AST 325 Lab 2: Spectroscopy

Juan Pablo Alfonzo
juanpablo.alfonzo@mail.utoronto.ca

November 6, 2019

Abstract

This report focuses on the spectroscopy of various light sources that were obtained using a Charged Coupled Device (CCD) image sensor and a SBGI spectrograph attached to a 16-inch telescope. Various spectra were collected from both man made light sources and astronomical light sources. These spectra were analyzed and a relationship between the pixel number on the CCD and the corresponding wavelength of light was found. This was done using a linear least square fitting on points that mapped the pixel number of the peaks versus the corresponding wavelengths. Error propagation was done on both the slope and the y-intercept of the linear least square fitting. The main conclusion of this experiment is that the relationship between pixel position and the corresponding wavelength of light very closely follows a linear relationship.

1 Introduction

Spectroscopy is an area of major interest in astrophysics as it allows us to study the composition of far away astronomical objects by simply observing their emitted light spectra. This lab focuses on collecting spectra of various man made and astronomical sources to analyze their compositions. The way this was done was using a CCD to collect incoming photons from these light sources. A relationship between the pixel recorded on the CCD (or SGBI spectrograph) and the corresponding wavelength was then found for certain spectra using a linear least square fitting. In this lab the spectra for 4 man made sources were recorded: an incandescent light bulb, a neon gas tube light, a hydrogen gas tube light, and a fluorescent tube light. Additionally, spectra for 5 stars were recorded: the Sun, Vega, Enif, Navi, and Scheat. A "Dark" spectrum was also taken to account for the radiation (mostly heat) given off by the SGBI spectrograph. The centroids of certain spectra were then calculated and graphed versus their respective wavelength, found on this lab handout[4], in order to find the mapping between pixel number and wavelength via a linear least square fitting. Using this, wavelengths found in the spectra could be compared to those of various elements to conclude which elements were present in the light sources. A black body diagram of the light sources could also be generated using this information.

Work for this lab was evenly split amongst the author (Juan Pablo Alfonzo) and the three other group members: Parampreet Singh, Lucas Louwerse, and Nicholas Clark. The code used to analyze the data sets were discussed and worked on together as a group, but each member wrote their own unique code and had their own unique approach to handling the data. In parts of the code very different approaches are taken in tackling similar problems providing the group with more support for our conclusions as different approaches yielded similar solutions.

2 Observations and Data

The main instruments used in this lab were a CCD and a 16-inch telescope with an attached SGBI spectrograph. The CCD was an "Ocean Optics" one which was used to collect the spectra of the man made sources as well as that of the Sun. This CCD like any other works via the photoelectric effect which causes a metal to release an electron when struck by a photon. This electron is then sent through a chain of sensors that end with an analog to digital converter so that a computer can record the results. The specifics of this process can be found in the reference section of this paper [1]. The photons were directed into the Ocean Optics CCD using a fiber optic cable. The SGBI spectrograph is a type of CCD and hence works in a very similar way. The 16-inch telescope to which the SGBI spectrograph was attached to is located on the roof of McLennan Physical Labs (60 St George St, Toronto, ON) and data was collected on October 28th, 2019. The photons were directed using the telescope lenses onto the SGBI spectrograph. The Ocean Optics CCD provided us with the spectra in a text file list, while the SGBI spectrograph gave us a `.fit` file both of which were plotted as follows:

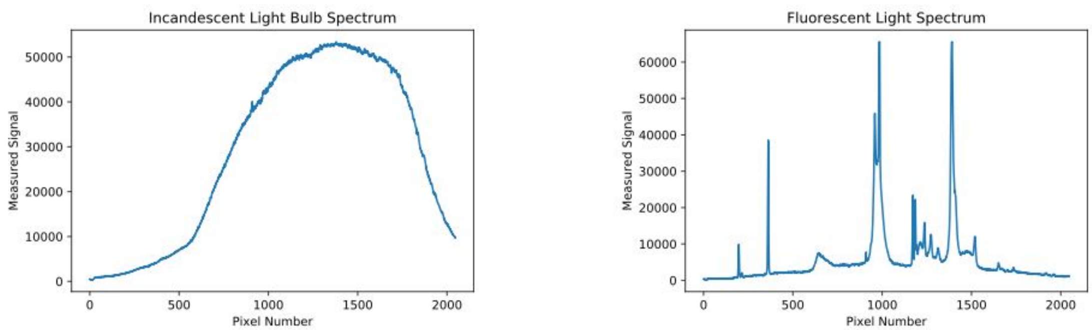

Figure 1: Spectra of Incandescent Bulb and Fluorescent Light [3]

