**Optimization of printing parameters to reduce warpage in fused deposition modeling**

**Pranav Addepalli**

**August 28th, 2019**

**Problem**

In large-scale 3D printing, one of the most substantial issues is the warpage problem. Warpage is the distortion of a 3D printed part due to variations in thermal cycles of heating and cooling corresponding to expansion and contraction of plastic. In this study, convex optimization methods are used to modify the four major process variables (infill percentage, infill type, layer height, and wall thickness) during the 3D printing by studying their effects on heat flow trajectories and spatiotemporal temperature variation in an attempt to reduce the likelihood of warpage.

**Hypothesis**

Previous work showed that increasing infill percentage led to a decrease in the likelihood of warpage. Other studies show the effect of each individual printing parameter on temperature variation with almost all of them implying that by increasing part density, the total natural convection will be reduced and will consequently increase conduction along the solid plastic. It is therefore hypothesized that any combination of the printing parameters that increases the density of plastic will lead to more uniform heat flow trajectories and temperature variations.

**Independent Variables**

The printing parameters that will be studied in this experiment include infill percentage (10%, 20%, or 30%), infill type (rectilinear, triangular, or hexagonal), layer height (2mm or 4mm), and wall thickness (0.8mm or 1.5mm). All combinations will be printed and tested such that there are 36 unique combinations of parameters.

**Dependent Variables**

The main measured variable is the temperature of a printed object at different positions over time. This data has both a spatial depth, since there are measurements throughout the object, and a temporal depth, since the data is over time – during and after the printing process. The temperature will be measured by calculating the temperature based on a thermistor’s voltage reading using the Steinhart-Hart Equation:

where *T* is the temperature in Kelvin, *R* is the resistance in ohms, and *A*, *B*, and *C* are the Steinhart-Hart coefficients that are constant to specific thermistors**.** Ultimately, each disc will produce spatiotemporal temperature variation data that will undergo extensive processing for comparison.

**Constants**

The constants in this experiment will be the material used (Hatchbox 1.75mm PLA), the 3D printer used (a gCreate gMax 1.5 XT+ printer), the printer settings (extruder temperature of 200°C; a bed temperature of 80°C; the time intervals between measurements (1 second); the type of NTC thermistor used; and all other printing parameters and process variables.

**Control**

There is no control in this experiment because the temperature variation data from each of the 36 parameter combinations will be compared with each other rather than against a standard.

**Repeated Trials**

There will be no repeated trials due to time constraints. On average, printing and measuring data from one parameter combination in this study will take 24 hours. If time permits, certain trials may be repeated.

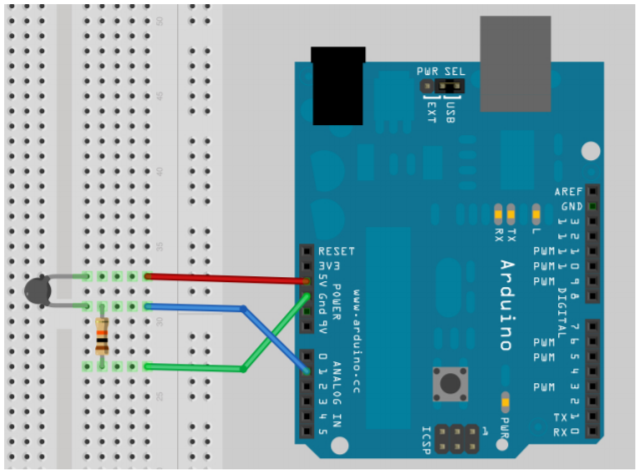
**Materials**

* Hatchbox 1.75mm PLA filament (SDS: ultimaker.com/download/13078/SDS%20PLA.pdf)
* gCreate gMax 1.5 XT+ 3D printer with Heated Bed Extension
* 500 Glass encapsulated 20kOhm NTC thermistors (NTCLG100E2203JB) from Vishay/BC Components
* 10kOhm resistors
* Arduino Mega
* Computer with Arduino, Python, and Fusion360 software

**Procedure**

***Phase 1: Equipment Setup***

1. Throughout any manufacturing or testing procedure, safety glasses, boots, and gloves will be worn.
2. I will set up an Arduino temperature sensor using thermistors, program it to calculate temperatures based on voltage readings, and test the sensor with room temperature and hot water.
   1. Wearing an anti-static wristband, the temperature sensor will be connected through a voltage divider circuit, shown in Figure 1.



