



# **TRANSIENT HEAT CONDUCTION**

December 2022

**Zakaria Kortam**

**Professor Abdollah Tabrizi**

**ENGR 50**

# Table of Contents

THC - Experiment

## Main Contents

- [1] Introduction
- [3] Procedure
- [4] Program
- [9] Results
- [11] Analysis

## Additional

- [13] Graphs
- [15] A.D.
- [16] Oath
- [17] Code

# Introduction

Within this experiment, a long, rectangular, and slender rod of steel is initialized to 20° Centigrade (293.15K) in temperature across its surface. Over the course of this experiment, each of the four sides of the surface will be exposed to varying and pre-determined temperatures, which will remain constant. The objective of this experiment is to measure and identify the locations, as well as the degree of temperature variation on the surface. In order to identify the changes and rate-of-change of temperature based on time and location, the following differential equation will be used and implemented.

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = \frac{1}{\alpha} \frac{\partial T}{\partial t}$$

The above equation is a condensed form of the following equation

$$\frac{\left( (x+1,y) - 2(x,y) + (x-1,y) \right) / dx^2 + \left( (x,y+1) - 2(x,y) + (x,y-1) \right) / dy^2}{1/a \left( [\text{current time}](x,y) - [\text{previous time}](x,y) \right) / dt}$$

The left-hand side of the equation, in this case the top, represents the data from the previous point in time of that same position, as well as the positions around it. The right side/bottom displays the result. One can already see that all that's needed is for [current time](x,y) to be isolated, which will give you the current temperature of that position in that point in time.

The automated calculations, for each individual point in time, will be implemented and executed through the utilization of the C++ programming language, which, through user input and interaction, will compose a specific object class for this surface, which will additionally augment its independent capability and versatility. In addition to the custom object-class, the program will utilize a 3-dimensional array for the storage of temperatures across various intervals of time.

In the following experiment, the data will be initialized to the following:

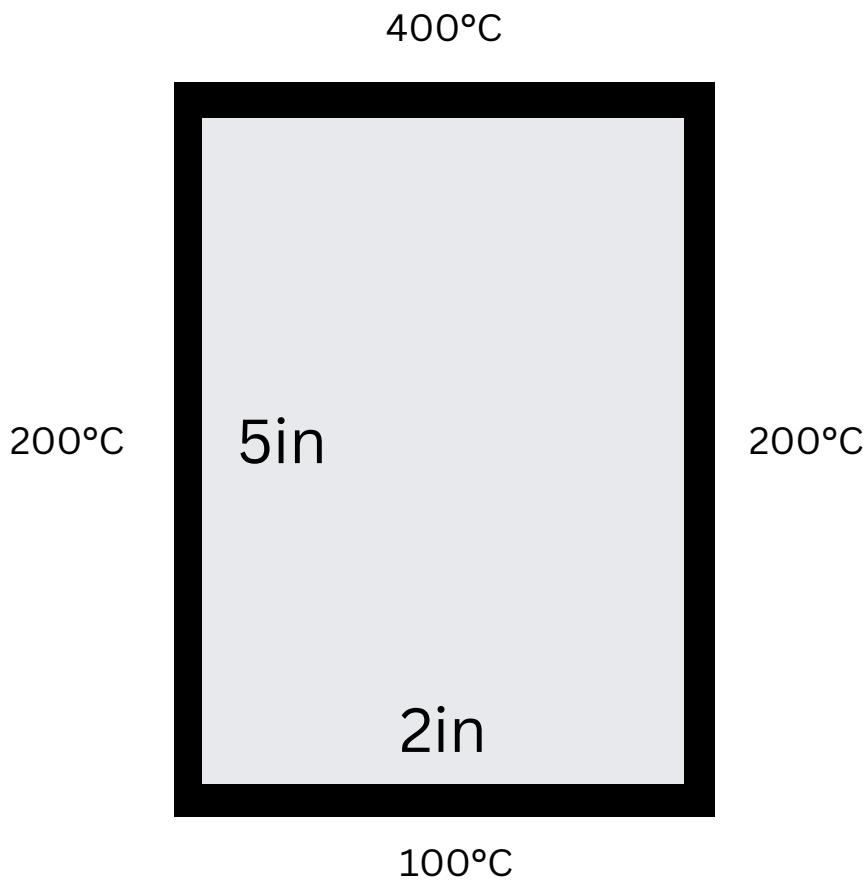
Top = 400°C

Bottom = 100°C

Left = 200°C

Right = 200°C

A (Alpha) =  $1.8 \times 10^{-5}$



# Procedure

- [1] Program opens with a header banner.
- [2] User enters Material Name.
- [3] User enters Material side temperatures.
- [4] User inputs initial temperature.
- [5] User inputs  $dt, dx, dy$ .
- [6] User inputs the number of points (X,Y,T).
- [7] User determines alpha and file name.
- [8] Program runs the calculations.
- [9] The data is stored in a 3D array.
- [10] Data is outputted into a file.