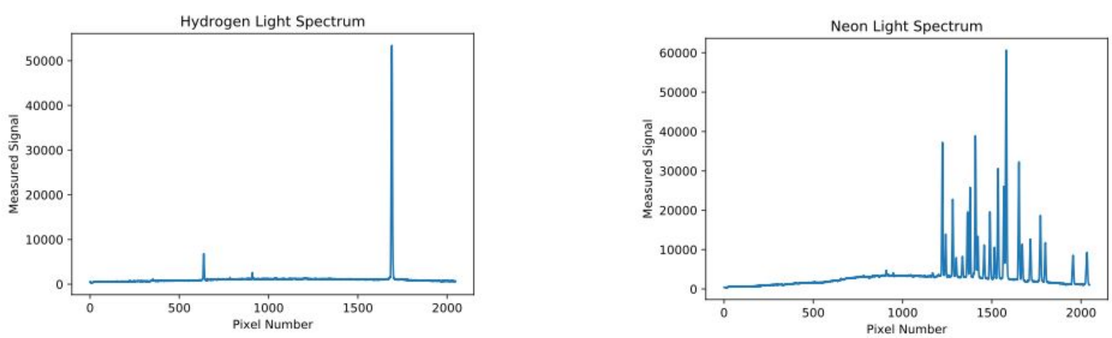

Figure 2: Spectra of Neon and Hydrogen [3]

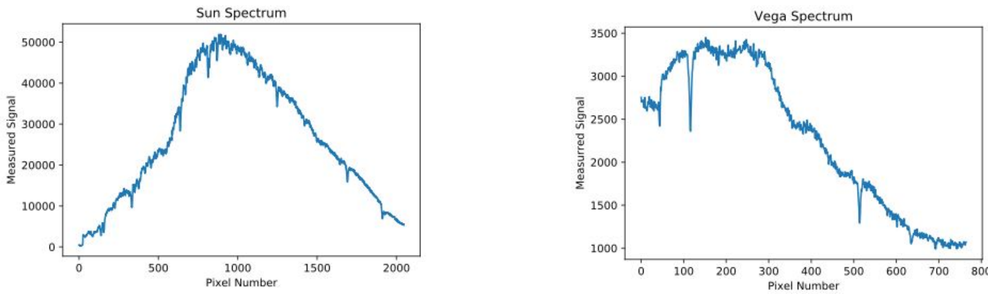

Figure 3: Spectra of the Sun and Vega [3]

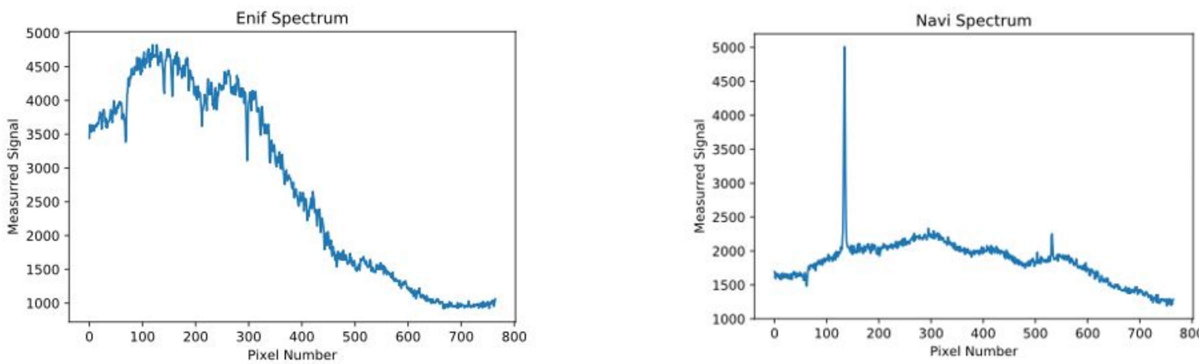

Figure 4: Spectra of Enif and Navi [3]

