ONE STEP FABRICATION: SOFTWARE-AUTOMATED PRINTED CIRCUITS VIA MULTI-MATERIAL ADDITIVE MANUFACTURING

GABRIEL SANCHEZ

Master's in Computer Engineering

APPROVED:

____
Rodrigo Romero, Ph.D., Chair

____
Jesus Gutierrez, Ph.D.,

____
Francisco Medina, Ph.D.,

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

Dedication

To the Creator of this complex system, whose beauty can’t be fully acquired but better understood. To my family and friends whom I have shared this journey with.

ONE STEP FABRICATION: SOFTWARE-AUTOMATED PRINTED CIRCUITS VIA MULTI-MATERIAL ADDITIVE MANUFACTURING

By

GABRIEL SANCHEZ RANGEL, B.S.

THESIS

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

MASTER OF SCIENCE

Department of Electrical and Computer Engineering
THE UNIVERSITY OF TEXAS AT EL PASO
December 2025

Acknowledgments

To my mentor, Dr. Romero, for his constant guidance and effective suggestions. To Dr. Gutierrez for his valuable insight. To Dr. Medina for reviewing this thesis and offering helpful feedback.

Abstract

As of now, the creation of 3D printed components faces many issues regarding stability, repeatability, and manual labor. Existing methods lack an integrated software framework that automates circuit generation while ensuring predictable behavior in printed geometries. In this work, circuits are modeled by leveraging distributed equations that reduce to lumped approximations when the printed geometry remains uniform. The system is implemented through a parametric, domain-specific programming workflow and evaluated across 100 Hz to 100 kHz to assess the accuracy of the predicted electrical values. Printed results were validated by empirical studies involving an RLC meter and by comparing both theoretical and actual values. Results proved consistent software reliability independent of material resistivity variation across samples. This framework aims to establish a practical approach for local fabrication of reliable printed circuits, positioning multi-material FDM printing as a viable candidate for on-demand circuit printing.

Table of Contents

List of Tables

List of Figures

List of Equations

Chapter 1: Introduction

1.1 Background

3D printing, a branch of additive manufacturing, has changed the way engineers work by enabling rapid prototyping, cost-effective production, and on-demand manufacturing. Fused Deposition Modeling (FDM) printing has been applied in electronics to fabricate passive electronic components such as resistors, inductors, capacitors, and printed circuit boards. However, creating functional and effective electronics has remained a challenge given the limitations in materials, design complexity, and fabrications processes.

The goal of this research is to bridge the gap in 3D-printed electronics by developing an automated tool that simplifies the design and fabrication of passive electronic components while ensuring reasonable performance. By addressing the limitations of traditional computer-aided design (CAD) software and incorporating automation, this tool aims to streamline the workflow for engineers, researchers, and hobbyists interested in 3D printing electronic components.

1.2 Problem Statement

Traditional CAD software is not designed with electronics in mind, making the process of 3D printing electronic components complex and time-consuming. For example, generating a resistor requires multiple steps: calculating the resistance, sketching, extruding the geometry, adding terminals, exporting, slicing, and manually assigning the appropriate filaments. This workflow is time-consuming and inefficient.

A major challenge in printed electronics especially in FDM printing is the integration of multiple filament types, such as conductive, non-conductive, and other specialized materials. Each filament has unique properties, including melting temperature, adhesion, and shrinkage, to mention a few, which complicates the printing process. Electronics require both conductive and insulating materials, making single-material fabrication impractical. For example, a capacitor made solely from plastic would serve no functional purpose beyond decoration. To produce functional components, multi-extrusion 3D printers are essential.

Beyond material challenges, 3D-printed electronic components such as resistors, capacitors, inductors, and printed circuit boards (PCBs) face additional technical hurdles. These include maintaining precise tolerances, achieving accurate resistivity, ensuring strong interlayer bonding along the x, y, and z-axes, and controlling material overflow or underflow. Temperature fluctuations during printing further affect quality and performance. Additionally, differences in G-code generation across slicing software impact print reliability, and the lack of electronic design automation (EDA) tools tailored for 3D-printed PCBs limits integration with conventional circuit design workflows.

1.3 Research Objectives and Significance

The objective of this thesis is to create software that mostly automates the creation of 3D printed passive RLC components and inter-connections while minimizing the manual work required for the user. To achieve this, we plan on using parametric modeling of components through the aid of OpenSCAD. The impact of this tool extends to:

- Enabling engineers and researchers to prototype electronic components more efficiently.
- Reducing the learning curve for individuals new to 3D-printed electronics.
- Facilitating the adoption of multi-material 3D printing for functional electronic applications.
- Contributing to advancements in additive manufacturing for electronics.

The software serves two primary purposes. First, it is intended to be used as generator of traces, resistors, capacitors, and inductors of different values and designs. And, second, to serve as an educational tool that supports learners to understand the relationship between electrical behavior and geometry.

1.4 Research Questions and Hypothesis

This research seeks to answer the following questions:

1. Can automation reduce the complexity and time required for generating 3D-printable electronic components?
2. Is OpenSCAD an effective software for designing 3D-printed electronics?
3. How can we ensure consistent print quality despite variations in filament properties?

We hypothesize that an automated design tool will significantly reduce the effort and expertise required to create functional 3D-printed electronic components, making the process more accessible and efficient.

1.5 Scope and Limitations

The scope of this research is limited to the development of a software tool for designing passive RLC components using multi-extrusion FDM 3D printers capable of handling four different filament types. The study focuses on the automation of component design rather than optimizing their electrical performance.

Key *limitations* include:

- The software will not incorporate electrical simulations.
- Only basic passive components: resistors, capacitors, and inductors will be supported.
- Hardware constraints will be limited to a layer height of 0.2mm, a nozzle diameter of 0.4mm, all printed using mainly PLA composites.
- Users will have minimal control over non-essential parameters beyond inductance, capacitance, and resistance.

Chapter 2: Literature Review

2.1 Overview

3D printing has transitioned from a specialized industrial capability into an accessible manufacturing platform. Historically, prototyping required long lead times, no guaranteed performance, high costs, and outsourcing to companies who offer their services. The emergence of additive technologies addressed these limitations by enabling rapid prototyping along with accessible prices. 3D printing is efficient today because of many cumulative historical events including the following:

1980s: Birth of 3D Printing

- **1981:** Dr. Hideo Kodama of Nagoya Municipal Research Institute files for the first patent for a rapid prototyping system using photopolymers. Laying the groundwork for Stereolithography. [1]
- **1984:** Charles Hull invents Stereolithography. He then files a patent in 1986.[2]
- **1989:** Scott Crump invents Fused Deposition Modeling (FDM) and co-founds Stratasys, another major player in the 3D printing industry.[3]

1990s: Commercialization and early applications

- **1992:** 3D Systems releases the first SLA machine for commercial use, targeting industrial prototyping.
- **1993:** The first Selective Laser Sintering (SLS) machine is developed by Carl Deckard at the University of Texas. SSL allows for the creation of parts from powdered materials, including metals and ceramics.[4]
- **1999:** The first 3D-printed organ, a human bladder, is created using bioprinting techniques. Significant milestone in medical applications.

2000s: Democratization and Expansion

- **2005:** Dr. Adrian Bowyer launches the RepRap Project, an open-source initiative to create a self-replicating 3D printer. Significantly reduces the cost of 3D printers and democratizes access to the technology.
- **2006:** The first commercially available SLS printer is released, enabling the production of complex and durable parts, for industrial use.
- **2009:** The FDM patents expire, leading to a surge in affordable, consumer-grade 3D printers. Companies like MakerBot emerge, making 3D printing accessible to hobbyists and small businesses.

2010s: Mainstream Adoption and Innovation

- **2012:** The first 3D-printed prosthetic limb is created, showcasing the potential of 3D printing in healthcare.
- **2014:** Carbon3D introduces Continuous Liquid Interface Production (CLIP), a breakthrough technology that speeds up the 3D printing process by up to 100 times.
- **2019:** The world's first 3D-printed neighborhood is built in Mexico, highlighting the potential of 3D printing in construction.

2020s: Advanced Applications and Future Trends

- **2020:** The COVID-19 pandemic accelerates the adoption of 3D printing for producing PPE, face shields, ventilator parts and medical equipment.
- **2022:** Advancements in multi-material and multi-color 3D printing enable more complex and functional designs.
- **2025 (Projected):** The global 3D printing market is expected to exceed $50 billion, driven by advancements in materials, speed, and scalability.

The historical progression of 3D printing shows that it evolved from a mechanical prototyping tool into a widely accessible manufacturing tool. Beyond its original scope, multi-material manufacturing introduced a paradigm shift regarding its traditional functionality. Recent advances in materials, specifically in composite polymers, show that permittivity, permeability, and conductivity can be strongly present in a polymer; then by manipulating its geometry, material placement, and other physical aspects one can manipulate the behavior of the object.

2.2 Theoretical Framework

2.2.1 Electromagnetic Foundations

Here are the four time-domain Maxwell equations that govern all electromagnetic interactions [5] :

$$
\begin{aligned}
\nabla \cdot \mathbf{E} &= \frac{\rho}{\epsilon} \\
\nabla \cdot \mathbf{B} &= 0 \\
\nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\
\nabla \times \mathbf{B} &= \mu \mathbf{J} + \mu\varepsilon \frac{\partial \mathbf{E}}{\partial t}
\end{aligned}
\tag{1}
$$

The electromagnetic response of a 3D-printed passive component can be fully described by Maxwell's equations, which couple both electrical and magnetic fields within the medium. These relationships determine how charge distributions, current densities, and other time varying fields change the reactive behavior of the components. In a conductive polymer, or better said, a composite polymer, the microscopic interaction within the charge carriers and the polymer matrix results in the transformation of electrical energy to heat. This phenomenon is expressed through the relation [6]:

$$\frac{J}{E} = \sigma \tag{2}$$

Where J is the current density, σ is the effective electrical conductivity, and E is the local electric field. Integrating this relationship into the following equation yields the macroscopic resistance of the printed object:

$$R = \frac{L}{\sigma * A} \tag{3}$$

The reactive component arises from Maxwell's curl and divergence laws[7], while the resistive component stems from the Ohmic term that introduces real power loss into the Poynting energy balance. Together they define the complex impedance of a printed object, and all conductors:

$$Z(\omega) = R + j\omega L + \frac{1}{j\omega C} \tag{4}$$

The formula above allows electrical characterization and of 3D-printed passive elements that arise from the material parameters[8]: conductivity, non-conductivity, permeability, permittivity and from geometrical design.

2.2.2 Lumped Vs Distributed Models

Multiple modeling approaches exist for quantifying electrical behavior. Lumped modeling assumes that resistance and reactance are concentrated at discrete locations, providing a simplified but useful way to analyze the system. Distributed modeling treats the system as a continuous arrangement of infinitesimal passive elements, allowing a higher fidelity analysis. In other words, lumped modeling and distributed modeling are not the same [9]:

- Distributed: Infinite amount of resistors, capacitors, and inductors distributed in the wire.
- Lumped: Resistance, Capacitance, and Inductance concentrated in a point.

