Lab 2: Spectroscopy

Daniel Chui: daniel.chui@mail.utoronto.ca
Amanat Brar, Dushan Izcsik, Sheridan Kerr, Ruyi Xu

Nov 3, 2023

ABSTRACT

We used the method of spectroscopy with an emphasis on wavelength calibration to conclude the surface temperatures of a selection of blackbodies and the recessional velocity of an FEII gas cloud. The method of calibration involved formulating a relation between the centroid locations of recognizable intensity peaks in the well known and studied spectra of Neon and OH telluric skylines with their corresponding wavelength values. This relation is then used as a function to obtain wavelength estimations for arbitrary pixel locations. The analysis led us to identify a blackbody temperature of 4450K $\pm$ 10K and a recessional velocity for the FEII cloud of $106\mathrm{km\,s^{-1}} \pm 42\mathrm{km\,s^{-1}}$. The additional blackbody surface temperatures found for Capella and Elnath disagree with scientifically backed and accepted values (5) and so the discrepancies will be discussed.

1. Introduction

Spectroscopy is an important tool in the field of astronomy because it allows astronomers to study and make predictions about physical properties of objects in space. The method of spectroscopy involves taking a spectograph of an object which relays information about the light of said object in form of an intensity-pixel spectrum. A crucial step in spectroscopy, and the underlying motivation of this paper, is the process of interpreting such spectra using wavelength calibration and solving for what is called the 'wavelength solution'.

The physics behind the spectograph and the information it relays are based on the principle of diffraction. Diffraction is a method of separating a light source into its many wavelengths; but, while a prism separates light by passing it through another medium (refraction), diffraction is done by reflecting light off of a grated surface. The grated surface causes different wavelengths of light to reflect at slightly different angles creating a complete spectrum. The spectrum, which contains information about the wavelengths and intensities, are recorded by a camera and analysed by astronomers.

Specifically, the information relayed by the spectrograph is three fold: a pixel's position on a light detector, a relative intensity of light that pixel experiences, and a relative wavelength of light that pixel experiences as compared to others. It returns this information for all pixels on the detector. With only information about relative wavelengths, the spectrum alone is meaningless, so astronomers make use of the spectrum of a well studied and understood light source to calibrate their instrument and interpret the raw measurement. In this paper, calibration is done with two understood spectra: a neon lamp (5), and OH telluric emission lines in the night sky (5). By taking a spectrum of these control measurements and understanding their impact on the spectograph's pixel intensity and pixel wavelength we can make sense of the relative information recorded from the target object's spectrum. This relationship between pixel position and pixel wavelength is called the 'wavelength solution'.

On top of wavelength calibration, a few more actions are taken in reducing the data. Though sometimes given to us, the bulk of the spectra in this paper is extracted from a .fit file. The extraction is done in DS9 where a projection is done down the length of the image which returns pixel number and intensity values (2.2). Additionally, when taking spectra from telescope measurements, calibration images are reduced with the use of 'flats' and 'darks'. flats are images of a constant spectrum like a lamp which have the purpose of overloading the CCD while darks are an attempt at taking images with no stimulus. flats and darks are done so astronomers can have an understanding of the potential flaws in their spectroscopic measurements.

2. Data Reduction

2.1. Sources and Equipment

The spectra presented in this paper were taken from reference papers and telescope measurements. Reference papers for a neon spectrum and OH telluric skylines were used to analyse the provided black body spectrum and the FEII emission line while the spectra for Capella and Elnath were taken from telescope measurements by Michael Williams.

The equipment used when taking spectra of Capella and Elnath were the following: A Meade Lx 200 10 telescope mounted on a Sky Watcher CQ350 German Equatorial mount, a Skelyak Alpy spectrograph with an expected one nanometer per pixel resolution, and two CCDs: an ATIK 314L 1397 used for imaging, and an ATIK Titan 796 guide camera.

