Lab 2: Introduction to Spectroscopy

Victoria Spada

October 31, 2022

Abstract

Spectroscopy is a tool that is ubiquitous in astronomy and astrophysics for determining wavelength properties of measured sources. In this report, wavelength calibration of 1-D and 2-D spectrograph measurements were completed. First, a 1-D spectrum from a USB4000 spectrometer was calibrated using a measured spectrum of a neon lamp, which had distinct peaks that were mapped to known wavelengths after determining their centroids. This calibration was done as an ordinary least squares regression, and was used to calculate the temperature of a blackbody whose spectrum was measured by the same spectrograph.

A 2-D dispersed image of OH telluric sky lines was used to calibrate 2-D spectrum measurements to wavelength. This was completed using first- and second-order least squares regression. This calibration was then applied to calculate the velocity of ionized iron gas from a supernova explosion. The resulting velocity was found to be 1060 ± 130 km/s.

Finally, measurements from the Shelyak Alpy spectrograph were taken and the standard data reduction process was used to extract the spectra of different astronomical bodies. The wavelength calibration, done with a neon spectrum, was found to be better suited to a quadratic fit, though uncertainties were found to produce smaller residuals but the uncertainties were high. Resulting spectra from Aldebaran, Elnath, Jupiter, and Mars were compared.

1 Introduction

Spectrometers use a light source, a dispersing element, and a detector to measure emitted wavelength spectra from source [1]. In particular, the tool is used in the field of astronomy to study and characterize astronomical bodies. From measured spectra, information on the chemical composition and physical characteristics of the source, such as temperature, pressure, and velocity, can be determined. An important step in extracting meaningful information from these spectral measurements is wavelength calibration, by which the relationship between pixel location on the detector is related to the measured wavelength [2, 3, 4].

The purpose of this report is to complete wavelength calibrations for three spectrographs and apply the calibration to make meaningful analysis of spectrum measurements. A blackbody spectrum measured by a spectrograph can be used to determine the blackbody temperature, using Wien's displacement law (eqn. (1)), where T is temperature in [K] and λ_{peak} is the wavelength of the spectrum corresponding to maximum intensity [5].

$$\lambda_{peak} T = 2.898 \times 10^{-3} m \cdot K \tag{1}$$

Using the calibration of OH telluric sky lines, the velocity of a source can be found. This can be done by using the concept of Doppler shifts, shown in eqn. (2), where the intrinsic wavelength of the source is λ_0, Δ is the Doppler shift in wavelength, and c is the speed of light, 3.0×10^8 m/s [4].

$$\frac{\Delta\lambda}{\lambda_0} = \frac{v}{c} \tag{2}$$

Measured spectra of astronomical bodies can also be used to infer absorption lines/features and the relative brightness of those bodies.

2 Observations and Data

A spectrum of a blackbody taken from a USB4000 spectrograph was provided. To calibrate the spectrograph, a 1-D spectrum measurement of a neon lamp (taken from the same spectrograph) was provided, along with a spectrum of neon calibrated to wavelength, which is well-established in the scientific community [2].

A data file of a 2-D dispersed image of ionized iron gas from an exploding supernova was also provided as a .fits file. In addition to this, a spectrum of OH telluric sky lines was provided for use in calibration [2, 3].

On the night of October 25, 2022, measurements of Aldebaran, Elnath, Jupiter, and Mars were taken from the Burton Tower at the University of Toronto, using a Shelyak Alpy spectrograph. Additionally, three .fits files of the spectrograph bias, one flat measurement, and three dark measurement for each time interval of the object measurements were collected. A spectrum measurement of neon was provided for calibration purposes.

3 Data Reduction and Methods

One of the main objectives was to perform wavelength calibration for 1-D and 2-D measurements from spectrographs. The methods involved for the three calibrations were similar and are discussed below.

From the provided 1-D USB4000 neon spectrum measurement, distinct peaks were identified using a reference table provided in the lab manual [2]. From the 28 local maxima in the measurement, 11 distinct peaks were identified using the reference spectrum. These peaks are listed in Table (1). The centroids of these peaks were then determined. The centroid is defined in eqn. (3),

$$x_{cm} = \frac{\sum x_i I_i}{\sum I_i} \tag{3}$$

where x_{cm} is the centroid pixel value, x is the pixel array, and I is the measured intensity array.

The full-width half-maximum (FWHM) values of the identified peaks were used to determine the peak widths (ie, the range of indices i used in computing the centroid). Relevant code for this aspect of data reduction is in Appendix A. One prominent peak at the high end of the spectrum had to be discarded because it did not reach it's half-max height on the left side, so the peak width could not be included (see Appendix B). The determined centroids are shown in Table (1).

Wavelength [nm]	Centroid Location [pixel]
540.056	53
585.249	241
640.225	475
650.653	522
692.947	698
703.241	742
717.394	804
724.512	830
743.89	916
748.887	928
753.577	954

Table 1: Wavelengths identified on neon lamp spectrum measured by the USB4000 spectrograph, and corresponding pixel locations of the peaks on the measurement. The centroids are plotted on Figure (7a).

Wavelength [Å]	Centroid Location [pixel]
16128.608	71
16194.615	84
16235.376	98
16317.161	115
16350.650	126
16442.155	157
16502.365	182

Table 2: Wavelengths identified on OH telluric sky spectrum and centroids of the corresponding peaks on the y-axis of the dispersed image measurement. The centroids are plotted on Figure (7b).

Wavelength [nm]	Centroid Location [pixel]
540.056	178
585.249	412
640.225	575
650.653	591
659.895	639
692.947	657
703.241	678
717.394	691

Table 3: Wavelengths identified on neon spectrum from the Shelyak Alpy spectrograph, and centroids of the corresponding peaks on the y-axis. The centroids are plotted on Figure (7c).

