AST326 Lab-02 Report
Introduction to Spectroscopy

Raina Irons[1]

[1] *University of Toronto*

ABSTRACT

In this report we introduce some introductory spectroscopic methods using Python and it's applications in the subject. One dimension is the jumping point for our analysis on a blackbody spectrum, conducting a basic model linear fitting to the pixel and wavelength values of a Neon spectrum. The best fit line fitting was done using *Scipy*'s *curve_fit*() and returned 0.2390 ± 0.0016 nm/pixel for the slope, and 527 ± 1.15 nm with a reduced $\chi^2 = 0.47$. Applying the wavelength calibration we find the peak wavelength occurs at 630 nm ± 1.65 nm, and using Wien's Law, the temperature of the blackbody is 4580 ± 1.2 Kelvin. Continuing to two dimensions, we introduce FITS files and their applications in data reduction and properties. We perform two-dimensional wavelength calibration on OH skylines and find a wavelength solution with 3.5 ± 0.15 nm/pixel for slope 15900 ± 24.15 nm with reduced $\chi^2 = 1.62$ by taking a section near an [FeII] emission line. We compare a quadratic and cubic model functions and determine the linear to be the best reflection of the data. We apply a two-dimensional Gaussian fitting on the emission to determine the center and scale the y-position to the wavelength dimension. This gave 6530 ± 38.3 A, or 1.65μm. Applying the Doppler equation, the radial velocity of the [FeII] emission to be 1577 ± 5 km/s. In addition, data reduction methods were discussed and used to extract four stellar spectra and the moon's spectra based off a Neon lamp wavelength solution.

1. INTRODUCTION

Spectroscopy serves as a fundamental tool used in astronomy, used to determine chemical composition of far away bodies only sci-fi films dream of visiting. Here we will introduce some basics in using spectroscopy to analyse several spectra of celestial objects, the most important of which is wavelength calibration using a known spectra. Initially, we will use a Neon spectrum to determine the pixel to wavelength relationship on the same spectrograph used to create the dispersed image a blackbody source. The importance of this comes from the ability to map pixel positions of Neon peaks to their corresponding wavelengths, allowing for the same solution to be applied to the blackbody source. From this, we can determine the peak wavelength of the blackbody, and as consequence, determine its temperature. This is modeled by Wien's Displacement Law, which states that blackbody radiation curves will peak at different wavelengths inversely proportional to their temperature. This is written formally as,

$$\lambda_{peak} = \frac{b}{T}. \tag{1}$$

Where λ_{peak} is the wavelength at which the peak of the spectrum occurs, b is the proportionality constant equal to 2.897×10^{-3} m·K, and T is the temperature of the source. With this basic idea of the wavelength solution, we can expand this idea from a one dimensional source and calibration, to a two dimensional spectra.

The wavelength solution of a two dimensional dispersed image is a bit more complicated than a one dimensional spectra, with more interesting characteristics to take into consideration when applying a wavelength solution. The file we will be working with now is not a one dimensional array, but a FITS (Flexible Image Transport System) file which is a standard data set used in spectroscopy. After some applied scaling to increase clarity, we show the file below.

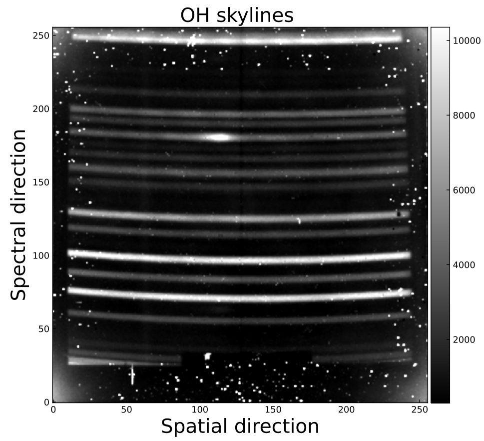

Figure 1: The above is the fits file we are interested in calibrating a wavelength solution for. Note the overarching horizontal bright OH telluric emission lines in our image, these are what is key in determining the pixel to wavelength relationship of the file as we can use external sources, described in 3, in order to determine the wavelength of each skyline. The bright source near the top of the file is ionized iron gas, FeII, emission from a supernova explosion, and our point of interest.

One key feature of the FITS file is the curvature of the OH skylines in the FITS file which is due to the non-zero optical aberrations, providing imperfect, or non-uniform, performance in the image. The simplest way to fix this is to consider a rectangular section of the FITS file near the iron emission and apply the wavelength solution found there to the entire data set. This method takes care of the issue of the curvature, as we only really care about the iron emission and not other sections. A more in depth analysis of this section will be discussed later along with the process. Finally, knowing the pixel to wavelength relationship, we can use this to calculate the wavelength of the FeII emission, and using the intrinsic wavelength of the FeII 1.644 μm line emission, 1.6439981 μm, we can find the velocity.

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

Where λ is the observed wavelength found from our FITS file, λ_0 is the intrinsic wavelength of the FeII 1.644 μm line emission, and c is the speed of light. Note that the velocity of the emission depends on the units of c, and not the observed or intrinsic wavelengths, as the units cancel out in the numerator and denominator of the fraction. Therefore we use $c = 2.99 \times 10^6$km/s in our calculation.

Spectra are obtained through data reduction methods that involve multi-level steps. This is one step further than slicing the FITS file of the telluric lines to determine the wavelength solution and applying it to the iron emission centroid. Now, we take into consideration flats, bias', and darks for varying exposure times. The first of three is the flat, which is correction for varying pixel sensitivity in CCD (Charged Couple Device) pixels. Within the CCD, when a photon within our desired wavelength range hits the surface, it produces an electron due to the photoelectric effect and detected. Flat-fielding is the method used to correct for non-uniform performance across CCD pixels due to different absorption and reflection efficiencies. We take an exposure shot of a uniform bright source with the same CCD we use for other sources in this analysis. The flat FITS file is then divided by its exposure time to normalize the exposure. Note that non-uniform performance is multiplicative, so removing the effect requires dividing the dispersed image by the flat. The bias of the data is time independent, caused by the transfer by the bias across the pixel to the output amplifier. The intensity is then measured in voltage before converted to a digital number using an analog-to-digital converter. The bias is this voltage to transfer electrons and included in the final data. Lastly, the dark is proportional to the exposure time, which takes into consideration electrons created inside CCD pixels independent of incoming photons,meaning there are no from the source, but from the CCD's independent detections due to internal photons. This effect is additive, so we subtract it from the raw dispersed image before dividing out the flat from the product. Note that the dark of our files already takes into consideration the bias, so we do not consider it in the reduction. In addition, we create master flat and dark files such that the three available darks and flats of each exposure time are median combined such that random noise is reduced. In general, we have the following method to reduce our image.

$$\text{Reduced image} = \frac{\text{Raw dispered image} - \text{Master Dark}}{\text{Master Flat}} \tag{3}$$

Note that for each raw image, we must make sure the exposure time matches the exposure time of the dark FITS file in order for the result to be consistent. We can divide out by the same normalized one second flat file for each image. This is the basic procedure for our data reduction we will follow. In addition to this, we will conduct data reduction and wavelength calibration using a Neon lamp for an eight second exposure time in order to convert our reduced spectra of the stars and moon to their wavelengths using similar methods to the blackbody calibration.

