University of Colorado - Boulder

ASEN 3802: Aerospace Sciences Laboratory II

March 21st, 2024

Lab 2: Heat Conduction

Author:
Reece Fountain

Author:
Nathan Malyszek

Professor:
Jeff Glusman

Author:
Jared Steffen

I. Introduction

The purpose of this Lab is to analyze the heat conduction of various materials. Additionally, the rate of heat generation applied to each material varies by data set, as the voltage applied to the heater is a variable. The materials used are 7075-T651 Aluminum, C360 Brass, and T-303 Annealed Stainless Steel. The voltages applied are 25V, 240mA, and 30V 290mA for the Aluminum, 25V, 237mA, and 30V, 285mA for the Brass, and 22V, 203mA for the Steel. The datasets are of thermocouple data, recorded every 10 seconds, spaced evenly every 0.5 inches (1.27 cm) after the first thermocouple, which is placed 1 3/8 inches (3.49 cm) beyond the heater. We use MATLAB to plot and perform analysis on the data, including comparison to analytical calculations of heat transfer.

II. Methodology

A. Initial Question 1

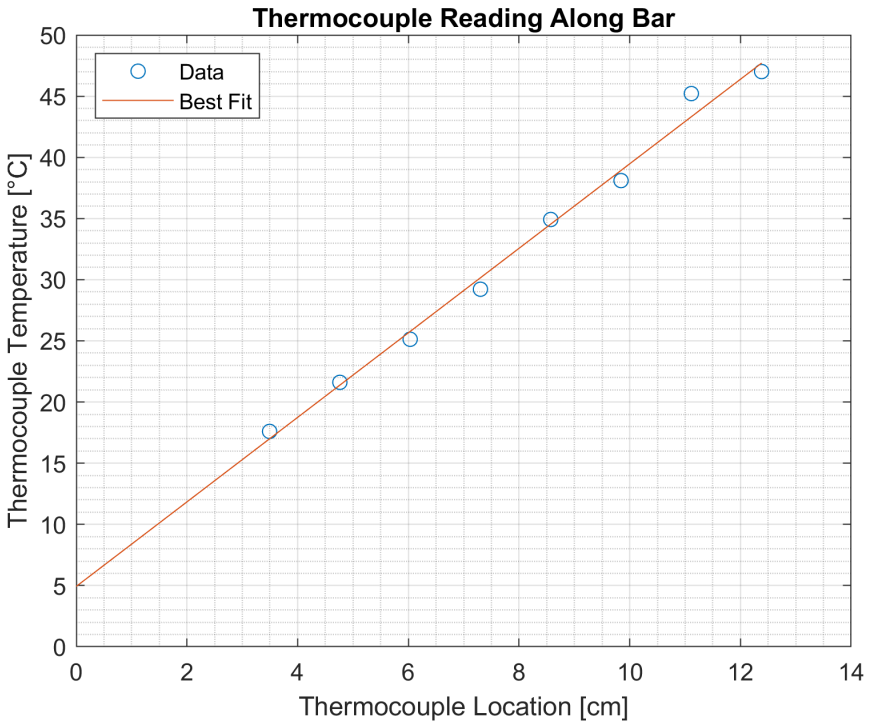

Fig. 1

Figure 1 shows the temperature distribution along for an arbitrary metal rod and the best-fit line for the data. This method was implemented in two different sections of the lab, which will be detailed later.

B. Initial Question 2

We derived the 1D Heat Equation as an analytical solution for modeling the heat transfer through each materials from **Equation 1** and **Equation 2**, where $n = 1, 2, 3, ...$

$$\lambda_n = \frac{(2n-1)\pi}{2L} \tag{1}$$

$$b_n = \frac{-2H}{L}\int_0^L x \sin(\lambda_n x)\, dx \tag{2}$$

First, we evaluate the integral portion of **Equation 2** through integration by parts, followed by substitution of **Equation 1** once simplified before simplifying further.

$$u = x \qquad du = dx$$
$$dv = \sin(\lambda_n x)dx \qquad v = -\frac{\cos(\lambda_n x)}{\lambda_n}$$

$$\begin{aligned}
\int_0^L x\sin(\lambda_n x)\,dx &= -\frac{x\cos(\lambda_n x)}{\lambda_n}\bigg|_0^L - \int_0^L \frac{\cos(\lambda_n x)}{\lambda_n}\,dx \\
&= \left[-\frac{x\cos(\lambda_n x)}{\lambda_n} - \frac{\sin(\lambda_n x)}{\lambda_n^2}\right]_0^L \\
&= -\frac{L\cos(\lambda_n L)}{\lambda_n} - \frac{\sin(\lambda_n L)}{\lambda_n^2} \\
&= -\frac{\lambda_n \sin(\lambda_n L) + \lambda_n^2 L\cos(\lambda_n L)}{\lambda_n^3} \\
&= -\frac{\sin(\lambda_n L) + \lambda_n L\cos(\lambda_n L)}{\lambda_n^2} \\
&= -\frac{\sin\left[L\frac{(2n-1)\pi}{2L}\right] + \frac{(2n-1)\pi}{2L}L\cos\left[L\frac{(2n-1)\pi}{2L}\right]}{\left[\frac{(2n-1)\pi}{2L}\right]^2} \\
&= -\frac{4L^2(-1)^n}{\pi^2(2n-1)^2}
\end{aligned}$$

This last simplification is possible because the $\sin\left[\frac{(2n-1)\pi}{2}\right]$ and $\cos\left[\frac{(2n-1)\pi}{2}\right]$ terms in the equation alternate between ± 1, which can be expressed more simply as $(-1)^n$. Finally, substitute this evaluation into **Equation 2** for final simplification.

$$\begin{aligned}
b_n = -\frac{-2H}{L}\int_0^L x\sin(\lambda_n x)\,dx &= -\frac{-2H}{L}\left[-\frac{4L^2(-1)^n}{\pi^2(2n-1)^2}\right] \\
&= \frac{8HL(-1)^n}{\pi^2(2n-1)^2}
\end{aligned}$$

This leaves us with the final expression **Equation 3**.

$$b_n = \frac{8HL(-1)^n}{\pi^2(2n-1)^2} \tag{3}$$

Equation 3 can then be used inside **Equation 4** to calculate the temperature along a bar at a given time.

$$u(x,t) = T_0 + H_x + \sum_{n=1}^{\infty} b_n \sin(\lambda_n x)e^{(-\lambda_n^2 \alpha t)} \tag{4}$$

$$u(x,t) = T_0 + H_x + \sum_{n=1}^{\infty} b_n \sin(\lambda_n x)e^{(-\lambda_n^2 \alpha t)} = T_0 + H_x + \sum_{n=1}^{\infty} \frac{8HL(-1)^n}{\pi^2(2n-1)^2}\sin\left[\frac{(2n-1)\pi}{2L}x\right]e^{\left[-\frac{(2n-1)\pi}{2L}^2 \alpha t\right]}$$

This provides our final Equation 5 for analytical analysis for comparison to our experimental results.

$$u(x,t) = T_0 + H_x + \sum_{n=1}^{\infty} \frac{8HL(-1)^n}{\pi^2(2n-1)^2}\sin\left[\frac{(2n-1)\pi}{2L}x\right]e^{\left[-\frac{(2n-1)\pi}{2L}^2 \alpha t\right]} \tag{5}$$

C. Initial Question 3

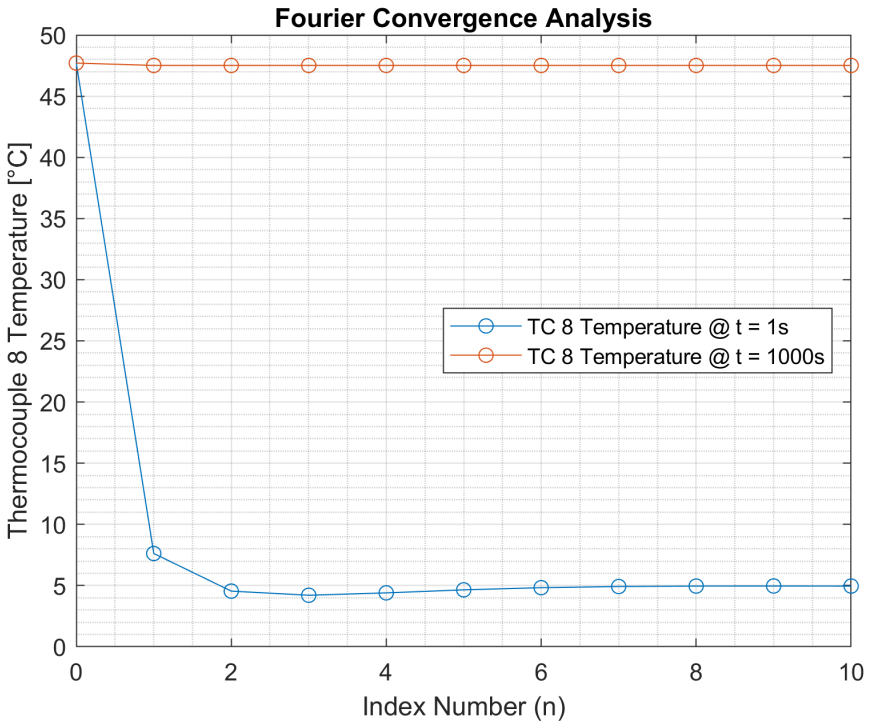

Fig. 2

Since the solution to the one-dimensional heat equation includes a summation of n from 1 to ∞, the number of terms to be sufficiently accurate needs to be determined. **Figure 2** shows the thermocouple temperature for the last thermocouple at $t = 1s$ and $t = 1000s$ for $1 \leq n \leq 10$. Since the analysis of this lab occurs at the steady state of the rod ($t \geq 1000$), it can be seen that one term of the summation will be sufficient for the analysis of the steady states of each test.

D. Initial Question 4

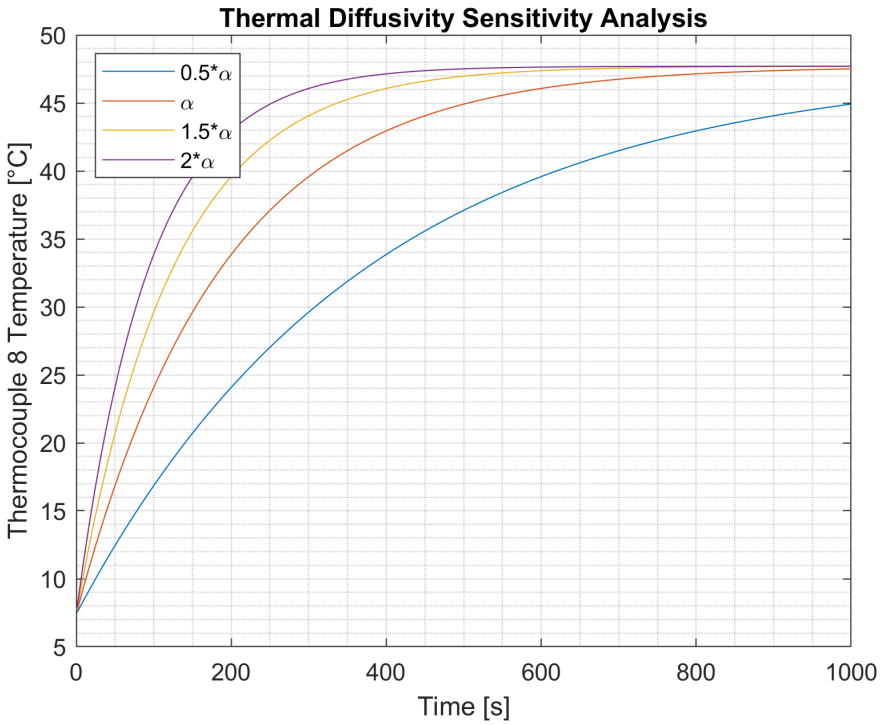

Fig. 3

Thermal diffusivity for a material can be calculated using **Equation 6** and given material properties shown in **Appendix A.A**.

$$\alpha = \frac{k}{\rho c_p} \tag{6}$$

Using the values given to calculate the thermal diffusivity for each material, it can be seen that thermal diffusivity can vary quite a bit among different materials. **Figure 3** shows how thermal diffusivity effect the time dependent temperature profiles for a given material. The specific material used in this case is Aluminum. The higher the thermal diffusivity, the faster that material reaches its steady state, and the lower the thermal diffusivity, the longer it takes to reach the steady state. From the i=given material values, it can be expected that Aluminum and Brass reach their steady states relatively quickly, while Steel will take much longer.

III. Results

A. Steady State Distribution

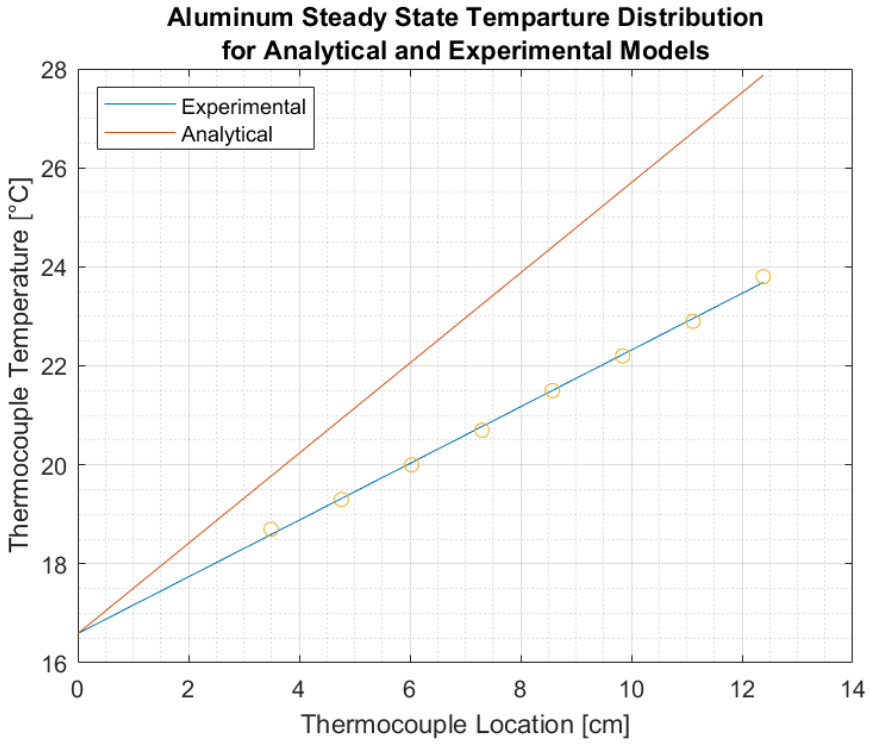

Fig. 4

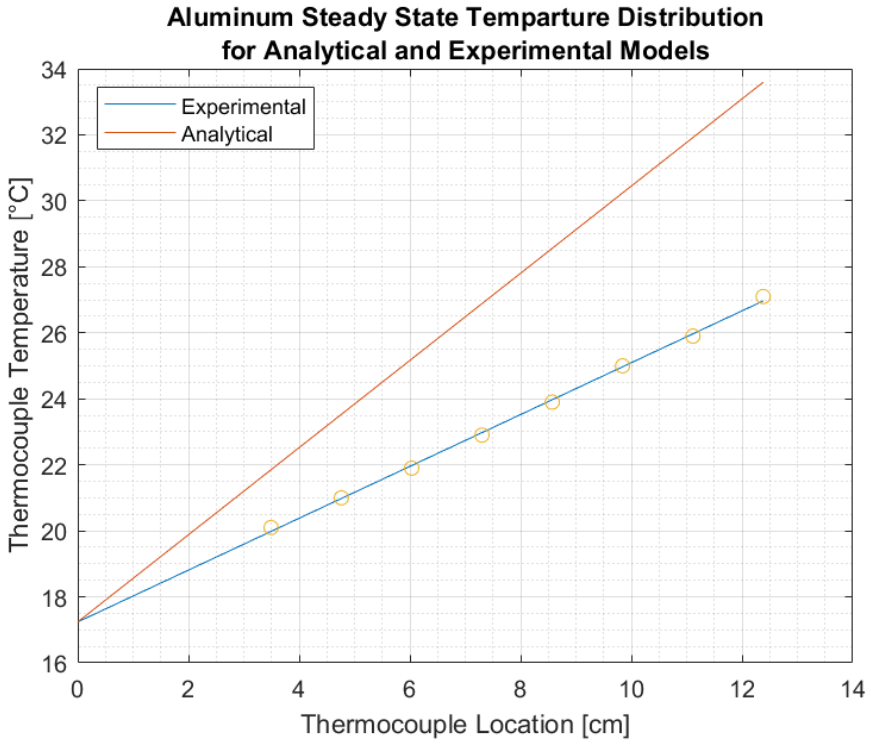
Aluminum Steady State Temparture Distribution
for Analytical and Experimental Models
Experimental
Analytical
Thermocouple Temperature [°C]
Thermocouple Location [cm]
34
32
30
28
26
24
22
20
18
16
0
2
4
6
8
10
12
14