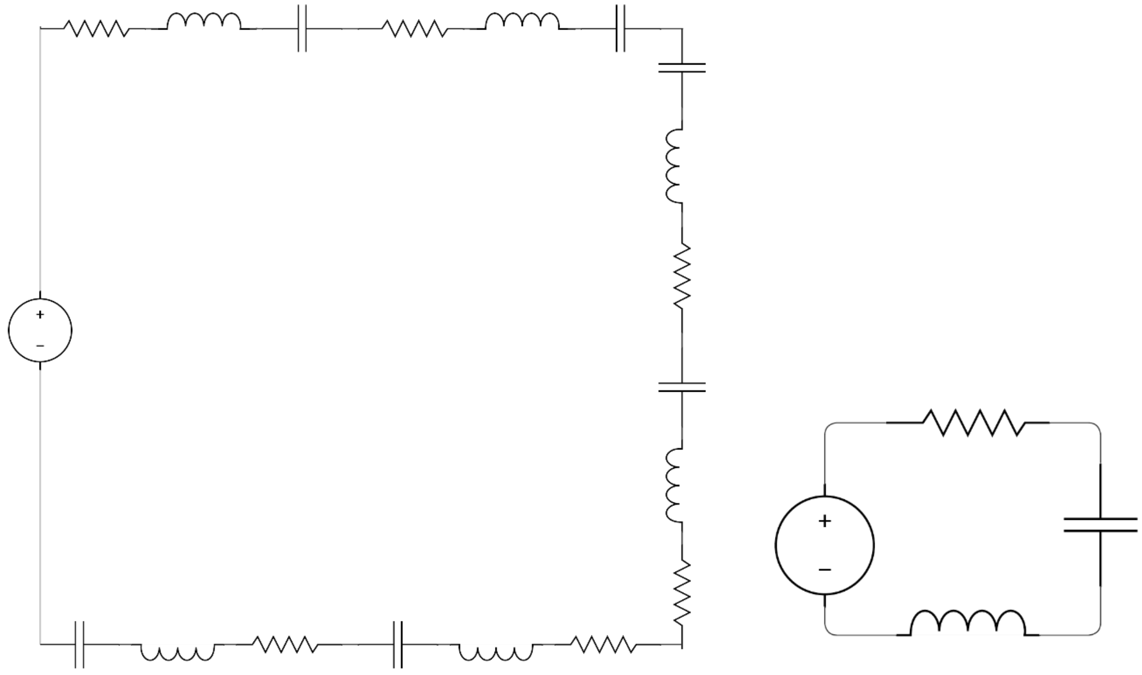


Figure 1. Distributed vs Lumped

Let's understand why a distributed equation is a lumped equation assuming same cross-sectional area, same material, and uniform geometry.

Lumped equation for resistance:

$$\mathrm{R} = \frac{(\rho * L)}{A} \tag{5}$$

Distributed equation under constants:

$$\Omega = \int_0^L \frac{p(x)}{A(x)} dx \tag{6}$$

$$\Omega = \int_0^L \frac{p}{A} dx$$

$$\Omega = \frac{p}{A} \int_0^L dx$$

$$\Omega = \frac{p}{A} \; [\,x]_0^L$$

$$\Omega = \frac{p * L}{A}$$

Therefore, resulting in the same equation as the lumped equation. Capacitance behaves the same way under constants:

$$C = \int \frac{\epsilon(x) * A(x)}{d(x)} dx = \frac{\epsilon * A}{d} \tag{7}$$

Inductance is different from the previous passives because it depends on local conductor geometry and global magnetic field distribution in space[7]. Assuming geometry is uniform, and the field is fully confined:

$$L = \int_{0}^{Length} \frac{u(x)}{g(x)} dx \approx \frac{u_o * u_r * N^2 * A}{l} \tag{8}$$

2.2.3 Material Science Theories

Conductive Polymers:

Conductive polymers and their composites such as carbon nanotubes-filled composite PLA, copper-filled PLA, and other carbon-based materials exhibit intrinsic or composite induced electronic conductivity. Unlike conventional thermoplastics, which have insulating properties, these functional composites form partial percolation networks inside the non-conductive material to form a conductive path within it. Their effective conductivity can be adjusted via key processing, slicer parameters, and material characteristics.

Filler content and distribution: Homogeneous and high amount of filler allow the material to reach a higher conductivity by creating more conductive paths (percolation) [10].

$$\sigma \propto (\phi - \phi_c)^t$$

where:

$\sigma: Effective\ conductivity$

$\phi: Volume\ fraction\ of\ conductive\ filler$

$\phi_C: Percholation\ threshold$

$t: Critical\ exponent$

In an ideal material, such is in the case of copper, the equation would look like:

$$\sigma \propto (1 - 0)^1$$

but in the case of composite PLA:

$$\sigma \propto (.05 - .02)^2$$

Thus, resulting in a smaller conductive network.

Dielectric Polymers

Dielectric polymers serve as the electrical insulation and energy storage medium in capacitors [11]. Their polarization determines the power loss, permittivity, and breakdown strength. The main polarization mechanisms are electronic, ion, and dipolar which govern overall dielectric behavior. Stored energy can be defined as:

$$U = \frac{1}{2} * \epsilon_0 * \epsilon_r * E_b^2$$

where:

$\epsilon_0 : Vacuum\ permittivity$

$\epsilon_r : Relative\ permittivity$

$E_b : Breakdown\ electric\ field\ strength$

Theoretically and in practice, it is more critical to take care of E_b given that small structural defects cause stored energy to drop drastically. Optimizing other variables are second priority once E_b is controlled.

Ferromagnetic Polymers

Ferromagnetic Polymers and polymer magnetic composites are materials that combine functionality and ease of printing into a product. Their integration into 3D printing enables new device architectures to take place: tunable inductors, electromagnetic absorbers, and magnetically responsive actuators. Pure polymers are intrinsically non-magnetic; ferromagnetic behavior arises when it is mixed with ferromagnetic particles such as: FE3O4, Fe2O3, NiCo, or other compounds are dispersed into the polymer matrix. The effective permeability equation is the following:

$$\mu_{eff} = \frac{\mu_m * (1 + 2\phi * (\left(\frac{\mu_f}{\mu_m}\right) - 1)}{(1 - \phi * (\left(\frac{\mu_f}{\mu_m}\right) - 1)}$$

where:

μ_{eff}: Effective magnetic permeability of the composite

μ_m: Permeability of the polymer matrix

μ_f: Permeability of the ferromagnetic filler

ϕ: Filler volume fraction

In principle, one would use ferromagnetic material as the filament itself, though this material does not provide ease of printing. Many manufacturers cannot exceed 30% ferromagnetic material infill volume given that ease of printing disappears with a higher infill. Therefore, many manufactures mix both carefully while offering good printability and functionality. PLA, PETG, or ABS are frequently used as the base polymer matrix to provide structural stability. PLA is frequently chosen given its ease of printability, which is the most important factor for many functional objects.

2.2.4 Domain Specific Programming

The computational Geometry Algorithm Library provides the mathematical foundations for reliable geometric modeling with OpenSCAD. OpenSCAD is the environment used to test and design the geometry, in this case passives, which allows prototyping to take place before printing. CGAL is an open-source library written in C++. It implements various robust algorithms such as, but not limited to, polygon and polyhedron operations, mesh generation, Boolean set operations, and surface reconstruction. Its precise geometry generation allows one to create geometry that is stable and precise even at small scales. Other more complex algorithms exist but this specific algorithm has been implemented into OpenSCAD given its relative simplicity, purely algorithmic, and because it is open source.

Historically, earlier versions of OpenSCAD implemented Constructive Solid Geometry and Binary Space Partitioning to define complex object with traditional Boolean operators such as union, difference, and intersections. While good, these algorithms had flaws. They implemented polygon clipping, plane-splitting and other functionality that led to unintentional geometry. The integration of CGAL replaced all these algorithms with more robust solutions. The implementation of Nef polyhedral representations allowed exact mathematical Boolean operations that ensured watertight, non-self-intersecting solids [12]. CGAL role extends beyond geometric generation, it provides the computational guarantee that the geometry generated is defined without any numerical inconsistencies. This guarantees that a printed passive, when exported as STL or 3MF for example, will conserve its volumetric integrity therefore guaranteeing electrical performance at least geometrically.

2.3 Challenges

While the creation of additive manufactured 3D passives is something that has already been done, there are still many challenges that need to be resolved.

Material challenges

One can use either plastic-based materials or conductive inks to create conductive paths. Both have their issues, especially PLA among the z axis as suggested in [13]. In the case of both, it is not uncommon to observe non-homogenous behavior in the material itself. Insulating materials, known as dielectrics, have low permittivities and low breakdown voltages. Thermal stability must be considered for this type of applications, given that thermal activity within the medium might alter its behavior. Material agglomeration might become an issue.

Geometric Resolution

Given the default hardware resolution nozzle of .4mm or 400 um, creating miniature structures is not feasible. Even then, the structures created exhibit small gaps between them that make them act as porous materials[14] which is not ideal for circuits. Layer height is determined by the nozzle, thus introducing different resolutions in xy-axis and the z-axis. For example, assuming a .4mm nozzle (xy-axis), a layer height could be of .2mm (z-axis) therefore there is a higher resolution in the z-axis. Normally the layer height is smaller than the nozzle diameter.

Process integration and automation

This is not the case with all the materials, but some materials require postprocessing once they have been deposited[15], such as UV treatment, therefore making them incompatible with FDM printing. When it comes to automation, providing continuous deposition requires three things: that the material itself is of high quality, that the software is robust, and that the hardware is good enough to provide the material required at the expected speed, temperature, volumetric flow, etc.

Contamination

When switching materials in the toolhead, it is important that a modest amount of material is purged. Otherwise, cross-contamination might occur when printing[16]. Other ways contamination might occur is by over-extruding then creating shorts within the object.

Electrical characterization issues

Contact resistance is not ideal when it comes to 3D printed circuits[17], the rough and porous surfaces generated by these methodologies provide poor contact area therefore poor characterization assuming a poor setup.

Reliability and repeatability

GCODE could result in different behavior assuming the environment of the printed passive is unfavorable. In rare occasions, a print could result in catastrophic failure where the printer continues to print without stopping. AI is being considered, as discussed by Chen et al [18].

Chapter 3: Methodology

3.1 Overview

This section introduces the back-end, the logical aspect, of the program. Other workflows that are for aesthetic, visual, and/or not related to logic are briefly introduced given these are not critical workflows for the program. Ubuntu, a Linux kernel OS, served as the ecosystem where work was implemented. We prioritized what is open source. Therefore, other alternatives such as Windows or MacOS were discarded for development purposes. The flowcharts presented show the intention of the workflow. To understand granularly what the workflow intends to do, one must look at the source code, which is likely available upon request.

All workflows were designed, in principle, to be modular, favoring a non-monolithic architecture. Small programs are chained together, where each program has a clear purpose. The modular design is intentional: it simplifies debugging, facilitates scalability, and supports maintainability. In the event of a bug, a modular program can be isolated then debugged rather than having to modify a large, monolithic program[19]. This of course takes greater discipline, development upfront time, analysis, and structure but it ultimately offers more flexibility and resilience long term.

3.2 Programming Languages and Libraries Implemented

For general-purpose computing, we use Python. Speed is not relevant given that the program fulfills its purpose. For example, C is fast, but many lines of code are required to get functionality out of it. The opposite is true for Python. Below is an example of both programs doing the same thing.

Table 1. Python and C functionality comparison

C	Python

#include <stdio.h>
int main() {
  FILE *fp;
  fp = fopen("data.txt", "r");
  if (!fp) {
    perror("Could not open file");
    return 1;
  }
  char buffer[256];
  int line_count = 0;
  int printed = 0;
  while(fgets(buffer, sizeof(buffer), fp)) {
    line_count++;
    if (printed < 3) {
      printf("%s", buffer);
      printed++;
    }
  }
  printf("\nTotal lines: %d\n", line_count);
  fclose(fp);
  return 0;
}
	with open("data.txt") as f:
	lines = f.readlines()
	print(len(lines))
	print(lines[:3])

For Python, we imported a library called Pyside, this library intends to create a graphical user interface. Creating a GUI is not our main concern, but it allows users to easily use the program without any terminal knowledge. Creation of geometry requires a domain-specific programming language capable of automation, standard exporting formats, importing geometry, and other technical functionality[20]. OpenSCAD is a declarative programming language. That means that you can't control exactly *how* it is done but you must write exactly what you *want* with the given syntax. Tradeoffs are accepted here given that Python complements it. OpenSCAD is minimal in syntax which allows elegant code to follow. Python and bash are both interpreted and the software developed for this thesis uses them differently. Python is used for complex operations, analysis, etc. Bash is used for its portability and capability for manipulating the OS. It is native in many OS systems that incorporate the Linux kernel.

