \frac{4l}{\pi d_{\mathrm{filament}}^2}\left(h(w - h) + \pi \left(\frac{h}{2}\right)^2\right) \tag{33}$$

The value l_{E} can be finely tuned by an extrusion factor, giving the final value as

$$l_{\mathrm{E}}' = f_{\mathrm{extrusion}} l_{\mathrm{E}} \tag{34}$$

Vita

Gilbert Thomas Carranza is an El Paso native who obtained both his bachelor's degree in 2017 and master's degree in 2019 from the University of Texas – El Paso. He began to work in the EM Lab under Dr. Raymond C. Rumpf in 2016 designing 3D circuits. Since then, his expertise has grown to include computational electromagnetics, computational geometry, software development, electromagnetics measurement, and advanced hybrid 3D printing. At the time of this dissertation, he is the first author of three peer-reviewed articles (two articles under review), a co-author of one article, and a holder of a U.S. patent for his contribution in 3D volumetric circuit design and manufacturing. Gilbert is also an avid musician and golfer.

Email: gtcarranza2@gmail.com