A similar data reduction approach was done for the 2-D dispersed image measurement of the ionized iron gas. The measurement dimension was brought down to 1 by taking the median value in the y-axis from pixels in $[135, 150]$, and taking the resulting median value from that section as the corresponding intensity for each pixel on the x-axis. This resulted in a 1-D spectrum of the same format as the neon lamp spectrum previously discussed. The same method as was used for the neon lamp spectrum was used to locate seven distinct OH telluric sky peaks [2]. The seven resulting peaks are shown in Table (2).

Spectra from the Shelyak Alpy spectrograph were reduced by dividing by a flat image and subtracting bias and dark. A measurement spectrum of neon was provided to complete calibration of the spectrograph. To accomplish this, the .fits data file had to be modified.

The bias, or voltage, of the charge-coupled detector (CCD) must be subtracted from the measurement. Bias is a time independent signal, so a bias measurement of any exposure level can be subtracted from a spectral measurement to properly account for it. Dark is another factor that must be subtracted from the measurement. The dark of a measurement includes the electrons produced by the silicon detector itself, not the source being measured. Dark is time-dependent, so to subtract the dark from a measurement, a dark measurement with the same exposure time as the measure-

ment of interest must be subtracted. Dark measurements also intrinsically include bias, so only dark measurements were subtracted for the data reduction stage [6]. Three dark measurements of the proper exposure time were combined, and the median was selected to create a "master" dark array that was subtracted from the measured spectra.

Then, measurements from the Shelyak Alpy spectrograph were flat-fielded. This was done to remove the problem of the spectrograph detector having a non-uniform sensitivity. The image for flat fielding is of a bright light, which is then divided out of the measurement of interest to calibrate it to a uniform sensitivity [6].

Finally, the measurements observed using the Shelyak Alpy spectrograph were compressed into one dimension by taking the median y-axis value over a band of the spectrum, for each pixel on the x-axis. The selected range on the y-axis was unique to each measurement, depending on where the spectrum visibly fell. The measurement of the neon spectrum included pixels $y \in [150, 250]$ in its medians. The Aldebaran spectrum included y-axis pixels within $y[225, 235]$ in its medians. The measurement of Elnath included y-axis pixels within $y \in [270, 280]$ in its medians. The measurements of the Jupiter and Mars spectra included y-axis pixels within $y \in [175, 200]$ and $y \in [150, 160]$ in its medians.

Peaks on the neon spectrum obtained from the Shelyak Alpy spectrograph were found using the same method as described above for the 1-D USB4000 neon spectrum measurement. The resulting peaks, which were used to calibrate the wavelength-pixel relation for the spectrograph, are shown in Table (3).

4 Data Analysis and Modelling

The 1-D USB4000 neon spectrum, the OH telluric line spectrum, and the 2-D Shelyak Alpy neon spectrum were fit with a linear wavelength solution, of the form $y = mx + c$. The parameters m and c were determined using Ordinary least squares (OLS) regression, as shown in eqn. (4), where N is the number of data points. This method of fitting minimizes the absolute difference between the fit and the data, hence the name "least squares" [7].

$$\begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & N \end{bmatrix}^{-1} \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix} \tag{4}$$

The uncertainties of the OLS fit parameters were determined using equations (5) and (6), where σ_m is the uncertainty of the parameter m, σ_c is the uncertainty of parameter c, and σ is the standard deviation of the fit. Since the standard deviation of the fit is not known a priori, it is estimated using eqn. (7), where $\sigma = \sigma_{O1}$ [7]. A more in-depth discussion of these uncertainties is in Appendix C.

$$\sigma_m^2 = \frac{N\sigma^2}{N\sum x_i^2 - (\sum x_i)^2} \tag{5}$$

$$\sigma_c^2 = \frac{\sigma^2 \sum x_i^2}{N\sum x_i^2 - (\sum x_i)^2} \tag{6}$$

$$\sigma_{O1}^2 = \frac{1}{N-2} \sum_i [y_i - (mx_i + c)]^2 \tag{7}$$

USB4000 Neon Spectrum [nm]	
Model: $y = mx + c$	
m	0.237 ±0.001
c	528 ±1
Δ_{abs}	0.426 ±0.123

Table 4: OLS fit results for calibration of the neon lamp spectrum.

OH Telluric Line Spectrum	
Model: $y = mx + c$ [Å]	
m	3.36 ±0.18
c	15900 ±500
Δ_{abs}	12.6 ±3.0

Table 5: OLS fit results for calibration of the OH telluric line spectrum.

Shelyak Alpy Neon Spectrum [nm]	
Model: $y = mx + c$	
m	0.332 ±0.039
c	465 ±23
Δ_{abs}	15.3 ±1.4

Table 6: OLS fit results for calibration of the Shelyak Alpy neon spectrum.

OH Telluric Line Spectrum [Å]	
Model: $y = qx^2 + rx + s$	
q	-0.012 ±0.262
r	6.29 ±4.16
s	15700 ±31
Δ_{abs}	3.77 ±1.45

Table 7: Quadratic fit results for calibration of the OH telluric line spectrum.

Shelyak Alpy Neon Spectrum [nm]	
Model: $y = qx^2 + rx + s$	
q	0.000616 ±0.006
r	-0.2103 ±5.17
s	560 ±890
Δ_{abs}	5.66 ±1.82

Table 8: Quadratic fit results for calibration of the Shelyak Alpy neon spectrum.

The results for the OLS fits are shown in Tables (4), (5), and (6). The mean absolute difference Δ_{abs} of the fit is defined in eqn. (8) where $y(x_i)$ is the data point at x_i and $f(x_i)$ represents the fit at x_i. The implementation of eqn. (8) is discussed further in Appendix D.

$$\Delta_{abs} = \frac{1}{N}\sum_{i}^{N} |y(x_i) - f(x_i)| \tag{8}$$

The parabolic wavelength solutions found for the OH telluric sky spectrum and the Shelyak Alpy neon spectrum were modelled using quadratic least squares regression for a solution of the form $y = qx^2 + rx + s$. The quadratic system shown in eqn. (14) was solved to determine the parameters of the model [8].