# Program C++

# Program

## Material Class

### Fields

- name: string
- temp: long long[][][]
- dt: double
- dx: double
- dy: double
- initial: double
- tx: int
- ty: int
- tt: int
- a: double
- s1: double
- s2: double
- s3: double
- s4: double
- + filename: string

### Functions

- + Material(0 args)
- + Material(14 args)
  - Accessors
  - Mutators
- + Initialization() : void
- + InitialOutput() : void
- + Calculations() : void

# Program

## Main

```
int main() {  
    Material item = introduction();  
    output.open(item.filename);  
  
    item.Initialization();  
    item.InitialOutput();  
    item.calculations();  
    system("Pause");  
  
}
```

Introduction() : Material - Initiates the user-input series.

output.open() : ofstream - Creates the output file with the name attached to the Material object (item).

Initialization() : void - Initializes the temperatures.

InitialOutput() : void - Prints the initialized state.

calculations() : void - Calculates and prints the output.



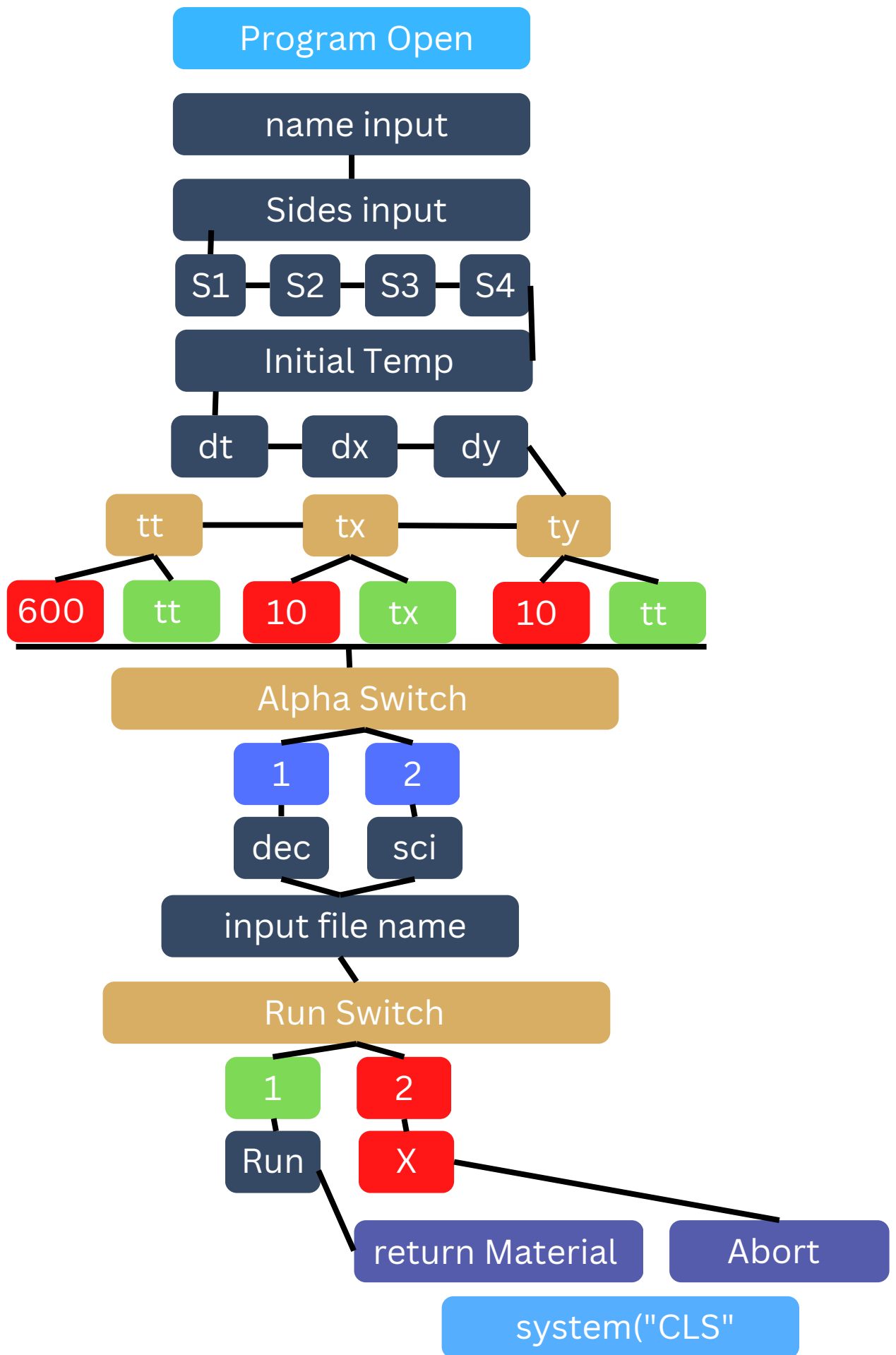
# Program

## Material:calculations()

```
cout<<setw(7);
  for (int time = 1; time < tt; time++) {
    for (int row = 1; row < ty-1; row++) {
      for (int col = 1; col < tx-1; col++) {
        //Formula Components
        double segment1 = (temp[row+1][col][time-1] - (2 * temp[row][col][time-1])
          +temp[row-1][col][time-1]) / pow(dx, 2);
        double segment2 = (temp[row][col + 1][time-1] - (2 * temp[row][col][time-1])
          + temp[row][col - 1][time-1])/ pow(dy, 2);

        //Formula Assembly
        temp[row][col][time] = dt * a * (segment1 + segment2) + temp[row][col][time-1];
        output<<setprecision(4)<<temp[row][col][time];
        output<<setw(7);
      }
      output<<endl;
    }
    output<<endl;
    output<<endl;
    output<<endl;
  }
}
```

Within this function, it has a nested for-loop that extracts and outputs the data for each column in each row in every point of time. Segment 1 and Segment 2 are the two components of the differential equation previously shown. In the assembly section, they are combined, along with the necessary calculations to create the output value, which is the temperature at that particular spot in that point in time.



# Results

**It took 0.0036 seconds for the temperature change to effect the mid-point, which it did by 1 degree.**

**From that point, it took 0.0006 seconds for the center point to double from 20°C to 40°C.**

Temp change from 20°C	Time it took to reach it from 0.	
	2x	0.0042
	4x	0.0110
	8x	0.0216

**For the points nearest to the side with 400°C, in the middle horizontally, it took only 0.002 seconds for the temperatures to increase 10x to 200°C.**

**For the points nearest to the side with 100°C, in the middle horizontally, it took only 0.02 seconds for the temperatures to increase 5x to 100°C.**

**It took 20x longer for the temperatures to increase, on average, in the lower temperature area, than the higher temperature area.**