3.3 Installation Workflow

The installation workflow can be categorized as in the following:

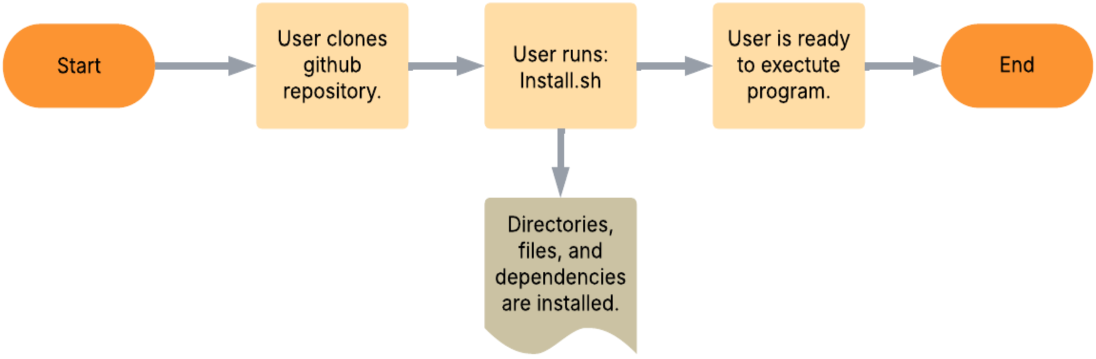

Start
User clones github repository.
User runs: Install.sh
User is ready to exectute program.
End
Directories, files, and dependencies are installed.

Figure 2. Installation workflow

The program is stored in a GitHub repository, given that it provides: version management, collaboration, code hosting, sharing, code reviews, and more. Install.sh is a bash script that requires sudo privileges to install the program within the OS. A series of things occur when installing the script: dependencies such as Python 3 and libraries Pyside and OpenSCAD are installed. After installation takes place, it is necessary to create virtual machines. The whole process is done without any verbosity. Given that the output is generated by scripts, it is likely unnecessary for the people who will be using the script. To prevent breaking the program, we stick to specific versions for the current installed dependencies.The install script is then destroyed, to guarantee that in the situation that the user decides to re-run the install.sh script, his files and other dependencies are not over-written.

3.4 Software Structure

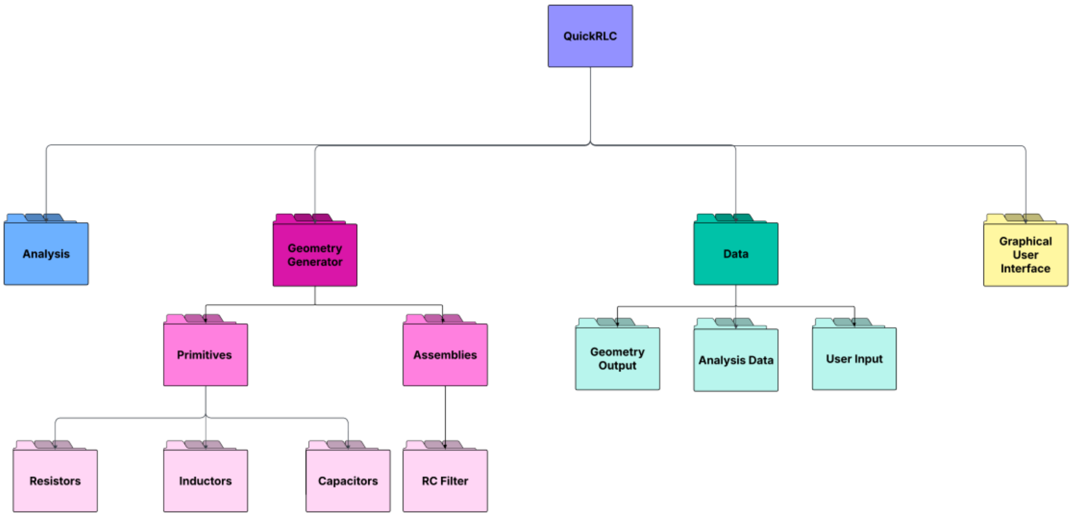


Figure 3. Tree structure of QuickRLC

The tree structure helps the code maintain a clear intention of where data and code are. This simplification from the actual software provides a clear understanding of how everything is organized. We follow the Unix like mentality on how to build our software. “Do only one thing and do it well”[21].

3.5 Geometry Analysis

Analysis of passive printed parts can be summarized in the following way:

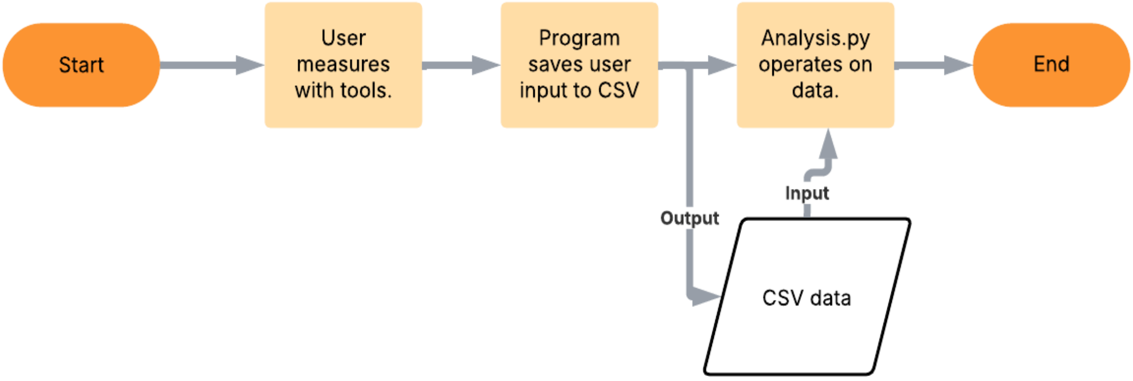

Start
User measures with tools.
Program saves user input to CSV
Analysis.py operates on data.
End
Output
Input
CSV data

Figure 4. Geometry analysis workflow.

Here we measure the physical reactive and resistance values that are present after they are 3D-printed. This is critical, in the sense that, it allows us to understand how the theoretical values differ from the actual values. First, the user must measure the components. The user could use a multimeter to measure resistance within the circuit, or other tool, but it would then be harder to analyze reactance from capacitor and inductors. So, when measuring reactance, it is recommended to use specialized instruments[22].

3.6 Geometry Creation Workflow

The creation of geometry workflow can be categorized as shown in Figure 5 below. This is the most complex part of the developed software. It is where user input is brought to life. Primitive passive components such as resistors, capacitors, and inductors are used to create more complex geometries such as RC filters. OpenSCAD doesn’t allow one to sketch then extrude for example, unlike SolidWorks or Fusion, but it allows you to describe your geometry in terms of shapes, dimensions, variables, and mathematical relationships through code.

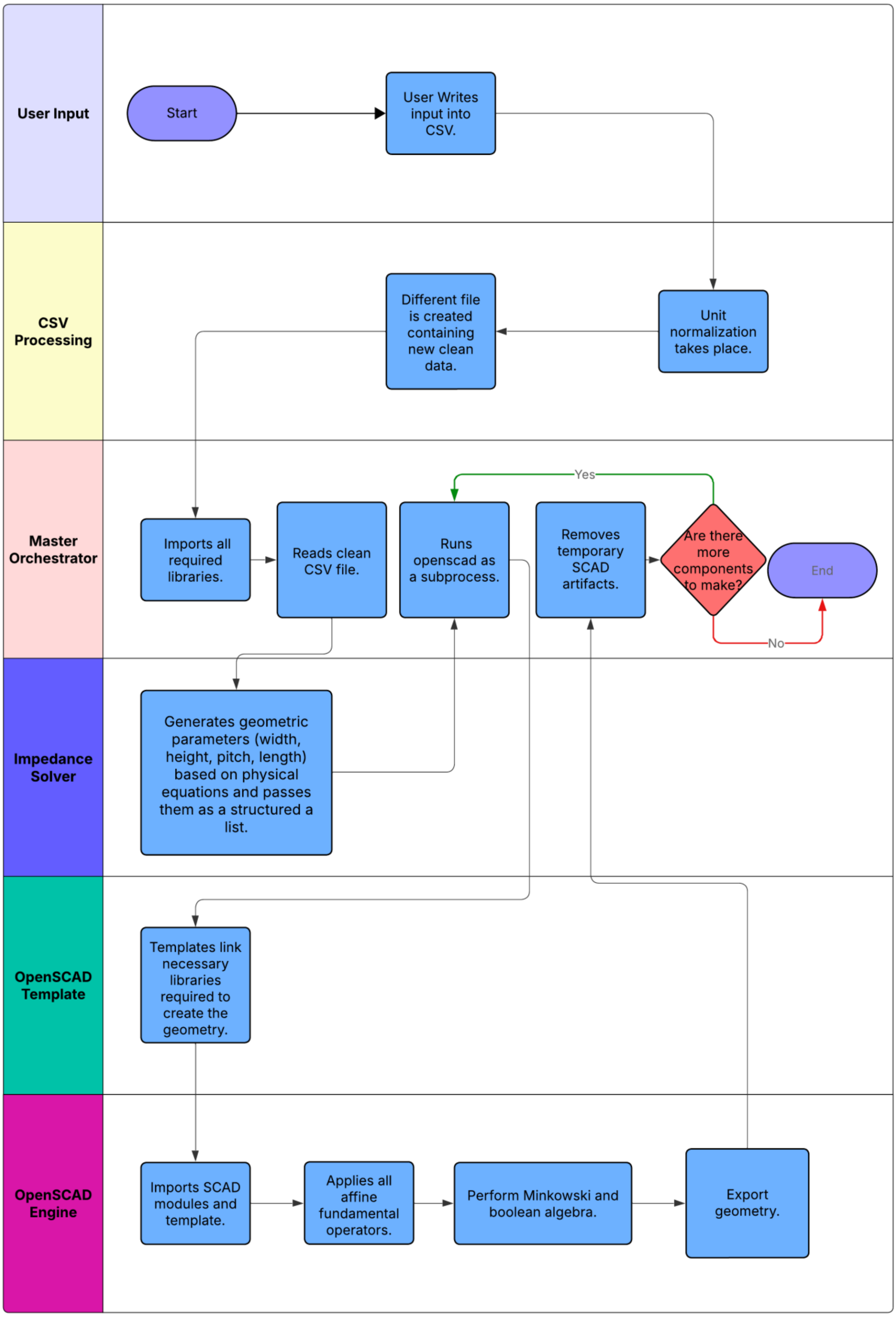

User Input
Start
User Writes input into CSV.
CSV Processing
Different file is created containing new clean data.
Unit normalization takes place.
Master Orchestrator
Yes
Imports all required libraries.
Reads clean CSV file.
Runs openscad as a subprocess.
Removes temporary SCAD artifacts.
Are there more components to make?
End
No
Impedance Solver
Generates geometric parameters (width, height, pitch, length) based on physical equations and passes them as a structured a list.
OpenSCAD Template
Templates link necessary libraries required to create the geometry.
OpenSCAD Engine
Imports SCAD modules and template.
Applies all affine fundamental operators.
Perform Minkowski and boolean algebra.
Export geometry.