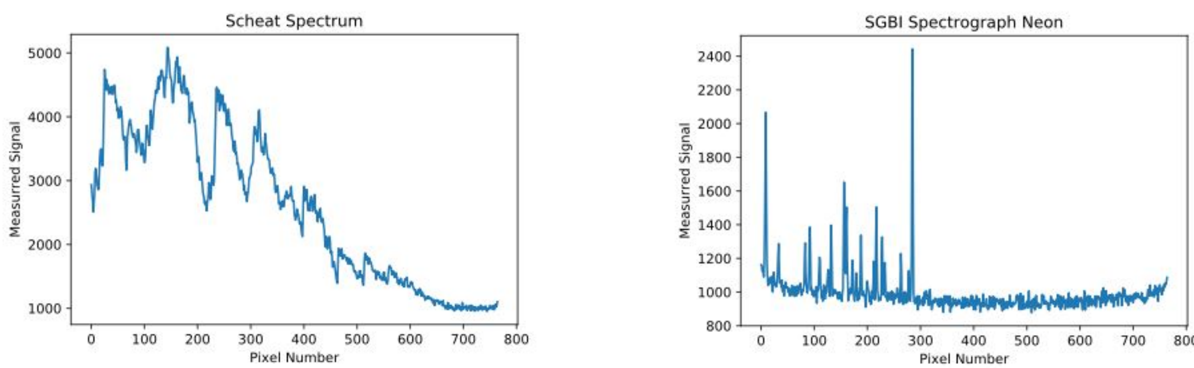

Figure 5: Spectra of Scheat and SGBI Neon [3]

While steps were taken to ensure no other light sources were presented while collecting these spectra there is always an inevitable amount of noise that will get through. Hence, all these spectra have some noise attached to it as a systematic error, but as will be discussed later in this paper specific steps were taken to account for these errors.

3 Data Reduction and Methods

All reduction and statistical manipulation of the manipulation were done using Python code. The Ocean Optic CCD data was copied from the text file into a spreadsheet column which was then imported into Python as a list using the `pandas` module. The SGBI spectrograph data was directly imported to Python as arrays from the `.fit` files using the `astropy` and `numpy` modules.

From here the two spectra that were analyzed in detail were the neon obtained using the Ocean Optics CCD and the neon obtained using the SGBI spectrograph. These two were the focus as they would allow us to find the mapping between pixel number and wavelength needed to understand all the other spectra. The first calculation done on these data sets were the calculation of finding their peaks (local maximums) and valleys (local minimums) using the code found in appendix A. For the SGBI spectrograph neon the data was first reduced using the "Dark" spectrum in order to get rid of the noise caused the by the actual SGBI spectrograph. With these points the pixel position of the centroids of the spectra could be calculated using as weighted mean as follows:

$$\bar{x_w} = \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} \tag{1}$$

Where x_i are the pixel position of of the valleys and the w_i are the measured signal of the valleys. The code for this calculation can be found in appendix B

The resulting centroids pixel positions were then compared to the Ocean Optics handout[2] to see their corresponding wavelengths. The corresponding pairs were plotted on Python, using the `matplotlib` module, with pixel position on the x-axis and wavelength on the y-axis. A linear least square fitting was then applied to this data set as follows:

$$\begin{pmatrix} m \\ c \end{pmatrix} = \begin{pmatrix} \sum_i^N x_i^2 & \sum_i^N x_i \\ \sum_i^N x_i & N \end{pmatrix}^{-1} \begin{pmatrix} \sum_i^N x_i y_i \\ \sum_i^N y_i \end{pmatrix} \tag{2}$$

Where m is the slope of the line, c is the y-intercept of the line, x_i are the pixel positions of the centroids, y_i are the corresponding wavelengths to each centroid, and N is the number of total points. The code for the generation of this line can be found in Appendix C.