2. DATA REDUCTION & METHODS FOR DISPERSED IMAGES

Standard data reduction processes require multiple steps in order to extract spectra from dispersed images. Our end goal is to extract the spectra of four stars (Aldebaran, Capella, Elnath-betaTau, and Mahasim-Theta-Aur), and the moon. Data reduction is described in Section 1 and follows Equation 3 for each dispersed spectra. The data was acquired from a standard data set from Micheal Williams [1] for each star spectra, moon spectrum, and eight second Neon lamp calibration file. We first correct for the Neon lamp spectra, as it is key in defining the wavelength calibration of the FITS files. The first step is to create our master flat, which can later be applied to the rest of the spectra regardless of exposure time, and the master dark (time dependant). This involves stacking three flat fits files of a one second exposure time and taking the median along the plane (axis 0) of each 2X2 file. This process results in reducing the random noise of each individual file, and creates the first of many master FITS files. Next we can apply the same method to the $1s$,$8s$, and $10s$ dark FITS files to create a $1s$ master dark, $8s$ master dark, and $10s$ master dark. Please refer to Section 5.1 for further details and Python implementation of stacking two-dimensional arrays to create the master flat and dark for varying times. To compare the difference when reducing random noise, we present a $8s$ dark FITS file compared to the $8s$ master dark.

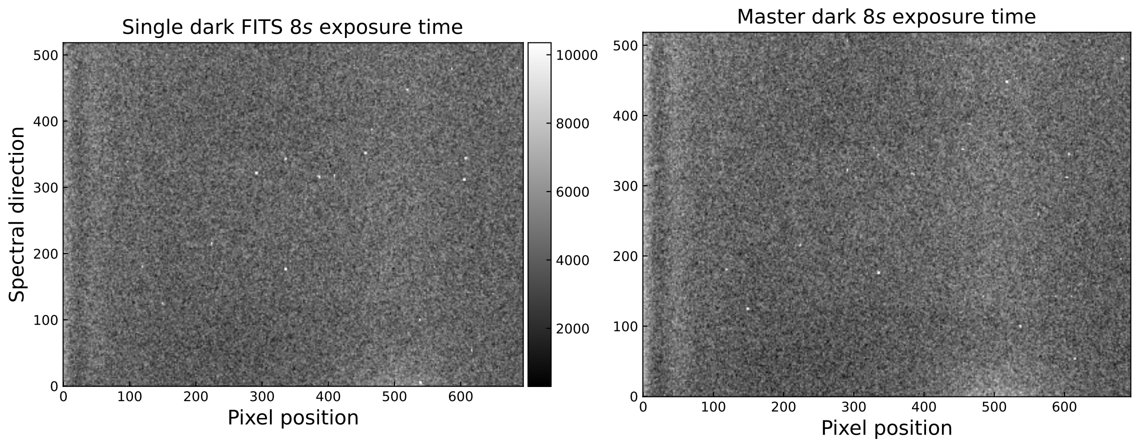

Figure 2: *Left*: A single dark FITS file takes into account the detections of internal photons for CCD pixels in the detector. Individual images for the same exposure time would vary across pixels. *Right*: To minimize the random noise of individual darks, we stack the FITS files behind one another and take the median along the plane axis. This results in possible extrema to be reduced in the overall master dark.

With the master flat and dark, we are able to apply Equation 3 to each respective exposure time. In the case of the $8s$ Neon dispersed spectrum, we first subtract off the $8s$ master dark, and then divide by the master flat.

3. RESULTS & DATA ANALYSIS

3.1. *One-dimensional wavelength calibration with Neon spectrum for blackbody temperature*

[1] TA Admin and Observatory Demonstrator

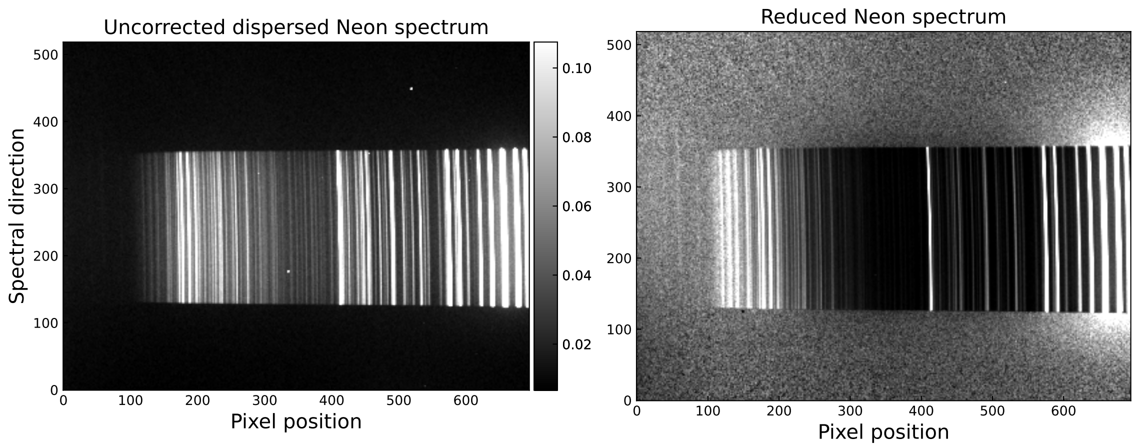

Figure 3: *Left*: The raw Neon 8s exposure. *Right*: Here we have applied the data reduction methods described following Equation 3. Note that we can ignore the surrounding background of the FITS file as it is not important, only the Neon spectrum lines matter.

The basic goal is to develop a method to relate pixel position to wavelength for a simple spectrograph calibration to apply to a blackbody spectrum taken from the same spectrograph. The blackbody's intensity initially is dependent on the pixel position, so what we can do is take a known spectra, in our case Neon, and map the pixel location of the peaks of the dispersed Neon spectrum to their corresponding wavelength.

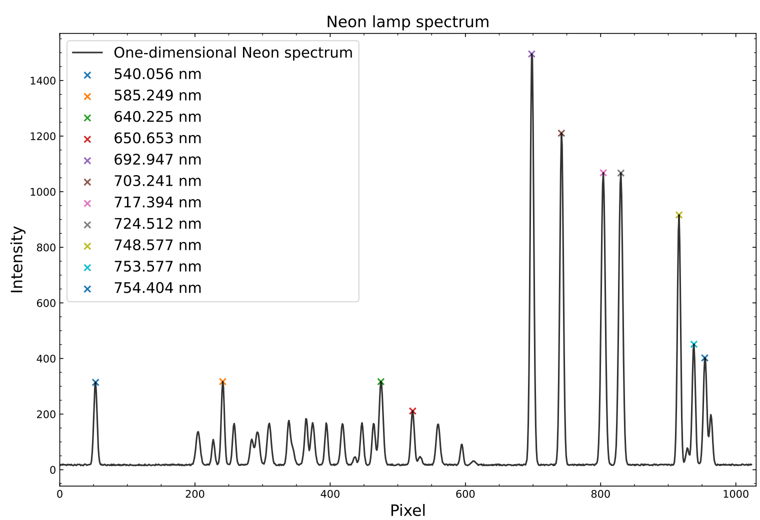

Figure 4: The Neon spetrum has several distinctive lines we can use to help determine the wavelength calibration. We list off several peaks above a threshold height of 200 for the intensity, and use *Scipy*'s libray function $find\_peaks()$ to return the peak heights and pixel positions.