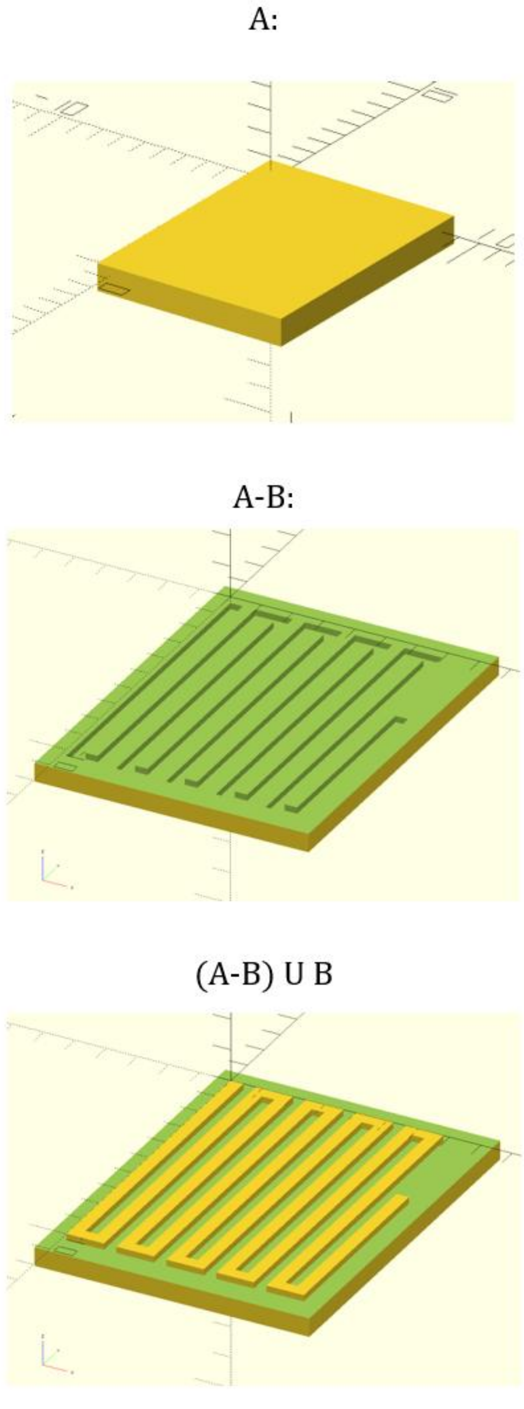
Figure 5. Geometry creation workflow.

3.7 Exporting Geometry

Even though OpenSCAD has 3MF enabled as of recent builds, colors only work as a visual guides, it doesn't export colors or materials, unfortunately. Therefore, a different solution had to be implemented to support multi-material circuits. This was implemented by exploiting small gaps between components. Basically, we apply any Boolean operation on the object that we want to export. For instance, for the expression:

$$(A\text{-}B) \cup B$$

The equation can be understood visually as shown in Figure 6. The key there is that we enlarge the item being subtracted (B) either by inflating it on all axis or using a Minkowski method. Then we place the original (B) into the gap created so that it is never touching the object. This basically makes the item "float," thus avoiding coplanar issues that CGAL geometry generators do not tolerate. Once exported, most modern slicers will allow you to divide the object into independent parts whose materials can be adjusted successfully. This allows STL files to export in a way that color can be later implemented into the slicer.



A:

A-B:

(A-B) U B

Figure 6. Sequence of Boolean operations

3.8 Software Validation and Verification

Before testing, we have to define our inputs, outputs, functional requirements, and non-functional requirements. For instance, here is a comparison of functional vs. non-functional requirements.

Functional requirements:

- The software must generate the correct physical parameters from input in the form of Ohms, Farads, and Henries.
- The software must imitate the functionality of the passive that is printed, even if it is not perfect, or is far from the wished reactance/resistance.

Non-functional requirements:

- Performance: All individual software modules must perform the required geometry in less than a minute.
- Usability: Graphical interface should facilitate the use of the program, but it is not the priority of the program. Not all command-line interface (CLI) functionality is required on the GUI.
- Reliability: All modules should work.
- Portability: It must run on Linux kernel-based OS.

- Maintainability: The program must follow a non-monolithic structure.

For software verification, we tried different techniques to verify the correct behavior of our software components. We first inspected the code, not robust but easy to implement, to see if there were any logical bugs. Then, tested the software by analyzing tests, that is, we tried inputs whose outputs are known.

3.9 Physical Validation and Verification

Once the software geometry has been created and visually inspected, it is technically the same to its physical geometry. There were no discrepancies between the software and the physical output of the 3D printer. All printed objects had less than +-.2mm tolerance, which was considered acceptable in our tests. The discrepancy between the software and results are mainly because of material warping. That is, material that is hot and then cooled happens to shrink.

3.10 Workflow Example

The following examples illustrate core behavior implemented by the software. Only essential steps are introduced for clarity; auxiliary helper functions and non-critical routines are skipped for compactness.

Appendix A — Graphical User Interface Overview

Appendix B — Impedance Solver

Chapter 4: Testing and Results

This chapter presents the experimental results of the software-generated geometries: resistors, capacitors, inductors and compound prints such as RC filters. The results were obtained under a controlled environment and repeatable printing conditions using the same conductive filament.

Setup:

Table 2. Control setup

Control	Description
Filament	Amolen Conductive PLA, 1.27 ohm/cm
Infill	100%
Infill-Type	Rectilinear
Line width	0.42mm
Fan speed	60%
Auxiliary fan speed	20%
Nozzle temperature	250C
Bed temperature	50C
Room temperature	29C
Volumetric flow	$7mm^3$/s
Nozzle speed	70mm/s

We froze all slicer parameters to ensure that all printed objects have the same configuration. We used Amolen given it was the only conductive filament available that was not expensive that had a known resistivity.

4.1 Individual Components

4.1.1 Resistor

Single Axis Resistor

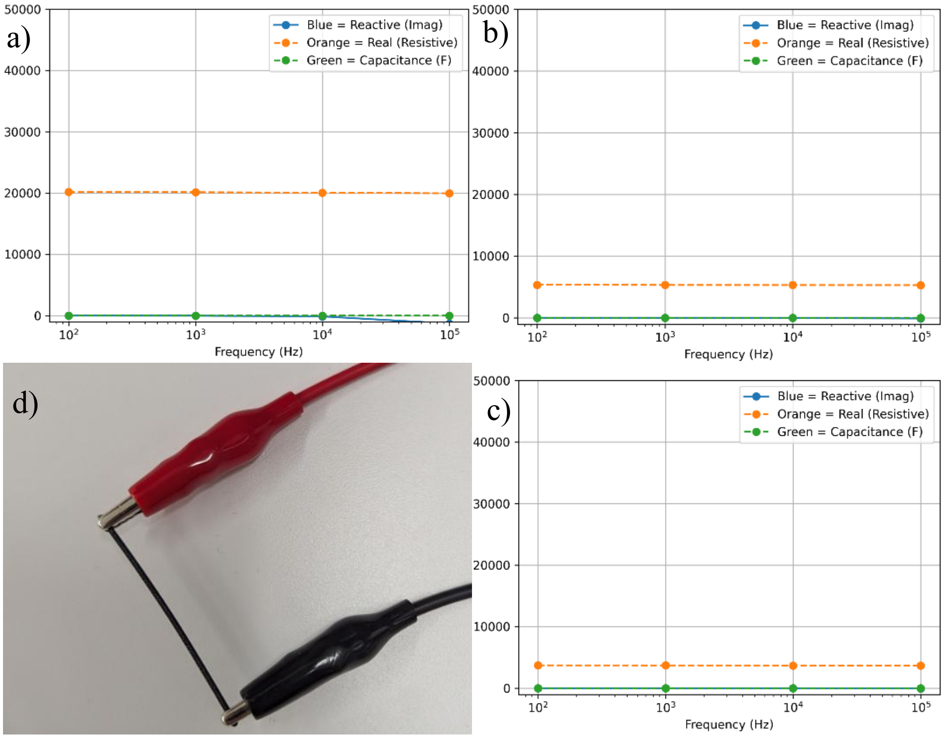


Figure 7. Single Axis resistor characterization

a) 1x1x50mm^3 b) 2x2x50mm^3 c) 3x3x50mm^3 d) Linear resistor

Statistical Frequency Results

Table 3. Single Axis Resistor Frequency Statistical Results

Size	R Mean (Ω)	R Std (Ω)	C Mean (F)	C Std (F)	ρ Mean	N
1×1×50	2.008×10^{4}	1.038×10^{2}	2.018×10^{-4}	3.960×10^{-4}	401.6	4
2×2×50	5.318×10^{3}	2.220×10^{1}	1.060×10^{-3}	2.015×10^{-3}	425	4
3×3×50	3.700×10^{3}	1.826×10^{1}	5.054×10^{-3}	9.895×10^{-3}	667	4

Serpentine Resistor

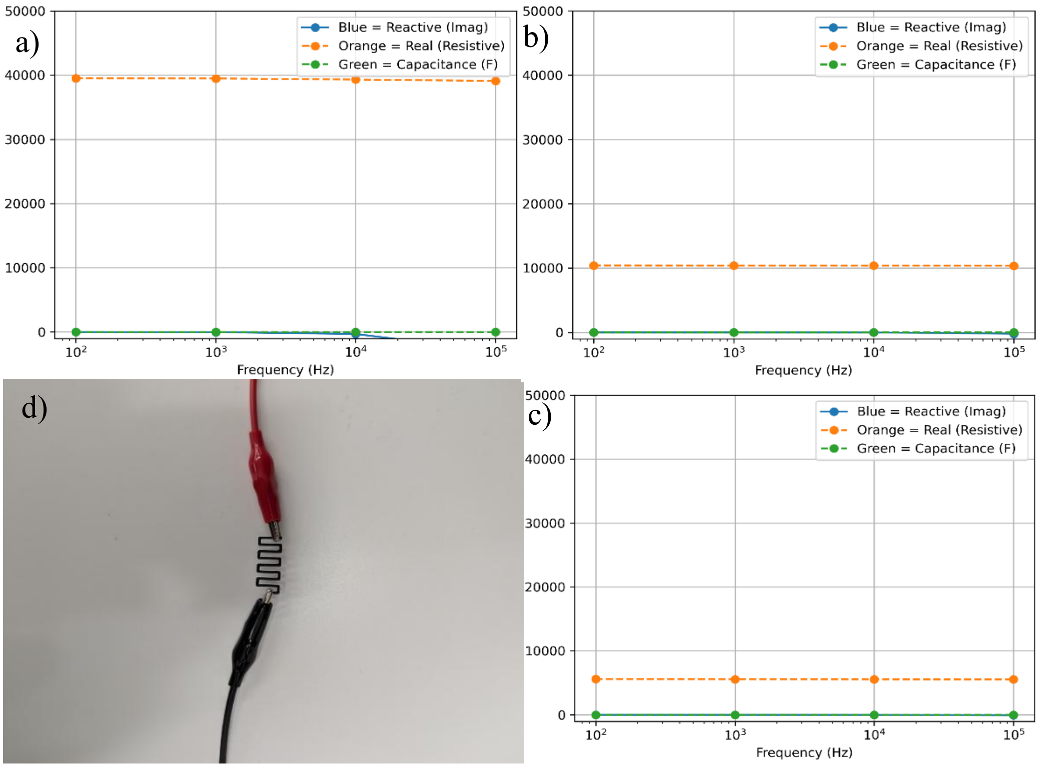


Figure 8. Serpentine resistor characterization

a) 1x1x100mm^3 b) 2x2x100mm^3 c) 3x3x100mm^3
d) Serpentine/Zigzag resistor

Statistical Frequency Results

Table 4. Serpentine Resistor Statistical Frequency Results

Size	R Mean (Ω)	R Std (Ω)	C Mean (F)	C Std (F)	ρ Mean ($\Omega * \mathrm{mm}$)	N
1×1×100	3.934×10^{4}	1.841×10^{2}	8.080×10^{-5}	1.584×10^{-4}	393	4
2×2×100	1.038×10^{4}	1.758×10^{1}	8.119×10^{-4}	1.581×10^{-3}	415	4
3×3×100	5.572×10^{3}	1.708×10^{1}	2.040×10^{-3}	3.946×10^{-3}	505	4