The R^2 value of the least square line fittings were computed using the following formula:

$$R^2 = 1 - \frac{\sum (y - y_i)^2}{\sum (y_i - \bar{y})^2} \tag{3}$$

R^2 represents the accuracy of the linear model, y represents the wavelength as calculated by the line created using equation 2, y_i represents the predicted wavelength, and $\bar{y}$ represents the average of all wavelengthsR^2 represents the goodness of the linear model, y represents the wavelength as calculated in Eq. 2, y_i represents the predicted wavelength, and $\bar{y}$ represents the average of all wavelengths

Two main avenues of error were explored in this lab. The first was to do with the line of least square fitting and finding the error of this line's slope and y-intercept follows:

$$\sigma^2 = \frac{1}{N-2} \sum_i^N [y_i - (mx_i + c)] \tag{4}$$

$$\sigma_m = \sqrt{\frac{N\sigma^2}{N \sum_i^N x_i^2 - \left[\sum_i^N x_i \right]^2}} \tag{5}$$

$$\sigma_c = \sqrt{\frac{\sigma^2 \sum_i^N x_i}{N \sum_i^N x_i^2 - \left[\sum_i^N x_i\right]^2}} \tag{6}$$

Where σ_m is the error in the slope of the line, σ_c is the error in the y-intercept of the line, and x_i, y_i, N are the same as above

4 Data Analysis and Models

4.1 Ocean Optics CCD Data

The spectrum that was of main focus was that of neon as finding a line of least squares for it would allowed us to create a mapping from pixel count to wavelength that could be applied to all other recorded spectra. The peaks and valleys of the neon spectrum were found using the weighted mean method mentioned in the previous part of the paper (equation 1). These points were matched with their corresponding wavelengths according to the lab handout [2]. A linear least square fitting was done on these points using the methods outlined in the previous section of this paper (equation 2). These points and the linear least square fitting line were then plotted using the `.mathplotlib` module on Python, the code for this plotting can be found in appendix C.

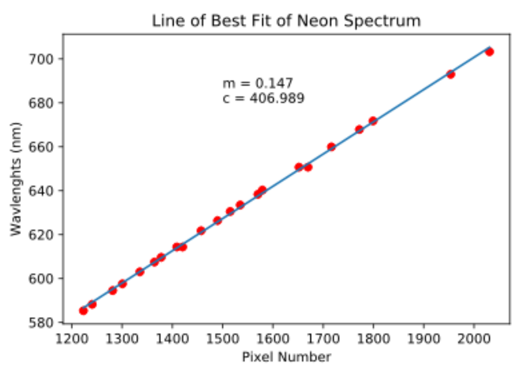

Figure 6: Pixel Number to Wavelength Mapping using Neon Spectrum Centroids

From least square fitting line we can see that the mapping between pixel number and wavelength is a linear one. It is also a very close relation with an R^2 value of 0.999999994821. The code used to calculate this value can be found in appendix D in accordance to equation 3. The errors calculated for m and c using equations 4,5 and 6 are also relatively small at 0.00097 and 1.51 respectively. The code for the error calculation can be found in appendix E.

4.2 SGBI Spectrograph Data

Similarly to the Ocean Optics neon spectrum the same calculations were done to the neon spectrum recorded with the SGBI spectrograph. The peaks and valleys were found using the code in appendix A and then the centroids were found in accordance with equation 1, using the code found in appendix B. The centroids were then matched with their respective wavelengths according to the lab handout[2]. With this a linear least square fitting could be found, using equation 2 and the code in appendix C, in order to find the mapping between pixel count and wavelength for the SGBI spectrograph.