A discussion of the uncertainties for these parameters is in Appendix E. The fit results are shown in Tables (7) and (8).

$$\begin{bmatrix} q \\ r \\ s \end{bmatrix} = \begin{bmatrix} \sum x_i^4 & \sum x_i^3 & \sum x_i^2 \\ \sum x_i^3 & \sum x_i^2 & \sum x_i \\ \sum x_i^2 & \sum x_i & n \end{bmatrix}^{-1} \begin{bmatrix} \sum x_i^2 y_i \\ \sum x_i y_i \\ \sum y_i \end{bmatrix} \tag{9}$$

Using the calibration of the spectrograph determined from the 1-D neon lamp measurement from USB4000, a measurement of a blackbody from the same spectrograph was calibrated to wavelength. This is shown in Figure (5). The temperature of a measured blackbody was found according to Wien's displacement law, eqn. (1) [5]. The blackbody temperature is found by using the maximum of the measurement, λ_{peak}. The temperature was found to be 3763 ± 4 K.

Using the calibration of the OH telluric lines, the velocity of the ionized gas, v, was found. This was done using the concept of Doppler shifts, shown in eqn. (2), where the intrinsic wavelength of the source is $\lambda_0 = 1.644\ \mu m$, $\Delta\lambda$ is the Doppler shift in wavelength, $\lambda_0 - \lambda_{obs}$, and c is the speed of

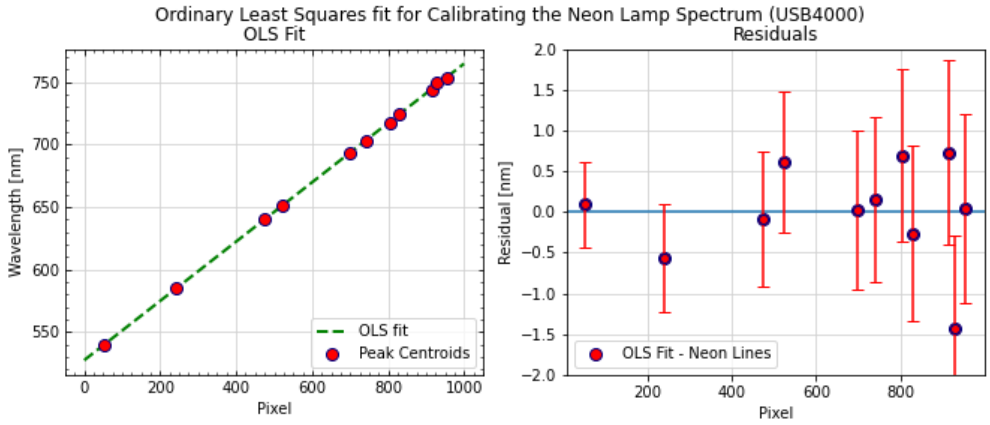


Figure 1: Centroids of neon spectrum peaks as a function of pixel number, fitted to a straight line using OLS regression, and the resulting residuals.

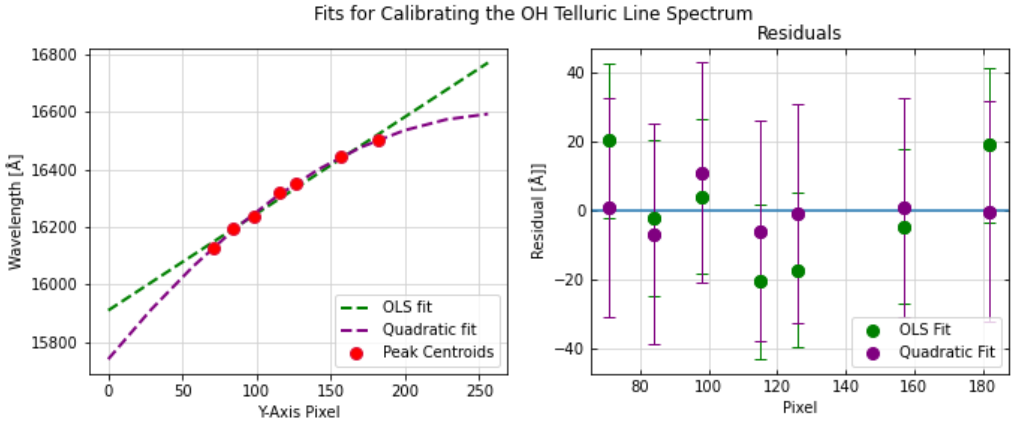


Figure 2: Centroids of OH telluric spectrum peaks as a function of pixel number, fitted to a straight line using OLS regression, and the resulting residuals.

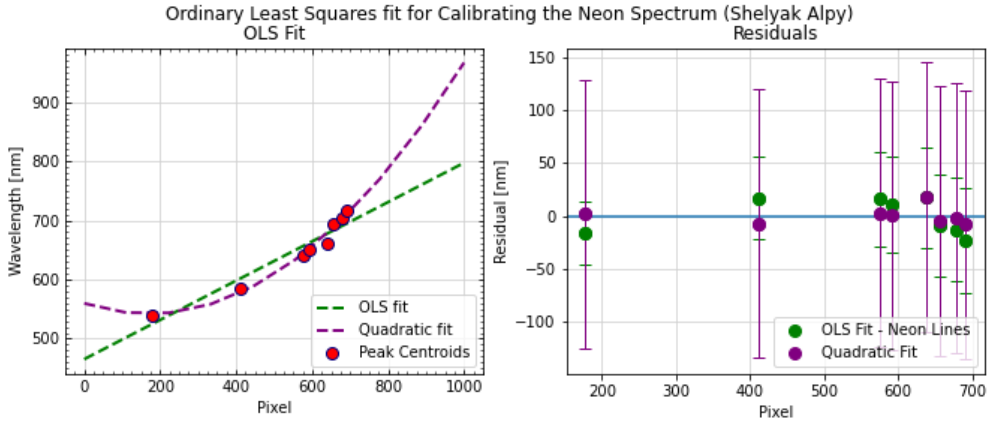


Figure 3: Centroids of neon spectrum peaks, measured by the Shelyak Alpy spectrograph, as a function of pixel number, fitted to a straight line using OLS regression, and the resulting residuals.

