Interpolation of Noisy Data

Jordan Rhoades

December 7, 2018

Abstract

For my final project in my engineering course, Microcontrollers, I wanted to find out if a computer that had dust ran at a higher temperature compared to the computer with no dust in it. After getting noisy data and not being able to look at the data on qualitative level, I needed to find a numerical method that could give a good approximated polynomial. The Lagrange Interpolation method approximation was a good fit to the experimental data and showed that a computer does indeed run at a slightly warmer temperature when a computer has dust in it compared to a computer with no dust.

Keywords— Interpolation, Lagrange Interpolation, Least Squares, Thermistor, Temperature, Raspberry Pi

1 Introduction

In an engineering course called Microcontrollers, I wanted to experiment and see if a computer that had dust in it would run at a hotter temperature than if the computer had no dust in it. For this class' project, I used a Raspberry Pi 3 and a circuit to read and write data from three locations in the computer for a period of fifteen minutes. During that time period, I had the computer run a stress test that would test my computer components and push them to their limits that would result in making the computer be at a warmer temperature. Using that data, I then created graphs showing the temperatures using a Python script. My setup of the microcontroller and circuit are shown below.

The components in my circuit included three thermistors and an integrated circuit. The thermistors interact with heat and depending on the the temperature, it will give a specific resistance and the voltage going across the thermistor will change accordingly. The integrated circuit allows me to use three of these thermistors and sends the change in voltage to my Raspberry Pi 3 where I can convert that voltage into a temperature. The three thermistors placed in the computer are measuring the in-flow temperature, the out-flow temperature, and the temperature in between the Central Processing Unit (CPU) and the Graphics Processing Unit (GPU). The two components are listed shown below.

The final product from project are shown below and show why I need to use numerical methods to be able to analyze the experimental data.

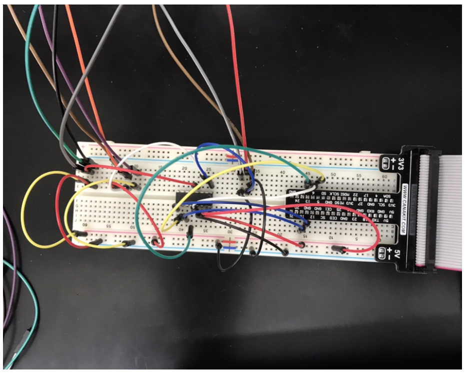

Figure 1: Image of the circuit attached to the microcontroller.

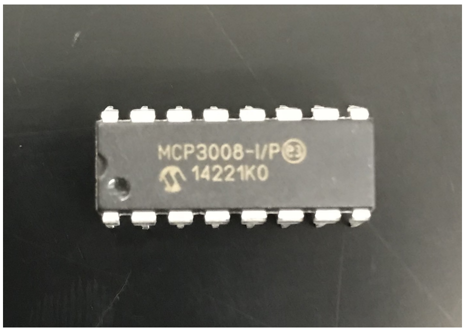

Figure 2: MCP3008 integrated circuit that allows the use of three thermistors and sends data to the Raspberry Pi 3.

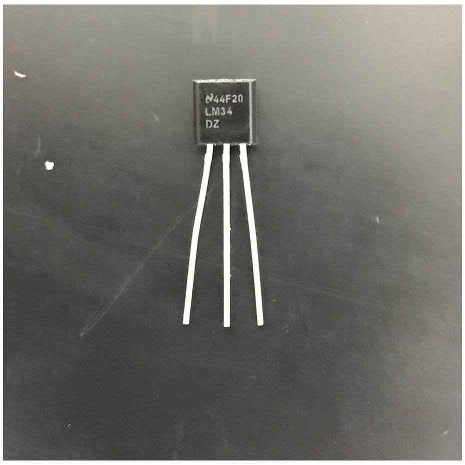

Figure 3: One of three thermistors used in the this Microcontrollers project.