Fig. 5

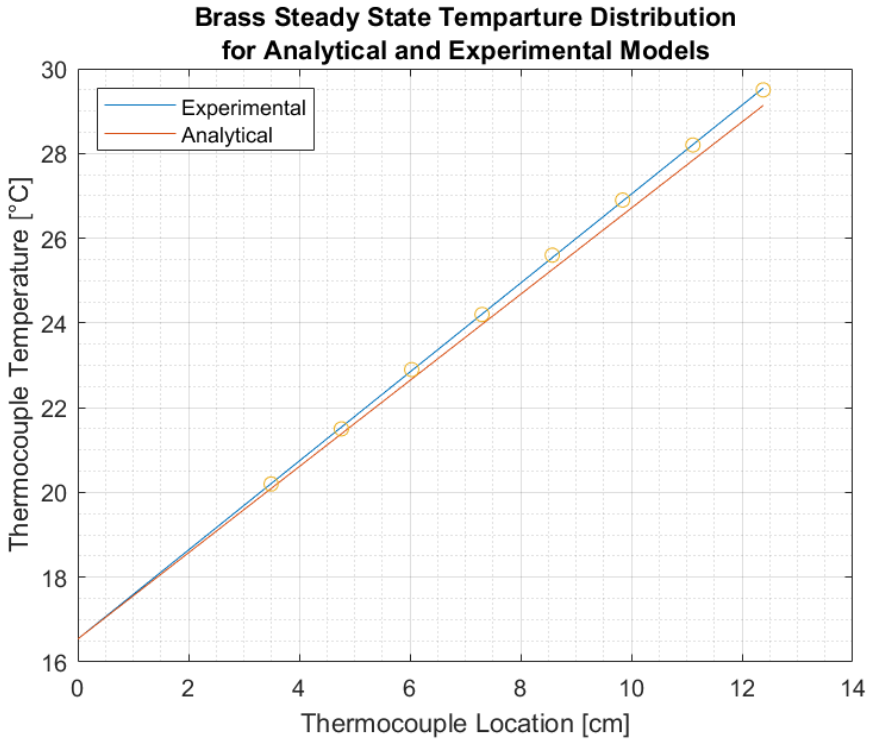
Brass Steady State Temparture Distribution
for Analytical and Experimental Models
Experimental
Analytical
Thermocouple Temperature [°C]
Thermocouple Location [cm]
30
28
26
24
22
20
18
16
0
2
4
6
8
10
12
14

Fig. 6

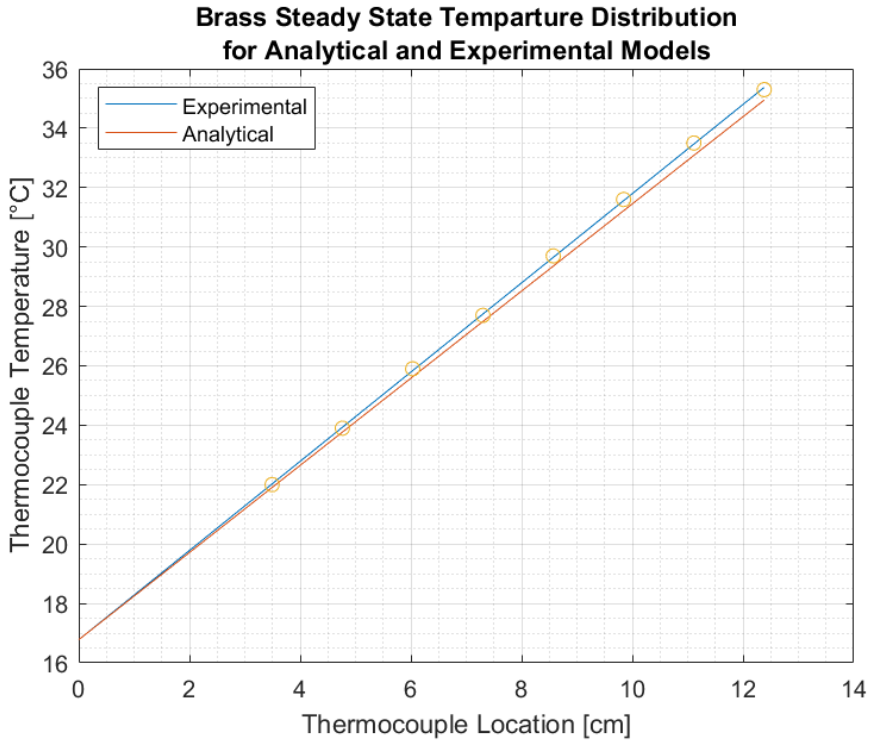
Brass Steady State Temparture Distribution
for Analytical and Experimental Models
Experimental
Analytical
Thermocouple Temperature [°C]
Thermocouple Location [cm]
36
34
32
30
28
26
24
22
20
18
16
0
2
4
6
8
10
12
14

Fig. 7

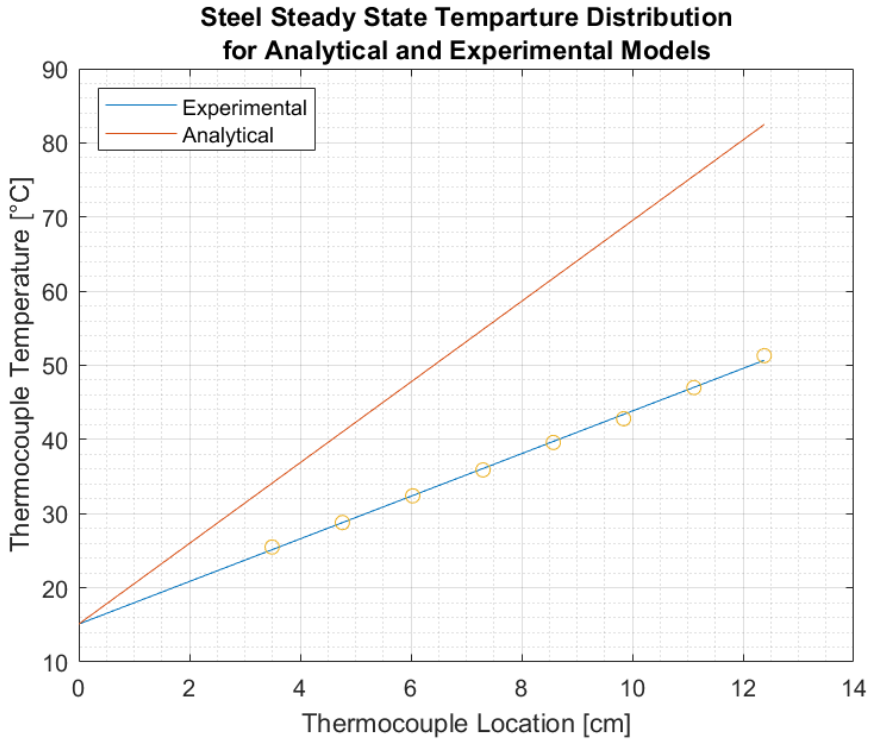
Steel Steady State Temparture Distribution
for Analytical and Experimental Models
Experimental
Analytical
Thermocouple Temperature [°C]
Thermocouple Location [cm]
90
80
70
60
50
40
30
20
10
0
2
4
6
8
10
12
14

Fig. 8

Figure 4 through **Figure 8** show the analytical and experiment steady state temperature distributions for all 5 test cases. The Aluminum and Steel analytical slopes vary from their experimental counterparts by a pretty large degree, while the Brass is fairly accurate. **Table 1** below has all of the slopes, as well as the temperature at the beginning of the bar (T_0). The analytical slope is determined from the equation $H_{an} = \frac{\dot{Q}}{kA}$ where $\dot{Q} = IV$ and the experimental slopes were determined by using MATLAB's polyfit function on the experimental data, which also gives T_0.

Table 1 Steady State Analytical Slope, Experimental Slope, and Starting Temperature

Test Case	H_{an} $[^oC/m]$	H_{exp} $[^oC/m]$	T_0 $[^oC]$
Aluminum at 25V 240mA	91.0858	52.2741	16.5914
Aluminum at 30V 290mA	132.0744	78.5527	17.2399
Brass at 25V 237mA	101.6795	104.9869	16.5417
Brass at 30V 285mA	146.7273	150.1687	16.7804
Steel at 22V 203mA	544.0595	287.3078	15.1074

B. Time Dependent Temperature Profiles

1. Model IA

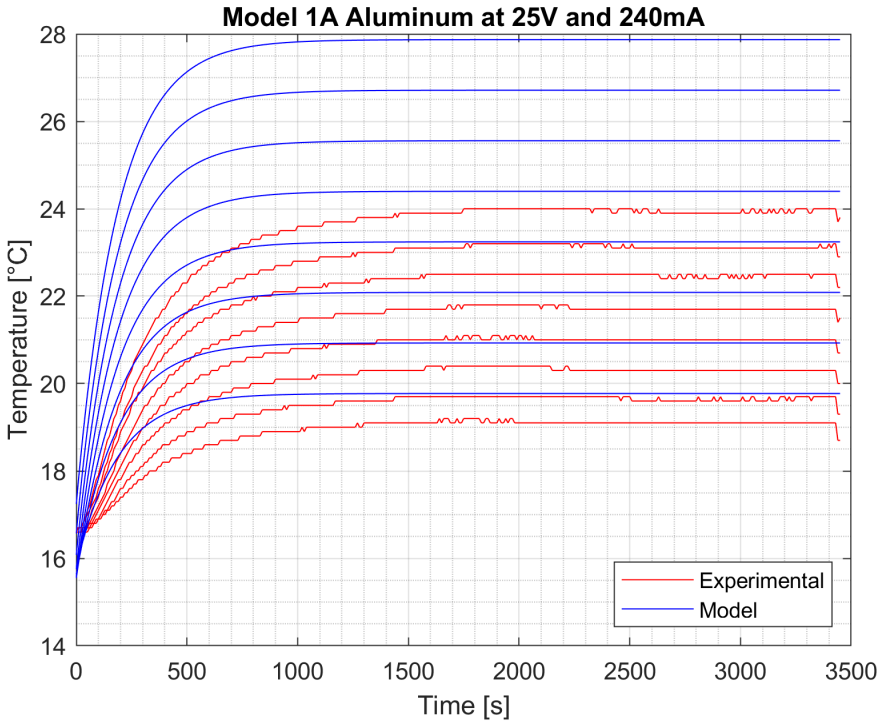

Fig. 9

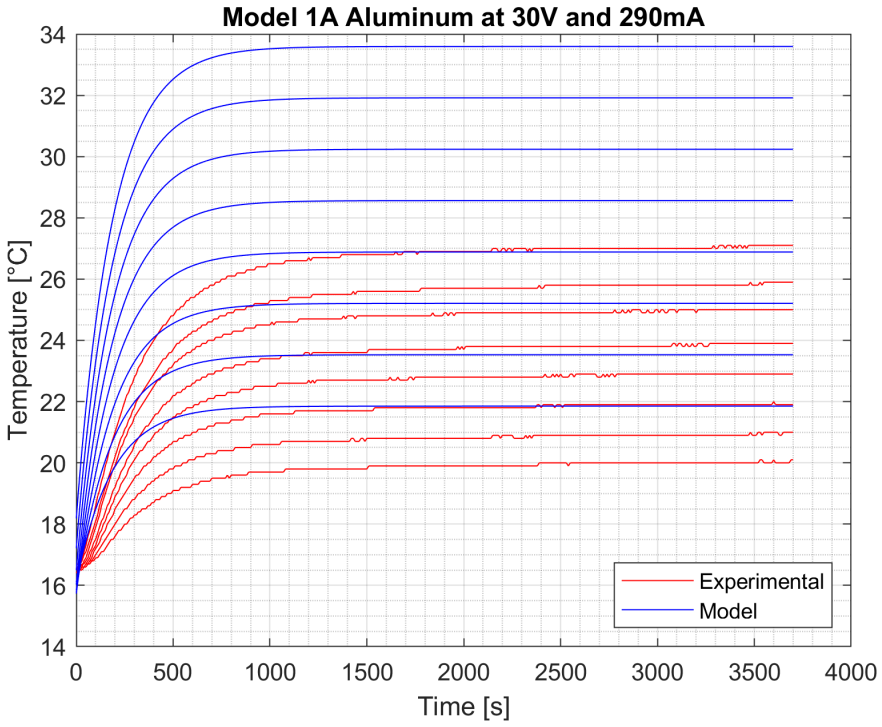
Model 1A Aluminum at 30V and 290mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 10

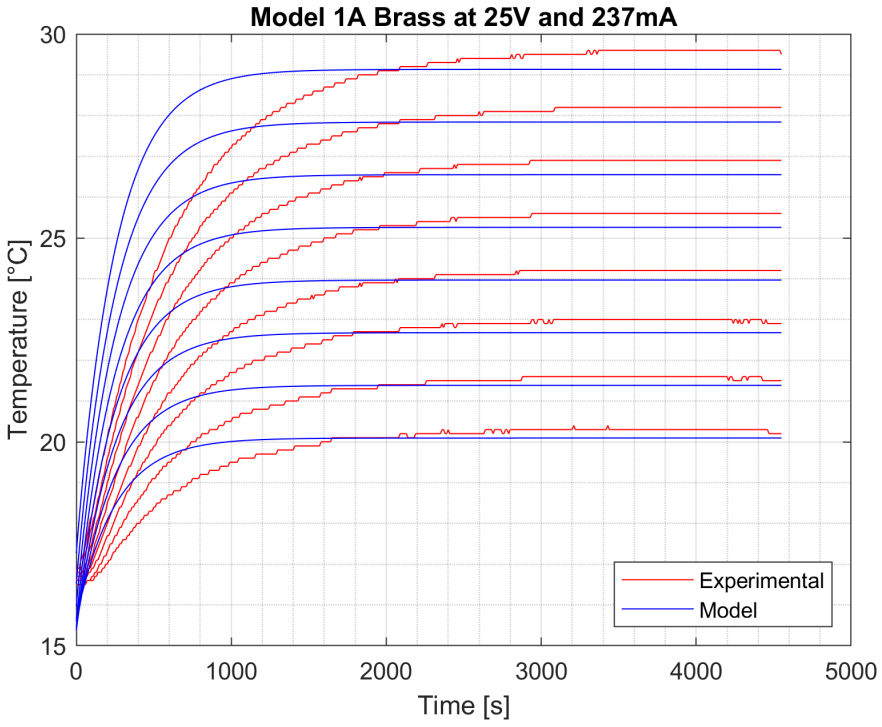
Model 1A Brass at 25V and 237mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 11

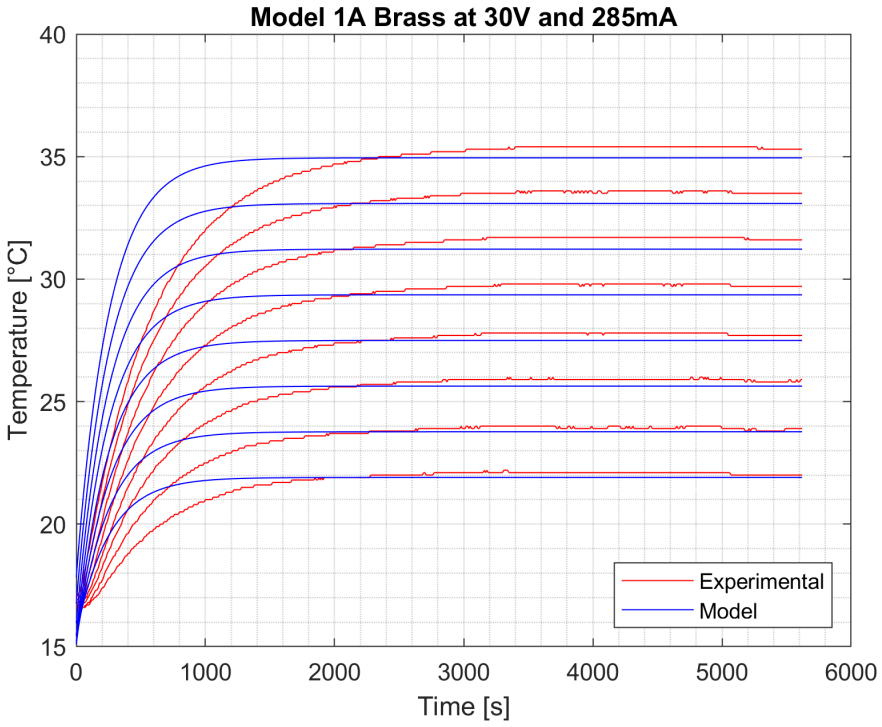
Model 1A Brass at 30V and 285mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 12

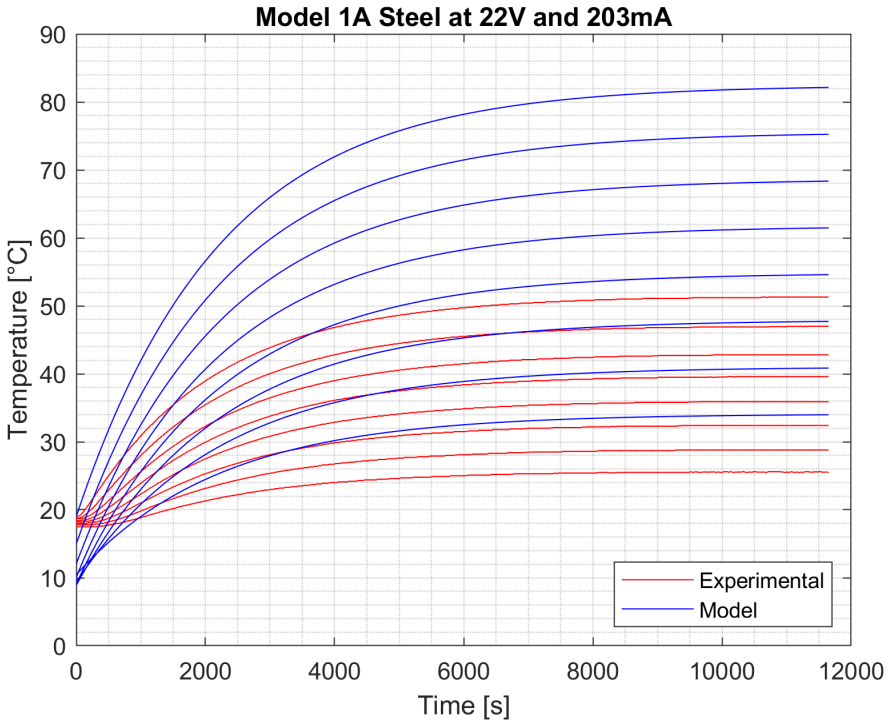
Model 1A Steel at 22V and 203mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 13

Figure 9 through **Figure 13** show the results for Model IA, which uses the analytical slope to plot the time dependent profiles of the thermocouple reading. Following the trend that the analytical slopes vary from the experiment slopes by quite a large degree, it can be seen that the temperature profiles for the experimental data are quite different from the analytical ones. Model IB was used to correct for this discrepancy.