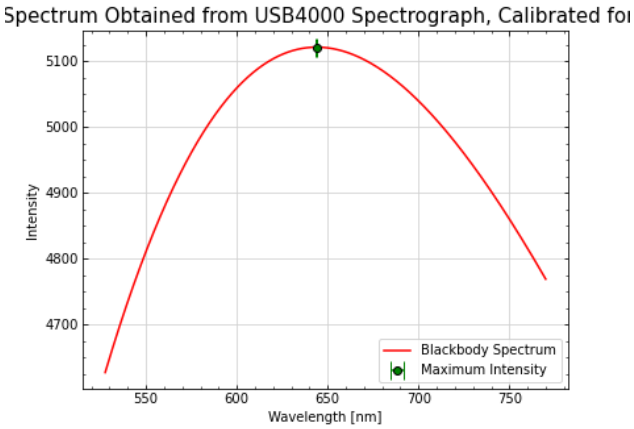


Figure 5: Blackbody spectrum measurement, calibrated from pixels to wavelength.

light, 3.0×10^8 m/s. To find the value of $_{obs}$ the measurement was first cropped to $y \in [178, 183]$, $x \in [103, 121]$. Then, the median values across the y axis were found for each x-pixel in the selected range. Finally, the centroid of the distribution of medians was used to determine the x-pixel corresponding to the centroid of the measurement. The quadratic fit in Table (7) was used to determine the corresponding wavelength, which was used in eqn. (2) to calculate λ_{obs}.

The velocity of the iron gas source was found to be 1060 ± 130 km/s with the quadratic calibration and 1370 ± 100 km/s. The cropped iron gas emission from the 2-D is shown in Figure (4).

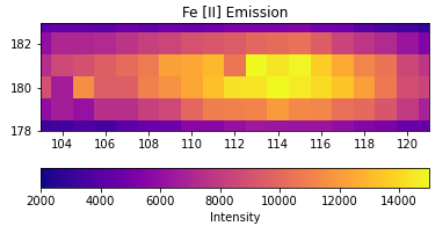


Figure 4: Fe II gas measurement, zoned in on pixels $y \in [178, 183]$, $x \in [103, 121]$.

5 Discussion

The USB4000 wavelength calibration using the 1-D neon spectrum was found to be a good fit. The fit shown in Figure (1) shows that the distribution appears linear, and the residuals of the fit show that all but one of the identified centroids intersects with the zero point, which demonstrates a good fit. The mean absolute difference was found to be 0.426 ± 0.123 nm, which is relatively small compared to order of magnitude of the wavelengths being studied, which are on the order of 500-800 nm. The fit could be improved by decreasing the uncertainty. This could be done by selecting more distinct peaks from the neon spectrum used for calibration. Only 11 peaks were identified for this calibration, selecting only the most distinct peaks, but with a larger dataset the fit would be more robust.

Figure (5) shows the resulting of calibrating the pixel number to wavelength. The wavelength corresponding to the maximum intensity was found to be 643 ± 0 nm. Using Wien's displacement Law, the blackbody temperature was found to be 3763 ± 4 K. This could correspond to a red or a yellow-orange star (spectral types K or M).