4.1.2 Inductor

Planar Inductor

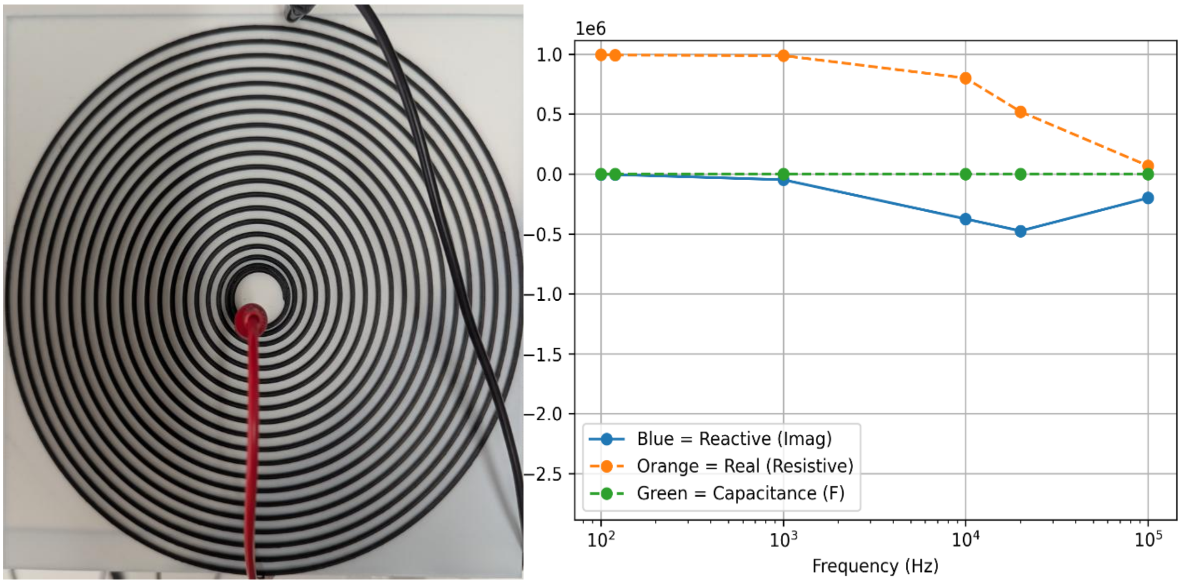


Figure 9. Inductor characterization

Statistical Frequency Results

Table 5. Planar Inductor Statistical Frequency Results

Type	R Mean (Ω)	R Std (Ω)	C Mean (F)	C Std (F)	N
L	7.271×10^{5}	3.721×10^{5}	1.331×10^{-7}	2.215×10^{-7}	6

4.1.3 Capacitor

Parallel Plate Capacitor

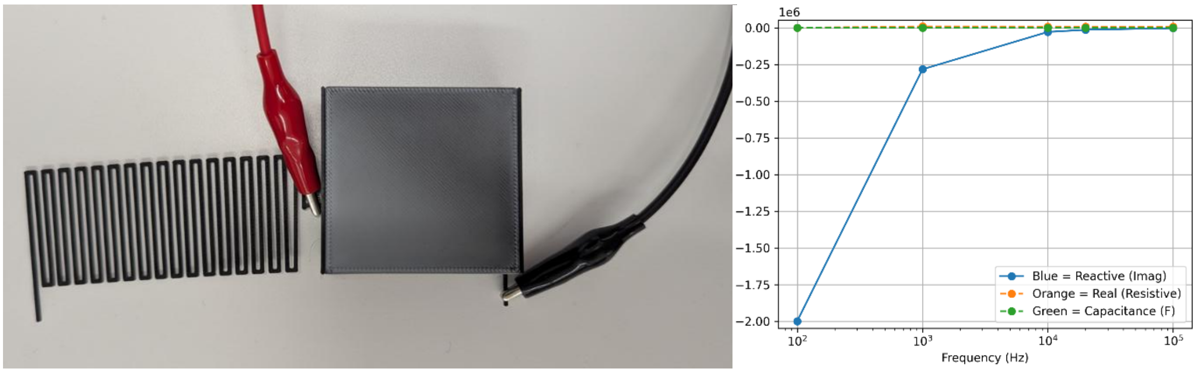


Figure 10. Capacitor characterization

Statistical Frequency Results

Table 6. Interdigitated Capacitor Statistical Frequency Results

Type	**R Mean (Ω)**	**R Std (Ω)**	**C Mean (F)**	**C Std (F)**	**N**
RC	**7.620×10^{3}**	**7.155×10^{2}**	**6.031×10^{-10}**	**1.087×10^{-10}**	5

4.2 Compound Components

4.2.1 RC Filter

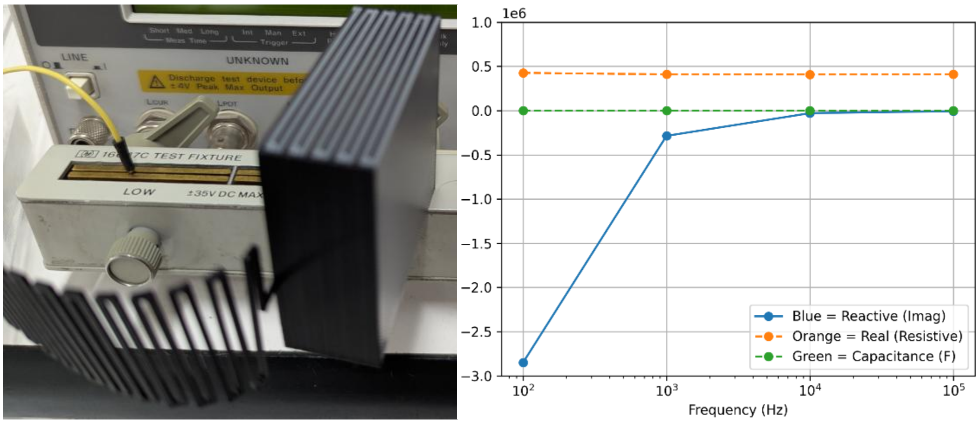


Figure 11. RC Filter characterization

Statistical Frequency Results

Table 7. RC Filter Statistical Frequency Results

Type	**R Mean (Ω)**	**R Std (Ω)**	**C Mean (F)**	**C Std (F)**	**N**
RC	4.141×10^{5}	8.290×10^{3}	$.473\times10^{-9}$	$.162\times10^{-9}$	4

Chapter 5: Discussion and Interpretation

The purpose of these tests is to determine component behavior across a broad range of frequencies, characterize their impedances, and to statistically find any correlations. The capacitor used in the RC filter is the same as the component in the individual section. All tested components had a negative reactive value, therefore indicating capacitive behavior.

5.1 Individual Components

5.1.1 Resistor

Recalling the formula used to obtain resistance from a resistor (5).

Let's take our current 1x1x50 mm resistor and see its theoretical value. Amolen datasheet states a resistivity of 1.42 $\Omega \cdot cm$ equivalent to 14.2$\Omega \cdot mm$. Plugging all values into the equation:

$$\frac{(14.2\Omega mm \cdot 50mm)}{(1mm \cdot 1mm)} = 710\ \Omega$$

Our measured resistance is 20kohms which is much higher than 710 ohms, let's find resistivity.

$$\frac{(\rho \cdot 50mm)}{(1mm \cdot 1mm)} = 20\text{k}\Omega$$

$$\rho = 400\Omega \cdot mm$$

Comparing the datasheet resistivity with our measured resistivity, there is a considerable difference between results. This corresponds to approximately 28 times more resistance than the stated datasheet value. Many factors come into play when it comes to resistivity, such as humidity, temperature, z-layer bonding, xy-layer bonding, speed, extrusion flow, geometry of infill used, and many others. Other postprocessing factors such as temperature annealing, chemical applications, electro-plating, just to mention a few that lower resistivity[23]. The discrepancy is because manufacturers test their products under ideal conditions and with different methodologies that lead to large deviations when applied outside controlled lab conditions. Interestingly, we got the same resistivity for most resistors, meaning we obtained stable resistances if the slicer implemented the same route. Results strongly suggest that having a consistent workflow, same building path, and consistent slicer parameters result in small resistivity changes. The 1x1x100 zigzag resistor resulted in 40kohm and the 1x1x50 linear resistor gave 20kohms, resistance essentially doubled given we doubled the length of our linear resistor, first zigzag resistor is 1x1x100mm which is the same as saying 1x1x(2x50)mm. These results are proof of correct program output and stability of the system itself.

Statistical Frequency Analysis

For zigzag geometries, the 2×2×100 mm^3 and 3×3×100 mm^3 resistors exhibited the lowest dispersion, with standard deviations around 17Ω corresponding to less than 0.3 % of their mean resistance. Linear geometries followed the same trend, showing deviations below 0.6 %. The 1×1 mm^2 cross-section in both families showed the highest variation, likely due to increased current density and contact sensitivity. Overall, the 17Ω standard deviation observed in the larger geometries shows strong geometrical stability and uniform electrical behavior. Thus, confirming that the fabricated resistors are governed by bulk geometry rather than capacitive or interfacial effects.

5.1.2 Inductor

Reality vs. Theoretical Values

where D_{in} and D_{out} are:

$$D_{in} = 20mm\, , D_{out} = D_{in} + \left(2 * Pitch * (Turns - 1)\right), D_{out} = 210mm$$

where D_{avg} is:

$$D_{avg} = \frac{(D_{in}+D_{out})}{2}\, , D_{avg} = \frac{(20+210)}{2} = 115\text{mm},$$

where A_{eff} is:

$$A_{eff} = \pi * \left(\frac{D_{avg}}{2}\right)^2 = 10386.89\ mm$$

Recalling our equation for inductance for this specific model:

$$L = \frac{u_o * u_r * N^2 * A_{eff}}{k * D_{avg}}$$

Placing the values in this inductor:

$$L = \frac{(4\pi * 10^{-7}) * 2.5 * 19^2 * .010386m^2}{2.71 * .115m}$$

L = 0.000037795340444

L = 37.79 uH

The inductor presented high levels of resistance, therefore measuring diminutive inductance is technically impossible given that resistance makes current flow negligible; one must have enough current flowing through the coil to measure a magnetic field. One can observe reactance drops at 10khz and rises again at 100khz. Resistance drops as frequency increases; this seems to be related to a measurement error given resistance cannot drop due to frequency changes. Other observation is that the inductor itself is capacitive throughout the tests, given the geometry of the inductor that introduces capacitive behavior along its trajectory.

Statistical Frequency Analysis

As we can observe, results yielded in an inconclusive manner.