2. Model IB

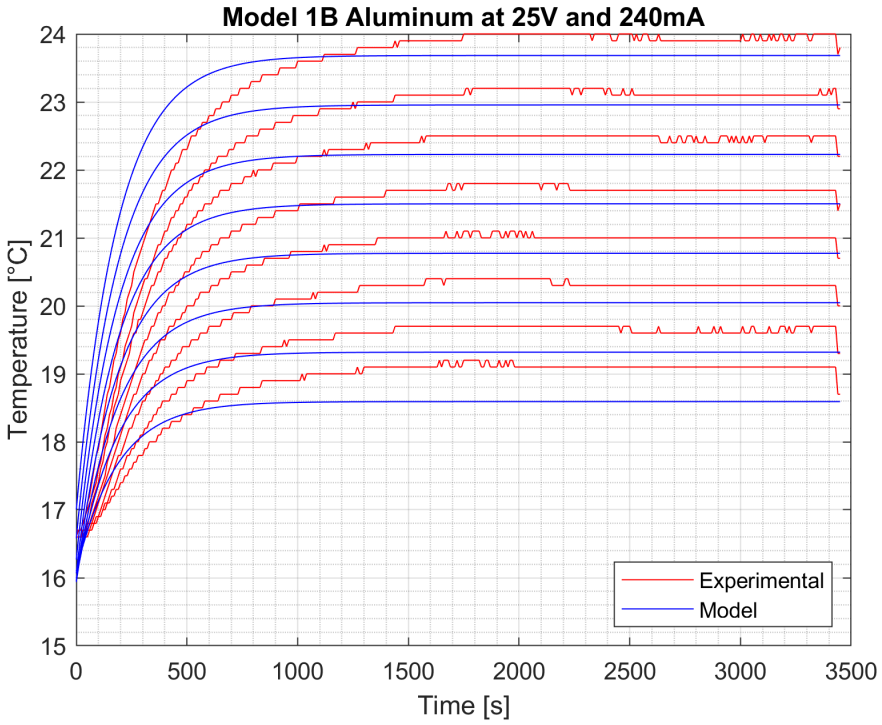

Fig. 14

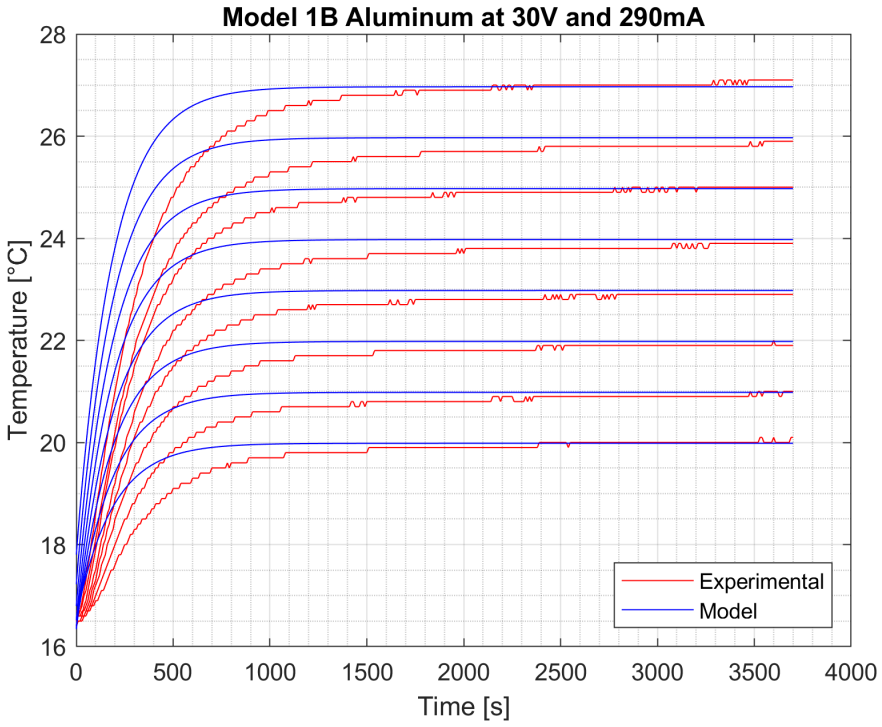
Model 1B Aluminum at 30V and 290mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 15

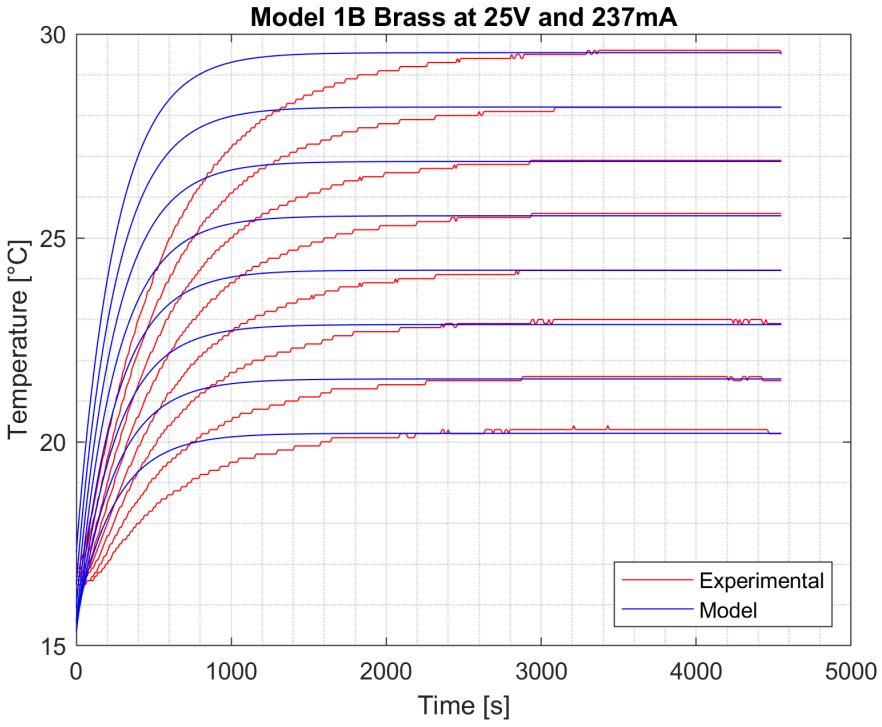
Model 1B Brass at 25V and 237mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 16

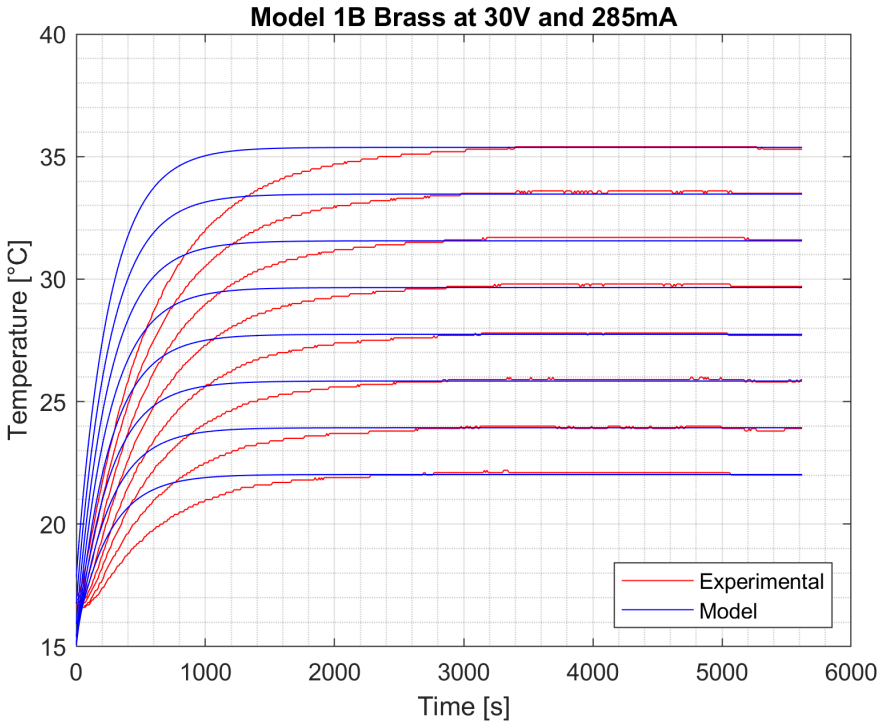
Model 1B Brass at 30V and 285mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 17

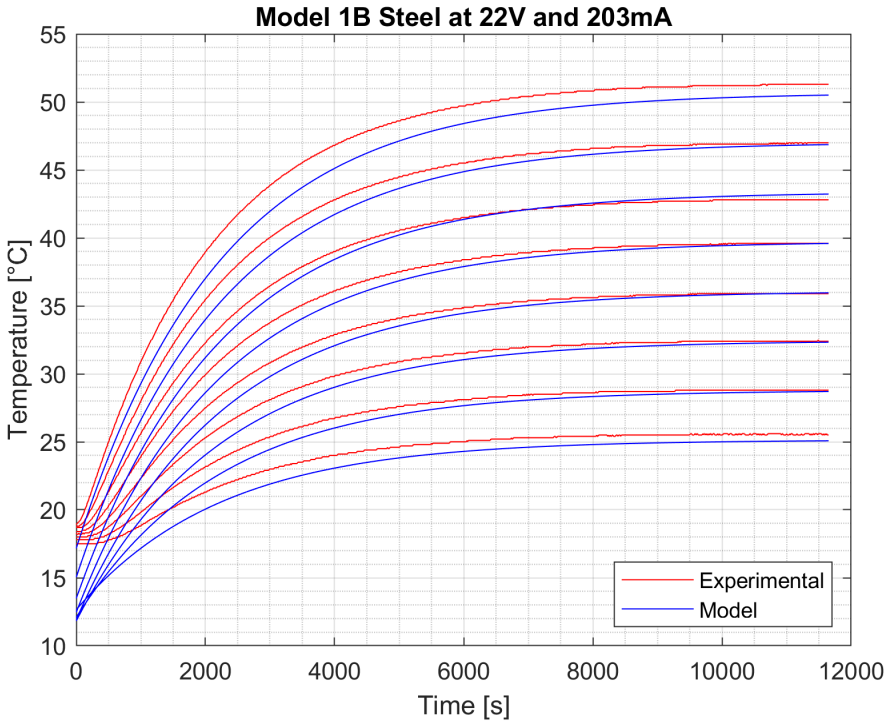
Model 1B Steel at 22V and 203mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 18

Figure 14 through **Figure 18** show the time dependent profile results for Model IB. The only difference between the code of the two models is that Model IA uses the analytical slopes while Model IB uses the experimental slopes. This correction sees more accurate results for all test cases.

C. Initial State Distributions

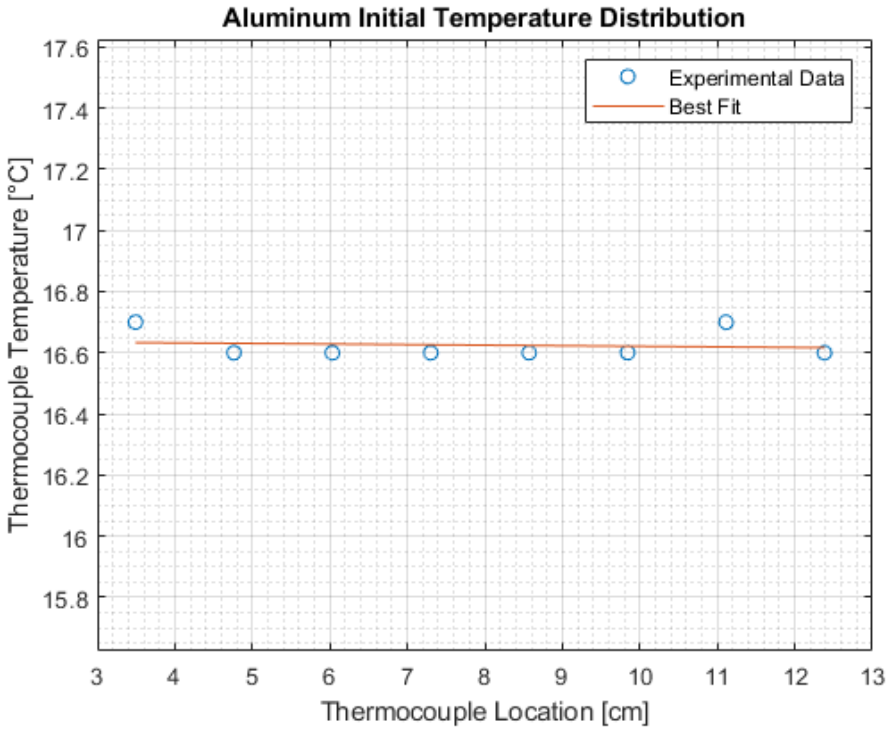

Fig. 19

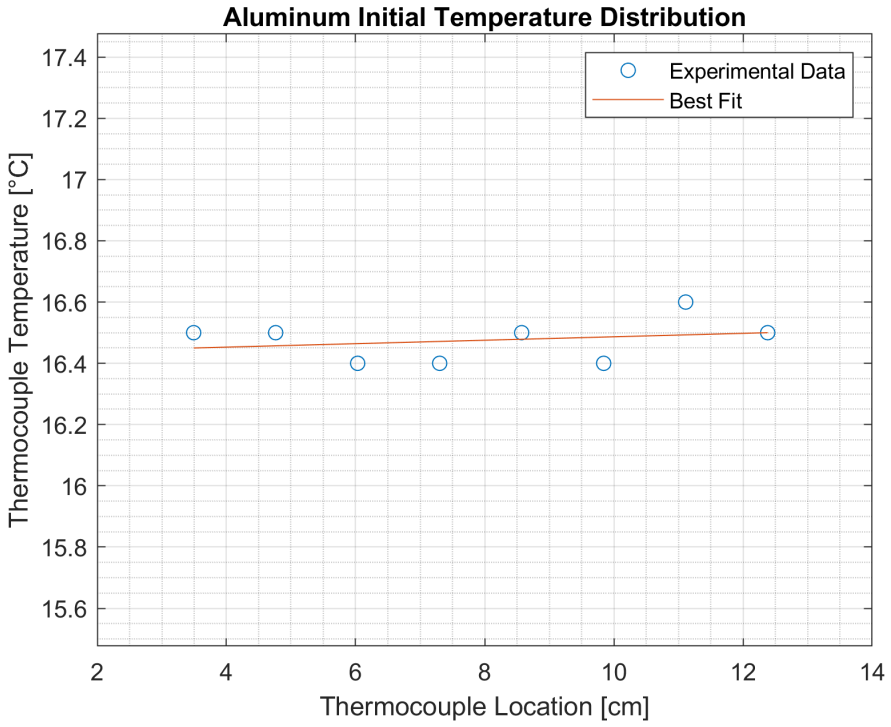
Aluminum Initial Temperature Distribution
Experimental Data
Best Fit
Thermocouple Temperature [°C]
Thermocouple Location [cm]
17.4
17.2
17
16.8
16.6
16.4
16.2
16
15.8
15.6
2
4
6
8
10
12
14