In order to identify the intensity peaks, we can use a simple peak finding algorithm that takes in a minimum threshold required for a peak, and compares them with its neighbors. This is found in the *Scipy* python library with the $find\_peaks()$ function. We can them return all corresponding peak heights and locations in a two dimensional array for analysis. As a general rule, we chose only peaks that were above an easily distinguishable height intensity of 200.

This allows for the peak positions to be compared to the Neon spectrum, and matching wavelengths selected manually.

Table 1: The output of *Scipy find_peaks*() function and selection of matching wavelengths corresponding to these peak intensities. Each respective column and row entry are in the same position in each section.

Pixel position (nm)		Intensity height (Pixel)		Wavelength (nm)	
53	241	314.17537	316.78071	540.056	585.249
475	522	316.72719	210.99634	640.225	650.653
698	742	1495.2875	1210.3815	692.947	703.241
804	830	1067.8626	1066.9936	717.394	724.512
916	938	916.43762	451.26308	748.577	753.577
954		402.17602		754.404	

We are now able to distinguish the relationship between the pixel position and wavelength by plotting them against one another. This allows us to see visually the relationship between them appears to be a linear relationship.

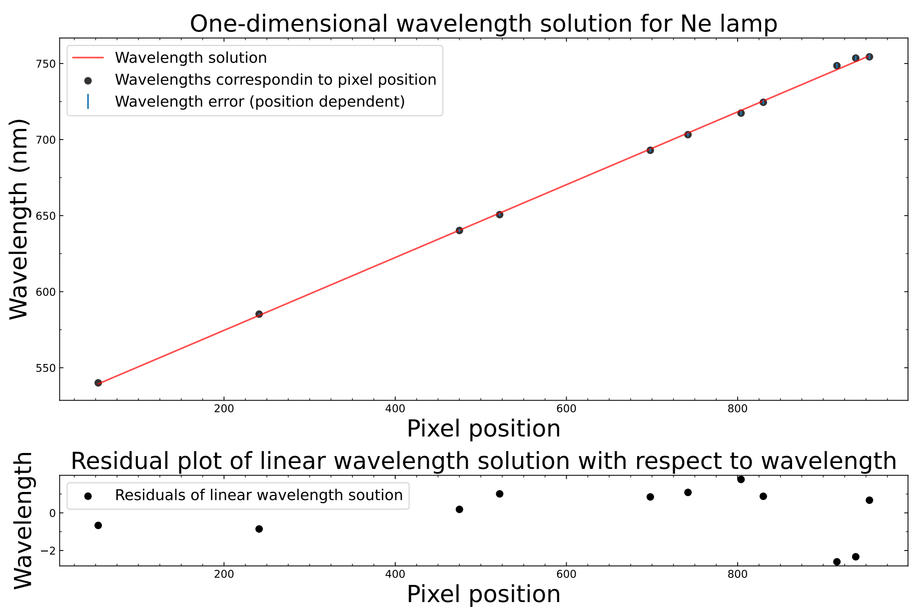

Figure 5: The goal of plotting the known wavelength of each peak of the Neon spectrum against its pixel position allows for a visualization between the two values, allowing us to determine the relationship. Immediately, we see they appear to have a linear relationship, and fitting a general linear function using *scipy*'s *curve_fit*() yields a reduced chi square of (insert here) with slope (insert) and intercept (insert with errors). In addition, the residuals of the fit are plotted below distributed about zero demonstrating an acceptable fit. The fitting parameters found were 0.2390 $\pm$ 0.0016 nm/pixel for the slope, and 527 $\pm$ 1.15 nm for the intercept with reduced chi-square of 0.47. This signifies an over fit in the data, and is further discussed in Section 5.4.

With this linear relationship determined, we are able to apply this method to the blackbody spectrum, converting from pixel position to wavelength. This is done by taking the best fit parameters returned in *curve_fit*() and scaling each of the pixels to return the corresponding wavelength.

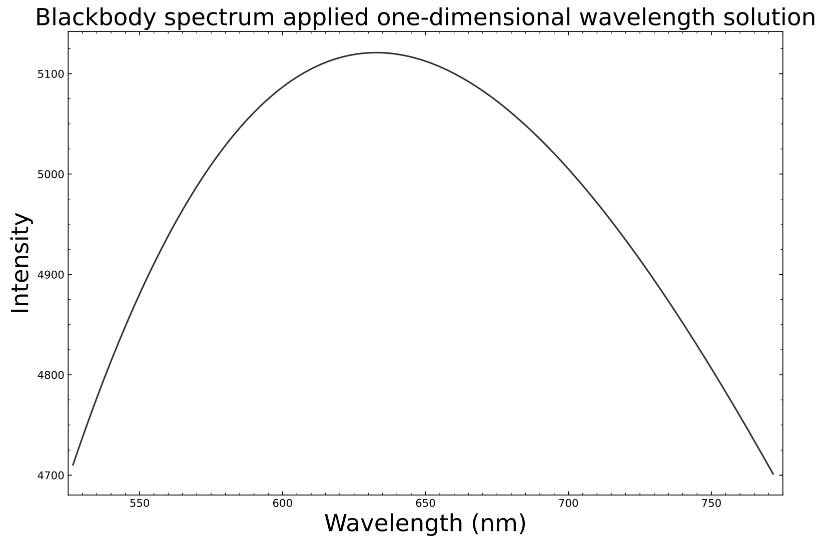

Figure 6: The resultant of the wavelength solution yields the intensity of the blackbody spectrum over its wavelengths. The consequent is the peak wavelength is now accessible, and its temperature.

The peak wavelength of the spectrum was calculated to be 630 nm $\pm$ 1.65 nm, and using Equation 1, the temperature of the blackbody is 4580 $\pm$ 1.2 Kelvin.

3.2. *Two-dimensional wavelength solution with OH skylines for [FeII] emission*

The natural extension from working in one dimension is to turn to two, which is our next extension in spectroscopy basics. We will now turn our attention to the FITS file as discussed in 1. We can employ a similar method as in one-dimension, and take a sectional rectangle of the FITS file close to the emission line, find the wavelength solution for that section, and employ it to the entire file. Since we are not concerned with any other source in the file, applying a solution that is specific to the [FeII] line is the simplest approach to deal with the curvature. We choose 15 columns of the file from pixel location 135 to 150 and take the median across the 0th axis, bypassing all "hot" or outliers in the row that may skew the average intensity we calculate for each row. With this new median taking into consideration 15 columns of information near the emission, we can display the OH skylines along with their peak locations and intensities. A table of the values returned $find\_peaks()$ and the selected wavelengths can be found in Section 5.2. From these values, we are finally able to determine the relationship between the pixel position and the telluric lines close to the iron emission.

To choose an appropriate fitting, we fit three curves of first to third degree to the wavelength vs pixel plot and compare their fitting. A key idea in determining our fit is to consider the number of degrees of freedom in each curve compared to the number of points we are fitting each curve to. For example, a general linear function has two degrees of freedom in the slope, commonly denoted m, and the y-intercept, denoted c. Comparing this to a cubic function, which has twice as many possible fitting parameters, the residuals of the fit may be better, but there may not be enough data points to our fit to conclude that it is the best fit. Over-fitting is also a possible issue with upping the degrees of freedom , where residuals would become small and result in the curve perfectly fitting our data. We present the three best fit lines ranging from linear to cubic along with their corresponding residuals.