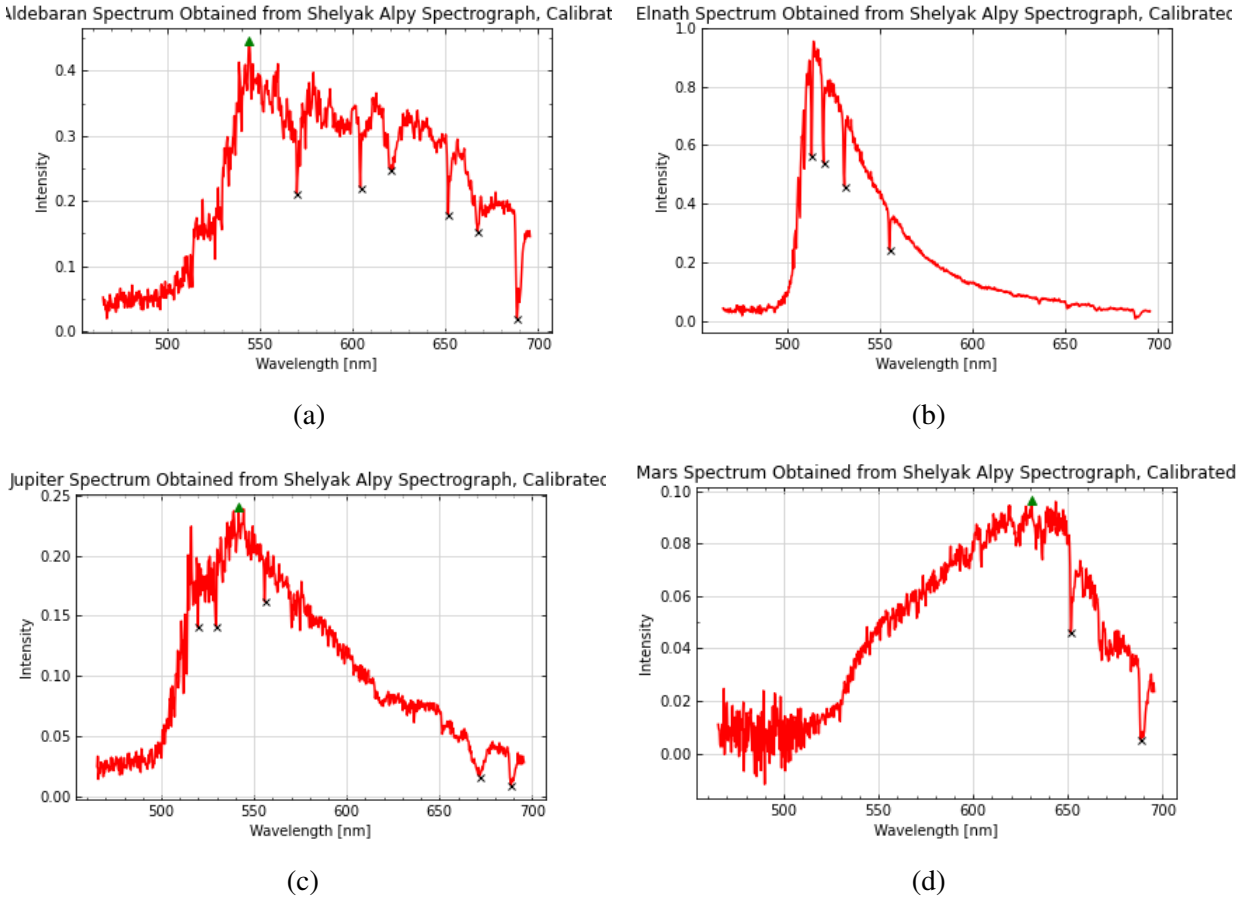


Figure 6: Spectra of astronomical bodies taken by Shelyak Alpy spectrograph. The pixel numbers were fit to the quadratic model shown in Table (8). Local minima showing absorption peaks are marked with x's. Global maxima are indicated with green triangles.

The 2-D spectrograph measurements were found to be better calibrated with a quadratic fit than a linear fit. The fits are shown in Figure (3). The residuals of the linear fit are generally larger than the residuals of the quadratic fit. They do not appear to be randomly scattered around zero and instead look quadratic, indicating that a second-order polynomial would be a better fit. The mean absolute difference for the quadratic fit was found to only be 3.77 ± 1.45, while for the linear fit, it is 12.6 ± 3.0. It is important to note that only 7 peaks were identified for this calibration, selecting only the most distinct peaks and excluding peaks that appeared to be lumped beside each other. With a larger dataset, the fit could be more robust, but the closeness/lumping together of the added peaks could also be a source of high uncertainties. Th quadratic fit appears to be more appropriate, but with a small sample of peaks being fitted, the uncertainty is quite high.

With a more robust fit, the value of λ_{obs} used to determine the galaxy velocity would be more accurate. There are many other sources of uncertainty for the calculated velocity that are not well-accounted for. For example, the section of pixels used to include in determing the iron gas velocity was selected by eyeballing the measurement of the iron gas emission, and a slightly different selection would yield a different final velocity.

Calibrating the Shelyak Alpy measurements with the neon spectrum provided, it was found that

the quadratic fit yielded more randomly scattered residuals and a lower mean absolute difference than the linear fit, however the resulting parameters are extremely uncertain compared to the OLS fit. Due to these high uncertainties, the linear fit was used to determine the spectra of Aldebaran, Elnath, Mars, and Jupiter. The resulting spectra are shown in Figures (6a), (6b), (6c), and (6d).

In Figure (6a), the star Aldebaran shows the most emissions between around 540-600 nm. These wavelengths correspond to the yellow/orange region on the visible spectrum. This is expected, as Aldebaran is of spectral type K5 III and is an orange giant [2]. The peaks marked with x's on the spectrum show wavelengths where absorption was high (ie, those wavelengths are not emitted, but absorbed by the measured object).

The Elnath spectrum is more pointed (Figure (6b)), with a global maximum at 515 ± 1 nm. The intensities captured in this measurement are on a greater scale than for Aldebaran, which indicates that Elnath is the brighter star and is also hotter. Elnath is of spectral class B7 III [2].

In Figure (6c), the spectrum of Jupiter shows a few absorption peaks in the yellow/green visible area. The global emission peak is located at 542 ± 1 nm. This is unexpected because this corresponds to a visible cyan or green colour, but Jupiter is an orange planet. This may be due to the poor fit of the spectrum (ie, the choice of using the linear fit). In the red/near-infrared region there are two major, broad absorption peaks.

The spectrum of Mars, shown in Figure (6d) shows that emissions are highest in the 600-650 nm region, which corresponds to orange/red visible light. This is expected because Mars is a red planet. The scale on this spectrum is only around half of the scale on the spectrum of Jupiter, which is indicative of how much brighter Jupiter is in the night sky compared to Mars.

6 Conclusions

Spectroscopy is a useful tool that has numerous applications in astronomy and astrophysics, as it enables researches to measure the spectra of astronomical objects and extract chemical and physical information about the measured source. In this report, wavelength calibrations for 1- and 2-dimensional measurements were performed for three spectrographs. With small datasets being calibrated, the results were more uncertain when fit to a second-degree polynomial because there were more variables in the system. The calibrations performed in this report were used to determine the temperature of a blackbody using Wien's displacement Law, as well as the velocity of iron gas from a supernova explosion using the Doppler Shift relation. Finally, measured spectra of four astronomical bodies were compared and contrasted for their brightness and temperatures.