Fig. 20

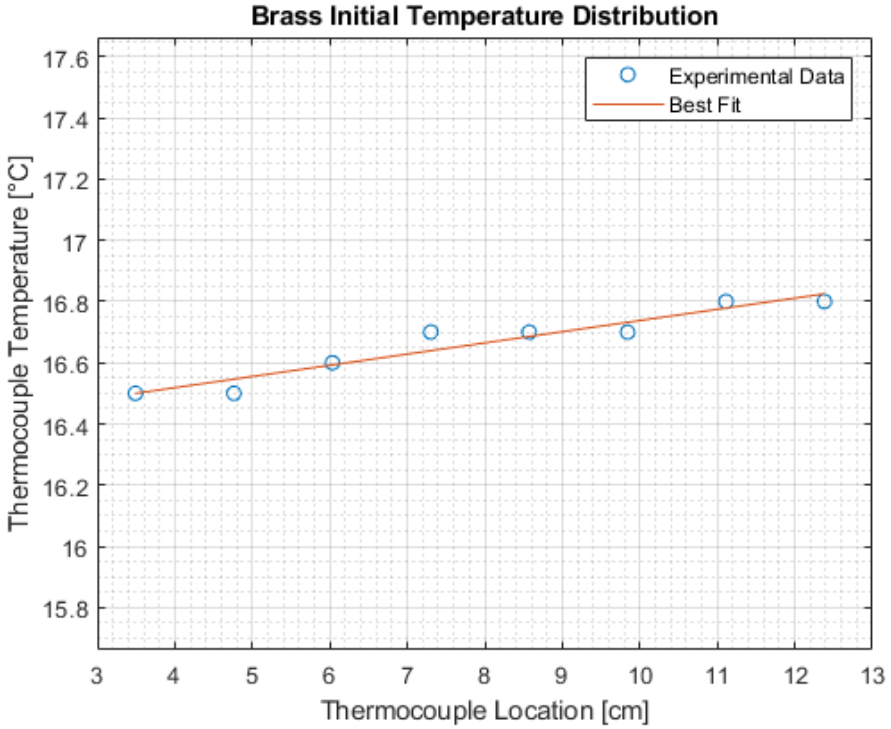
Brass Initial Temperature Distribution
Experimental Data
Best Fit
Thermocouple Temperature [°C]
Thermocouple Location [cm]
17.6
17.4
17.2
17
16.8
16.6
16.4
16.2
16
15.8
3
4
5
6
7
8
9
10
11
12
13

Fig. 21

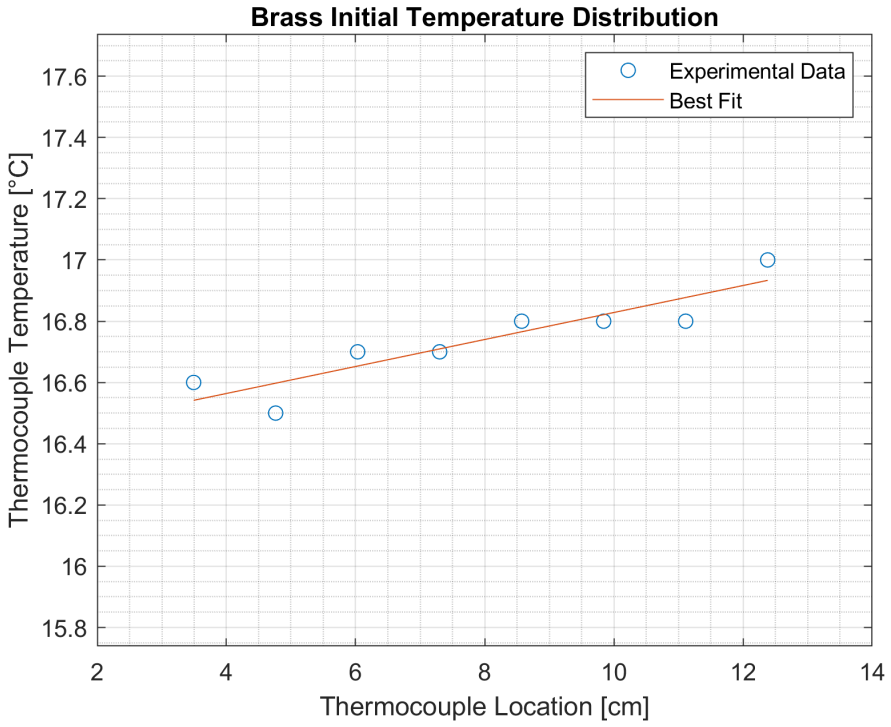
Brass Initial Temperature Distribution
Experimental Data
Best Fit
Thermocouple Temperature [°C]
Thermocouple Location [cm]
17.6
17.4
17.2
17
16.8
16.6
16.4
16.2
16
15.8
2
4
6
8
10
12
14

Fig. 22

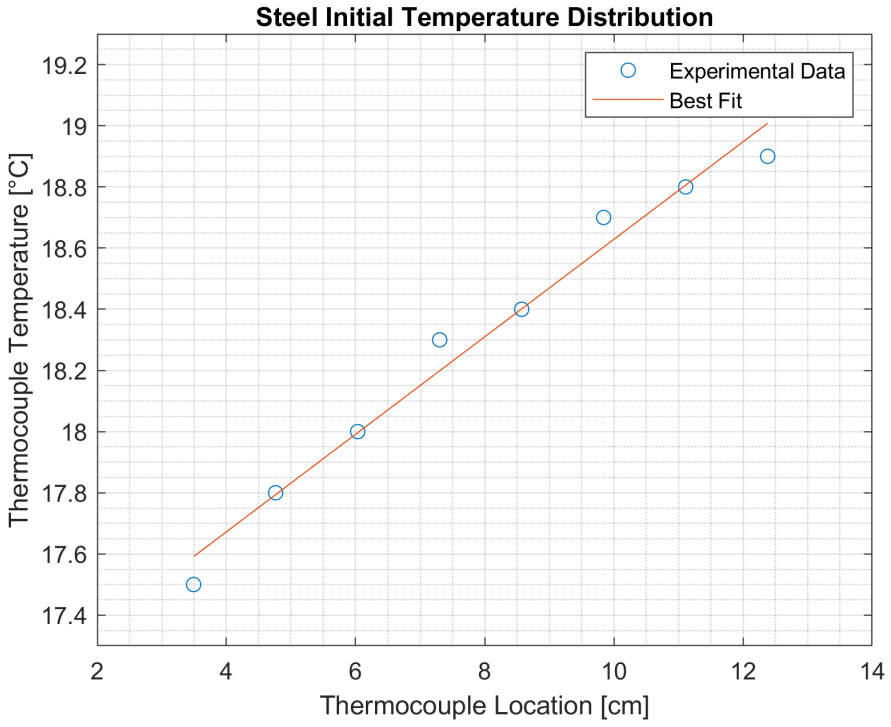
Steel Initial Temperature Distribution
Experimental Data
Best Fit
Thermocouple Temperature [°C]
Thermocouple Location [cm]
19.2
19
18.8
18.6
18.4
18.2
18
17.8
17.6
17.4
2
4
6
8
10
12
14

Fig. 23

There was an assumption that was made in the original derivation of the solution to the one dimensional heat equation that states that the initial temperature for the whole rod was equal to the temperature on the cool end of the rod. While this assumption is mostly valid for Aluminum and possibly even Brass, it is not the case for Steel. This can be seen in **Figure 19** through **Figure 23**. In order to correct for this an additional model was developed to account for the initial temperature distributions. Returning to the derivation from the second initial question in the lab and using the same result for the the integral in that equation, the value for b_n can be corrected. Instead of the integral being multiplied by $\frac{-2H}{L}$, it is now multiplied by $\frac{2(M-H)}{L}$, where M is the slope of the initial temperature distribution. This now makes the solution to the one dimensional heat **Equation 7**:

$$u(x,t) = T_0 + H_x + \sum_{n=1}^{\infty} \frac{-8(M-H)L(-1)^n}{\pi^2(2n-1)^2} \sin\left[\frac{(2n-1)\pi}{2L}x\right] e^{\left[-\frac{(2n-1)\pi}{2L}^2 \alpha t\right]} \tag{7}$$

The results for the initial temperature distribution for each test case are displayed in **Table 2** below.

Table 2 Initial Temperature Experimental Slope

Test Case	M_{exp} $[^oC/m]$
Aluminum at 25V 240mA	-0.1875
Aluminum at 30V 290mA	0.5624
Brass at 25V 237mA	3.6558
Brass at 30V 285mA	4.4057
Steel at 22V 203mA	15.9355

1. Model II

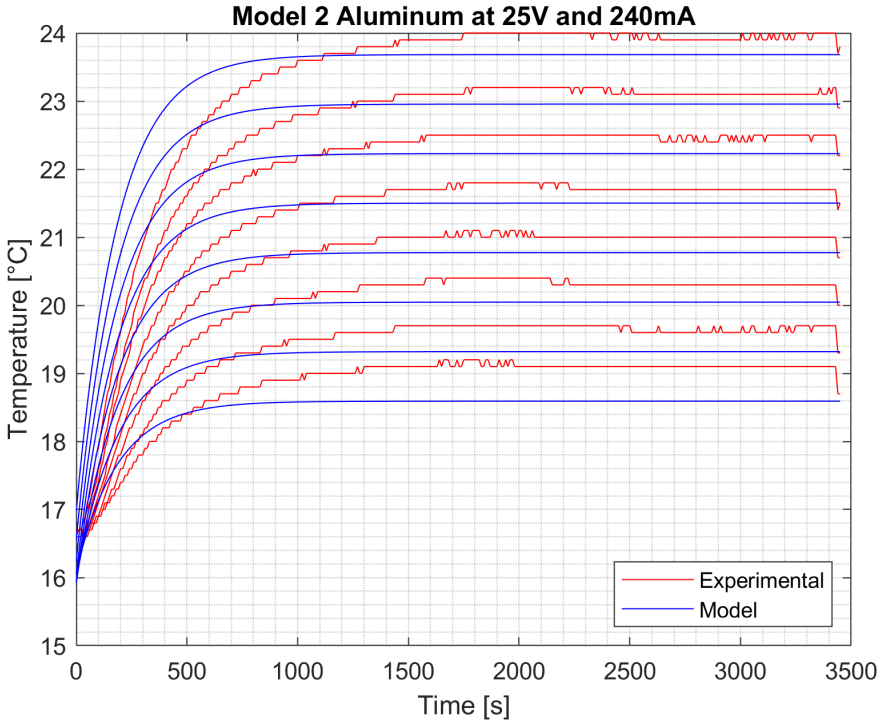

Fig. 24

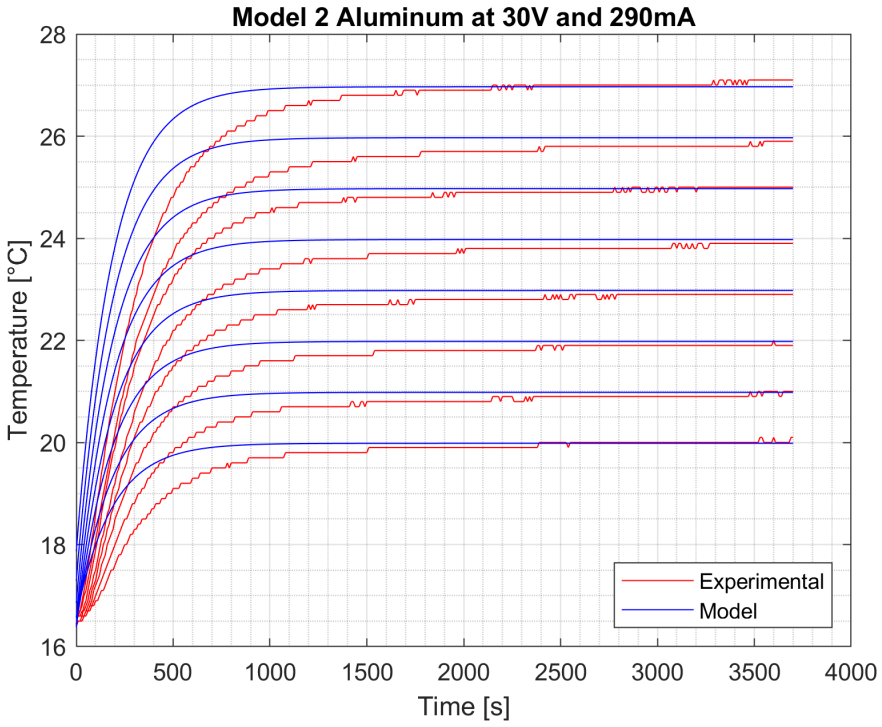
Model 2 Aluminum at 30V and 290mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 25

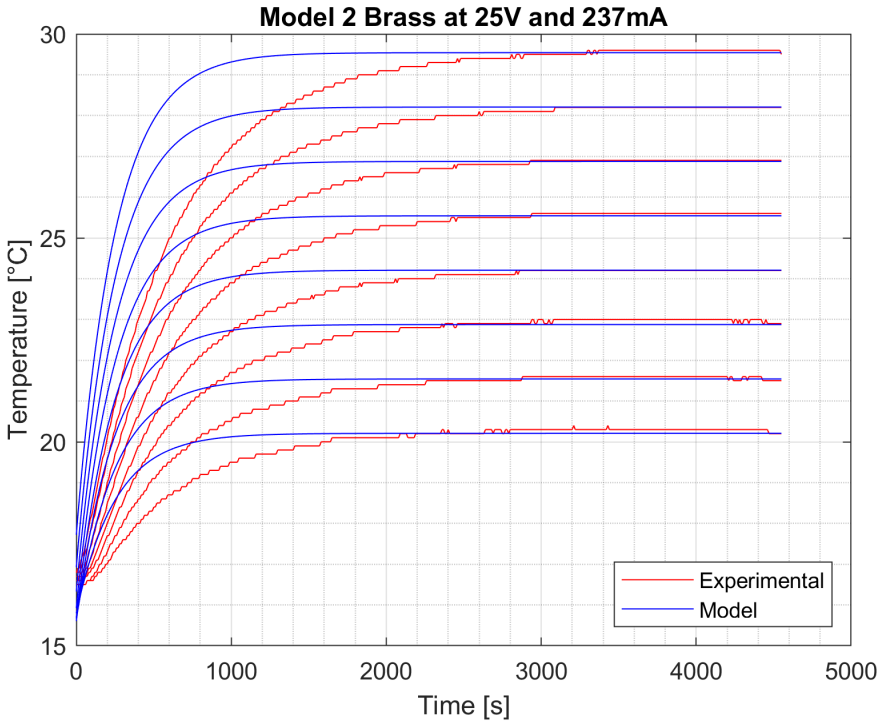
Model 2 Brass at 25V and 237mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 26

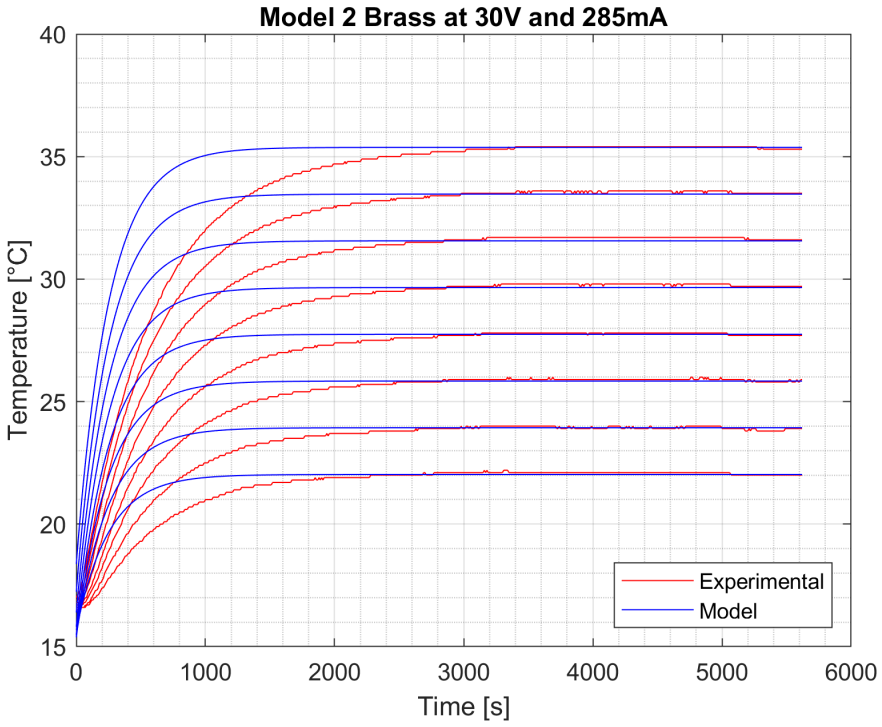
Model 2 Brass at 30V and 285mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 27