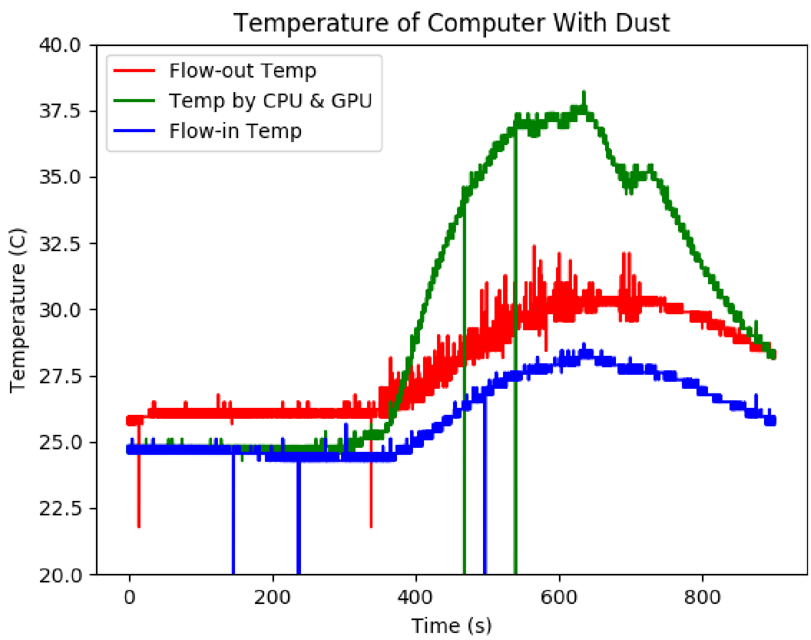

Figure 4: Experimental data for a dusty computer.

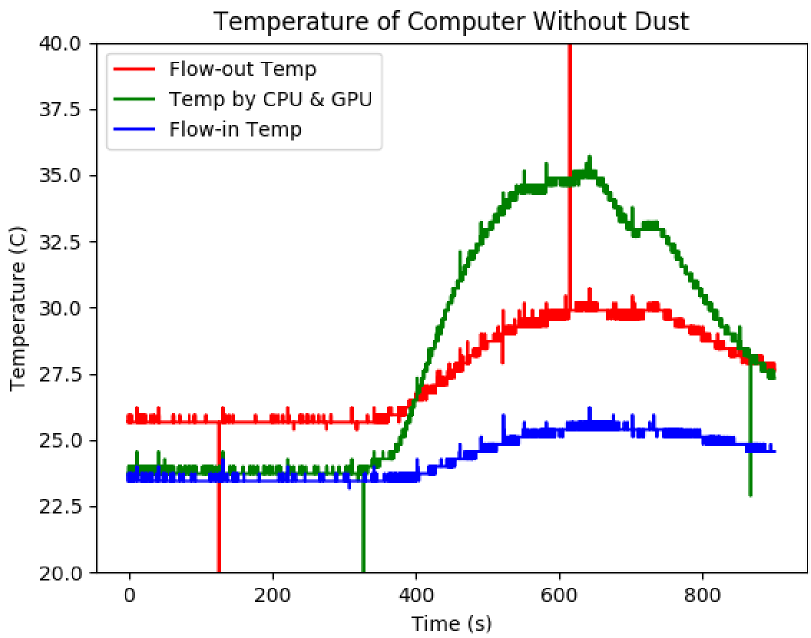

Figure 5: Experimental data for a computer without dust.

2 Problem Formation

Due to the amount of noise gathered from this past project, it is very hard to do any kind of analysis besides a general one that can show that one graph is a higher temperature than the other. I would like to be able to see the difference between the dusty data and not dusty data numerically. From this numerical course, I would like to use a numerical method that can create a "best fit" polynomial that can give me a close and good approximation of this noisy data.

3 Numerical Methods

In this course, we learned about a numerical method called interpolation that can use my experimental data and create good approximated polynomials. The two methods that I used for this project are Lagrange Interpolation and Least Squares

3.1 Lagrange Interpolation

I used Lagrange Interpolation because I can set the amount of points that it has using "linspace." I also used this method because its points are on an evenly spaced interval and unlike cubic spline, I can create a polynomial that does not accumulate noisy data. The general equation for Lagrange Polynomial is:

$$P(x) = \sum_{j=1}^{n} y_j \prod_{i=1; i \neq j}^{n} \frac{x - x_i}{x_j - x_i} \tag{1}$$

3.2 Least Squares

I also used Least Squares because even with error in my experimental data, the approximated polynomial does not get distorted or have much noise as other approximation. The general equation for Least Squares is:

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = \sum_{i=0}^{n} a_i x^i \tag{2}$$

4 Results

For the Lagrange Interpolation, I used 50 evenly spaced nodes and because of this, the graphs below show very good approximations. The best fit lines on each of the three sections are following the data without noise in an accurate manner. The two experimental data then have the Lagrange Interpolation placed on top and show how well the polynomial fits are below.

For the Least Squares approximated polynomial is fitted for a polynomial of order ten. This polynomial had the best fit and any other order did not make an approximation that fit any of the three temperature data. This approximated polynomial overall is good but it lacks a good approximation for the temperature for the CPU and GPU. The two polynomial approximations fitting the experimental data are below.