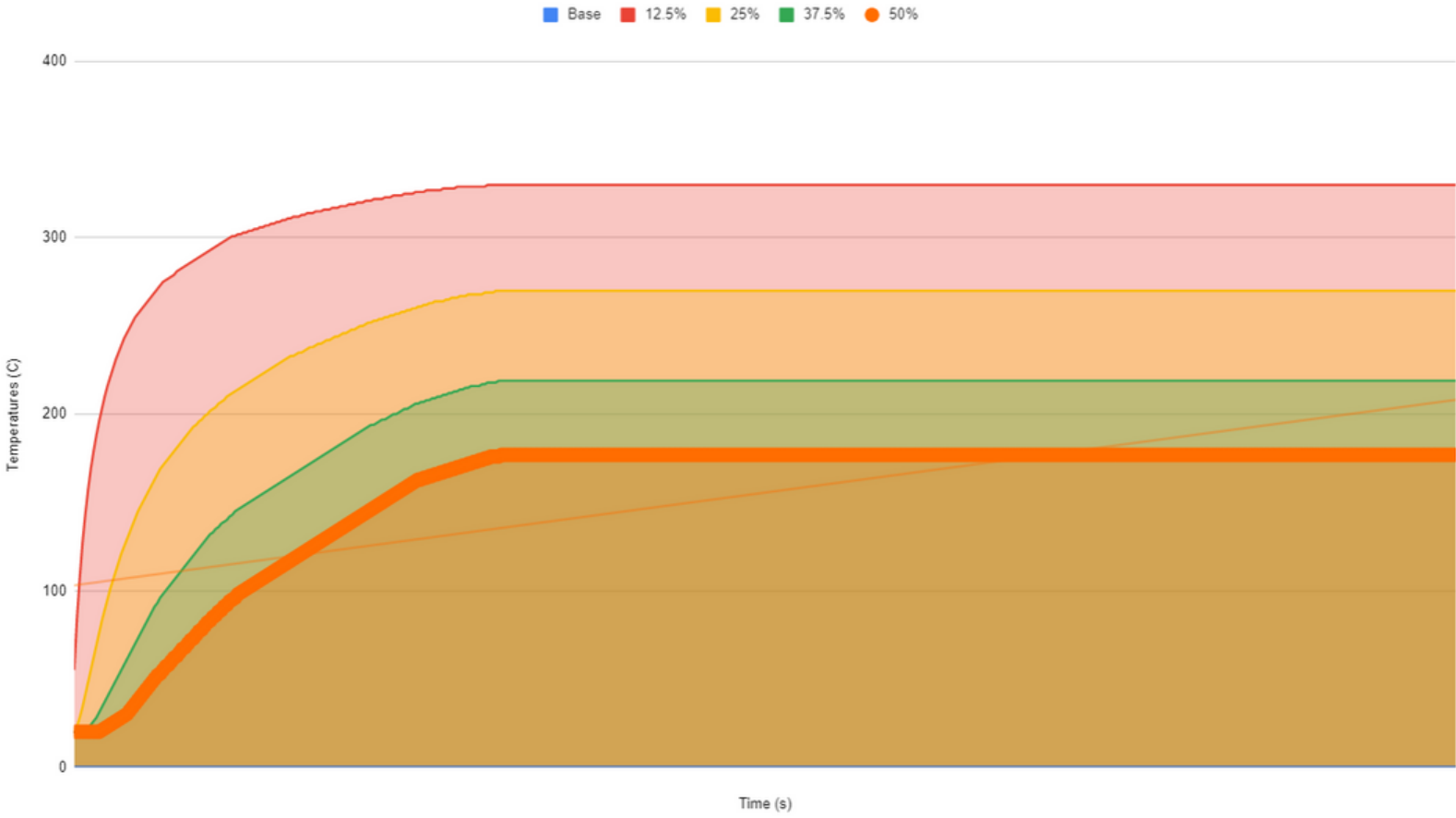
# Analysis

Within this experiment, as expected, the regions that are closer to the side that has the higher temperature, have higher temperatures. However, a major take-away would be the rate of change based upon the side-temperature. As we saw in the results section, the rate of change was substantially slower in the side that approaches 100 degrees, when compared to the side that approaches 400. This was likely due to the fact that the side-temperature acts as its own limit, thus meaning that when the difference between it and the initial temperature is lower, it acts as its own ceiling. Therefore, the amount of resistance achieved at a lower temperature is higher. We saw this too in the chart displayed in the previous page, where, initially, the rate of change was extremely quick, and then after a certain point, with time, it started to slow down as it approached a certain temperature, due to the fact that the temperature can not exceed the temperature of the highest temperature side. Consecutively, this is all relative. For the center point, for instance, it can not exceed the highest temperature of the highest-temperature point that surrounds it.