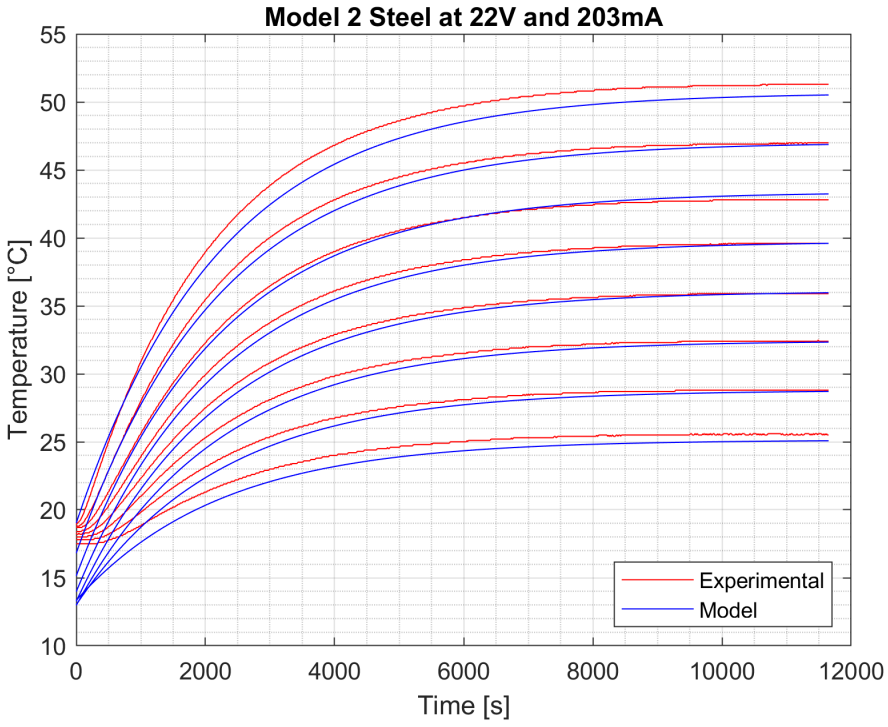
Model 2 Steel at 22V and 203mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 28

The new solution to the one dimensional heat equation was implemented into a new model, Model II. **Figure 24** through **Figure 28** show the new time dependent temperature profiles with the new solution. The results from this model look identical to the results from Model IB, indicating that the assumption that the initial temperature of the entire rod is constant is a good assumption for developing the model, despite it not being entirely true.

D. Variance in Thermal Diffusivities

1. Model III

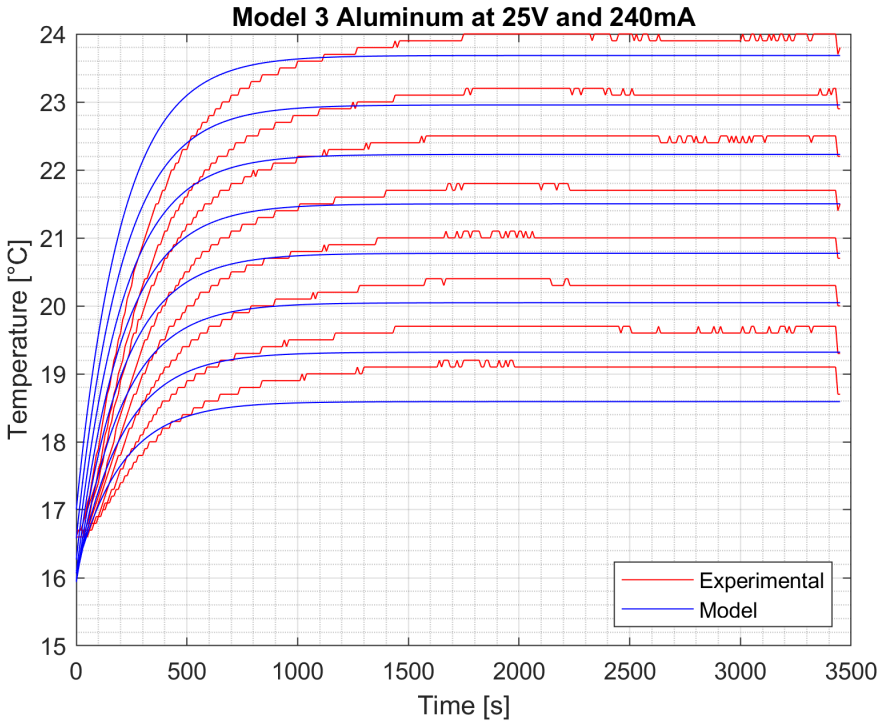

Fig. 29

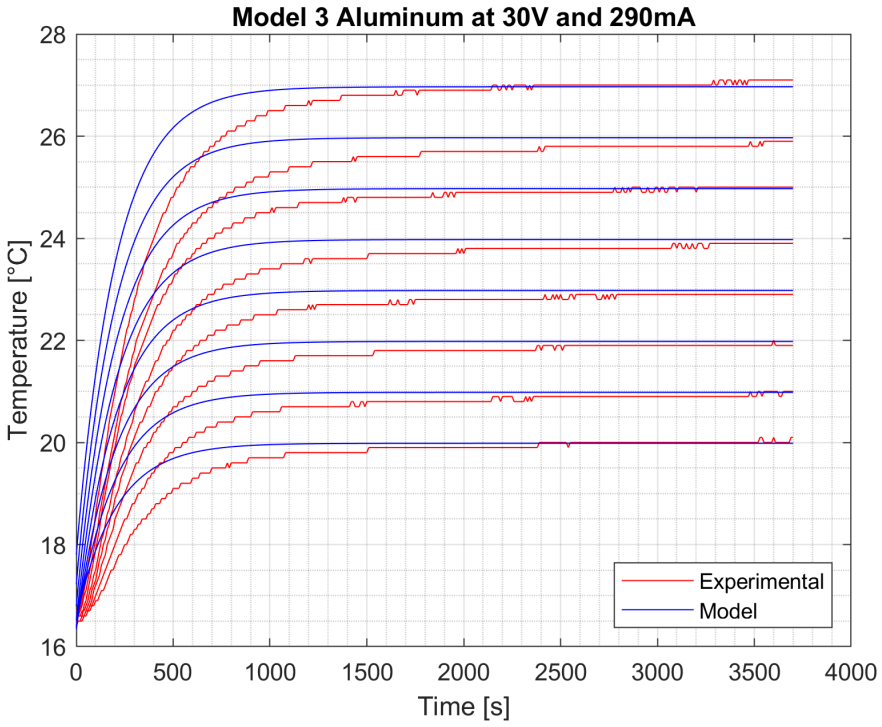
Model 3 Aluminum at 30V and 290mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 30

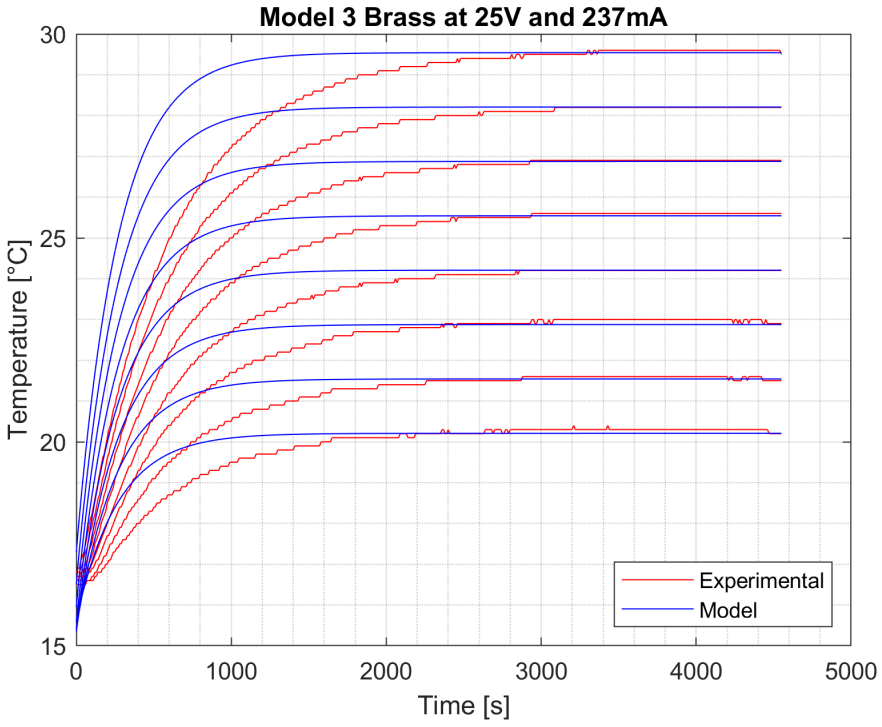
Model 3 Brass at 25V and 237mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 31

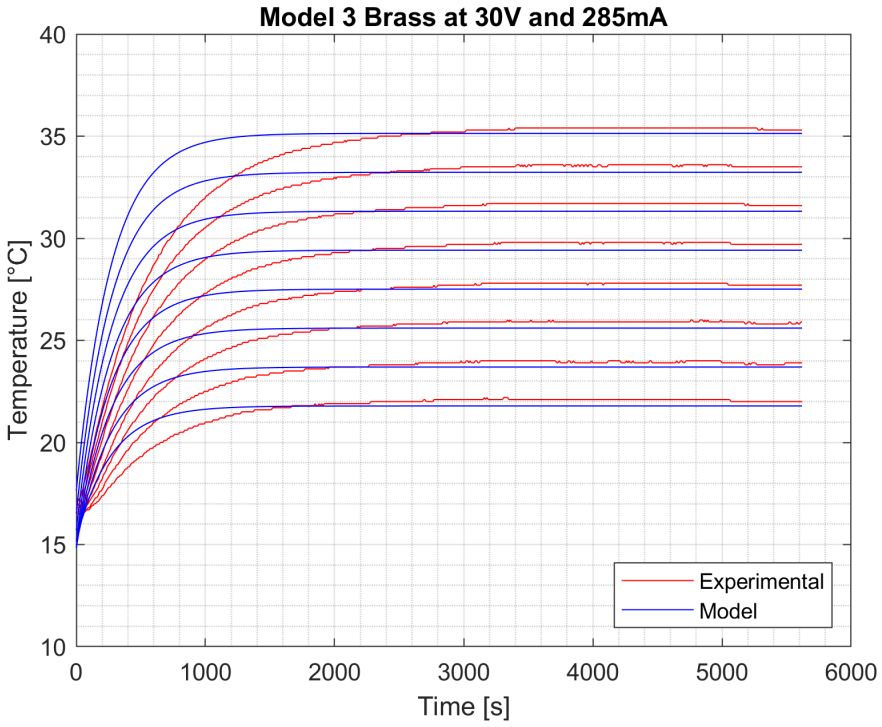
Model 3 Brass at 30V and 285mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 32

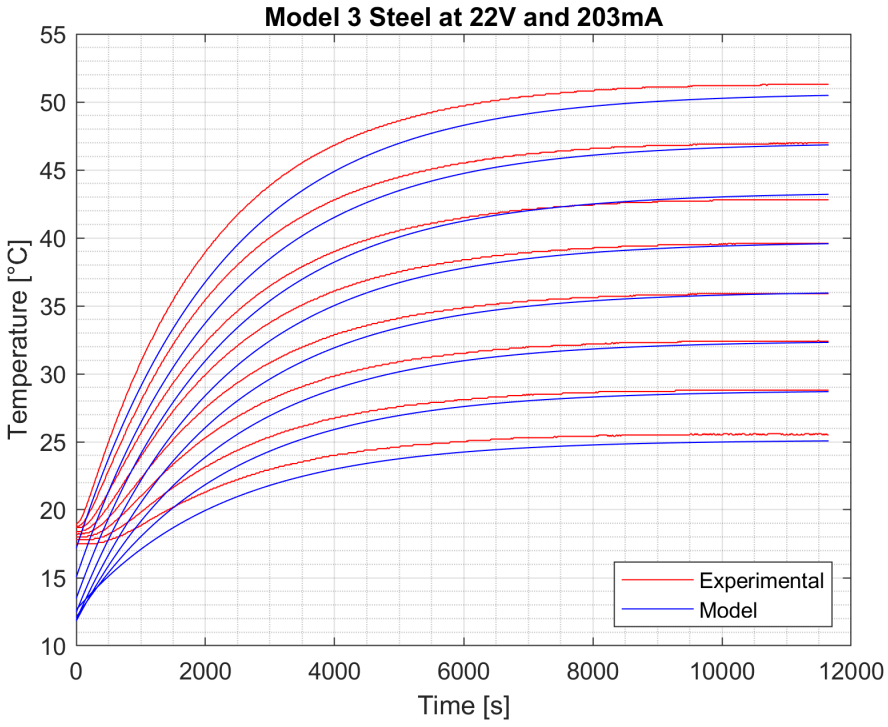
Model 3 Steel at 22V and 203mA
Temperature [°C]
Time [s]
Experimental
Model

Fig. 33

The final model that was developed was one that accounts for imperfections in the manufacturing process of the metal rods. These imperfections would alter the true thermal diffusivties of the metal rods, so Model III implements a different thermal diffusivity values for each metal and the results can be seen in **Figure 29** through **Figure 33** as well as the new thermal diffusivity values used in **Table 3** below. There will be further discussion on how these thermal diffusivity values were determined in a later section of the report.

Table 3 Original and Adjusted Thermal Diffusivities

Material	$\alpha\ [m^2/s]$	$\alpha_{adj}\ [m^2/s]$
Aluminum	4.8191×10^{-5}	4.3950×10^{-5}
Brass	3.5604×10^{-5}	3.3325×10^{-5}
Steel	4.0500×10^{-6}	3.9650×10^{-6}

E. Time to Steady State

Table 4 Time to Steady State and Corresponding Fourier Number

Material	$t_{ss}\ [s]$	Fo
Aluminum	1020	2.0132
Brass	1340	2.0054
Steel	11240	2.0013

Table 4 shows the time to reach steady state for each material and the Fourier number with corresponding that time and the new thermal diffusivity value used.

IV. Discussion

A. Compare Experimental vs. Analytical Solutions

The experimental steady-state slopes were calculated using MATLAB's polyfit function, while the analytically derived steady-state slopes were calculated from the heater's voltage, current draw, the bars' cross-sectional areas, and thermal conductivity. The experimental and analytical steady-state distributions were plotted against one another and the end of the time span on the same plots (**Figure 4** through **Figure 8**), and all of the steady-state slopes discussed below are sourced from these plots and were placed in **Table 1**.

For the Aluminum bar, it was found that the experimentally calculated steady-state slope was nearly half that of the analytically derived steady-state slope. At the 25V 240mA test the experimental steady-state slope, H_{exp}, was 52.2741 $^oC/m$ while the analytically derived steady-state slope, H_{an}, was 91.0858 $^oC/m$. Similar results were obtained for the test at 30V and 290mA, which produced values of H_{exp} = 78.5527 and H_{an} = 132.074 $^oC/m$, respectively. Brass had different results, having nearly equal if slightly lower, analytical slopes than the experimental values were calculated to be. For Brass at 25V and 237mA the experimental and analytical slopes are H_{exp} = 104.987 and H_{an} = 101.680 $^oC/m$. At 30V and 285mA the experimental slope, H_{exp} = 146.727, and analytical slope, H_{an} = 150.169 $^oC/m$, continue to follow this relationship. The Steel test at 22V and 203mA returns to nearly the same relationship between H_{exp} and H_{an} as Aluminum had, with values experimental slope being nearly half the analytical slope, at 287.308 and 544.060 $^oC/m$.

This leads to the conclusion that a certain combination of corresponding thermal conductivity and density determines the accuracy of this sort of modeling. Brass has both a high thermal conductivity and density and is the most accurate. Aluminum has a high conductivity and low density, while Steel has a low conductivity and high density, and both greatly overshoot the experimental data. Additionally, there are likely defects in the materials that result in further errors being introduced into all of the experimental data, making models less applicable as a whole.