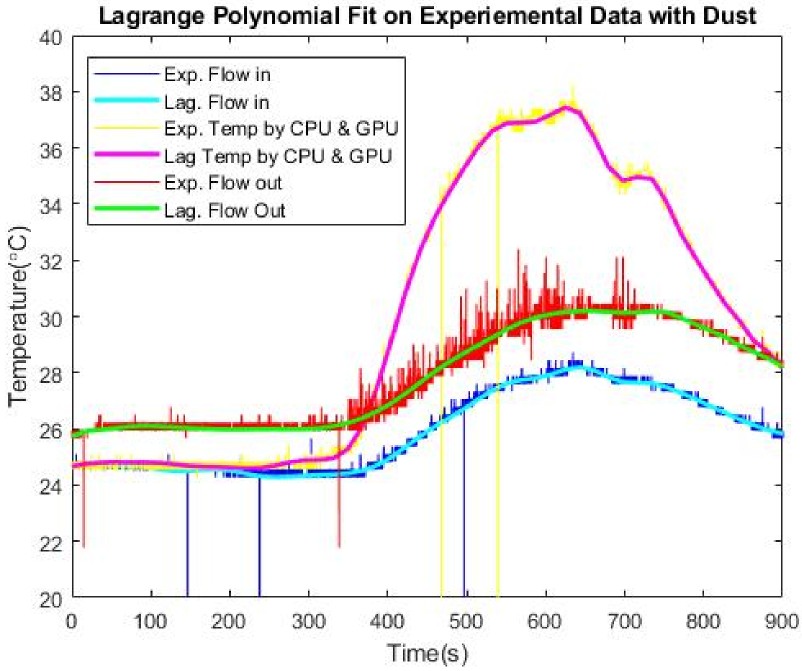

Figure 6: Lagrange Interpolation polynomial placed over the experimental data with dust to show that it's a good approximation.

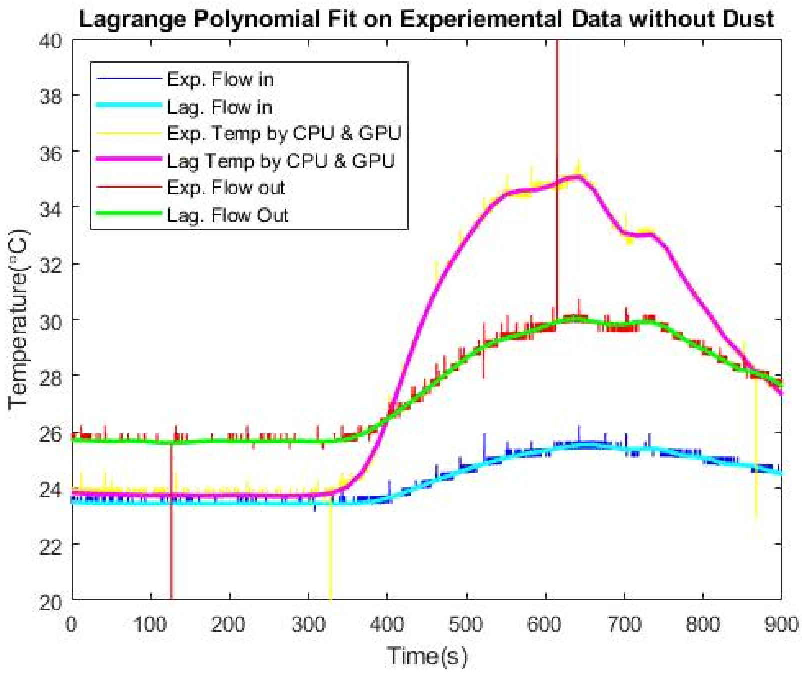

Figure 7: Lagrange Interpolation polynomial placed over the experimental data without dust to show that it's a good approximation.

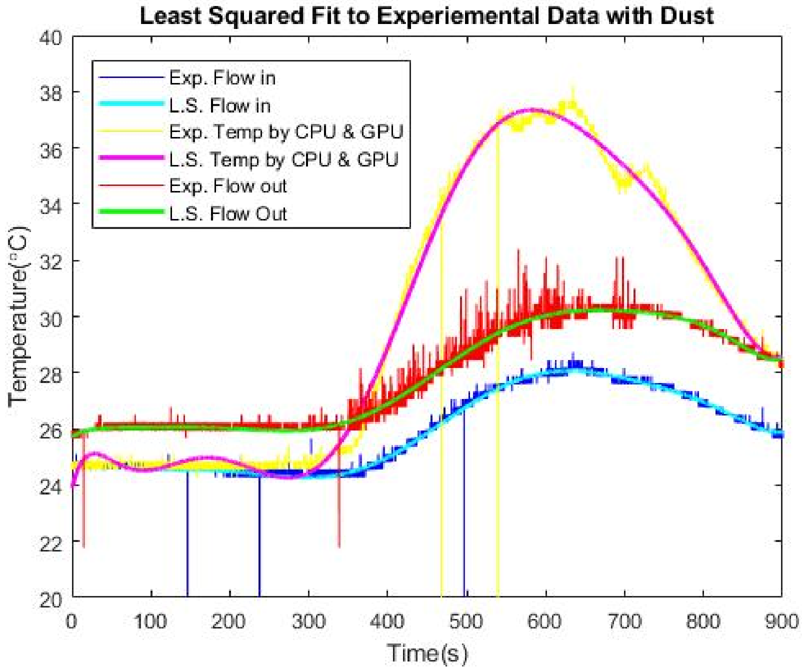

Figure 8: Least Squares polynomial placed over the experimental data with dust to show that it's an overall good approximation.