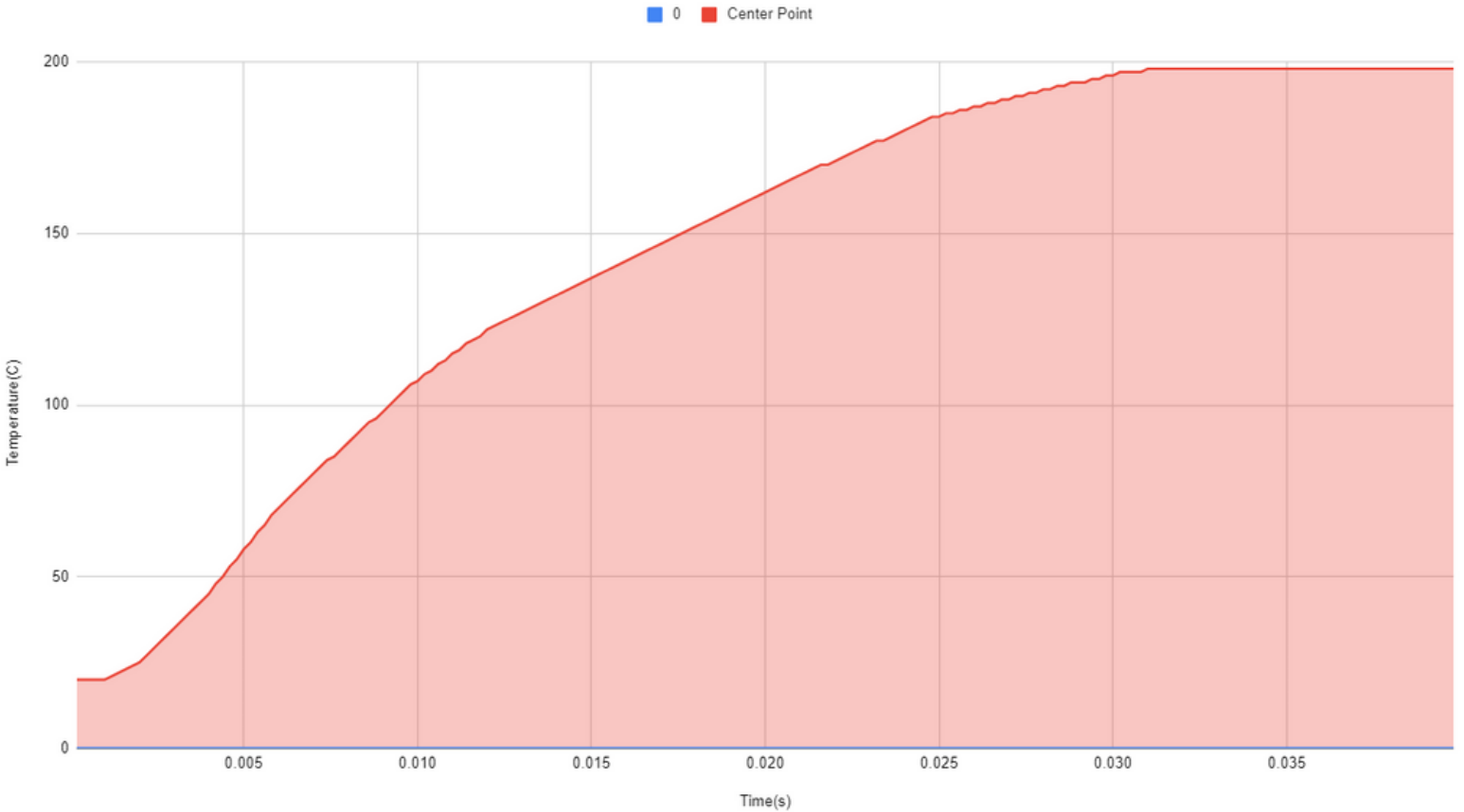
# Analysis

Additionally, while the higher temperatures had higher rates of initial increase, the duration of severe increase was substantially shorter. In simpler terms, higher temperature areas lost steam substantially quicker than lower temperature ones, since they approached the limit much sooner and more intensely. One more observation would be that temperature points on the same y position are much more