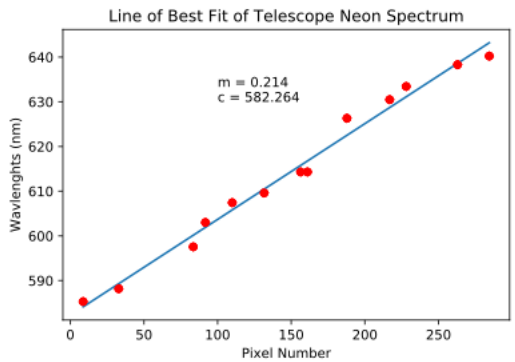

Figure 7: Pixel Number to Wavelength Mapping using Neon Spectrum Centroids (SGBI spectrograph)

From least square fitting line we can see that the mapping between pixel number and wavelength is a linear one. It is also a very close relation with an R^2 value of 0.999999923811. The code used to calculate this value can be found in appendix D in accordance to equation 2. The errors calculated for m and c using equations 4,5 and 6 are also relatively small at 0.0075 and 1.28 respectively. The code for the error calculation can be found in appendix E.

5 Discussion

5.1 Ocean Optics CCD Data

From the previous section of this paper we can conclude the Ocean Optics CCD maps the pixel count to wavelength linearly. More specifically the mapping is given by the function $y = 0.147x + 406.989$ where x is the pixel count of a peak (or valley depending if its an emission spectrum or absorption spectrum) on the spectrum and y is the corresponding wavelength. This relationship was found to be quite strong as it had an R^2 value very close to 1, 0.999999994821, meaning the line was almost a perfect fit to the data. Additionally, the error calculated on the slope and y-intercept were both also very small compared to their calculated value. The slight shift making this data not perfectly follow a linear fit can be attributed to noise when the data collecting was taking place. Such noise from other light sources might have lead to peaks appearing where they should not be, or making them bigger or smaller by some factor making slight differences in the spectrum. Other group members also computed polynomial fits to the data but found them to be only marginally better than a linear fit, but considering the simplicity of a linear relation it was used as the preferred fit.

With this information we could then find information about the emitted wavelengths of the other sources that were recorded using the Ocean Optics CCD. By simply running the code found in appendix A and appendix B we could find the centroids of all the other spectra. These centroids could then be plugged in to the line of best fit to see which wavelengths were being emitted by these sources.

5.2 SGBI Spectrograph Data

Similarly to the Ocean Optics CCD we can conclude the SGBI spectrograph maps the pixel count to wavelength linearly from the previous section of this paper. More specifically the mapping is given by the function $y = 0.214x + 584.264$ where x is the pixel count of a peak (or valley depending if its an emission spectrum or absorption spectrum) on the spectrum and y is the corresponding wavelength. This relationship was found to be quite strong as it had an R^2 value very close to 1,0.99999992381, meaning the line was almost a perfect fit to the data. Additionally, the error calculated on the slope and y-intercept were both also very small compared to their calculated value. The slight shift making this data not perfectly follow a linear fit can be attributed to noise when the data collecting was taking place. Such noise from other light sources might have lead to peaks appearing where they should not be, or making them bigger or smaller by some factor making slight differences in the spectrum. Other group members also computed polynomial fits to the data but found them to be only marginally better than a linear fit, but considering the simplicity of a linear relation it was used as the preferred fit.

With this information we could then find information about the emitted wavelengths of the other sources that were recorded using the SGBI spectrograph. By simply running the code found in appendix A and appendix B we could find the centroids of all the other spectra. These centroids could then be plugged in to the line of best fit to see which wavelengths were being emitted by these sources. To demonstrate this we found the valleys of the spectrum of Vega using the code in appendix A and then plotted as follows:

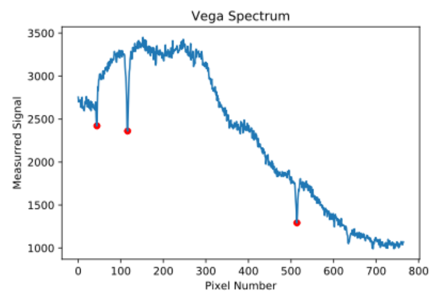

Figure 8: Valleys of Vega Spectrum

Printing out the valley locations from the code we get they are at pixel number 44, 116, and 514. If we plug these values into our line of best fit we find the corresponding wavelengths to be $593.68nm, 609.088nm$ and $694.26nm$ respectively. These three wavelengths line up with hydrogen and hellium emission wavelengths which makes sense considering Vega is main sequence star which is fusing hydrogen into helium at its core.