APPENDIX A: Calculating the Centroid of a Peak

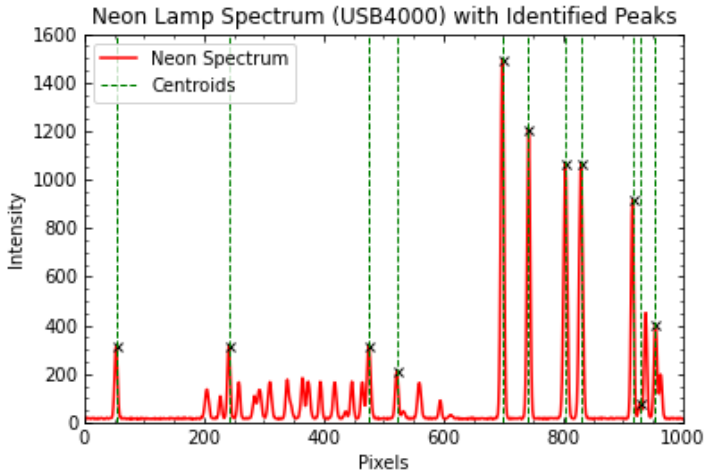
Distinct peaks corresponding to known wavelegnths were identified using a reference spectrum provided in the lab manual. The centroids of these peaks can be determined to better select the pixel value of the peak. The full-width half-maximum (FWHM) values of the identified peaks were used to determine the peak widths (ie, the range of indices i used in computing the centroid). A snippet showing how this is accomplished in python is shown below. The code was taken and edited from [9].

```
# First locate local maxima
peaks = [] # Pixel number at peak
peak_heights = [] # Signal value at peak
threshold = 50

# Iterate over signal array in search of neon peaks
for j in range(len(neon)- 4):
    i = j+2
    # Conditions that need to be met
    threshold_met = neon[i] > threshold
    higher_than_neighbour = (neon[i] > neon[i-1])&(neon[i] > neon[i+1])
    higher_than_neighbour2 = (neon[i] > neon[i-2])&(neon[i] > neon[i+2])
    if threshold_met & higher_than_neighbour & higher_than_neighbour2:
        peaks.append(pixels[i])
        peak_heights.append(neon[i])

# Locate the centroids
centroids = []
peakwidth = 6
fwhms = []

# Find FWHM for each peak
peaks_covered = 0
for i in range(0,len(neon),1):
    curr_peak = peaks[peaks_covered]
    curr_peak_height = peak_heights[peaks_covered]
    curr_half_maximum = curr_peak_height/2
    curr_peak_width = 0
    if neon[i] >= (curr_half_maximum):
        while neon[i] >= (curr_half_maximum):
            curr_peak_width += 1
            i += 1
        fwhms += [curr_peak_width]
        peaks_covered += 1
        if peaks_covered == len(peaks):
            break
```

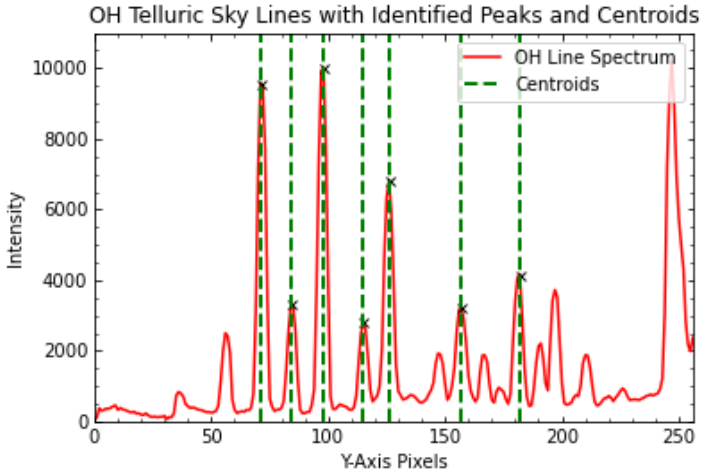
```
# Now compute the centroid using the FWHMs
for i, peak in enumerate(peaks):
    peakwidth = fwhms[i]
    indexmin = int(np.where(pixels==peaks[i])[0][0] - peakwidth/2)
    indexmax = int(np.where(pixels==peaks[i])[0][0] + peakwidth/2)
    x_range = pixels[indexmin: indexmax+1]
    I_range = neon[indexmin: indexmax+1]
    x_range = np.array(x_range)
    I_range = np.array(I_range)
    xcm = np.sum(x_range*I_range) / np.sum(I_range)
    centroids += [xcm]
```

APPENDIX B: Identifying Neon and OH Telluric Spectrum Lines

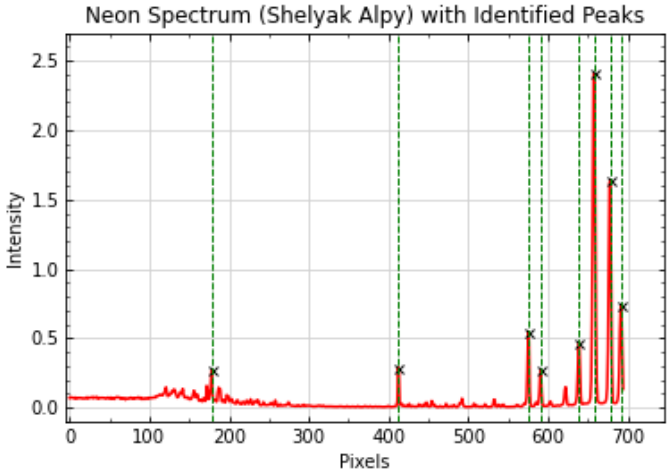
Below are the three spectra that were used for calibration, each labelled with the peaks and their corresponding centroids that were identified for calibration.



(a) Spectrum of neon lamp obtained from the 1-D USB4000 spectrograph.



(b) Spectrum of OH telluric lines obtained from 2-D measurement.



(c) Neon spectrum obtained from Shelyak Alpy spectrograph.

Figure 7: The centroids of selected local maxima are indicated in green and the corresponding local maxima are x's.