consistent than points at the same x position. For instance if  $(2, 5)$  and  $(9, 5)$  were compared, they would have a similar temperature than  $(2, 8)$  and  $(2, 3)$ . This is due to the real temperature fluctuation happening as a result of the top and bottom temperatures  $(400, 100)$ , instead of the side temperatures, which are consistently 200.

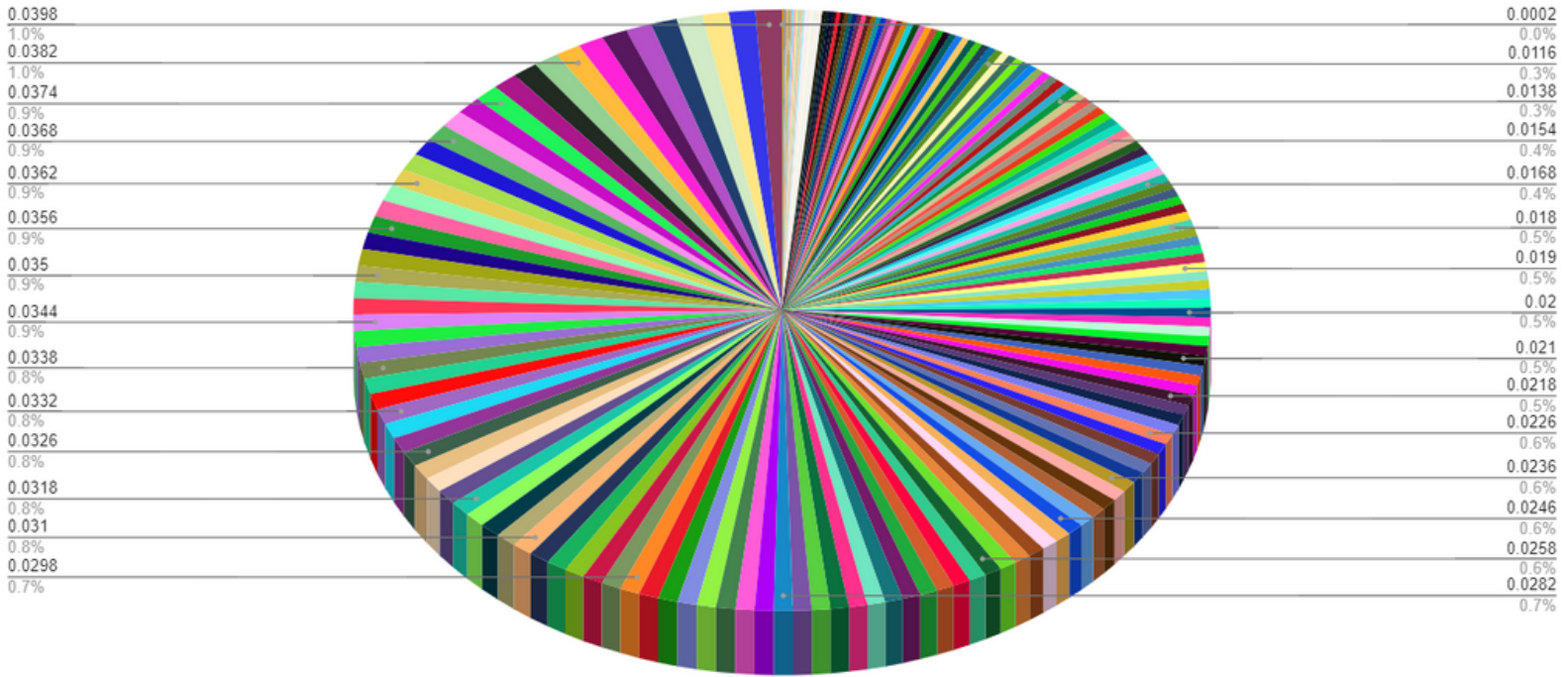
### Mid-X Temperatures | Varying distance from the top