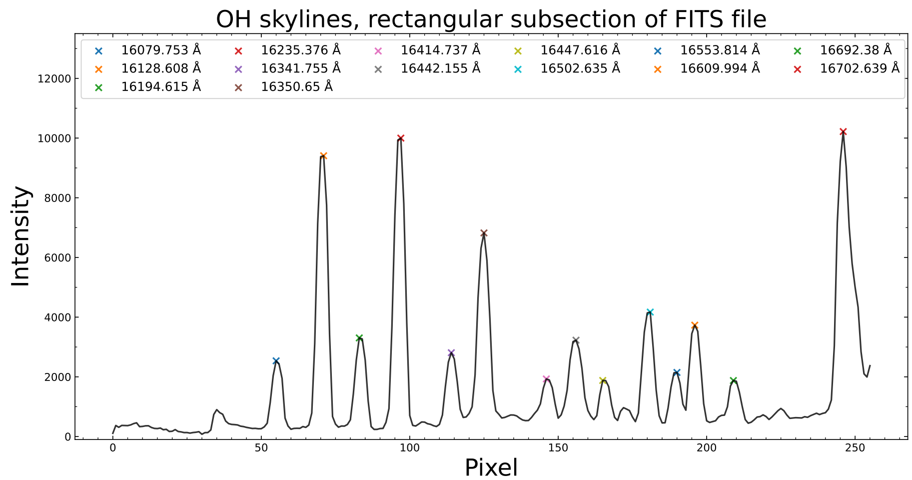

Figure 7: The OH skylines will come into use for determining the pixel and wavelength relationship in two dimensions. The $find\_peaks()$ algorithm returns the peak intensities and pixel location of the telluric lines for a height threshold of 1500. The legend of this plot depicts the selected OH wavelengths from Rousselot et. al[1]. Figures 17 and 18 were used to compare and select the wavelengths of the peaks.

With our two-dimensional wavelength solution, we can apply the calibration to each FITS file. One obstacle remains in our path: determining the pixel location of the iron emission exaclty. It is not necessary to know the x position, as we chose a rectangular section close enough to the emission that we can apply to the entire file, although it is nice to have. To find the position, we take a 40X40 pixel subsection of the FITS file centered roughly about the emission and apply a two-dimensional Gaussian fitting to the subsection. This was done using *centroid_2dg*() from the *photutils.centroids*[2], which calculates the centroid of a two-dimensional array by fitting a two-dimensional Gaussian (plus a constant) to the subsection. This returns the position of the center of the iron gas emission, of which we can round down to the nearest whole integer for a realistic pixel value. This results in a position of (22., 20.) in the subsection, and translates to (112.0, 180.0) after a translation. A visualization of this can be found in Section 5.3.

We apply the linear wavelength solution to the iron centroid located at y pixel position of 180. After applied scaling, the wavelength of the iron emission is 16530 $\pm$ 38.3 Å, or 1.652 μm. Finally, we can apply Equation 2 to this observed wavelength using the intrinsic wavelength of iron 1.6439981 μm to find the velocity to be 1577 $\pm$ 5 km/s.

3.3. *Extracting spectra from reduced images*

Our end goal is to extract the spectra of four stars (Aldebaran, Capella, Elnath-betaTau, and Mahasim-Theta-Aur), and the moon from the reduced dispersed images. To begin, we must conduct wavelength calibration of the FITS files along the pixel axis by taking our reduced Neon spectrum as outlined in Section 2. Much like in Section 3.2, we will extract a rectangular subsection about the center across the reduced dispersed Neon image 30 rows tall and median combine the columns such that we are able to plot the resultant intensities and bypass any "hot" pixels.

A table summarizing the pixel positions, intensity heights, and selected wavelengths of the Neon spectrum from the above Figure will be presented in Section 5.2. We can use the selected wavelengths along with the pixel positions to examine the relationship between them. We choose to fit a linear function to this data after examining the data points and show the line of best fit in Figure 10.

With our wavelength calibration complete, we can apply this scaling correction to the pixel positions in each of the corrected spectra of the stars and moon sectioned about the source, where the intensity is the sum of each intensity along each column in the FITS files. This leaves us to present the final spectra.

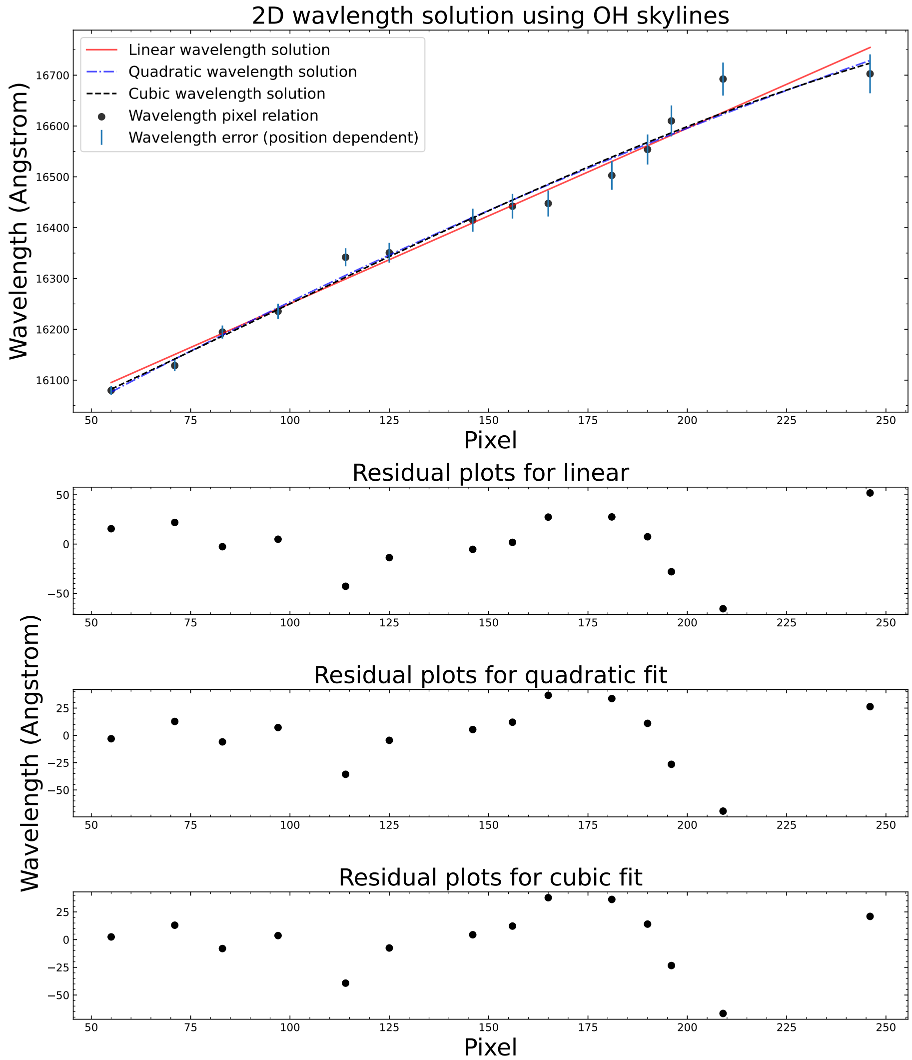

Figure 8: Each fit ranging from linear to cubic has been plotted and their residuals below. The key distinction that will determine the correct fit is the error reduced chi-square values for each of the fits: 1.62, 3550.5, and 10303188.7 for the linear, quadratic, and cubic respectfully. We choose linear as it allows the errors to be great enough that the reduced chi-square is desirable. For further details of these calculations please refer to Section 5.4, Table 4. The returned best fit parameters for the linear model is 3.5 ± 0.15 nm/pixel for slope 15900 ± 24.15 nm.