B. Justify Initial Temperature

During the first part of the development of the model, it was assumed that the initial temperature across the whole of the test bars was constant. Initially, it is believed this was a reasonable assumption because the bars were held at a constant temperature by the water chiller, at 10 ^oC. However, reviewing the results from models IA (**Figure 9** through **Figure 13**) and IB (**Figure 14** through **Figure 18**), it is concluded that this assumption was invalid. This is because despite adjusting the steady-state slope from the analytical solution to the experimental solution, it is shown that there were still large divergences in the model when compared to the experimental results. Upon plotting the initial temperature states recorded by the thermocouples in the experimental data, it is found that for Aluminum and Brass tests, there were very small trends in the data across the thermocouple locations on the bars, while for the Steel there was a significant increase in temperature across the bar. These trends are shown by a line of best fit in **Figure 19** through **Figure 23**. From these lines of best fit, the initial state temperature distribution was determined and it was decided that the models would have to be adjusted, which is shown in 2. For the Aluminum tests the slopes were determined to be of $-0.1875\ ^oC/m$ for the test conducted at 25V and 240mA, while at 30V and 290mA, the slope was $0.5624\ ^oC/m$. Over the length of the bar with thermocouples on it, this worked out to be between a $-0.0166\ ^oC$ and $0.05\ ^oC$ temperature δ across each bar, respectively. For the Brass test at 25V and 237mA, the slope worked out to be much higher than either Aluminum test at $3.6558\ ^oC/m$ and the second test at 30V 285mA was similarly even higher at $4.4057\ ^oC/m$. The crossbar temperature delta for each case worked out to be $0.3250\ ^oC$ for the first test and $0.3916\ ^oC$ for the second. For both the Aluminum and Brass tests the temperature δs are small enough to fall within the thermocouple error of $\pm 2\ ^oC$. For Steel, a much larger slope of $15.9355\ ^oC/m$ and a correspondingly much higher temperature δ across the bar for $19.0083\ ^oC$.

The transient phase between the initial and steady-state temperatures is shifted by these temperature δs which results in additional error for the Aluminum and Brass tests and a large amount of additional error for the Steel test. Once the model was updated the Steel stayed within the $\pm 1\ ^oC$ range across each thermocouple for the entire time span.

The changes in Model II to use these initial temperature slope values in **Equation 7** resulted in modeled values that were much closer to the experimental data, and can be seen in **Figure 24** through **Figure 28**. This attempt to match the initial slope of the experimental data with the model largely did not impact the steady-state values, except in the case of the first Aluminum model, where the model undershoots the experimental data in a change from Model IB where it was closer to the experimental data. This could be attributed to the error in thermocouple measurements, which are only accurate to $\pm 2\ ^oC$, or to the model not being accurate for heat transfer through Aluminum at lower heater outputs.

C. Discuss

1. Thermal Diffusivity Methods

In order to vary the thermal diffusivity, the 8th thermocouple and Fourier numbers during Model IB were examined. The 8th thermocouple was chosen because it was the last thermocouple to reach the steady state. In order to get the new thermal diffusivity, the data for each test case using the 8th thermocouple was run through the code used to generate Model IB using a range of alpha for each corresponding material from the range $\frac{\alpha}{1000} \leq \alpha_{adj} \leq 1.5\alpha$ using a spacing of $\frac{\alpha}{1000}$. The final temperature for each heat equation solution was compared to the steady-state temperature results from Model IB, and the index location of when the two temperatures were equal to one another was saved. That index location was used in the range of tested thermal diffusivities to find the newly adjusted thermal diffusivity. The adjusted thermal diffusivities are relatively close to the calculated values with the given material properties, and are slightly lower, which can be seen in **Table 3**. This is what would be expected if there were impurities in the manufacturing process, which is what was attempting to be modeled in Model III.

2. Fourier Number

The Fourier number indicates if enough time has passed in order for the other side of the rod to reach a steady state. For this specific experiment, once $Fo \geq 2$, the test was considered to reach a steady state. For the Aluminum and Brass, the Fourier number increases rather quickly, indicating that the time to reach a steady state is relatively quick. Both Aluminum cases reach $Fo \geq 2$ at around $1020s$, both Brass cases reach $Fo \geq 2$ at $1340s$ For Steel, the Fourier number increases at a much slower rate, indicating the time to reach steady-state is much longer in comparison. The Steel case reaches $Fo \geq 2$ at $11240s$. The conclusions drawn from the significance of the Fourier number are supported by the data and the Fourier Numbers and t_{ss} and all displayed in **Table 4**.

3. Time to Steady-State

Using the Fourier Number calculated earlier, the equation $Fo = \frac{\alpha\, t_{ss}}{L^2}$ was used, where α is the thermal diffusivity of each material and L is assumed to be 5.875 inches. The corresponding α and To were used for each material in order to find the steady-state time in seconds. Looking at the results, it can be seen that Aluminum has the lowest time to steady-state and Brass is in the middle of the two materials. Steel has the largest steady-state time by far, being over ten times greater than Aluminum t_{ss}. This can be seen visually by looking at **Figure 33** in comparison to the figures for both Aluminum and Brass. This makes sense since it is known that α for Steel is much lower in comparison to α for both Aluminum and Brass, meaning that it will take longer for thermal energy to diffuse through the rod due to the lower thermal diffusivity.

If it is assumed that $\frac{\alpha\, t_{ss}}{L^2}$ is kept constant, changing the length of the rod and the thermal diffusivity will have different effects on the value for t_{ss}. Increasing the length of the bar will increase t_{ss} since there is more physical mass for the heat to transfer into, requiring more time to reach a steady state. Decreasing the length will have the inverse effect, decreasing t_{ss}. Increasing the value for α will decrease t_{ss}, and decreasing α will increase t_{ss}. This is because α affects how easy or hard it is for energy to distribute itself through the mass. Making it "easier" to transfer heat will make the heat transfer faster, decreasing t_{ss}.

D. Hypothesize

For Aluminum, Model III appears to be more accurate across the transient solution than Model II. It could be argued that Model II is equivalent to Model III for the steady-state conditions. It is believed Model III is the best for the transient solution since Model III alters α, which affects the exponential decay of of the transient solution, as well as the fact that the transient solution for Aluminum is the most inaccurate of the two solutions. These conclusions are supported by **Figure 34** and **Figure 35**, as it can be seen that the transient solution is outside of the error of the thermocouples.

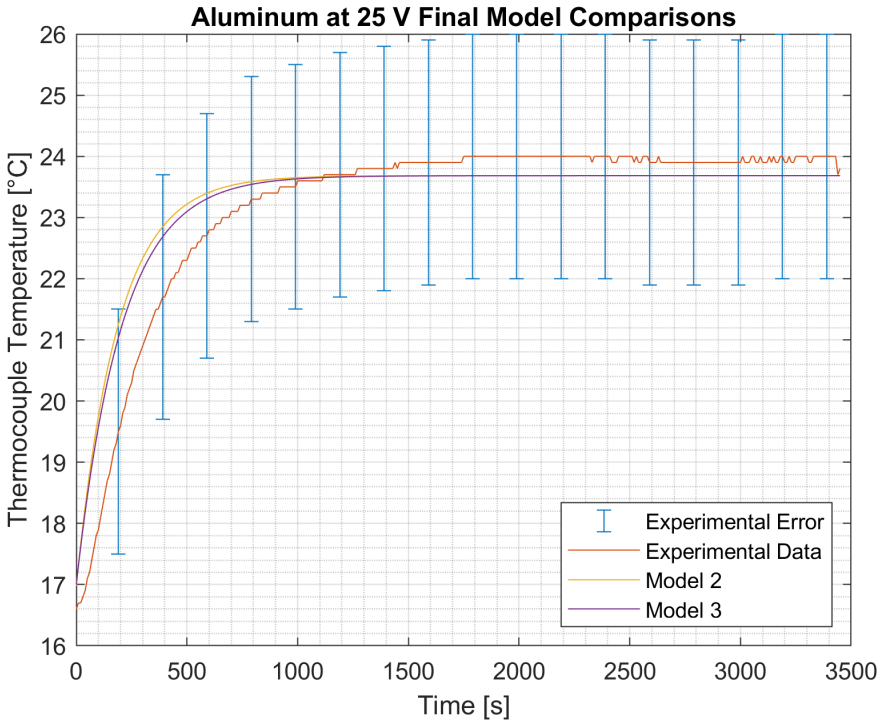

Fig. 34

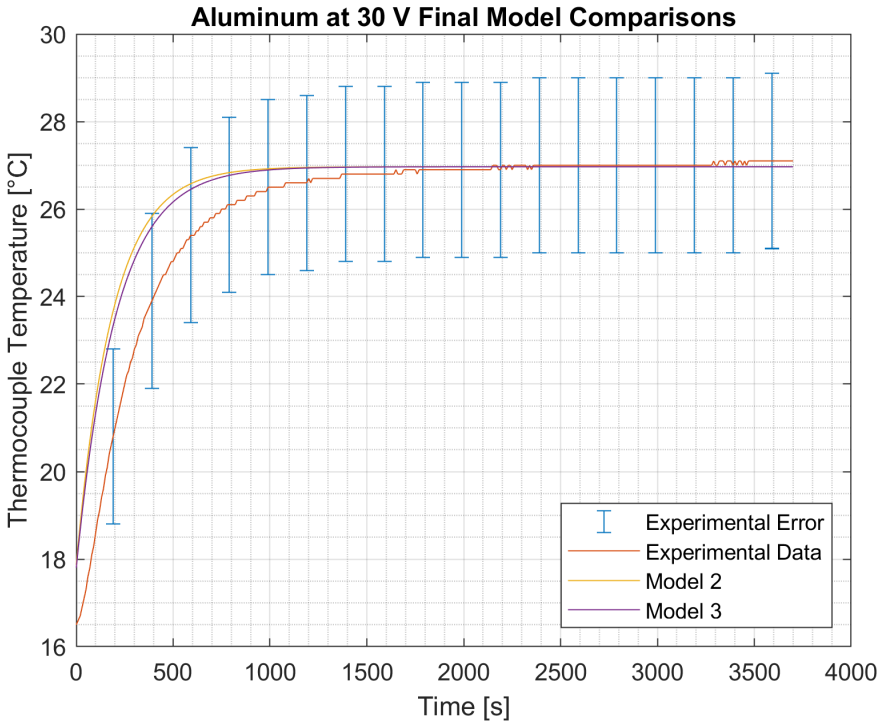

Fig. 35

For Brass, Model III is more accurate than Model II for the transient solution. Once again, it could be argued that Model II is equivalent to the steady-state condition for both models. The reasoning for why Model III is better for Brass is the same reasoning as it is for Aluminum; the transient solution for Brass is the most inaccurate, so addressing it by altering the thermal diffusivity of the material will have the greatest impact. These conclusions are also supported by **Figure 36** and **Figure 37**, which show the Brass transient solution is also outside of the error of the thermocouples.

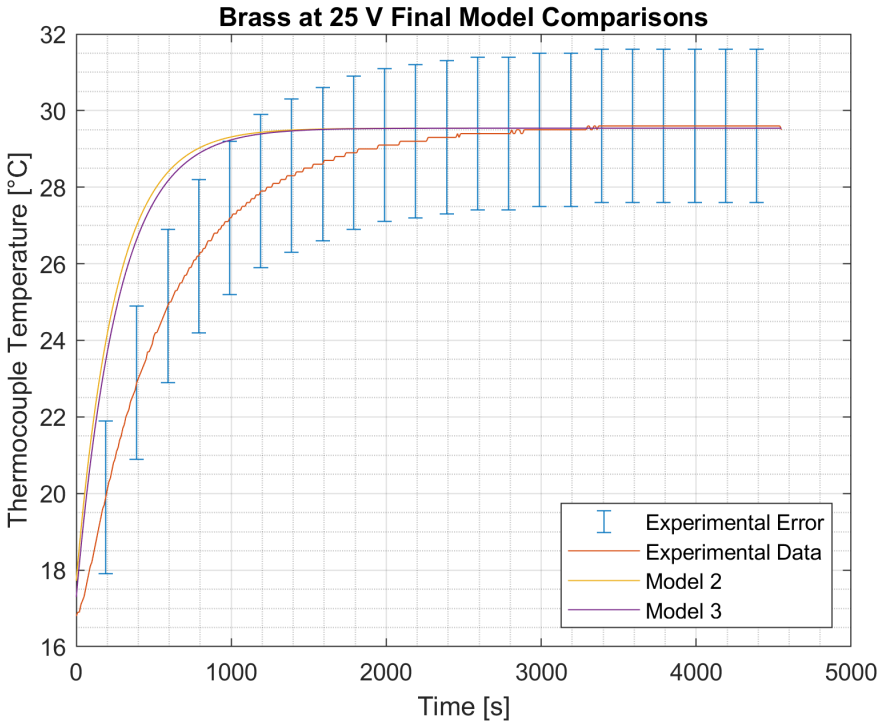
Brass at 25 V Final Model Comparisons
Thermocouple Temperature [°C]
Time [s]
Experimental Error
Experimental Data
Model 2
Model 3

Fig. 36

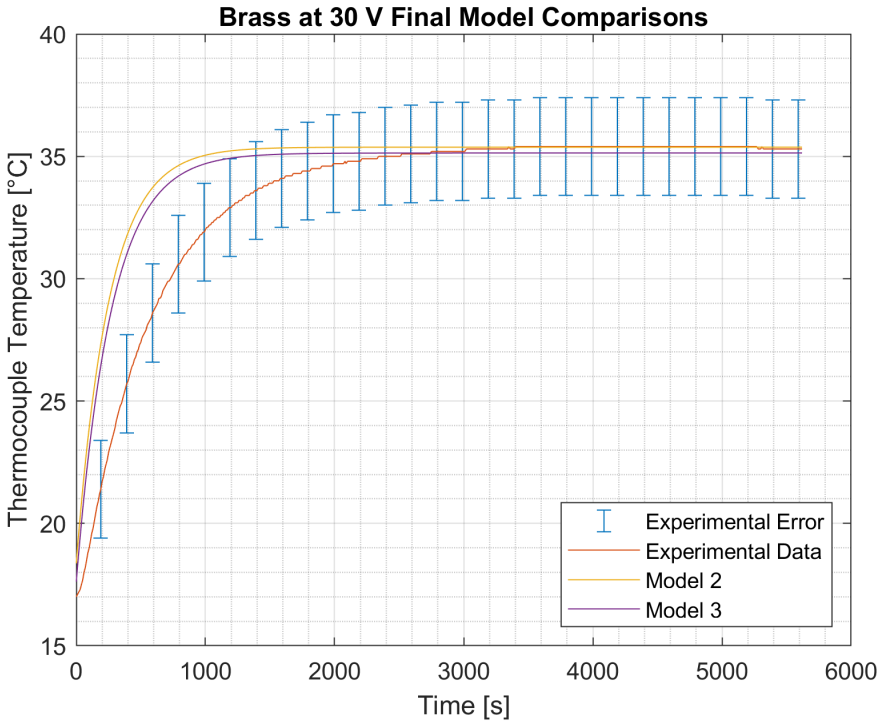
Brass at 30 V Final Model Comparisons
Thermocouple Temperature [°C]
Time [s]
Experimental Error
Experimental Data
Model 2
Model 3

Fig. 37

For Steel, the opposite is true in that Model II is more accurate than Model III across both the transient and steady-state solutions. This suggests that the calculated thermal diffusivity for steel is very close in approximation to the actual thermal diffusivity. Since Model II addresses the initial temperature distribution, and steel has the largest initial temperature distribution, it would make sense for Model II to have more impact than Model III since Model III does not address the initial temperature distribution. Another reason why Model II is better for Steel is the fact that Model III addresses the transient solution as mentioned above. Steel has the most accurate transient solution, so attempting to correct for the small discrepancy in it is not as effective as addressing the assumption that the initial temperature distribution along the bar is constant. This conclusion is supported by **Figure 38**, which shows the Steel transient solution is inside the thermocouple error.