APPENDIX C: Ordinary Least Squares Regression

Below is an implementation of OLS regression in python. This function was initially developed and used in Lab 1 [10].

```
def least_squares_fit(x,y):
    """
    This function gives the slope and y intercept
    of a 2-array dataset for an ordinary least
    squares fitting.
    Parameters
    ----------
    x : 1D array of floats or ints
        x-axis points.
    y : 1D array of floats or ints
        Mean of the Gaussian distribution.
    Returns
    -------
    m : float
        Resulting slope for ordinary least squares fitting.
    c : float
        Resulting y-intercept for ordinary least squares fitting.
    sigma_m : float
        The standard deviation of the slope m.
    sigma_c : float
        The standard deviation of the slope m.
    """
    # Define variables for slope and y intercept
    m, c, sigma_m, sigma_c = 0, 0, 0, 0

    length = np.size(x)
    # Check that input arrays are the same length
    if length == np.size(y):
        # Construct the matrices, [m c] = A^-1 * B
        # Start with matrix A
        a, b, d, e = 0, 0, 0, length
        for i in range(0, length, 1):
            a += x[i]**2
            b += x[i]
            d += x[i]
        A = np.array([[a, b], [d, e]])
        # Get the inverse of matrix A
        A_inv = np.linalg.inv(A)

        # Construct matrix B
```

```
    f, g = 0, 0
    for i in range(0, length, 1):
        f += x[i]*y[i]
        g += y[i]
    B = np.array([[f], [g]])

    # Find m and c resulting from [m c] = A_inv * B
    h, j, k, l = A_inv[0,0], A_inv[0,1], A_inv[1,0], A_inv[1,1]
    m = h*f + j*g
    c = k*f + l*g

    # Now find the errors for the slope and y intercept
    # First estimate the standard deviation sigma
    C = 0
    for i in range(0, length, 1):
        C += (y[i] - (m*x[i] + c))**2
    sigma_2 = (1/(length-2))*C
    sigma = np.sqrt(sigma_2)

    # Now find sigma_m
    sigma_m_2 = length*sigma_2 / ( length*a - (sum(x))**2 )
    sigma_m = np.sqrt(sigma_m_2)

    # Now find sigma_c
    sigma_c_2 = sigma_2*a / ( length*a - (sum(x))**2 )
    sigma_c = np.sqrt(sigma_c_2)

return np.array([m, c, sigma_m, sigma_c])
```

The uncertainties of the parameters, σ_m and σ_c, are derived in [7]. The derivation is discussed below because it mirrors the derivation done for the uncertainties in quadratic regression parameters.

By propagation of error it is known that for a dataset $z = z(x_1, x_2, ..x_N)$ of uncorrelated measurements, the standard deviation of z is:

$$\sigma_z^2 = \sum_i (\partial z/\partial x_i)^2 \sigma_i^2 \tag{10}$$

The diagonals of the covariance matrix of $z(x)$ and x_i can be used to find the standard deviations of the fit parameters. If it is assumed that the covariance is zero, then the diagonal values are equal to the standard deviations σ_{ii}^2. Thus, the standard deviations of the parameters m and c are as shown below.

$$\sigma_m = \sum_j (\partial m/\partial y_j)^2 \sigma_j^2 \quad \text{and} \quad \sigma_c = \sum_j (\partial c/\partial y_j)^2 \sigma_j^2$$

Taking the partial derivatives of the parameters above (using the parameter equations in eqn.

(4)), the individual standard deviations are found below.

$$\frac{\partial m}{\partial y_j} = \frac{\partial}{\partial y_j}\frac{\sum x_i \sum y_i - N\sum x_i y_i}{(\sum x_i)^2 - N\left(x_i^2\right)} = \frac{\sum x_i - N\sum x_i}{(\sum x_i)^2 - N\left(x_i^2\right)}$$

$$\frac{\partial c}{\partial y_j} = \frac{\partial}{\partial y_j}\frac{\sum x_i \sum x_i y_i - \sum x_i^2 y_i}{(\sum x_i)^2 - N\left(x_i^2\right)} = \frac{x_j\sum x_i - N\sum x_i^2}{(\sum x_i)^2 - N\left(x_i^2\right)}$$

These partial derivatives are plugged into equations () and () to yield:

$$\sigma_m^2 = \sigma_{O1}^2 \sum_j \left(\frac{\sum x_i - N\sum x_i}{(\sum x_i)^2 - N\left(x_i^2\right)}\right)^2 \tag{11}$$

$$\sigma_c^2 = \sigma_{O1}^2 \sum_j \left(\frac{x_j\sum x_i - N\sum x_i^2}{(\sum x_i)^2 - N\left(x_i^2\right)}\right) \tag{12}$$

where σ_{O1} is taken as the standard deviation of the fit, as shown in eqn. (7). Equations (11) and (12) are then simplified to yield equations (5) and (6).

7 APPENDIX D: Computing Mean Absolute Difference

The mean absolute difference between datasets and their fits was computed using the python function below.

```
def mean_abs_diff(array1, array2):
    """
    Return the mean absolute difference between two datasets
    Parameters
    ----------
    array1 : array of floats
    array2 : array of floats
    """
    differences = abs(array1-array2)
    delta_abs = np.mean(differences)
    count = len(differences)
    variance = 0

    for i in range (0,count,1):
        variance = variance + (1/count)*(differences[i] - delta_abs)**2
    stddev = np.sqrt(variance)
    err_abs = stddev/np.sqrt(count)

    return delta_abs, err_abs
```