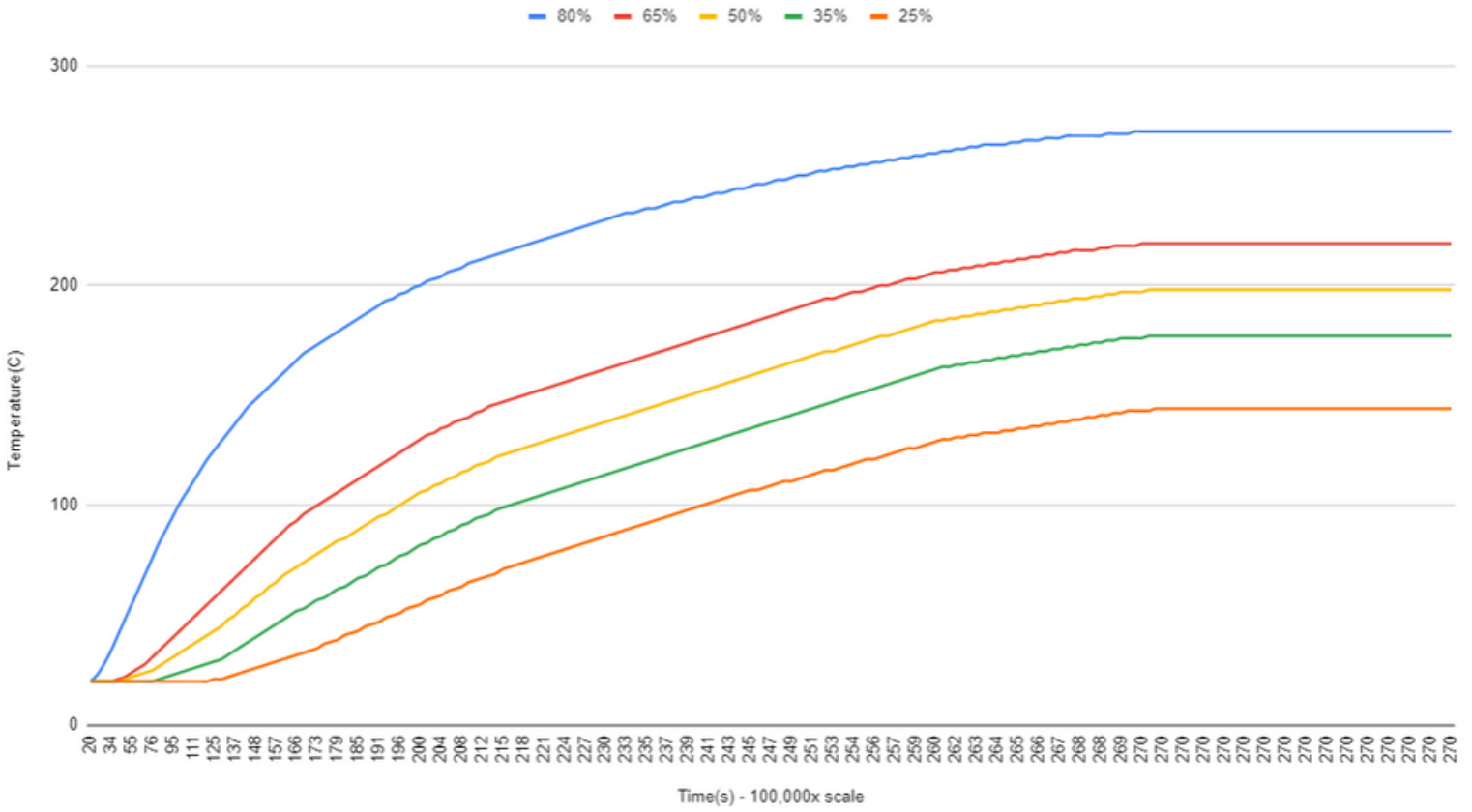
### Center point temperature change over time