6 Conclusions

In closing, this lab focused on measuring the spectra of different light sources. An Oceans Optics CCD was used to measure the spectra of various man-made light sources, while an SGBI spectrograph that was attached to a 16-inch telescope was used to measure the spectra of various astronomical sources. In both cases a neon light source was observed in order to find a mapping between the pixel number on the CCD and the corresponding wavelength of light. It was found that both CCDs have a mapping between pixel number and wavelength of light that closely follows a linear relation using a linear least square fitting method. The error of this method was also calculated and found to be relatively small, hence making the linear relation a good fit for this mapping.

7 Appendix

7.1 Appendix A: Peak and Valley Code

This code is run on the list that contains the spectrum which you are trying to find the local maximums and minimums of.

```
#Peaks
k=1 #index for while loop
NP=[] #empty list for measured intensity of peaks to be placed in
NPL=[] #empty list for pixel positon of peak to be placed in
thresh2=7800 #threshold that should be slightly less than your lowest peak's
             #measured intensity
```

```
#While loop to find the peak's measured intensity and peak's pixel position
while k<len(N)-1:
    if N[k]>N[k+1] and N[k]>N[k-1] and N[k]>thresh2: #defining conditions for a peak
        NP.append(N[k]) #Adding peaks' intensity to appropriate list
        NPL.append(k) #Adding peaks' pixel position to appropriate list
    k=k+1
print('These are peaks for this spectra:',NP)
print('These are peak positions for this spectra:',NPL)


#Define two functions that will help us find valleys

#Shift inputed value 7 units to the left
def LS1(z):
    LeShift1=z-7
    return (LeShift1)

#Shift inputed value 8 units to the right
def RS1(o):
    LeShift2=o+7
    return (LeShift2)


#Valleys
NV=[] #empty list for measured intensity of valleys to be placed in
NVL=[] #empty list for pixel position of valleys to be placed in
i=0
while i<len(NPL):
    NV.append(N[LS1(NPL[i])]) #Shifts peak pixel position left and finds
                              #corresponding measured intensity
    NV.append(N[RS1(NPL[i])]) #Shifts peak pixel position right
                              #and finds corresponding measured intensity
    NVL.append(LS1(NPL[i])) #Shifts peak pixel position left
    NVL.append(RS1(NPL[i])) #Shifts peak pixel position right
    i=i+1

print('These are valleys for this spectra:',NV)
print('These are valley positions for this spectra:',NVL)
```

7.2 Appendix B: Centroid Pixel Value Code

This code is run on the list containing your valleys pixel position and the list containing their measured signal. The code in turns returns the pixel position of the centroids of the spectrum

```
#Weighted Mean for Some Spectra
d=0 #Create an index
Centroids=[] #Empty list to record centroid pixel positons
#While loop that excecutes the weighted mean calculation
#Refer to Appendix A to see what the NV and NVL list contain
while d<len(NV):
    #Weighted mean calculation with the valley positions being the x values
    #and their measured intensity being the weights
    Centroids.append(((NV[d]*NVL[d])+(NV[d+1]*NVL[d+1]))/(NV[d]+NV[d+1]))
    d=d+2 #pairing of valleys to find only one centroid per pair
print('These are centroids of this spectra:', Centroids)
```