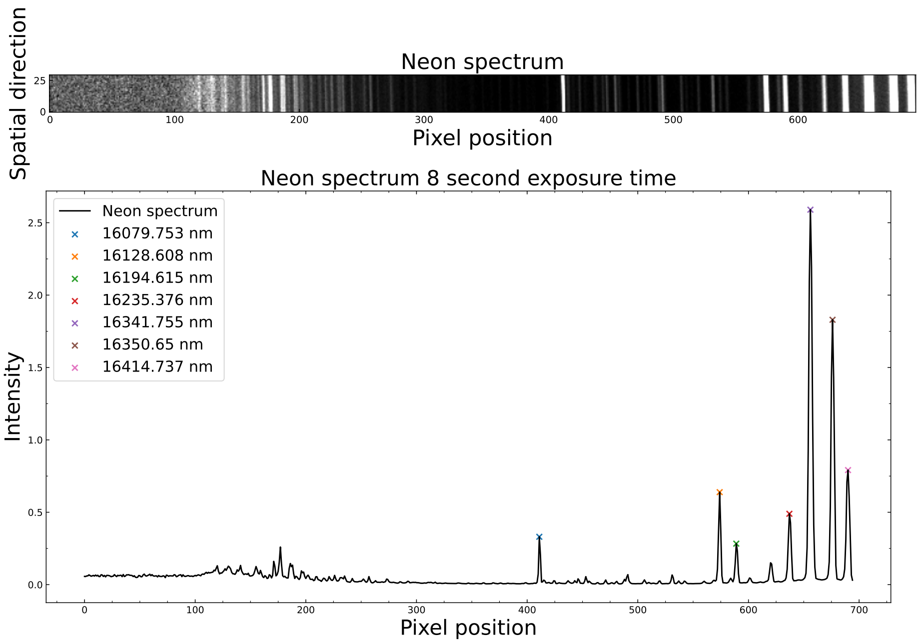

Figure 9: *Top*: The rectangular section taken into consideration when conducting the wavelength calibration. *Bottom*: The resultant intensities of the reduced Neon spectrum with 30 total rows median combined about the center of the FITS file. Here we also display the peaks of the spectrum found using $Numpy$ $find\_peaks()$ function with a height threshold of 0.27. This was chosen in specific to highlight easily identifiable peaks in the spectrum and avoid mislabeling any peaks. Note that this method does not take into account the slight offset in the Neon spectrum vertical lines, which is a negligible factor to take into consideration the tops and bottoms of each line differ by a few pixels. Peak wavelength values were taken from Rousselot et. al[1].

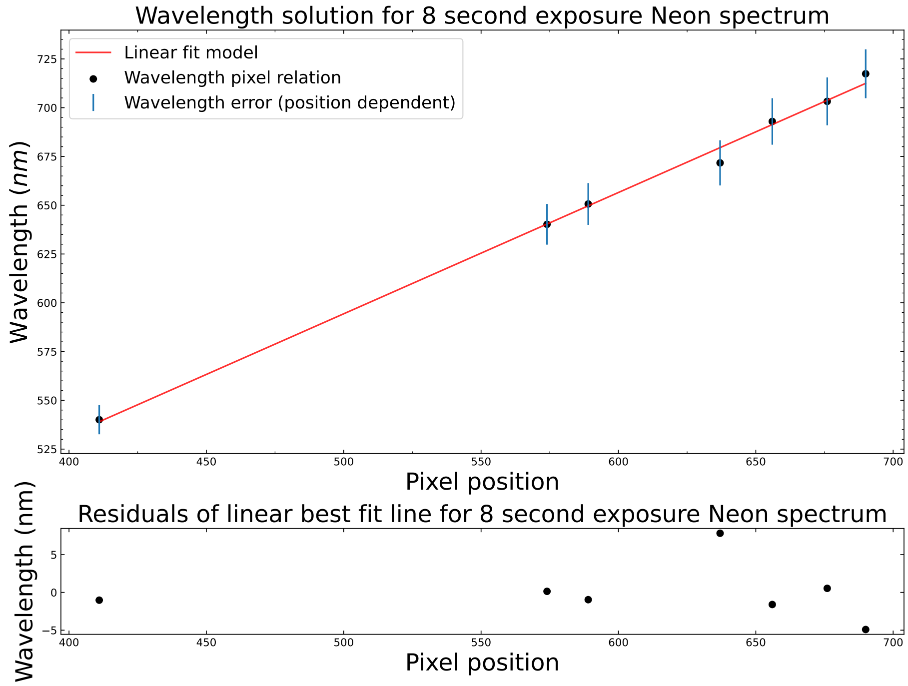

Figure 10: *Top*: The relationship between the wavelengths and pixels are shown to have a close linear relationship. The fitting was done through *Scipy*'s *curve_fit*() function and returned fitting parameters 1 ± 0.16 nm/wavelength for the slope and 280 ± 11.1 nm, and a reduced chi-square was calculated to be 0.1. This is an over fitting in the data, and is most likely due to the large errors in the figure. *Bottom*: Residual plot of the best fit line.

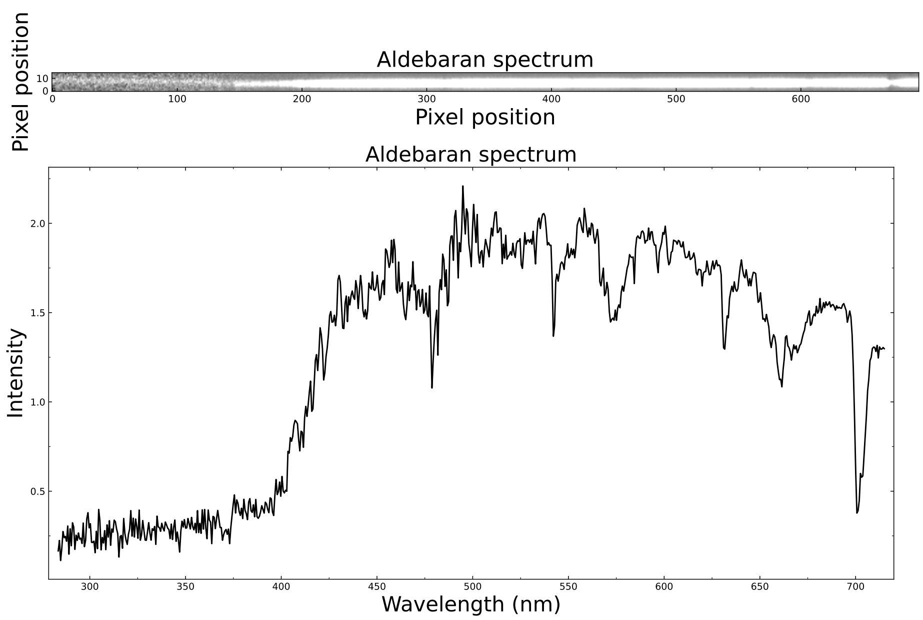

Figure 11: Aldebaran is the brightest star in the zodiac constellation of Taurus, a giant star that is cooler than the Sun with surface temperature of 3,900K, and a radius about 44 times. This is the reduced stellar spectra, which differs slightly from other online searches of the spectra. These differences can be accounted for with knowing the spectrums online may not be taken from the same geographical location, and not have the level of Toronto light pollution affecting its spectrum.