APPENDIX E: Quadratic Regression

Below is an implementation of quadratic regression in python.

```
def quadratic_regression_fit(x,y):
    """
    This function gives the quadratic fit of a 2-array
    dataset for an R^2 quadratic fitting, ax^2 + bx +c = y.
    Parameters
    ----------
    x : 1D array of floats or ints
        x-axis points.
    y : 1D array of floats or ints
        Mean of the Gaussian distribution.
    Returns
    -------
    a, b, and c : floats
        Resulting parameters from quadratic regression.
    sigma_a, sigma_b, sigma_c : floats
        Resulting ucnertainties of errors for quadratic regression
    """
    # Define variables for slope and y intercept
    a, b, c = 0, 0, 0
    sigma_a, sigma_b, sigma_c = 0, 0, 0

    length = np.size(x)
    # Check that input arrays are the same length
    if length == np.size(y):
        # Construct the matrices, [m c] = A^-1 * B
        # Start with matrix A
        n, o, p = 0, 0, 0
        q, r, s = 0, 0, 0
        t, u, v = 0, 0, 0
        for i in range(0, length, 1):
            n += x[i]**4
            o += x[i]**3
            p += x[i]**2
            q += x[i]**3
            r += x[i]**2
            s += x[i]
            t += x[i]**2
            u += x[i]
            v = length
        A = np.array([[n, o, p], [q, r, s], [t, u, v]])
        # Get the inverse of matrix A
```

```
    A_inv = np.linalg.inv(A)

    # Construct matrix B
    j, k ,l = 0, 0, 0
    for i in range(0, length, 1):
        j += (x[i]**2)*y[i]
        k += x[i]*y[i]
        l += y[i]
    B = np.array([[j], [k], [l]])
    # Use numpy.matmul()
    a, b, c = np.matmul(A_inv, B)

    # Calculate the standard deviation of the fit
    # for the uncertainties
    C = 0
    for i in range(0, length, 1):
        C += (y[i] - (x[i]**2*a + x[i]*b + c))**2
    sigma_2 = (1/(length-2))*C
    sigma = np.sqrt(sigma_2)

    # Determine parameter uncertainties
    aa = 0
    bb = 0
    cc = length
    for i in range(0, length, 1):
        aa += 2*x[i]
        bb += x[i]
    D = np.array([[aa], [bb], [cc]])
    # Use numpy.matmul()
    a_, b_, c_ = np.matmul(A_inv, D)

    sigma_a_2 = sigma_2*(a_[0]**2)
    sigma_b_2 = sigma_2*(b_[0]**2)
    sigma_c_2 = sigma_2*(c_[0]**2)
    sigma_a = np.sqrt(sigma_a_2)
    sigma_b = np.sqrt(sigma_b_2)
    sigma_c = np.sqrt(sigma_c_2)

return np.array([a[0], b[0], c[0], sigma_a, sigma_b, sigma_c])
```

The uncertainties of the parameters q, r, and s found in eqn. (14) are derived using a similar method to what is done for ordinary least squares linear regression. If we take the vector of parameters as follows:

$$A = \begin{bmatrix} q \\ r \\ s \end{bmatrix} \tag{13}$$

then the uncertainty of each parameter in A can be computed as shown below, using eqn. (10).

$$\begin{aligned} \frac{\partial}{\partial y_j}\begin{bmatrix} q \\ r \\ s \end{bmatrix} &= \frac{\partial A}{\partial y_j} = \frac{\partial}{\partial y_j}\begin{bmatrix} \sum x_i^4 & \sum x_i^3 & \sum x_i^2 \\ \sum x_i^3 & \sum x_i^2 & \sum x_i \\ \sum x_i^2 & \sum x_i & n \end{bmatrix}^{-1}\begin{bmatrix} \sum x_i^2 y_i \\ \sum x_i y_i \\ \sum y_i \end{bmatrix} \\ &= \begin{bmatrix} \sum x_i^4 & \sum x_i^3 & \sum x_i^2 \\ \sum x_i^3 & \sum x_i^2 & \sum x_i \\ \sum x_i^2 & \sum x_i & n \end{bmatrix}^{-1} \frac{\partial}{\partial y_j}\begin{bmatrix} \sum x_i^2 y_i \\ \sum x_i y_i \\ \sum y_i \end{bmatrix} \\ &= \begin{bmatrix} \sum x_i^4 & \sum x_i^3 & \sum x_i^2 \\ \sum x_i^3 & \sum x_i^2 & \sum x_i \\ \sum x_i^2 & \sum x_i & n \end{bmatrix}^{-1}\begin{bmatrix} \sum x_i^2 \\ \sum x_i \\ N \end{bmatrix} = A' \end{aligned} \tag{14}$$

Analytically evaluating or simplifying eqn. (14) is difficult, so numpy.linalg.inv() is used to invert the matrix, and then numpy.matmul() is used to perform the required multiplication. Eqn. (14) can be subbed into eqn. (10) to yield:

$$\begin{bmatrix} \sigma_q^2 \\ \sigma_r^2 \\ \sigma_s^2 \end{bmatrix} = \sum_j A'^2 \sigma_{O2}^2 \tag{15}$$

where σ_{O2} is the defined as:

$$\sigma_{O2}^2 = \frac{1}{N-2}\sum_i [y_i - (qx_i^2 + rx_i + s)]^2 \tag{16}$$

References

1. D.-S. Moon, ”Lecture 4: Introduction to Scpectroscopy”. 2022.
2. D.-S. Moon, ”Lab 2: Introduction to Spectroscopy”. 2022.
3. P. Rousselot, C. Lidman, J.-G. Cuby, G. Moreels, and G. Monnet, “Night-sky spectral atlas of OH emission lines in the near-infrared,” Astronomy and Astrophysics, pp. 1134–1150, 1999.
4. D.-S. Moon, ”Lecture 3: Introduction to Spectroscopy”. 2022.
5. D.-S. Moon, ”Lecture 5: Introduction to Lab 2”. 2022.
6. D.-S. Moon, ”Lecture 6: Analysis of Spectroscopic Data from the Campus Telescope”. 2022.
7. D.-S. Moon, ”Lab 1: Basic Photon Statistics with Python.” 2022.
8. Statistics - quadratic regression equation. Tutorials Point. (n.d.). Retrieved October 22, 2022, from https://www.tutorialspoint.com/statistics/quadratic_regression_equation.htm
9. Spectral Centroid Fitting (1D Centroiding and Peak Finding Algorithms). (n.d.). Retrieved October 17, 2022, from https://gist.github.com/seery-chen/50782af597a9ba3636ada9fd59b595c4
10. V. Spada, ”Lab 1: Basic Photon Statistics with Python.” 2022.