Thermistor

Resistor

Arduino Uno R3

**Figure 1:** *Diagram for a Voltage Divider circuit and sample breadboard wiring for thermistor resistance calculations.*

* 1. The temperature will be calculated using an Arduino program based on Equation 3, a modification of the Steinhart-Hart relationship, where *T* is the thermistor temperature in Celsius; *TA*, *TB*, and *TC* represent the thermistor alpha, beta, and gamma coefficients respectively; *R0* represents the fixed resistance in the circuit, and *V* represents the analog thermistor reading.
  2. The temperature sensor will be tested by running the program, recording the calculated temperature values, and comparing these values with actual thermometer temperature measurements.

1. I will use Fusion 360 CAD software to create discs for the 36 different parameter combinations.
   1. Each disc will be designed with a single channel for wiring and small channels leading from the single channel to thermistor locations in the disc. Each thermistor will be placed at 1 inch intervals radially throughout the entire 8 inch diameter disc.
   2. Each combination will have a radius of 4 inches and a thickness of 0.5 inches.

***Phase 2: Testing***

1. Safety equipment in Phase 1, Step 1 will be worn.
2. To find the temperature variation data, the discs designed in Phase 1, Step 2 will be 3D printed and thermistors placed into the discs while printing.
   1. While the disc is printing in accordance with the CAD model, once the printer has finished printing the channels (at a height of 8.89 mm), the print will be paused and thermistors will be fixed and connected to the Arduino.
   2. The wires will be led out of the disc, print bed, and printer area and connected to an Arduino voltage divider circuit assembled in Step 1 of Phase 1.
   3. Extreme caution must be taken during Phase 2, Step 2.2 by not touching the print in order to prevent external modification of the print and temperature as well as to prevent burns.
   4. The program will begin to read the thermistor outputs and calculate temperatures in 1 second intervals.
   5. Once completed, the thermistors will be disconnected but the Arduino circuit will be preserved such that it can be used to test another disc.
3. This process will be repeated for each of the discs.

***Phase 3: Data Analysis and Optimization* THIS IS STILL IN PROGRESS.**

1. After collecting the raw spatiotemporal temperature data for a printed disc, the data can be segmented by time (since measurements are taken at every second). There will be 6 thermistors with temperature measurements at each time value, and each has a unique X and Y position. Using the temperatures and the position, the "center of temperature" can be calculated similar to the process used last year with partial derivative analysis.
2. Each center of temperature will also have a calculated temperature value at that point. This allows the creation of comparable tensors of form (Horizontal center position X, Vertical center position Y, Temperature, and time). This means that the center of temperature data has a spatial depth (since the centers are moving) and a temporal depth.
3. Since there is a spatial and temporal depth to the center of temperature, a trajectory can be created to represent the heat flow for a certain disc. The trajectories can then be compared. **WILL FIND OUT HOW SOON**
4. The original four order tensors with X, Y, temperature, and time can be given additional dimensions unique to each disc - the printing parameters. This creates an eighth-order tensor with X, Y, temperature, time, infill percentage, infill type, layer height, and wall thickness.
5. These eighth-order tensors can be compared to the tensor of the actual "center" in terms of position for the disc. Each tensor can be given an additional dimension that represents an "evenness score" - simply the distance from the original center. This is the factor that we are going to be optimizing for, so the algorithm used to compute the score may need to be changed from simply a distance factor. **EVENNESS SCORE COULD BY RESULTS FROM THE DYNAMIC TIME WARPING ALGORITHM OR IT COULD BE A COMPLEX DISTANCE CALCULATION BASED ON GRAPH THEORY – WE WILL FIND OUT SOON, BUT IT MAY END UP BEING I FIGURE THIS OUT ONCE I START GETTING ACTUAL DATA**

**No statistics? I’m just going to be doing an optimization, if time permits, I will print one with the optimized values and compare the temperature distribution with the other prints and show qualitiatvely how its better to reduce warpage / uniform temperature / etc.**