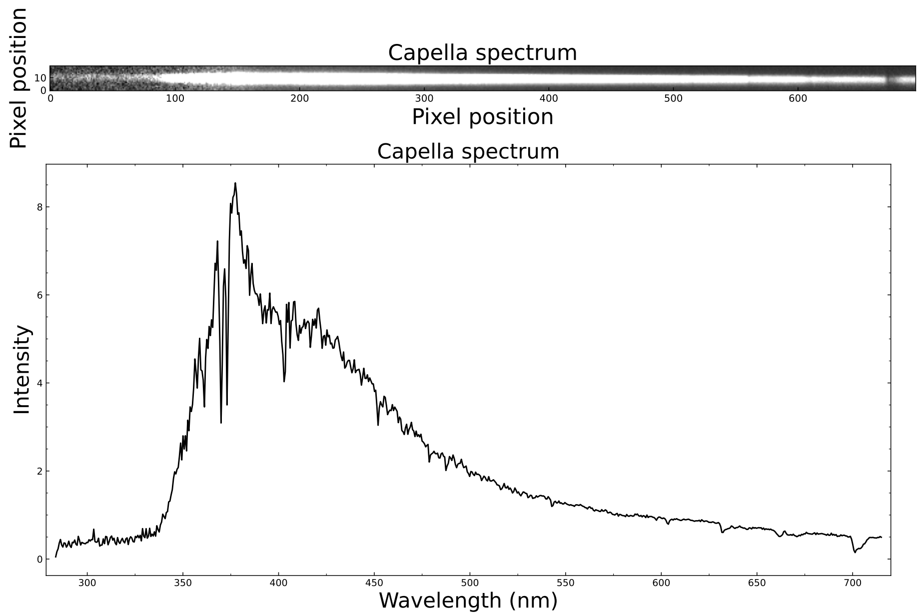

Figure 12: Capella is brightest star in the northern constellation of Auriga, and the sixth brightest star in the night sky. This is not a single star, but a quadruple star system organized in two binary pairs, made up of the stars Capella Aa, Capella Ab, Capella H and Capella L. They are taken into consideration as one body in our spectrum, as we do not acquire the capabilities to single out the individual stars, and instead treat it as a point source.

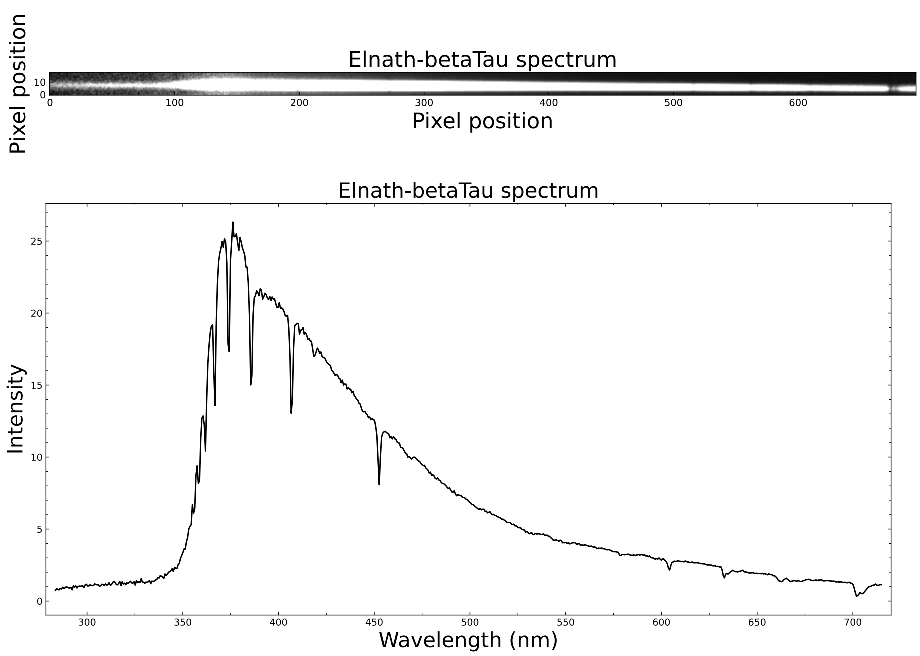

Figure 13: Elnath-betaTau is a B-class giant and the second-brightest star in the constellation of Taurus, with a luminosity 700 times our sun. This is by far one of the best spectrum achieved, having a clear blackbody trend and clear absorption lines.

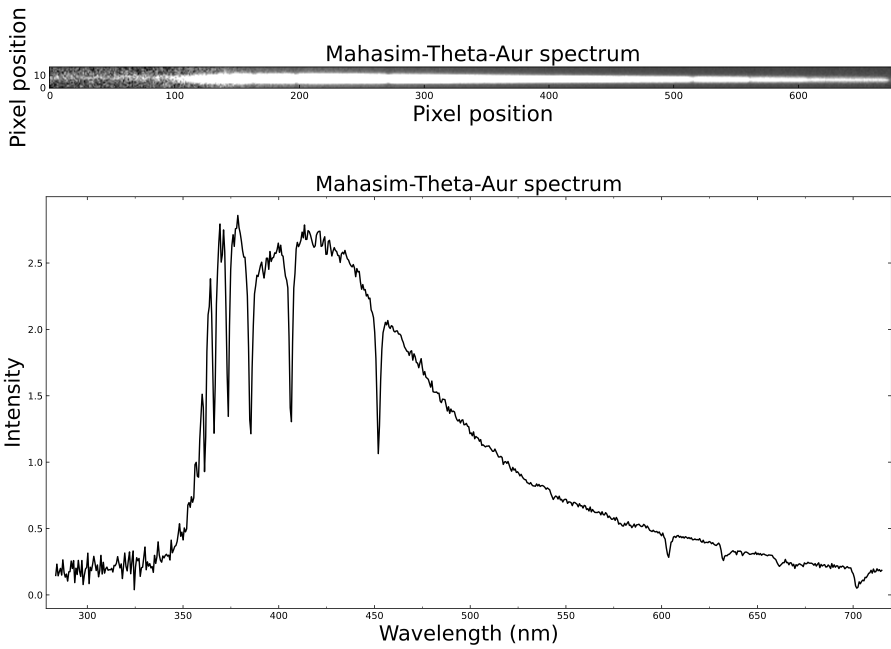

Figure 14: Mahasim-Theta_Aur is actually a binary star system in the constellation of Auriga located about 166 light-years from Earth. As in Capella, we treat this as a point source since we are not able to currently resolve the two stars.