7.3 Appendix C: Linear Least Square Fitting Code

This code uses your list containing the centroids pixel positions and your list containing the respective wavelengths in order to create a line of best fit

```
import numpy as np
import matplotlib.pyplot as plt

#Define a List with Centroids Pixel Position called "Centroids"
#Define a List that has every element in Centroids but squared called CentroidsSquared
#Define a List that contains all the corresponding wavlengths of the centroids
#called "Wavelengths"



#Define Matricies for Linear Fitting
ma=np.array([[sum(CentroidsSquared),sum(Centroids)],[sum(Centroids),len(Wavelength)]])
mc=np.array([[sum(CentroidsWavlenght)],[sum(Wavelength)]])
#Inverse of ma
mai=np.linalg.inv(ma)
#Slope and y-intercept
md=np.dot(mai,mc)

# Overplot the best fit
#Extract m and c using their matrix element
m = md[0,0]
c = md[1,0]

#While loop that generates the line of best fit
i=0
TLB=[] #empty list where y values of line will go
while i<len(Centroids):
    TLB.append(m*Centroids[i] +c) #Generating y values for the line
    i=i+1

#Creating Plot for Line of Best Fit
plt.subplot()
plt.plot(Centroids,LB) #Plots line of best fit
plt.plot(Centroids,Wavelength, 'ro') #Plots points of centroids versus wavelength
#Prints m and c value on the graph
plt.text(100,630,'m = {:.3f}\nc = {:.3f}'.format(m,c))
plt.title('Line of Best Fit of Neon Spectrum')
plt.xlabel('Pixel Number')
plt.ylabel('Wavlenghts (nm)')
plt.show()
```

7.4 Appendix D: R^2 Value Code

This code uses the list containing the y values of your line of best fit and the list containing your centroids pixel positions as well as the list containing their respective wavelengths in order to calculate the R^2 of your line of best fit.

```
#R Squared Value Calculation
averageY=sum(LB) #Using Line of Best Fit to Calculate Average Y
```

```
w=0
ExpPred=[] #Empty List where numerator of line of R^2 eqaution will be put
while w<len(LB):
    #Wavelenght is the same list defined in the previous appendix
    ExpPred.append((LB[w]-Wavelenght[w])**2)
    w=w+1

w=0
PredAv=[] #Empty List where denominator of line of R^2 eqaution will be put
while w<len(LB):
    PredAv.append((Wavelenght[w]-averageY)**2)
    w=w+1

rSquared=1-(sum(ExpPred)/sum(PredAv))
print("The R-Squared value for the Line Of Best Fit is",rSquared)
```

7.5 Appendix E: Linear Least Square Fitting Error Code

This code uses the list containing the y values of your line of best fit and the list containing your centroids pixel positions as well as the list containing their respective wavelengths in order to calculate the error in the slope and y-intercept of the line of best fit.

```
#Error Propogation for Line Of Best Fit
ErrorList=[] #Empty list for calculated error to go into
g=0
while g<len(LB):
    ErrorList.append((LB[g]-NW[g])**2)
    g=g+1

#Computation for sigma that appears in equation 3
sigmaSquared=(1/(len(LB)-2))*sum(ErrorList)
sigma=(sigmaSquared)**(1/2)
print("The Error Propogation Value (sigma) is", sigma)

#Computation for sigma M that appears in equation 4
#Sigma M
sigmaM2=((len(CentroidsN))*(sigmaSquared))/((len(Centroids))
*(sum(CentroidsSquared))-(sum(Centroids))**2)
sigmaM=(sigmaM2)**(1/2)
print('This is the error of m:',sigmaM)

#Computation for sigma C that appears in equation 5
#Sigma C
sigmaC2=((sigmaSquared)*(sum(Centroids)))/((len(Centroids))
*(sum(CentroidsSquared))-(sum(Centroids))**2)
sigmaC=(sigmaC2)**(1/2)
print("This is the error of c:", sigmaC)
```

References

[1] Spectrometer guide. URL http://www.astro.utoronto.ca/ astrolab/.

[2] Ocean optics calibration data. URL http://www.astro.utoronto.ca/ astrolab/.

[3] Group-G. Ast325 lab 2 spectroscopy raw data. URL https://bit.ly/2qid8r5.

[4] D.-S. Moon. Astronomy 325/326 course website. URL `http://www.astro.utoronto.ca/ astrolab/`.