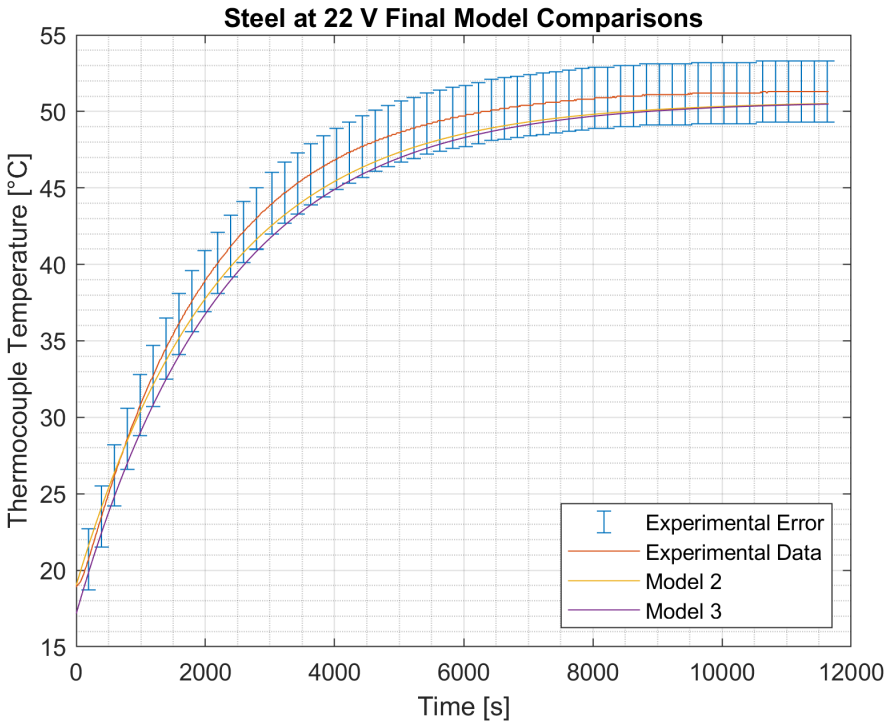

Fig. 38

E. Debate

For a researcher attempting to estimate the duration of the transient phase, Model III is the most accurate when determining the length of the transient phase. In comparison to Model II, the length of the transient phase for Model III tends to be closer in length relative to the experimental data. This would mainly be due to the fact that Model III changes α, which varies the exponential decay component of the transient solution. Specifically, Model III is most accurate in the transient phase when analyzing Steel. These models still have room for improvement, specifically in adjusting the thermal diffusivities of each material, and are not completely accurate to the physical behavior of each bar. However, Model III is the most accurate in general in comparison to Model II.

A. Appendix

A. Material Properties

Table 5 Given Material Properties

Material	Density ρ $[kg/m^3]$	Specific Heat Capacity c_p $[J/(Kg*K)]$	Thermal Conductivity k $[W/(m*K)]$
Aluminum	2810	960	130
Brass	8500	380	115
Steel	8000	500	16.2