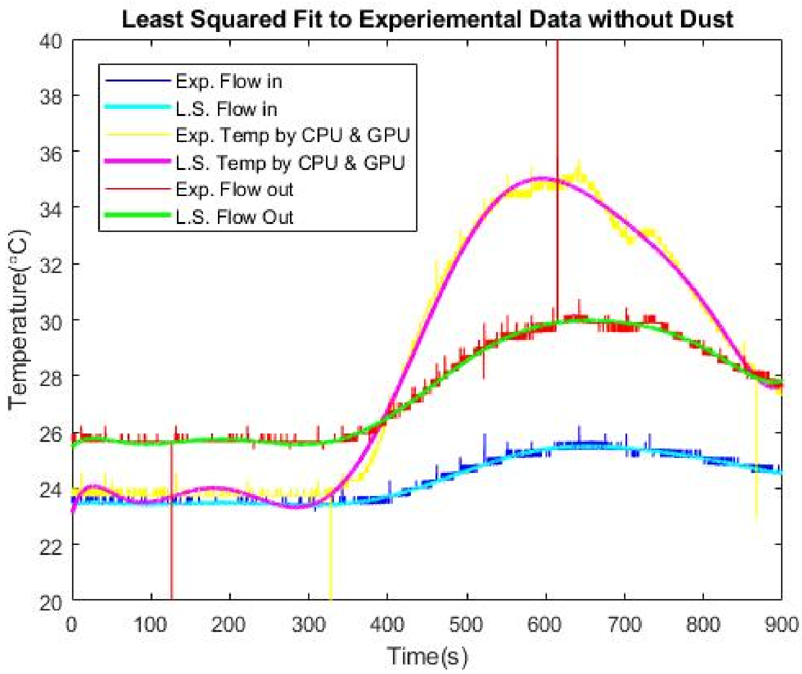

Figure 9: Least Squares polynomial placed over the experimental data with dust to show that it's an overall good approximation.

Thermistor	5 min	10 min	15 min
In-Flow	$24.3458°C$	$27.8791°C$	$25.8349°C$
CPU and GPU 2	$24.3458°C$	$37.0771°C$	$28.1484°C$
Out-Flow	$25.9955°C$	$30.0495°C$	$28.2139°C$

Table 1: Lagrange Interp. Data with Dust

Thermistor	5 min	10 min	15 min
In-Flow	$24.3519°C$	$25.3291°C$	$24.6429°C$
CPU and GPU	$24.3072°C$	$34.6846°C$	$27.0372°C$
Out-Flow	$25.6667°C$	$29.7726°C$	$27.6429°C$

Table 2: Lagrange Interp. Data without Dust

Since the Lagrange Interpolation fits the data extremely well, I can now see numerically if a computer with dust in it runs at a hotter temperature than if the computer runs without dust. I created a table that shows the temperatures for every five minutes starting at the five minute mark going to fifteen minutes.

Overall the data shows that there is a lower temperature when there is less dust in the computer. There is not a huge difference between the data with dust and the data without dust but it does show that if you were to clean out your dusty computer that it would run at lower temperature of a few degrees.

5 Conclusion

In conclusion, the two interpolation methods overall make fantastic approximations of the noisy experimental data. The Lagrange Interpolation makes near perfect approximations and the Least Squares method makes decent approximations. The data with the Lagrange Interpolation does show that there is a small difference in temperature between the dust and not dusty data. Due to using interpolation methods, I was finally able to qualitatively analyze the data that I collected for my Microcontrollers' final project.

6 Future Work

The next step for this project is to first test more interpolation methods and see if I can find another method that fits as good as the Lagrange Interpolation method did. I would also need to change what type of thermistors I used because that would get rid of some of the error that we acquired initially. The other thing that I would need to change is where exactly the thermistors are located. It would take further testing to see where a good location would be for the in-flow thermistors and the thermistor by the CPU and GPU.

7 References

Burden, Richard L., et al. Numerical Analysis. Cengage Learning, 2016.

"MathWorks - Makers of MATLAB and Simulink." Random Number Generation - MATLAB amp; Simulink, www.mathworks.com/.

Montalvo, Dr. Carlos, director. YouTube. YouTube, YouTube, 17 Mar. 2015, www.youtube.com/watch?v=5usW7aqo-bU.