5.1.3 Capacitor

Reality vs. Theoretical Values

Capacitor Ideal capacitance according to its geometry:

$$\mathrm{C} = \frac{\epsilon_{abs} \cdot Area \cdot (N-1)}{d}$$

$$\mathrm{C} = \frac{\left(8.85*10^{-12}\right)*3.0*\left(2.5 \cdot 10^{-3} m \cdot (10-1)\right)}{1 \cdot 10^{-3} m}$$

C = .597375*10^-9

Measured capacitance at 1khz:

$$C = \frac{1}{2\pi f|Xc|}$$

$$C = \frac{1}{2\pi * (1000) * | - 283100|}$$

C = .562nF

Statistical Frequency Analysis

The capacitor showed close to ideal capacitance, even in a broad range of frequencies resulting in a spread value of 18%. Resistance had a spread value of 9.3%, this indicates an anomaly given the previous results from purely resistive passives such as resistors.

5.2 Compound Components

5.2.1 RC Filter

Reality vs. Theoretical Values

For simplicity, the real part found in a filter is due to the zigzag resistor and the capacitor parasitic resistance. The reactive part is from the resistor reactive parasitics and the capacitor's intended reactive. Total impedance is the summation of the capacitor and resistor impedance as:

$$Z_{capacitor} = R_{capacitor} + jx_{capacitor}$$

$$Z_{zigzag} = R_{zigzag} + jx_{zigzag}$$

$$Z_{rc} = Z_{capacitor} + Z_{zigzag}$$

The capacitor itself presents a much lower real value compared to the resistor and higher reactive value. The resistor presents a higher real value compared and a small reactive value. The summation of their resistances gives their total impedance.

Finding R_{zigzag} empirical resistance, using known resistances from previous empirical results:

$$R_{zigzag} = R_{rcfilter} - R_{capacitor}$$

$$R_{zigzag} = 414.1 * 10^3 - 7.623 * 10^3$$

$$R_{zigzag} = 406.4 * 10^3$$

Finding resistivity for resistance R_{zigzag}:

$$\rho = \frac{R_{zigzag} * A}{l}$$

$$\rho = \frac{(406.4 * 10^3) * 1.5^2 mm}{1200mm}$$

$$\rho = 762.15\Omega * mm$$

Resistivity is much higher than previous resistive values. This was caused by our unintentional bending of the zigzag resistor; we connected the resistor the best way possible and that meant bending it in the moment. It seems conductive geometry is sensible to any physical manipulation. Resistivity on the capacitor should be like in previous resistors given it didn't experiment any force.

Finding resistance for the vertical plate:

$$R_{bus} = \rho * \frac{L_{bus}}{A}$$

$$R_{bus} = \rho * \frac{20mm}{25mm * 1mm}$$

$$R_{bus} = .8\rho$$

Finding resistance for horizontal plates, where L_plate is half the distance of the plate area:

$$R_{plate} = \rho * \frac{L_{plate}}{A}$$

$$R_{plate} = \rho * \frac{25mm}{50mm * 1mm}$$

$$R_{bus} = .5\rho$$

Capacitor resistance is equal to:

$$R_{side} = R_{plate} + R_{bus}$$

$$R_{side} = 1.3\rho$$

$$R_{capacitor} = \frac{2 * R_{side}}{5}$$

$$R_{capacitor} = \frac{2*1.3\rho}{5}$$

$$R_{capacitor} = \frac{2.6\rho}{5} * \left(\frac{10}{10}\right) = \frac{26\rho}{50}$$

$$R_{capacitor} = \frac{26\rho}{50}$$

$$\rho_{capacitor} = 7.623 * 10^3 * \frac{50}{26}$$

$$\rho_{capacitor} = 14.66 * 10^3$$

As observed, the resistivity found in this scenario is much higher than the resistivity present in the resistors. Linear resistors and zigzag resistors present behave predominantly as one-dimensional conductors, but capacitors structures are not 1D objects but 2D objects. In other words, lumped equations can't be used. Capacitor plates are wide, planar conductors in which current spreads laterally from the bus into the plate area. This behavior is fundamentally distributed, where current density varies across the plate. A more suitable expression given the current circumstances would be the following:

$$\rho_{effective} = \rho_{capacitor} * F(geometry)$$

Not a rule, but a general rule of thumb[24]:

1D (resistors, single axis)

$$F(geometry) \approx 1$$

2D (Capacitor plates)

$$F(geometry) \sim 1 - 100$$

3D (Fully volumetric)

$$F(geometry)\ \ depends\ on\ boundary\ conditions$$

Even though we can use lumped equations to derive capacitance with a relatively good precision on capacitors, we cannot implement lumped equations to find its resistance given its geometry is not uniform, especially for interdigitated geometry[25].

Statistical Frequency Analysis

Capacitance had a moderate spread among the tested frequencies, relative spread was of 34.2%. Resistance had a spread of 2% across the whole filter.

Chapter 6: Conclusions and Future Work

FDM printing is a serious candidate regarding 3D-printed circuits. Though not at par to conventional technology, it is probably one of the few technologies that can implement local, economic, and on-demand circuits that are just “good enough” for most prototypes requiring proof of concept. Results strongly indicate repeatable, consistent behavior on simple geometry. Even though circuits imitate what the software aims to do, they require more work and understanding. This work suggests that FDM should be considered as a serious candidate when it comes to building functional, reliable, and on demand circuits.

There is much incredible pending research.

Tool Related

- Building RF antennas
- Internal cavities for cooling and heat sinks given plastic cannot sustain high temperatures.
- Black Box systems: Creating systems that are extremely hard to reverse engineer.
- Creation of transistors, this requires semiconductive material.
- Objects that share a dual mechanical and electrical purpose.
- Improving OpenSCAD export functionality, specifically, exporting colors from geometry to 3MF directly.

- Understanding and optimizing slicer parameters for better prints.
- Implementing a slicer into the tool.
- Finding effective post-processing methods: heating, compression, etc.

Material Related

- Creating semiconductive filament.
- Improving the conductivity of filament.
- Creating cheaper filament.

Systems Related

- System that recycles plastic that is later converted to filament at scale.

References

[1] H. Kodama, “Automatic method for fabricating a three-dimensional plastic model with photo-hardening polymer,” *Rev. Sci. Instrum.*, vol. 52, no. 11, pp. 1770–1773, Nov. 1981, doi: 10.1063/1.1136492.

[2] C. W. Hull, “Apparatus for production of three-dimensional objects by stereolithography,” US4575330A, Mar. 11, 1986 Accessed: Nov. 08, 2025. [Online]. Available: https://patents.google.com/patent/US4575330A/en

[3] S. S. Crump, “Apparatus and method for creating three-dimensional objects,” US5121329A, June 09, 1992 Accessed: Nov. 08, 2025. [Online]. Available: https://patents.google.com/patent/US5121329A/en

[4] J. J. Beaman and C. R. Deckard, “Selective laser sintering with assisted powder handling,” US4938816A, July 03, 1990 Accessed: Nov. 08, 2025. [Online]. Available: https://patents.google.com/patent/US4938816A/en

[5] D. J. Griffiths, *Introduction to electrodynamics*, 4. ed., International ed. in Always learning. Boston: Pearson, 2013.

[6] “4.2: The Ohm Law,” Physics LibreTexts. Accessed: Nov. 08, 2025. [Online]. Available: https://phys.libretexts.org/Bookshelves/Electricity_and_Magnetism/Essential_Graduate_Physics_-_Classical_Electrodynamics_(Likharev)/04%3A_DC_Currents/4.02%3A_The_Ohm_Law

[7] “Microsoft Word - EDCh 10 passives.doc.” Accessed: Nov. 20, 2025. [Online]. Available: https://www.analog.com/media/en/training-seminars/design-handbooks/Basic-Linear-Design/Chapter10.pdf

[8] “Impedance Control in PCB | PCBCart.” Accessed: Nov. 20, 2025. [Online]. Available: https://www.pcbcart.com/article/content/impedance-control-in-pcb.html?

[9] “Lumped vs. Distributed Systems.” Accessed: Nov. 08, 2025. [Online]. Available: https://ccrma.stanford.edu/~jos/NumericalInt/Lumped_vs_Distributed_Systems.html

[10] “DietrichStaufferAmmonAharony-Introductiontopercolationtheory(1994).pdf.” Accessed: Nov. 08, 2025. [Online]. Available: https://www.eng.uc.edu/~beaucag/Classes/Properties/Books/DietrichStaufferAmmonAharony-Introductiontopercolationtheory(1994).pdf

[11] Z.-M. Dang, J.-K. Yuan, J.-W. Zha, T. Zhou, S.-T. Li, and G.-H. Hu, “Fundamentals, processes and applications of high-permittivity polymer–matrix composites,” *Prog. Mater. Sci.*, vol. 57, no. 4, pp. 660–723, May 2012, doi: 10.1016/j.pmatsci.2011.08.001.

[12] M. Granados, P. Hachenberger, S. Hert, L. Kettner, K. Mehlhorn, and M. Seel, “Boolean Operations on 3D Selective Nef Complexes: Data Structure, Algorithms, and Implementation,” in *Algorithms - ESA 2003*, G. Di Battista and U. Zwick, Eds., Berlin, Heidelberg: Springer, 2003, pp. 654–666. doi: 10.1007/978-3-540-39658-1_59.

[13] A. Pentek *et al.*, “The Effect of Printing Parameters on Electrical Conductivity and Mechanical Properties of PLA and ABS Based Carbon Composites in Additive Manufacturing of Upper Limb Prosthetics,” *Crystals*, vol. 10, no. 5, p. 398, May 2020, doi: 10.3390/cryst10050398.

[14] E. G. Gordeev, A. S. Galushko, and V. P. Ananikov, “Improvement of quality of 3D printed objects by elimination of microscopic structural defects in fused deposition modeling,” *PLOS ONE*, vol. 13, no. 6, p. e0198370, June 2018, doi: 10.1371/journal.pone.0198370.

[15] R. Syed and P. A. S., “Optimization parameters effects on electrical conductivity of 3D printed circuits fabricated by direct ink writing method using functionalized multiwalled carbon nanotubes and polyvinyl alcohol conductive ink,” *Int. J. Simul. Multidiscip. Des. Optim.*, vol. 12, p. 7, Jan. 2021, doi: 10.1051/smdo/2021007.

[16] A. Verma, A. Kapil, D. Klobčar, and A. Sharma, “A Review on Multiplicity in Multi-Material Additive Manufacturing: Process, Capability, Scale, and Structure,” *Materials*, vol. 16, no. 15, p. 5246, July 2023, doi: 10.3390/ma16155246.

[17] I. Verlangieri, T. G. de Oliveira, F. S. Lopes, I. G. R. Gutz, L. Angnes, and C. L. do Lago, “Handling electric connections in 3D-printed electrodes and sensors,” *Mikrochim. Acta*, vol. 192, no. 4, p. 265, Mar. 2025, doi: 10.1007/s00604-025-07122-z.

[18] J. Chen, Y. Yuan, Q. Wang, H. Wang, and R. C. Advincula, “Bridging Additive Manufacturing and Electronics Printing in the Age of AI,” *Nanomaterials*, vol. 15, no. 11, p. 843, Jan. 2025, doi: 10.3390/nano15110843.

[19] R. Su and X. Li, “Modular Monolith: Is This the Trend in Software Architecture?,” Jan. 22, 2024, *arXiv*: arXiv:2401.11867. doi: 10.48550/arXiv.2401.11867.

[20] “The Computational Geometry Algorithms Library.” Accessed: Nov. 20, 2025. [Online]. Available: https://www.cgal.org/

[21] “Unix philosophy,” *Wikipedia*. May 23, 2025. Accessed: Nov. 20, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Unix_philosophy&oldid=1291775239

[22] Keysight, “Power Reference Measurement with N432A Thermistor Power Meter,” Keysight. Accessed: Nov. 20, 2025. [Online]. Available: https://www.keysight.com/us/en/assets/7018-03130/application-notes/5990-9078.pdf

[23] K. Dembek, B. Podsiadły, and M. Słoma, “Influence of Process Parameters on the Resistivity of 3D Printed Electrically Conductive Structures,” *Micromachines*, vol. 13, no. 8, p. 1203, Aug. 2022, doi: 10.3390/mi13081203.