B. Matlab Code

```
%% ASEN 3802 Lab 2 -- Heat Conduction

% Jared Steffen
% Reece Fountain
% Nathan Malyszek

% Housekeeping
clc
clear
close all


%% Material Properties

% Aluminum 7075-T651
rho_al = 2810; %[kg/m^3]
cp_al = 960; %[J/(kg*K)]
k_al = 130; %[W/(m*K)]
alpha_al = k_al / (rho_al*cp_al); %[m^2/s]

% Brass C360
rho_br = 8500; %[kg/m^3]
cp_br = 380; %[J/(kg*K)]
k_br = 115; %[W/(m*K)]
alpha_br = k_br / (rho_br * cp_br); %[m^2/s]

% Stainless Steel T-303 Annealed
rho_ss = 8000; %[kg/m^3]
cp_ss = 500; %[J/(kg*K)]
k_ss = 16.2; %[W/(m*K)]
alpha_ss = k_ss / (rho_ss * cp_ss); %[m^2/s]

% Cross Sectional Area for All Rods
d = 1 * 0.0254; %[in] -> [m]
A = pi * (d/2)^2; %[m^2]

%% Question 1

% Exmaple Thermocouple Temps and Locations
Th1 = [17.6;21.61;25.13;29.22;34.92;38.10;45.21;47.01]; %[K]
Th_loc1 = [(11/8):0.5:(39/8)]' .* 0.0254; %[m]

% Line of Best Fit
```

```
poly = polyfit(Th_loc1,Th1,1);

% T0 (y intercept) and H (slope)
T0 = poly(2);
H = poly(1);

% Equaiton of Best Fit Line
best_fit = H .* [0;Th_loc1] + T0;

% Plot
figure()
plot(Th_loc1 .* 100,Th1,'o')
hold on
plot([0;Th_loc1] .* 100,best_fit)
xlabel('Thermocouple Location [cm]')
ylabel('Thermocouple Temperature [C]')
grid on; grid minor
hold off
title('Thermocouple Reading Along Bar')
legend('Data','Best Fit','Location','northwest')

%% Question 2

% Hand-written derivation achieved results that bn = (8*H*L*(-1)^(n))/(pi^2*(2*n-1)^2)
%% Question 3

% Variables
n = 0:10;
L = 5.875 * 0.0254; %[m]
x = 4.875 * 0.0254; %[m]

% Fourier Coefficients
t1 = 1; %[s]
t2 = 1000; %[s]
Fo1 = (alpha_al * t1)/L^2;
Fo2 = (alpha_al * t2)/L^2;

% Solve u(x,t) for t=1s and t=1000s
for i = 1:length(n)
    [u_xt1(i),~] = heat_eq_sol(x,t1,T0,H,alpha_al,L,n(i));
    [u_xt1000(i),~] = heat_eq_sol(x,t2,T0,H,alpha_al,L,n(i));
end

% Note: 1 term is sufficient for t = 1000s (steady state)

% Plot to Compare
figure()
plot(n,u_xt1,'o-')
hold on
plot(n,u_xt1000,'o-')
grid on; grid minor
xlabel('Index Number (n)')
ylabel('Thermocouple 8 Temperature [C]')
hold off
title('Fourier Convergence Analysis')
legend('TC 8 Temperature @ t = 1s','TC 8 Temperature @ t = 1000s','Location','east')


%% Question 4
```

```
103
104 % Solving u(x,t) w/ Various Thermal Diffusivity
105 alpha_range = alpha_al/2:alpha_al/2:2*alpha_al;
106 t = 1:1000; %[s]
107 n = 1;
108 for j = 1:length(alpha_range)
109     for i = 1:length(t)
110         [u_xt2(i,j),~] = heat_eq_sol(x,t(i),T0,H,alpha_range(j),L,n);
111     end
112 end
113
114 % Plot
115 figure();
116 plot(t,u_xt2)
117 grid on; grid minor
118 xlabel('Time [s]')
119 ylabel('Thermocouple 8 Temperature [C]')
120 title('Thermal Diffusivity Sensitivity Analysis')
121 legend('0.5*\alpha','\alpha','1.5*\alpha','2*\alpha','Location','northwest')
122
123 % Note: Increasing alpha leads to faster steady state of thermocouples
124
125 %% Plotting all test cases
126
127 % use dir and strsplit to make extracting material, voltage, and amperage
128 % easy to use for plotting titles
129 a=dir('*mA');
130 datafiles = dir("*mA");
131
132 for i=1:length(a)
133     load(a(i).name)
134 % how to get voltage and amperage from file names?
135 % - options include strsplit, regex, etc.
136 % ultimately, we need to use the format of each file name
137 % 'material'_'volts'V_'amps'mA
138 b = strsplit(a(i).name,'_'); % gives a cell array (b) that is 1x3
139 % {'material','voltsV','ampsmA'} -- now split by 'V' and 'mA'
140 %mat = strsplit(b{1});
141 v = strsplit(b{2},'V'); % volts are always in the second portion
142 ampval= strsplit(b{3},'mA'); % amps are always in the third portion
143 materials(i) = b(1);
144 volts(i) = str2num(v{1}); % convert string to number (vector)
145 amps(i) = str2num(ampval{1});
146 end
147
148 for i = 1:size(a,1)
149 data = load(a(i).name);
150 data_Th_steady(i,:) = data(end,3:10)';
151 material = materials(i);
152 voltage = volts(i);
153 current = amps(i);
154
155 time_end(i) = start_location_plot(data,material,voltage,current);
156 end
157
158 %% Find Analytical and and Experimental Slopes and Plot to Compare
159
160 data_Th_steady = data_Th_steady';
161
```

```
% Slopes/Graphs For Aluminum, Brass, Steel
 for i = 1:length(find(strcmp('Aluminum',materials) == 1))
     [H_an_al(i),H_exp_al(i),T0_al(i),best_fit_exp_al(:,i),best_fit_an_al(:,i)] =
         slopes(Th_loc1,data_Th_steady(:,i),A,amps(i),volts(i),k_al);
     figure();
     plot([0;Th_loc1]*100,best_fit_exp_al(:,i))
     hold on
     plot([0;Th_loc1]*100,best_fit_an_al(:,i))
     plot(Th_loc1.*100,data_Th_steady(:,i),'o')
     xlabel('Thermocouple Location [cm]')
     ylabel('Thermocouple Temperature [C]')
     grid on; grid minor
     title('Aluminum Steady State Temparture Distribution for Analytical and Experimental Models')
     legend('Experimental','Analytical','Location','northwest')
 end
 hold off


 for i = 1:length(find(strcmp('Brass',materials) == 1))
     [H_an_br(i),H_exp_br(i),T0_br(i),best_fit_exp_br(:,i),best_fit_an_br(:,i)] =
         slopes(Th_loc1,data_Th_steady(:,i+2),A,amps(i+2),volts(i+2),k_br);
     figure();
     plot([0;Th_loc1]*100,best_fit_exp_br(:,i))
     hold on
     plot([0;Th_loc1]*100,best_fit_an_br(:,i))
     plot(Th_loc1.*100,data_Th_steady(:,i+2),'o')
     xlabel('Thermocouple Location [cm]')
     ylabel('Thermocouple Temperature [C]')
     grid on; grid minor
     title('Brass Steady State Temparture Distribution for Analytical and Experimental Models')
     legend('Experimental','Analytical','Location','northwest')
 end
 hold off

  for i = 1:length(find(strcmp('Steel',materials) == 1))
     [H_an_ss(i),H_exp_ss(i),T0_ss(i),best_fit_exp_ss(:,i),best_fit_an_ss(:,i)] =
         slopes(Th_loc1,data_Th_steady(:,i+4),A,amps(i+4),volts(i+4),k_ss);
     figure();
     plot([0;Th_loc1]*100,best_fit_exp_ss(:,i))
     hold on
     plot([0;Th_loc1]*100,best_fit_an_ss(:,i))
     plot(Th_loc1.*100,data_Th_steady(:,i+4),'o')
     xlabel('Thermocouple Location [cm]')
     ylabel('Thermocouple Temperature [C]')
     grid on; grid minor
     title('Steel Steady State Temparture Distribution for Analytical and Experimental Models')
     legend('Experimental','Analytical','Location','northwest')
  end
  hold off


%% Model 1A

% Time Vectors
time_al1 = 0:10:time_end(1);
time_al2 = 0:10:time_end(2);
time_br1 = 0:10:time_end(3);
time_br2 = 0:10:time_end(4);
time_steel = 0:10:time_end(5);
```

```
218
219 % Solve 1D Heat Equation For All Cases
220 for j = 1:length(Th_loc1)
221     for i = 1:length(time_al1)
222         [u_xt_al1(i,j),~] = heat_eq_sol(Th_loc1(j),time_al1(i),T0_al(1),H_an_al(1),alpha_al,L,1);
223     end
224 end
225
226 for j = 1:length(Th_loc1)
227     for i = 1:length(time_al2)
228         [u_xt_al2(i,j),~] = heat_eq_sol(Th_loc1(j),time_al2(i),T0_al(2),H_an_al(2),alpha_al,L,1);
229     end
230 end
231
232 for j = 1:length(Th_loc1)
233     for i = 1:length(time_br1)
234         [u_xt_br1(i,j),~] = heat_eq_sol(Th_loc1(j),time_br1(i),T0_br(1),H_an_br(1),alpha_br,L,1);
235     end
236 end
237
238 for j = 1:length(Th_loc1)
239     for i = 1:length(time_br2)
240         [u_xt_br2(i,j),~] = heat_eq_sol(Th_loc1(j),time_br2(i),T0_br(2),H_an_br(2),alpha_br,L,1);
241     end
242 end
243
244 for j = 1:length(Th_loc1)
245     for i = 1:length(time_steel)
246         [u_xt_steel(i,j),~] = heat_eq_sol(Th_loc1(j),time_steel(i),T0_ss,H_an_ss,alpha_ss,L,1);
247     end
248 end
249
250 % Load Data
251 data_al1 = load(a(1).name);
252 data_al2 = load(a(2).name);
253 data_br1 = load(a(3).name);
254 data_br2 = load(a(4).name);
255 data_ss = load(a(5).name);
256
257 % Plot Model 1A
258 plot_model(u_xt_al1,time_al1,data_al1,materials(1),volts(1),amps(1),'1A');
259 plot_model(u_xt_al2,time_al2,data_al2,materials(2),volts(2),amps(2),'1A');
260 plot_model(u_xt_br1,time_br1,data_br1,materials(3),volts(3),amps(3),'1A');
261 plot_model(u_xt_br2,time_br2,data_br2,materials(4),volts(4),amps(4),'1A');
262 plot_model(u_xt_steel,time_steel,data_ss,materials(5),volts(5),amps(5),'1A');
263
264 %% Model 1B
265
266 % Solve 1D Heat Equation For All Cases w/ Experimental Slope
267 for j = 1:length(Th_loc1)
268     for i = 1:length(time_al1)
269         [u_xt_al1_B(i,j),~] = heat_eq_sol(Th_loc1(j),time_al1(i),T0_al(1),H_exp_al(1),alpha_al,L,1);
270     end
271 end
272
273 for j = 1:length(Th_loc1)
274     for i = 1:length(time_al2)
275         [u_xt_al2_B(i,j),~] = heat_eq_sol(Th_loc1(j),time_al2(i),T0_al(2),H_exp_al(2),alpha_al,L,1);
276     end
```

```
end

for j = 1:length(Th_loc1)
    for i = 1:length(time_br1)
        [u_xt_br1_B(i,j),~] = heat_eq_sol(Th_loc1(j),time_br1(i),T0_br(1),H_exp_br(1),alpha_br,L,1);
    end
end

for j = 1:length(Th_loc1)
    for i = 1:length(time_br2)
        [u_xt_br2_B(i,j),~] = heat_eq_sol(Th_loc1(j),time_br2(i),T0_br(2),H_exp_br(2),alpha_br,L,1);
    end
end

for j = 1:length(Th_loc1)
    for i = 1:length(time_steel)
        [u_xt_steel_B(i,j),~] = heat_eq_sol(Th_loc1(j),time_steel(i),T0_ss,H_exp_ss,alpha_ss,L,1);
    end
end

% Plot Model 1B
plot_model(u_xt_al1_B,time_al1,data_al1,materials(1),volts(1),amps(1),'1B');
plot_model(u_xt_al2_B,time_al2,data_al2,materials(2),volts(2),amps(2),'1B');
plot_model(u_xt_br1_B,time_br1,data_br1,materials(3),volts(3),amps(3),'1B');
plot_model(u_xt_br2_B,time_br2,data_br2,materials(4),volts(4),amps(4),'1B');
plot_model(u_xt_steel_B,time_steel,data_ss,materials(5),volts(5),amps(5),'1B');

%% Model 2

% Initital Temp Distribution Slope
M_exp_al1 = init_dist_plot(data_al1,Th_loc1,materials(1));
M_exp_al2 = init_dist_plot(data_al2,Th_loc1,materials(2));
M_exp_br1 = init_dist_plot(data_br1,Th_loc1,materials(3));
M_exp_br2 = init_dist_plot(data_br2,Th_loc1,materials(4));
M_exp_ss = init_dist_plot(data_ss,Th_loc1,materials(5));

% Solve 1D Heat Equation For All Cases w/ Experimental Slope and Initial Temp Distributions
for j = 1:length(Th_loc1)
    for i = 1:length(time_al1)
        u_xt_al1_2(i,j) =
            heat_eq_sol2(Th_loc1(j),time_al1(i),T0_al(1),H_exp_al(1),alpha_al,L,1,M_exp_al1);
    end
end

for j = 1:length(Th_loc1)
    for i = 1:length(time_al2)
        u_xt_al2_2(i,j) =
            heat_eq_sol2(Th_loc1(j),time_al2(i),T0_al(2),H_exp_al(2),alpha_al,L,1,M_exp_al2);
    end
end

for j = 1:length(Th_loc1)
    for i = 1:length(time_br1)
        u_xt_br1_2(i,j) =
            heat_eq_sol2(Th_loc1(j),time_br1(i),T0_br(1),H_exp_br(1),alpha_br,L,1,M_exp_br1);
    end
end

for j = 1:length(Th_loc1)
```

```
    for i = 1:length(time_br2)
        u_xt_br2_2(i,j) =
            heat_eq_sol2(Th_loc1(j),time_br2(i),T0_br(2),H_exp_br(2),alpha_br,L,1,M_exp_br2);
    end
end

for j = 1:length(Th_loc1)
    for i = 1:length(time_steel)
        u_xt_steel_2(i,j) =
            heat_eq_sol2(Th_loc1(j),time_steel(i),T0_ss,H_exp_ss,alpha_ss,L,1,M_exp_ss);
    end
end

% Plot vs Experimental Data
plot_model(u_xt_al1_2,time_al1,data_al1,materials(1),volts(1),amps(1),'2')
plot_model(u_xt_al2_2,time_al2,data_al2,materials(2),volts(2),amps(2),'2');
plot_model(u_xt_br1_2,time_br1,data_br1,materials(3),volts(3),amps(3),'2');
plot_model(u_xt_br2_2,time_br2,data_br2,materials(4),volts(4),amps(4),'2');
plot_model(u_xt_steel_2,time_steel,data_ss,materials(5),volts(5),amps(5),'2');

%% Model 3

% Find Adjusted Thermal Diffusivity
alpha_adj_al = alpha_var(u_xt_al1_B(end-50:end,8),alpha_al,x,time_al1,T0_al(1),H_exp_al(1),L,1);
alpha_adj_br = alpha_var(u_xt_br1_B(end-50:end,8),alpha_br,x,time_br1,T0_br(1),H_exp_br(1),L,1);
alpha_adj_ss = alpha_var(u_xt_steel_B(end-50:end,8),alpha_ss,x,time_steel,T0_ss,H_exp_ss,L,1);

% Solve 1D Heat Equation For All Cases w/ Experimental Slope and Adjusted Thermal Diffusivity
for j = 1:length(Th_loc1)
    for i = 1:length(time_al1)
        [u_xt_al1_3(i,j),Fo_al1(i)] =
            heat_eq_sol(Th_loc1(j),time_al1(i),T0_al(1),H_exp_al(1),alpha_adj_al,L,1);
    end
end

for j = 1:length(Th_loc1)
    for i = 1:length(time_al2)
        [u_xt_al2_3(i,j),Fo_al2(i)] =
            heat_eq_sol(Th_loc1(j),time_al2(i),T0_al(2),H_exp_al(2),alpha_adj_al,L,1);
    end
end

for j = 1:length(Th_loc1)
    for i = 1:length(time_br1)
        [u_xt_br1_3(i,j),Fo_br1(i)] =
            heat_eq_sol(Th_loc1(j),time_br1(i),T0_br(1),H_exp_br(1),alpha_adj_br,L,1);
    end
end

for j = 1:length(Th_loc1)
    for i = 1:length(time_br2)
        [u_xt_br2_3(i,j),Fo_br2(i)] =
            heat_eq_sol(Th_loc1(j),time_br2(i),T0_br(1),H_exp_br(2),alpha_adj_br,L,1);
    end
end

for j = 1:length(Th_loc1)
    for i = 1:length(time_steel)
```

```
        [u_xt_steel_3(i,j),Fo_ss(i)] =
            heat_eq_sol(Th_loc1(j),time_steel(i),T0_ss,H_exp_ss,alpha_adj_ss,L,1);
    end
end

% Plot Results
plot_model(u_xt_al1_3,time_al1,data_al1,materials(1),volts(1),amps(1),'3');
plot_model(u_xt_al2_3,time_al2,data_al2,materials(2),volts(2),amps(2),'3');
plot_model(u_xt_br1_3,time_br1,data_br1,materials(3),volts(3),amps(3),'3');
plot_model(u_xt_br2_3,time_br2,data_br2,materials(4),volts(4),amps(4),'3');
plot_model(u_xt_steel_3,time_steel,data_ss,materials(5),volts(5),amps(5),'3');

% Fourier Number and Time to Steady State
[t_al_ss,Fo_al_new] = FoNum(Fo_al1,alpha_adj_al,time_al1,L);
[t_br_ss,Fo_br_new] = FoNum(Fo_br1,alpha_adj_br,time_br1,L);
[t_steel_ss,Fo_steel_new] = FoNum(Fo_ss,alpha_adj_ss,time_steel,L);

%% Comparison of Models

% Gathering Thermocouple 8 Data for Models 2 and 3
% Aluminum 25V
al1_mod2 = u_xt_al1_2(:,end);
al1_mod3 = u_xt_al1_3(:,end);

% Aluminum 30V
al2_mod2 = u_xt_al2_2(:,end);
al2_mod3 = u_xt_al2_3(:,end);

% Brass 25V
br1_mod2 = u_xt_br1_2(:,end);
br1_mod3 = u_xt_br1_3(:,end);

% Brass 30V
br2_mod2 = u_xt_br2_2(:,end);
br2_mod3 = u_xt_br2_3(:,end);

% Steel 22V
ss_mod2 = u_xt_steel_2(:,end);
ss_mod3 = u_xt_steel_3(:,end);

% Plot
plot_errorbars(data_al1,al1_mod2,al1_mod3,time_al1,materials(1),volts(1))
plot_errorbars(data_al2,al2_mod2,al2_mod3,time_al2,materials(2),volts(2))
plot_errorbars(data_br1,br1_mod2,br1_mod3,time_br1,materials(3),volts(3))
plot_errorbars(data_br2,br2_mod2,br2_mod3,time_br2,materials(4),volts(4))
plot_errorbars(data_ss,ss_mod2,ss_mod3,time_steel,materials(5),volts(5))

%% Functions
function [u_xt,Fo] = heat_eq_sol(x,t,T0,H,alpha,L,n)
%-----------------------------------------------------------------
% Function to solve u(x,t) for a given n and material properties
% Inputs:
% x - thermocouple location(s) [m]
% t - time vector [s]
% T0 - initial temperature [C]
% H - steady state temperature distrubtion slope [C/m]
% alpha - thermal diffusivity [m^2/s]
% L - rod length (not true length) [m]
% n - number of iterations [unitless]
```

```
%
% Outputs:
% u_xt - 1D heat equation solution [C]
%--------------------------------------------------------------

% n Variable Functions
bn = @(n) (8*H*L*(-1)^(n))/(pi^2*(2*n-1)^2);
lambda_n = @(n) (2*n-1)*pi/(2*L);

% Run Summation for n Terms
gx_sum = 0;
for i = 1:n
    gx_sum = gx_sum + bn(i)*sin(lambda_n(i)*x)*exp(-(lambda_n(i)^2*alpha*t));
end

% Calculate u(x,t)
u_xt = T0 + H*x + gx_sum;


FoFunc = @(a,t,l)(a*t)/((L)^2); %anon fn, a "alpha", t "time", L "length"

% Fourier calculation
Fo = FoFunc(alpha,t,L);


end

function time_end = start_location_plot(DataSet, Material, Voltage, Current)
%--------------------------------------------------------------
% Function to solve from end of time vector and plot experimental data
% Inputs:
% DataSet - data for a specific test case [several units]
% Material - material type
% Voltage - test voltage value [V]
% Current - test current value [mA]
%
% Outputs:
% time_end - end of time vector [s]
%--------------------------------------------------------------

% extract time vector from first column
time = (DataSet(:,1));

% look at last thermocouple data in order to find where to start plot
val = DataSet(:,end);

% find where to start plot using zeroth, first, and second derivatives of
% line
x_start = find(val == val(1),1,"last");

dx = diff(val);
dx_start = find(dx > 0.1,1);

d2x = diff(diff(val));
d2x_start = find(d2x > 0.2,1);

starting_vals = [x_start dx_start d2x_start];

% find the smallest starting index out of the three methods to use for plotting
```

```
start_val = min(starting_vals);

% change time vector to reflect new starting point
time = time(start_val:end);
time = time - time(1);
time_end = time(end);

% plot, ignoring first and second columns to only plot relative data
figure()
for i = 3:size(DataSet,2)
    plot(time, DataSet(start_val:end,i))
    hold on
end
grid on; grid minor
xlabel("Time [s]")
ylabel("Temperature [C]")
title("Test Case for " + Material + " at " + Voltage + "V and " + Current + "mA")

end

function [H_an,H_exp,T0,best_fit_exp,best_fit_an] = slopes(Th_loc,Th,A,I,V,k)
%-----------------------------------------------------------------
% Function to solve for steady temp distribution slope, initial temp, and
% best fit lines
% Inputs:
% Th_loc - thermocouple location(s) [m]
% Th - test case temperature [C]
% A - cross-sectional area of rod [m^2]
% I - test case current [mA]
% V - test case voltage [V]
% k - thermal conductivity of material [W/(m*K)]
%
% Outputs:
% H_an - analytical steady state temperature distribution slope [C/m]
% H_exp - experimental steady state temperature distribution slope [C/m]
% T0 - initial temperature [C]
% best_fit_exp = line of best fit equation for experimental data
% best_fit_an = line of best fit equation for analytical data
%-----------------------------------------------------------------

% Polyfit Data
poly = polyfit(Th_loc,Th,1);

% Experimental Slope and Y-Intercept
H_exp = poly(1);
T0 = poly(2);

% Analytical Slope
I = I / 1e3;
Q_dot = V*I;
H_an = Q_dot / (k*A);

% Best Fit Line
best_fit_an = H_an .* [0;Th_loc] + T0;
best_fit_exp = H_exp .* [0;Th_loc] + T0;

end

function time_end = plot_model(u_xt,time_mat,DataSet, Material, Voltage, Current, Model)
```

```
%-----------------------------------------------------------------
% Function to solve from end of time vector and plot models
% Inputs:
% u_xt - 1D heat equation solution [C]
% time_mat - original time vector for a specific material [s]
% DataSet - data for a specific test case [several units]
% Material - material type
% Voltage - test voltage value [V]
% Current - test current value [mA]
% Model - model number [1A,1B,2,3]
%
% Outputs:
% time_end - end of time vector [s]
%-----------------------------------------------------------------

% extract time vector from first column
time = (DataSet(:,1));

% look at last thermocouple data in order to find where to start plot
val = DataSet(:,end);

% find where to start plot using zeroth, first, and second derivatives of
% line
x_start = find(val == val(1),1,"last");

dx = diff(val);
dx_start = find(dx > 0.1,1);

d2x = diff(diff(val));
d2x_start = find(d2x > 0.2,1);

starting_vals = [x_start dx_start d2x_start];

% find the smallest starting index out of the three methods to use for plotting
start_val = min(starting_vals);

% change time vector to reflect new starting point
time = time(start_val:end);
time = time - time(1);
time_end = time(end);

% plot, ignoring first and second columns to only plot relative data
figure()
for i = 3:size(DataSet,2)
    p1 = plot(time, DataSet(start_val:end,i),'-r');
    hold on
end
p2 = plot(time_mat, u_xt,'-b');
hold off
grid on; grid minor
xlabel("Time [s]")
ylabel("Temperature [C]")
title("Model " + Model + " " + Material + " at " + Voltage + "V and " + Current + "mA")
legend([p1(1);p2(1)],{'Experimental';'Model'},'Location','southeast')

end

function M_exp = init_dist_plot(data,Th_loc,Material)
%-----------------------------------------------------------------
```

```
% Function to solve for initial temp distribution slope
% Inputs:
% data - data for specific test case [C]
% Th_loc - thermocouple location(s) [m]
% Material - material type
%
% Outputs:
% M_exp -initial temperature distribution slope [C/m]
%-------------------------------------------------------------------

% Find and Polyfit Initial Temps
init_temp = data(1,3:end);
polyfit_data = polyfit(Th_loc,init_temp,1);

% Slope/Y-Intercept
M_exp = polyfit_data(1);
y_int = polyfit_data(2);

% Best Fit Line and Average (for y-scale)
best_fit = M_exp .* Th_loc + y_int;
avg = mean(init_temp);

% Plot
figure()
plot(Th_loc .* 100,init_temp,'o')
hold on
plot(Th_loc .* 100,best_fit)
grid on; grid minor
xlabel('Thermocouple Location [cm]')
ylabel('Thermocouple Temperature [C]')
ylim([avg-1 avg+1])
title(Material + " Initial Temperature Distribution")
legend('Experimental Data','Best Fit')

end

function u_xt = heat_eq_sol2(x,t,T0,H,alpha,L,n,M)
%-------------------------------------------------------------------
% Function to solve u(x,t) for a given n and material properties
% Inputs:
% x - thermocouple location(s) [m]
% t - time vector [s]
% T0 - initial temperature [C]
% H - steady state temperature distrubtion slope [C/m]
% alpha - thermal diffusivity [m^2/s]
% L - rod length (not true length) [m]
% n - number of iterations [unitless]
% M - initial temperature distribution slope [C/m]
%
% Outputs:
% u_xt - 1D heat equation solution [C]
%-------------------------------------------------------------------

% n Variable Functions
bn = @(n) (-8*(M-H)*L*(-1)^(n))/(pi^2*(2*n-1)^2);
lambda_n = @(n) (2*n-1)*pi/(2*L);

% Run Summation for n Terms
gx_sum = 0;
```

```
for i = 1:n
    gx_sum = gx_sum + bn(i)*sin(lambda_n(i)*x)*exp(-(lambda_n(i)^2*alpha*t));
end

% Calculate u(x,t)
u_xt = T0 + H*x + gx_sum;

end


function alpha_adj = alpha_var(u_xt,alpha,x,t,T0,H,L,n)
%-----------------------------------------------------------------
% Function to solve for adjusted thermal diffusivity
% Inputs:
% u_xt - 1D heat equation solution [C]
% alpha - thermal diffusivity [m^2/s]
% x - thermocouple location(s) [m]
% t - time vector [s]
% T0 - initial temperature [C]
% H - steady state temperature distrubtion slope [C/m]
% alpha - thermal diffusivity [m^2/s]
% L - rod length (not true length) [m]
% n - number of iterations [unitless]
%
% Outputs:
% alpha_adj - adjusted thermal diffusivity [m^2/s]
%-----------------------------------------------------------------

% Find Average of Steady State Variations
d = mean(u_xt);

% Range of Thermal Diffusivity to Test
alpha_range = alpha/1000:alpha/1000:1.5*alpha;

% Test Last Thermocouple with Alpha Range
for j = 1:length(alpha_range)
    for i = 1:length(t)
        [u_xt_alpha_var(i,j),~] = heat_eq_sol(x,t(i),T0(1),H(1),alpha_range(j),L,n);
    end
end

% Find Which Alpha Leads to Closest Steady State Average
e = find(u_xt_alpha_var(end,:) >= d, 1, 'first');
alpha_adj = alpha_range(e);

end

function [t_ss,Fo] = FoNum(Fo_exp,alpha_adj,time,L)
%-----------------------------------------------------------------
% Function to solve for time to steady state and fourier number
% Inputs:
% Fo_exp - fourier number from experimental data [unitless]
% alpha_adj - adjusted thermal diffusivity [m^2/s]
% time - time vector [s]
% L - rod length (not true length) [m]
%
% Outputs:
% t_ss - time to steady state [s]
% Fo - corresponding fourier number [unitless]
```

```
%-----------------------------------------------------------------

Fo_pos = find(Fo_exp > 2,1,'first');

t_ss = time(Fo_pos);

Fo = (alpha_adj * t_ss)/L^2;

end

function [] = plot_errorbars(DataSet,mod2,mod3,tmod,Material,Voltage)
%-----------------------------------------------------------------
% Function to plot thermocouple 8 final temp w/ error bars
% Inputs:
% DataSet - data for a specific test case [several units]
% mod2 - model 2 final thermocouple 8 data for a specific test case [C]
% mod3 - model 3 final thermocouple 8 data for a specific test case [C]
% t_exp - experimental time vector [s]
% tmod - model time vector [s]
% Material - material type
% Voltage - test voltage value [V]
%
% Outputs:
% Thermocouple 8 final temperature comparison bar plot w/ error bars
%-----------------------------------------------------------------

% extract time vector from first column
time = (DataSet(:,1));

% look at last thermocouple data in order to find where to start plot
val = DataSet(:,end);

% find where to start plot using zeroth, first, and second derivatives of
% line
x_start = find(val == val(1),1,"last");

dx = diff(val);
dx_start = find(dx > 0.1,1);

d2x = diff(diff(val));
d2x_start = find(d2x > 0.2,1);

starting_vals = [x_start dx_start d2x_start];

% find the smallest starting index out of the three methods to use for plotting
start_val = min(starting_vals);

% change time vector to reflect new starting point
time = time(start_val:end);
time = time - time(1);


% Error Vectors
err = 2 * zeros(length(DataSet(start_val:end,end)),1);
for i = 1:length(err)
    if rem(i,20) == 0 % if remainder or i/20 is zero sets value to 2
        err(i) = 2;
    else
        err(i) = NaN; % otherwise removes data point
```

```
    end
end

% Plot Comparison
figure();
errorbar(time,DataSet(start_val:end,end),err,"LineStyle","none")
hold on
plot(time,DataSet(start_val:end,end))
plot(tmod,mod2)
plot(tmod,mod3)
grid on; grid minor
xlabel('Time [s]')
ylabel('Thermocouple Temperature [C]')
legend('Experimental Error','Experimental Data','Model 2','Model 3','Location','southeast')
title(Material + " at " + Voltage + " V Final Model Comparisons")
hold off

end
```