Center point Rate of Change per 0.0002 seconds.

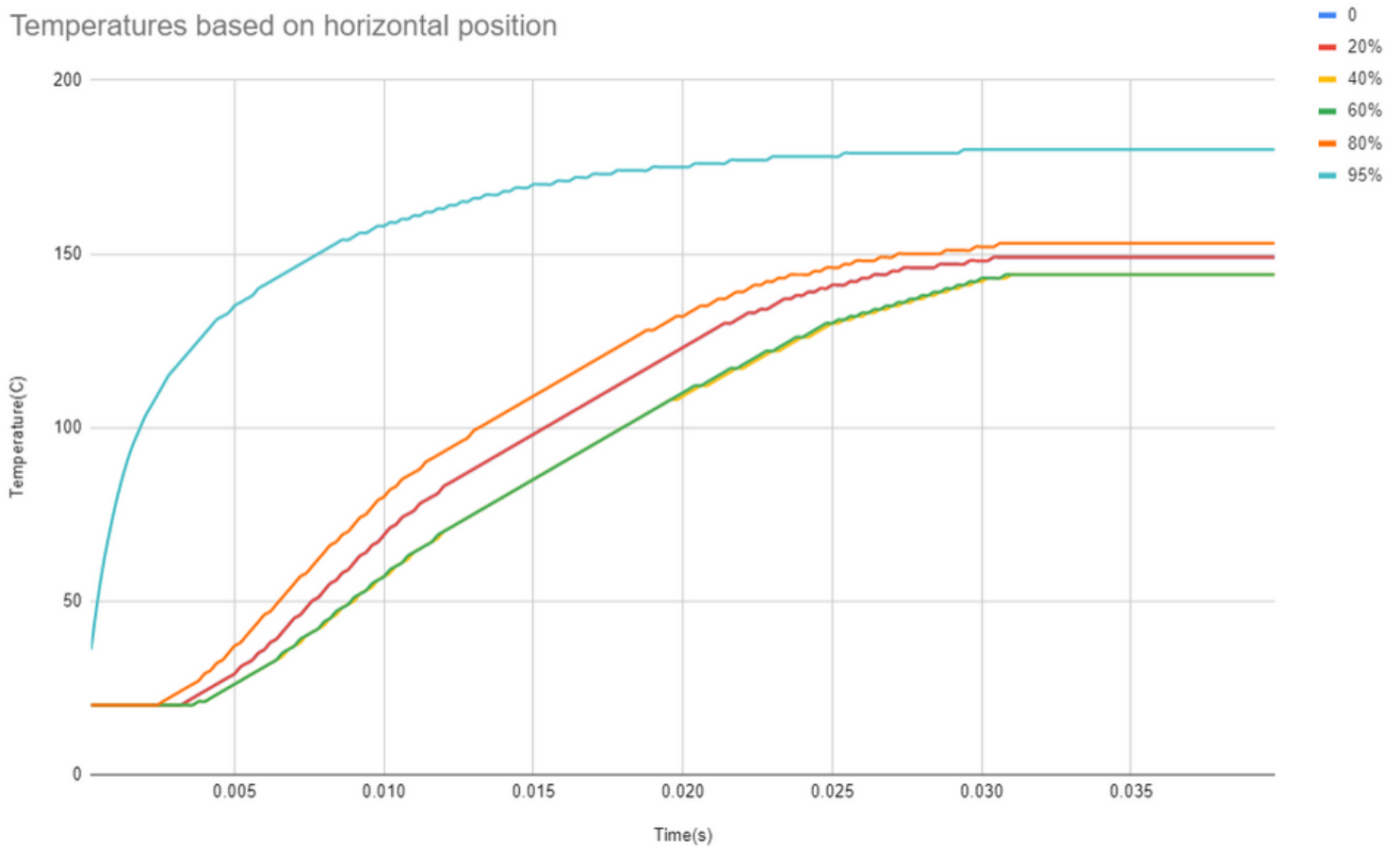


Different points | Varying vertical position

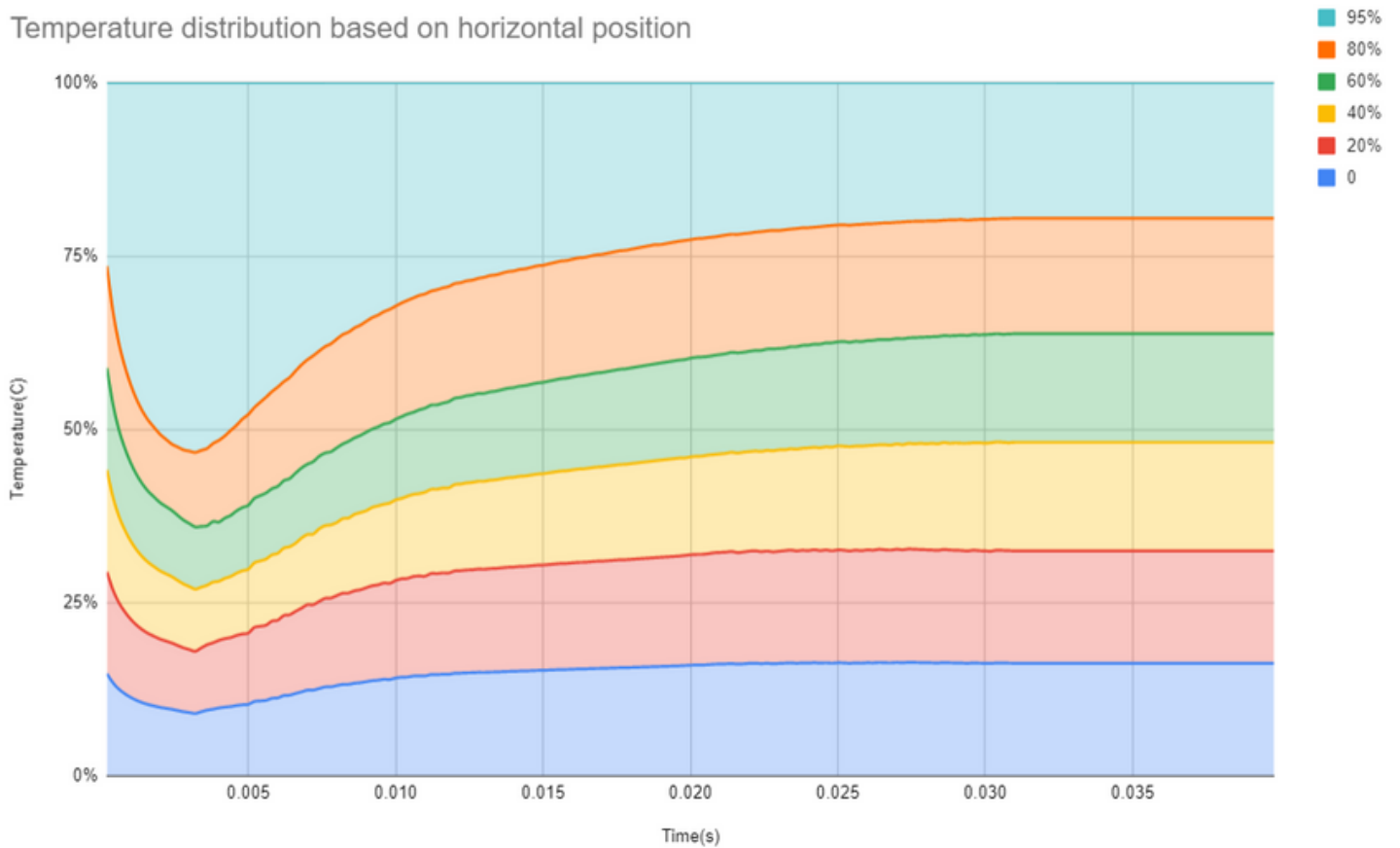




Temperatures based on horizontal position



Temperature distribution based on horizontal position



# Oath of Integrity

**“This project has been completed in its entirety, including the program design/development and the written report, by me. I have not shared any portion of my program with anyone. I know that plagiarism of any kind may result in a grade of ‘ F ‘ in the course.”**

Zakaria Kortam

12/11/2022

A handwritten signature in black ink, appearing to read 'Zakaria Kortam', written in a cursive style.

# Source Code