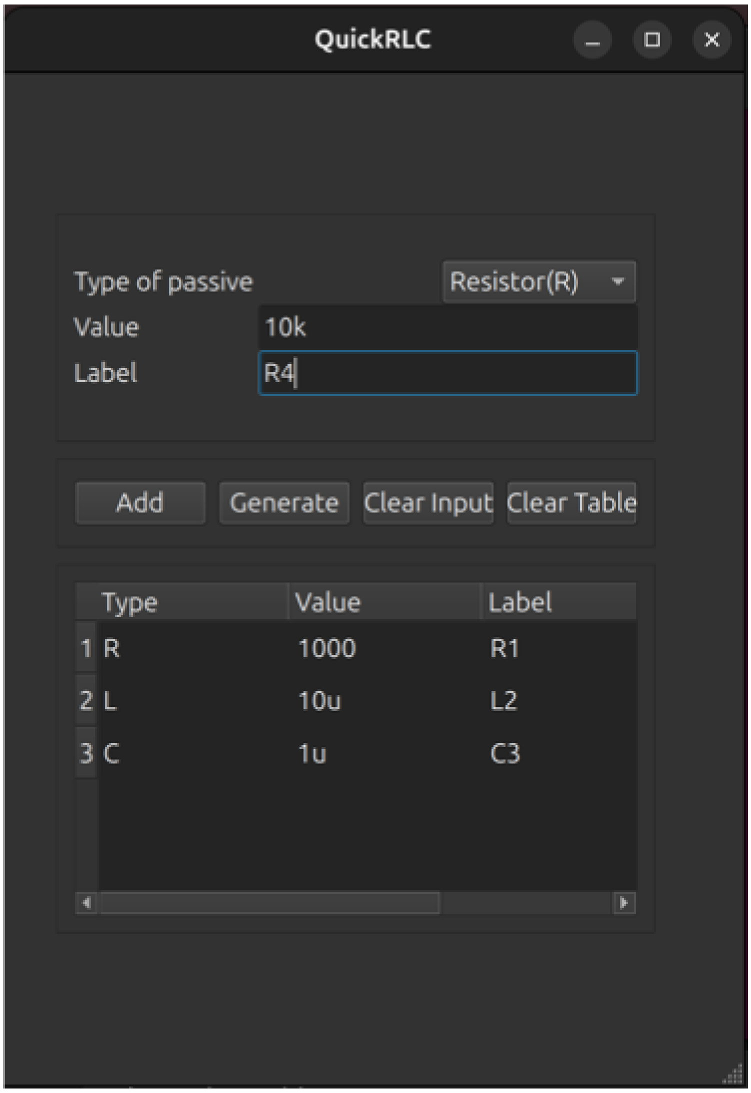
[24] C. Terquem, “University of Oxford Second Year, Part A2”.

[25] “The Feynman Lectures on Physics Vol. II Ch. 22: AC Circuits.” Accessed: Nov. 20, 2025. [Online]. Available: https://www.feynmanlectures.caltech.edu/II_22.html?

Appendices

Appendix A — Graphical User Interface Overview

The graphical user interface (GUI) is the only available interface that connects all necessary components for basic use that doesn't require deep knowledge. However, it doesn't have all functionality that would be available if the user was to use the command line interface. The following is a friendly user interface that allows users to input their wished values.



Type of passive: Resistor(R)

Value: 10k

Label: R4

	Type	Value	Label
1	R	1000	R1
2	L	10u	L2
3	C	1u	C3

Figure 12. Graphical User Interface

An important consideration is that the generated component creates either the wanted real or reactive value from the component, it doesn't control both, at least from the GUI. That is a .5uF capacitor will indeed provide the wanted capacitance, but its resistance depends highly on the material and predefined program parameters. To have granular control, the use of the CLI is required.

Appendix B —Impedance Solver

Once the master orchestrator is ready, it passes the input to the impedance solver.

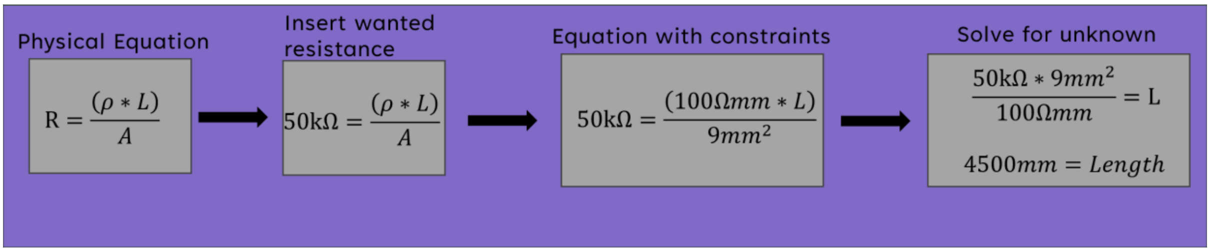

Physical Equation
$R = \frac{(\rho * L)}{A}$
Insert wanted resistance
$50\text{k}\Omega = \frac{(\rho * L)}{A}$
Equation with constraints
$50\text{k}\Omega = \frac{(100\Omega mm * L)}{9mm^2}$
Solve for unknown
$\frac{50\text{k}\Omega * 9mm^2}{100\Omega mm} = \text{L}$
$4500mm = Length$

Figure 13. Impedance solver substeps.

The constraints placed here are conservative, it is convenient to constrain the area in the sense that a very small cross section would decrease reliability. Resistivity should be known or empirically found. Now that resistance, resistivity, and area are constrained, length can be found. There are many ways to constrain an equation, knowing how to constrain it depends on the context. The impedance solver only outputs physical dimensions, that is, only area and length. While it is useful to know resistivity, area, and wanted resistance, these inputs are only necessary to the impedance solver. The impedance solver output is fed to the OpenSCAD module that creates the wanted geometry.

Appendix C — Correct Model Selection

Choosing the appropriate model is critical. Otherwise, you might not be able to print it.

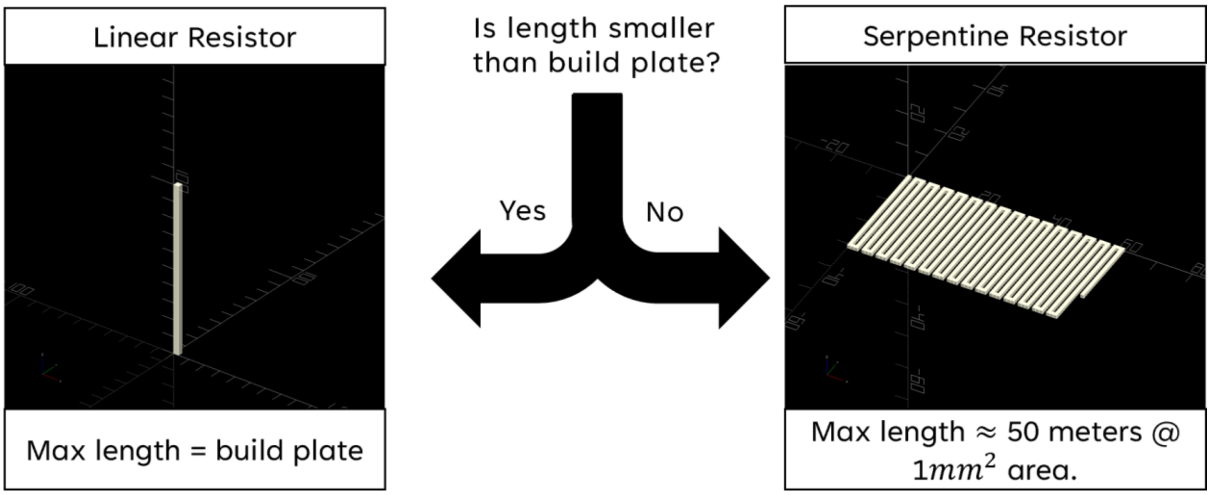

Linear Resistor
Is length smaller than build plate?
Serpentine Resistor
Yes
No
Max length = build plate
Max length ≈ 50 meters @ $1mm^2$ area.

Figure 14. Resistor model selection

Using 4500mm as length, a serpentine resistor is the ideal candidate for this specific resistor.

A 4500 linear resistor would not be able to print on most build plates.

Appendix D — Algorithmic Structure of the Serpentine Model

All current modules were written in OpenSCAD. Geometry creation is unique for each model; therefore, each model has its own algorithm. The serpentine model can be thought of as a linear resistor that is cut into many pieces, then reassembled into a serpentine structure. The following algorithm phases have been classified into colors so they can be understood easily when looking at the colored serpentine resistor.

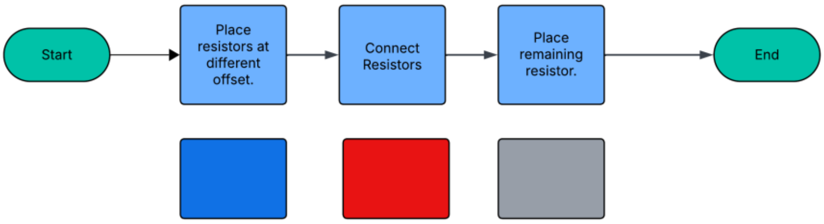

Start
Place resistors at different offset.
Connect Resistors
Place remaining resistor.
End

Figure 15. Serpentine resistor workflow.

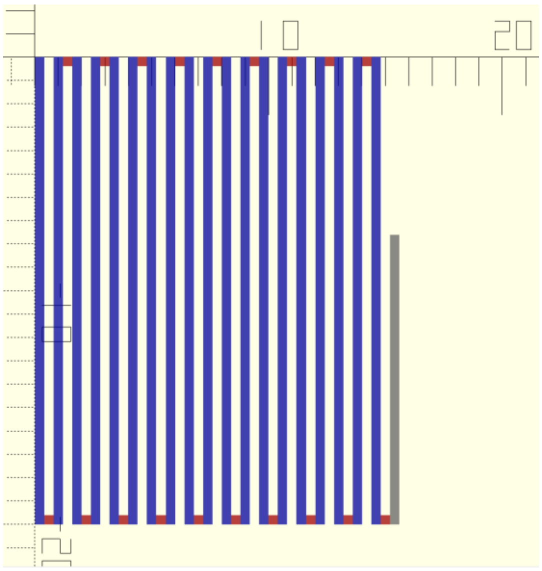


Figure 16. Colored Serpentine resistor.