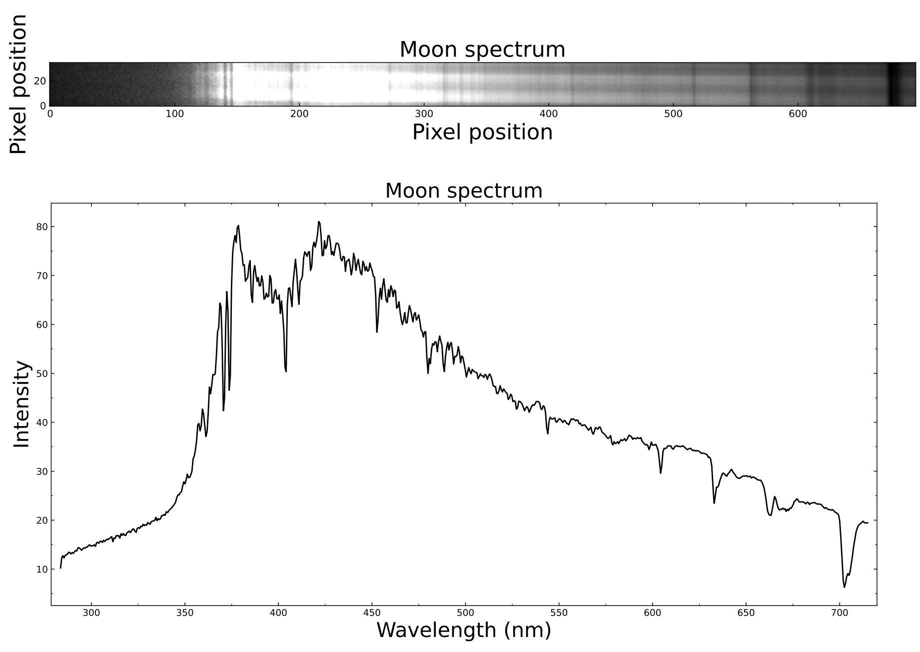

Figure 15: The moon is our closest target and one where we would expect a spectrum similar to our own sun due to reflecting off solar light.

4. DISCUSSIONS & CONCLUSIONS

CCD images and spectroscopy go hand-in-hand with one another. using them , we were able to create master flats and darks through median combining like exposure times along their planes. This allowed us to reduce the spectroscopic data given to us by Micheal Williams [2] and extract stellar spectra and the spectra of the moon from FITS files of their dispersed image. To do so, we first built up using a one-dimensional wavelength solution from a Neon lamp spectrum illustrated in Figure 4. From there wavelengths were selected from Rousselot et. al.[1] Figures 16 and 17 to obtain the wavelengths of the peaks. Comparing the wavelength to the pixel position in Figure 5, we create the wavelength solution using a linear model with slope 0.239385521 nm/pixel and intercept 526.702060 nm. We can then apply this calibration to a blackbody spectrum in Figure 6 and find the peak wavelength, which yields the temperature of the black body using Equation 1 of 4583.2 K.

Expanding from one dimension to two is relatively straight forward when producing the wavelength solution for an iron emission line. We extract 15 columns from the FITS file near the emission and median combine the columns along the rows to avoid any hot pixels within the data. This is used to procure Figure 7, and is used in determining the OH skyline wavelengths to conduct wavelength calibration against the pixel position of the peaks. In testing the best model to base our solution on, Figure 8 contains three model functions: linear, quadratic, and cubic. Of these, we found the linear model is the best fit with the lowest chi-square of 1.62 with slope 3.45 nm/pixel and intercept 15905.5 nm. To calculate the wavelength of the [FeII] emission, we must first determine the central position, which was done using a two-dimensional Gaussian fit with Photutils[2] Python library, then applying the wavelength calibration to the resulting center peak. This gave us a wavelength of 16526.71 Å, and applying Equation 2, has radial velocity 1577 km/s.

Now we consider corrections into our files by conducting the master flat and darks as outlined in Section 2. With these corrections, we reduce and extract the Neon spectrum to conduct our wavelength solution once more on this new detector. The wavelength solution was over a 30 row section centered in the FITS file and applied to the entire image regardless of the minor 1-2 pixel differences in the top of the line vs the bottom. See Figure 10 for the full linear wavelength solution. Finally, we can apply our corrections on each of the spectrum obtained, and apply correction **??** to each of the files before selecting sections of the rows about the spectra to be integrated over and plotted. The spectra are presented in Figures 11, 12, 13, 14, and 15. Each of these follow a general black body shape with Aldebaran being the least similar, and Elnath-betaTau having the most characteristics.

5. APPENDICES

5.1. *Data reduction Python implementation of dispersed spectra*

In order to create our master flat and darks for varying exposure time, we use the Python library *glob.glob*() to collect and organize FITS files of similar name style. For this example we use the $8s$ exposure dark FITS files.

```
dark8s = glob('/Users/.../Dark-000*_8s.fit')
dark8s_all = []
for name in dark8s:
    dark8sopen = fits.open(name)
    dark8s_data = dark8sopen[0].data
    dark8s_all.append(dark8s_data)
dark8s_all = np.array(dark8s_all)
master_dark8s = np.median(dark8s_all,axis=0)
print(master_dark8s.shape)
```

Following this, we open and append each of the FITS files data using the appropriate indexing and take the median across the zeroth axis. What this equivalently does is median combine the three files into one master dark for an $8s$ exposure, and reduces the random noise in the files. We can then repeat this process for the $1s$ and $10s$ darks, and the flat $1s$ files. Once we have done this, we use a general function that implements Equation 3 dependent on the exposure time done.

[2] TA Admin and Observatory Demonstrator

```
def image_correction(image,dark,flat):
    '''This function takes in an image and appropriate dark and flat
    exposure times and returns the corrected image'''
    image_corr = (image - dark)/flat
    return(image_corr)
```

And we can apply this to each of the spectra distributed.

5.2. *Tables of returned values and selected wavelengths*

We present here the table values used to produce Figure 7. Referenced table of values from the two-dimensional

Table 2: The output of *Scipy find_peaks*() function for the median of a 15 column section of the FITS file with the corresponding selected wavelength from Rousselot et. al[1]. Each respective column and row entry are in the same position in each section.

Pixel position		Intensity height		Wavelength	
(nm)		(Pixel)		(nm)	
55	71	2536.	9408.	16079.753	16128.608
83	97	3304.	9999.	16194.615	16235.376
114	125	2808.	6822.	16341.755	16350.650
146	156	1929.	3228.	16414.737	16442.155
165	181	1878.	4170.	16447.616	16502.635
190	196	2147.	3731.	16553.814	16609.994
209	246	1876.	10218	16692.380	16702.639

wavelength solution using a Neon lamp from Figure 10.

Table 3: The output of *Scipy find_peaks*() function for the median of a 30 column section of the reduced Neon FITS file with the corresponding selected wavelength from Rousselot et. al[1].

Pixel position	Intensity height	Wavelength
(nm)	(Pixel)	(nm)
411	0.33050863	540.056
574	0.63886052	640.225
589	0.28318632	650.653
637	0.48946093	671.704
656	2.58992175	692.947
676	1.82956876	703.241
690	0.79117535	717.394

5.3. *Pixel translation of [FeII] emission centroid*

We can confirm visually the central position of the 2D Gaussian fitting worked by plotting both the subsection of the OH skylines FITS file and the untouched FITS file.