Appendix E — Serpentine Resistor Source Code

OpenSCAD source code for the serpentine resistor.

```
module zigzag_resistor(Length, Area) {
  Width = sqrt(Area);
  Height = sqrt(Area);
  Unit_Length = sqrt(Length);
  Gap_Size = 2;
  Spacing = Width* Gap_Size;
  Turns = Length / (Unit_Length+(Gap_Size -1)*Width);
  for(i = [0:Turns - 1]) {
    translate([i * Spacing, 0, 0])
    rotate([0, 0,-90])
    cube([Unit_Length, Width, Height]);
    translate([ i * Spacing + Spacing / Gap_Size,
          (i % 2 == 0) ? -  Unit_Length : -Width,0])
    cube([Width*(Gap_Size-1), Width, Width]);
    if ( i == floor(Turns)-1){
      if((Turns - floor(Turns))* Unit_Length < sqrt(Area)){
        translate([(i+1) * Spacing,
                i % 2 == 0 ? -Unit_Length :0, 0])
        rotate([i % 2 == 0 ? 0:90,0,0])
        cube([Unit_Length*(Turns-floor(Turns)),Width, Height]);
      }else{
        translate([i%2==0? (i+1)*Spacing+Width:(i+1)*Spacing,
                (i % 2 == 0) ? -Unit_Length : 0, 0])
        rotate([0,0,i %2 == 0 ? 90:-90])
        cube([(Unit_Length+(Gap_Size-1)*Width)*
              (Turns-floor(Turns)),Width, Height]);
      }
    }
  }
}
//Invoke example
zigzag_resistor(100,10);
```

Appendix F — Exporting and Boolean Operations

Once the geometry is ready, one can export it “as is.” If for some reason, the user wished to wrap it in a protective film, for example or add different materials into the resistor, the cross-sectional area would look (red is resistor, blue is protective film):

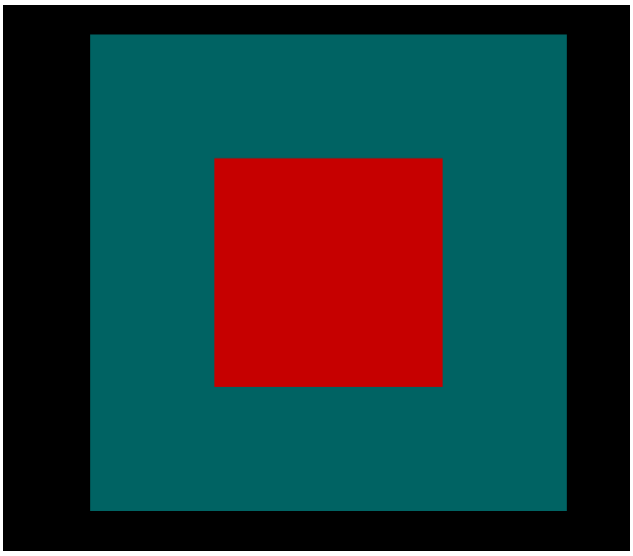


Figure 17. Resistor wrapped in a protective film.

If we were to export the geometry as it currently looks then we would not be able to export it, this is because there would be coplanar faces, resistor faces are touching the protective film, which would cause our CGAL geometry generator to crash. Therefore, the use of Boolean operations is needed so that we can successfully export our geometry. By implementing the necessary Boolean operations, we “shrink” the cross-sectional area of the resistor thus making it float. The gap between the resistor and the protective film is thin, usually in microns. This exports a single object that can be decomposed into parts later.

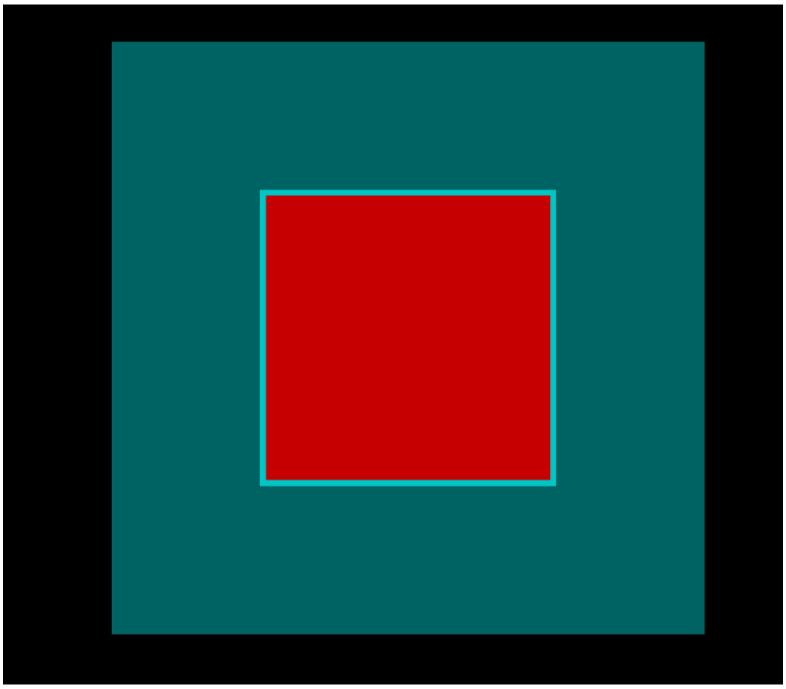


Figure 18. Serpentine resistor floating inside protective film.

Appendix G — Models Supported by QuickRLC

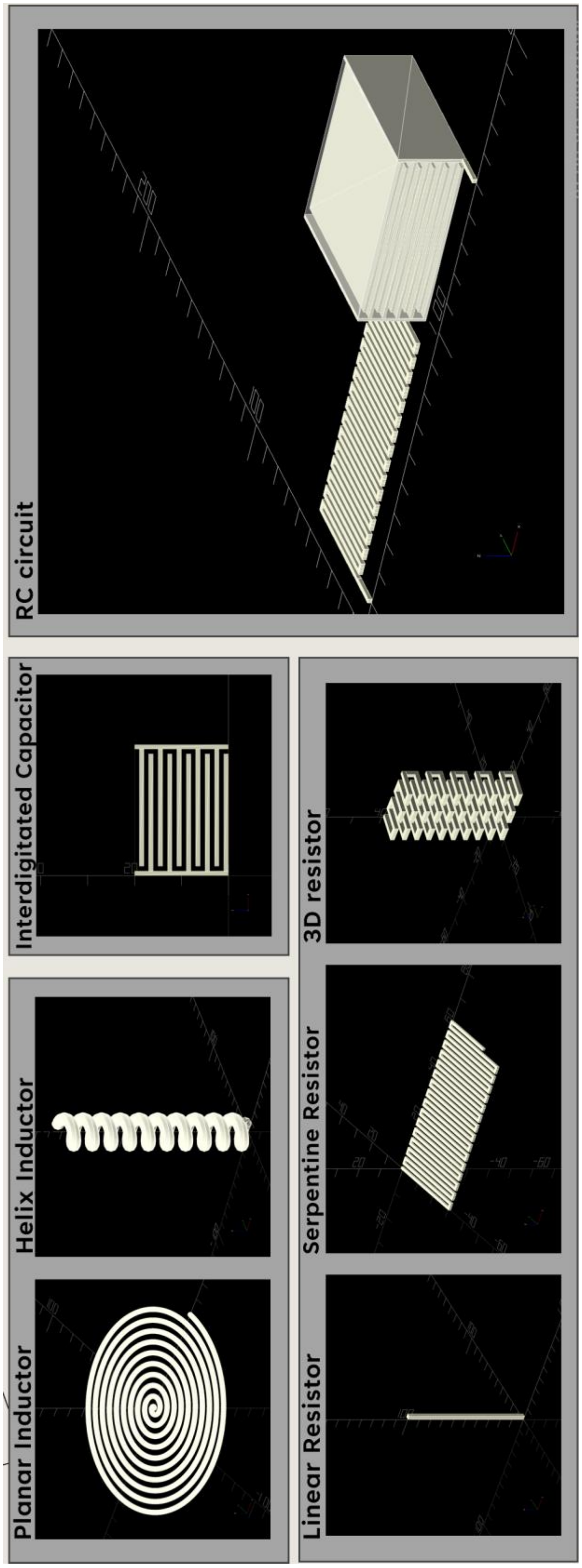


Figure 19. Different models supported by the tool.

Appendix H — Material Breakdown

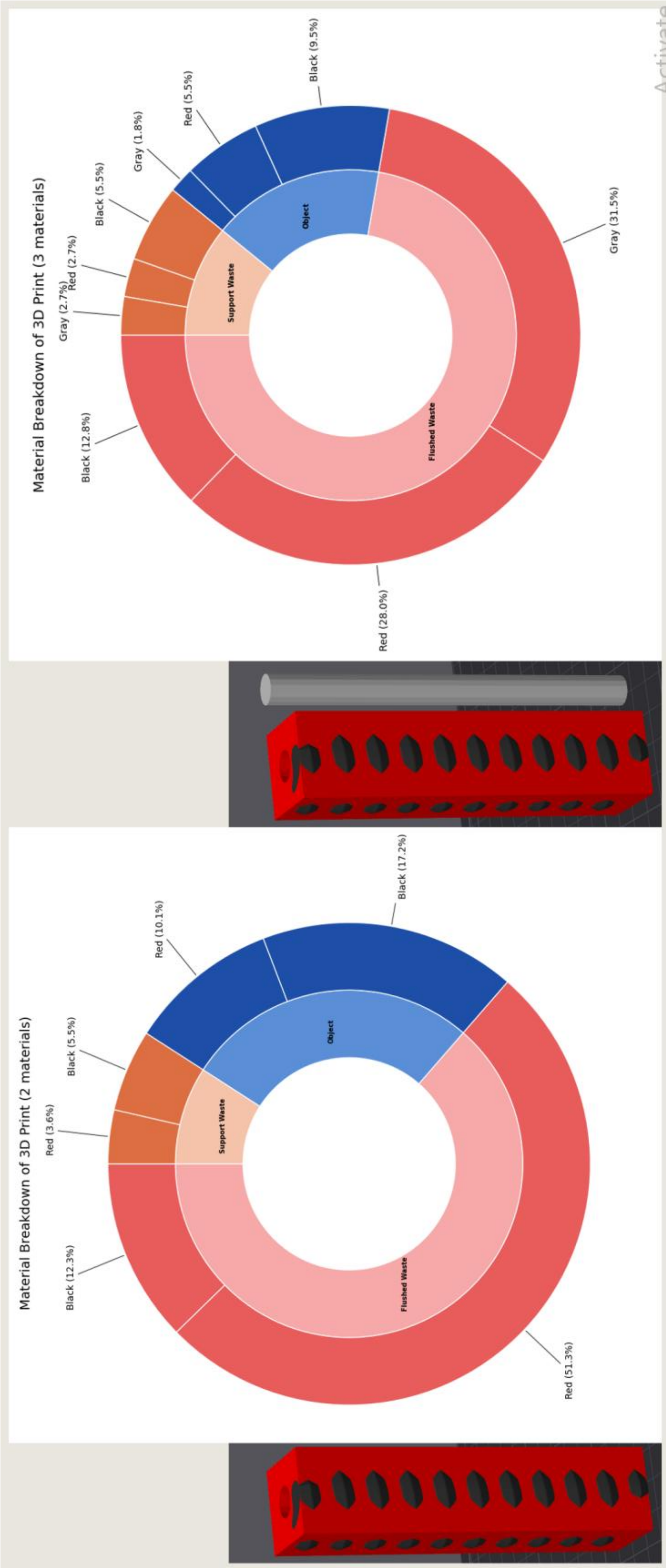


Figure 20. Material breakdown

Vita

Gabriel Sanchez Rangel is a graduate student in Computer Engineering at The University of Texas at El Paso, where he completed his Master of Science degree with a thesis focused on software-automated fabrication of multi-material additive printed circuits. He previously earned a Bachelor of Science in Electrical Engineering from UTEP, graduating *Cum Laude*, and was recognized with the Best Senior Project Award for his work on a hardware-encrypted chaotic shift key communication system. His academic and research experience includes developing embedded systems and hardware/software integration solutions, printed circuit board and sensor designs with associated manufacturing documentation, and high-speed particle detection systems as a Master's Research Assistant at the W.M. Keck Center for 3D Innovation. In addition to his research activities, Gabriel has founded startups including 3dforads.com and Lumolith.com, and has consulted on CNC automation and furniture software development for manufacturing-focused projects. His professional interests include embedded systems, additive manufacturing, circuit design, and hardware-software integration. Additional information about his projects and experience is available at gabrielresume.com.