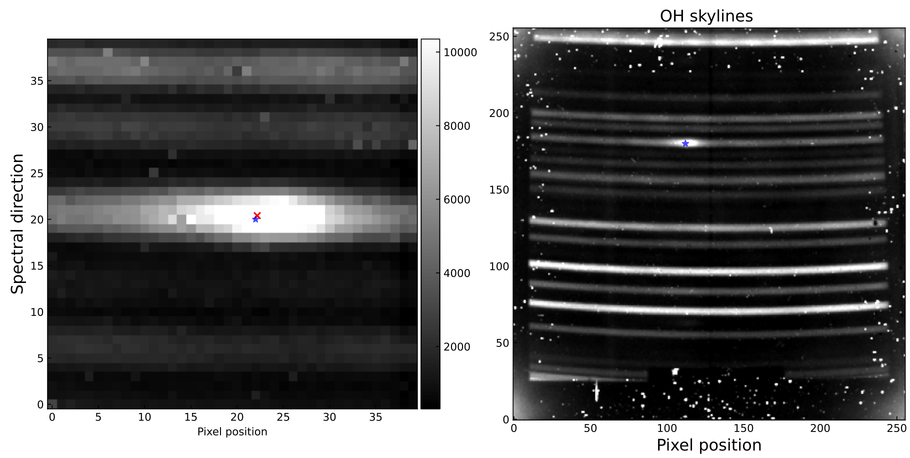

Figure 16: *Left*: The subsection of the FITS file shows the position of the centroid iron emission. We compare the returned value of *centroid_2dg*(), in red, to the rounded pixel value, in cyan. *Right*: The translated position of the centroid to the full FITS file in blue. From here we are ready to apply our wavelength solution to determine the wavelength of the [FeII] emission.

Going from left to right, we can translate between the two using,

```
centroid_x = centroid_int_position[0] + 90
centroid_y = centroid_int_position[1] + 160
```

Where we simply add the sectional slices we removed from the array. This returned a central position of (112, 180) for the [FeII] emission on the FITS file.

5.4. *Error analysis for wavelength solutions*

With finding the wavelength solution and providing a best fit that suits the data, we utilize many functions used to fit the data.

```
# define linear model function
def lin_fit(x,m,c):
    '''Defines a linear function to model the wavelength solution using curve_fit()'''
    return m*x + c
def quad_fit(x,a,b,c):
    '''Defines a quadratic function to model the 2D wavelength solution using curve_fit()'''
    return a*x**2 + b*x + c
def cubic_fit(x,a,b,c,d):
    '''Defines a cubic function to model the 2D wavelength solution using curve_fit()'''
    return a*x**3 + b*x**2 + c*x + d
# defining chi square function
def chi_square(y_measured, y_expected,errors):
    return np.sum( np.power((y_measured - y_expected),2) / np.power(errors,2) )
# define chi square reduced
```

```
def chi_square_reduced(y_measured,y_expected,errors,number_parameters):
    return chi_square(y_measured,y_expected,errors)/(len(y_measured - number_parameters))
```

The first of three of are our model functions used throughout Figures 5, 8, and 10, with the quadratic and cubic fit only used for 8. The last two are the χ^2 and χ^2 reduced. The *chi_square*() function takes in the error of the wavelengths, and returns the χ^2 of the fit and data. The reduced corrects for degrees of freedom in a line, and returns a more desirable value close to one. For each of the three figures, the errors of the wavelength were found as follows.

```
popt, pcov = sco.curve_fit(lin_fit,pixel_position,wavelengths)
# uncertainty calculations
merr = np.sqrt(np.diag(pcov)[0])
cerr = np.sqrt(np.diag(pcov)[1])
Ne_chisq = chi_square(wavelengths,lin_fit(pixel_position, *popt),wavelength_err)
Ne_chisqred = chi_square_reduced(wavelengths,lin_fit(pixel_position, *popt),2,wavelength_err)

resd_Ne = lin_fit(pixel_position, *popt) - wavelengths
```

Where popt and pcoc are the best fit parameters, and pcov is the covariance matrix, whose diagonals are the square of the standard deviation for each parameter. In addition the residuals can be calculated in the last line. This returns an array of wavelength errors dependent on the value of each wavelength of which we can pass into our *chi_square*() and *chi_square_reduced*() functions to get the fitting of our line. This returns a χ^2 of 77 and reduced of 0.47. We can note that values under one indicate an over fit of the data, but in our case our solution covers the whole domain, and does not extrapolate outside, making it a reasonable fit.

We can repeat this process for Figures 8 and 10, and summarize the results of the three figures in the table below. Note that of the three possible fit choices for Figure 8, we choose a linear function to fit the data. This maximizes

Table 4: The summarized error of three out of four methods for model fitting. We see here the error is dependent on the wavelength position and increases over time. We can use these values to pass into our chi-square functions to receive the error.

Figure 5 Error (nm)		Figure 8 Error in linear fit (nm)		Figure 8 Error in quadratic fit (nm)		Figure 8 Error in cubic fit (nm)	
0.08545772	0.38859077	8.55575335	11.04469978	0.15201743	0.19624068	0.00288685	0.00372666
0.76589467	0.84167793	12.9114096	15.08923772	0.22940812	0.26810347	0.00435651	0.00509135
1.12546206	1.19640809	17.7337433	19.44489397	0.31509067	0.34549416	0.00598364	0.00656101
1.2963775	1.33830015	22.71163616	24.26722768	0.40353717	0.43117671	0.00766326	0.00818815
1.4769674	1.51244041	25.66726005	28.15620648	0.45605229	0.50027554	0.00866054	0.00950035
1.53823897		29.55623884	30.48959375	0.52515112	0.54173484	0.01097001	0.01291208
		32.51186273	38.26755134	0.57766623	0.6799325	0.00997274	0.01028767

the error and reduces the chi-square to the best chi-square of the options within the data domain, acting as the best fit out of all the possibilities. For each of the fits we had reduced chi-square of 1.62, 3550.5, and 10303188.7 for the linear, quadratic, and cubic fits respectively. The fitting parameters for Figure 5 were 0.24 for the slope and 526 for the intercept. The fittings for Figure 8 are listed below in increased power starting with linear.

$$
\begin{aligned}
&3.45, \quad 1.6 \times 10^4, \\
&-3.59 \times 10^{-3}, \quad 4.50, \quad 1.58 \times 10^4, \\
&-1.88 \times 10^{-5}, \quad 4.95 \times 10^{-3}, \quad 3.31, \quad 1.59 \times 10^4.
\end{aligned}
$$

Table 5: The last of four methods. We see here the error is dependent on the wavelength position and increases over time. We can use these values to pass into our chi-square functions to receive the error.

Figure 10 Error (nm)	
7.46363902	10.42367104
10.69606662	11.5677325
11.9127669	12.27596102
12.53019689	

The reduced chi-sqaure for the last method was 0.1, which is an over fitting in the wavelengths. This value can be explained by the large errors of the data points, leading to a decrease in the reduced chi-square. The fitting parameters for this linear fit were 0.62 for the slope and 283 for the intercept.

6. REFERENCES

1. Rousselot, P. and Lidman, C. and Cuby, J. -G. and Moreels, G. and Monnet, G., "Night-sky spectral atlas of OH emission lines in the near-infrared",ADS Link

2. Photutiles Python library