2.2. Spectrum Extraction in DS9

We took spectra from .fit files using the 'projection' feature in DS9. For all the files worked on in DS9 the colour map was inverted so the spectral lines of interest produced peaks rather than troughs. The contrast of the images was also adjusted to allow the spectral lines to stand out in the projection. A projection in DS9 takes a row of pixels on an image and gives corresponding intensity values for each pixel. Since the relative wavelengths of each pixel decrease from left to right, we obtain the necessary information of a spectrum. DS9 then provides ths spectrum in a .dat file which can be analysed. An image of an example spectrum extraction is provided:

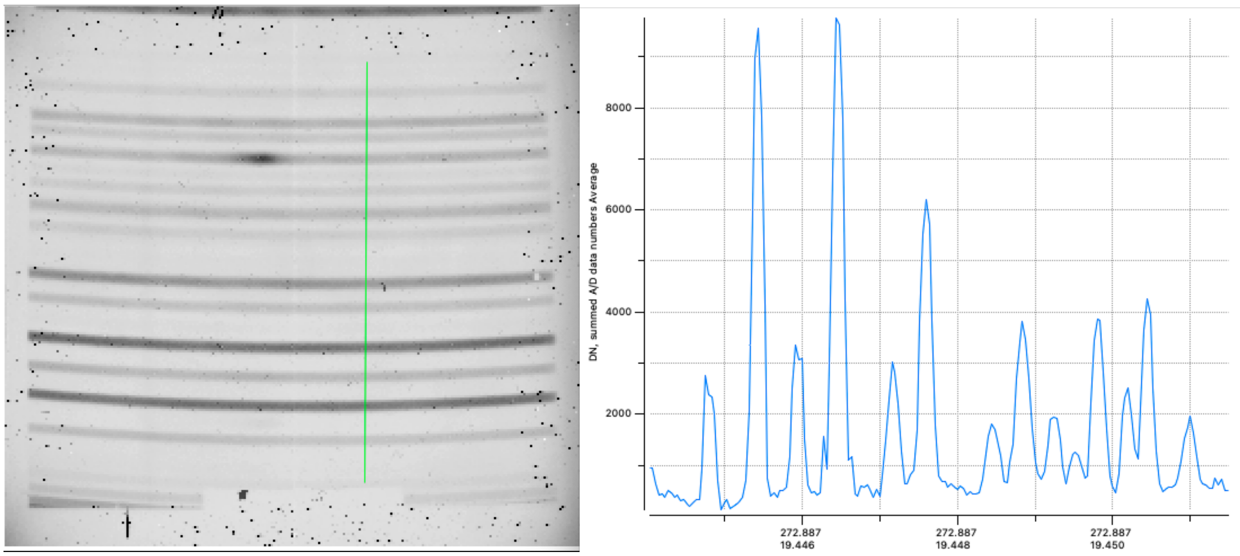

Fig. 1.—: The left image is a fit file in DS9 of OH telluric skylines with a projection taken in green and the right image depicts the corresponding spectrum for the projection. When saved as a '.dat' file, python reads the spectrum as two arrays: an array of pixels and an array of intensity values.

2.3. Dark Subtraction and Flat-field Correction

Since the spectra for Capella and Elnath were given as raw telescope images they had not undergone dark subtraction and flat-field corrections like the other spectra looked at. The dark subtraction was done by taking the median intensity values of three separate ten second dark exposures pixel by pixel in a .fit file. The ten second Capella and Elnath exposures had this new dark subtracted pixel by pixel giving us a dark-reduced image. Next, to perform a flat-field correction, we divided the dark-reduced image by a dark-reduced flat image. The dark-reduced flat image was found using the same procedure where a one second median flat image was subtracted off with a one second median dark image. The Capella and Elnath files were then ready for analysis.

3. Data Analysis

The first computational goal of this research was finding the peak temperature of a given blackbody spectrum using a wavelength solution. The blackbody spectrum was given as *Group_B_BB.dat* along with the neon calibration data, *Ne_calib.dat* and the 33 greatest peak wavelength values of neon spectrum in the form of a table, *Figure 4*, from the reference manual (5). To do this, we created a linear model that relates the spectograph's pixel intensity value to a wavelength and then fed the pixel number for the peak of the blackbody spectrum into the solution. Then we applied Wein's Law to get the temperature.

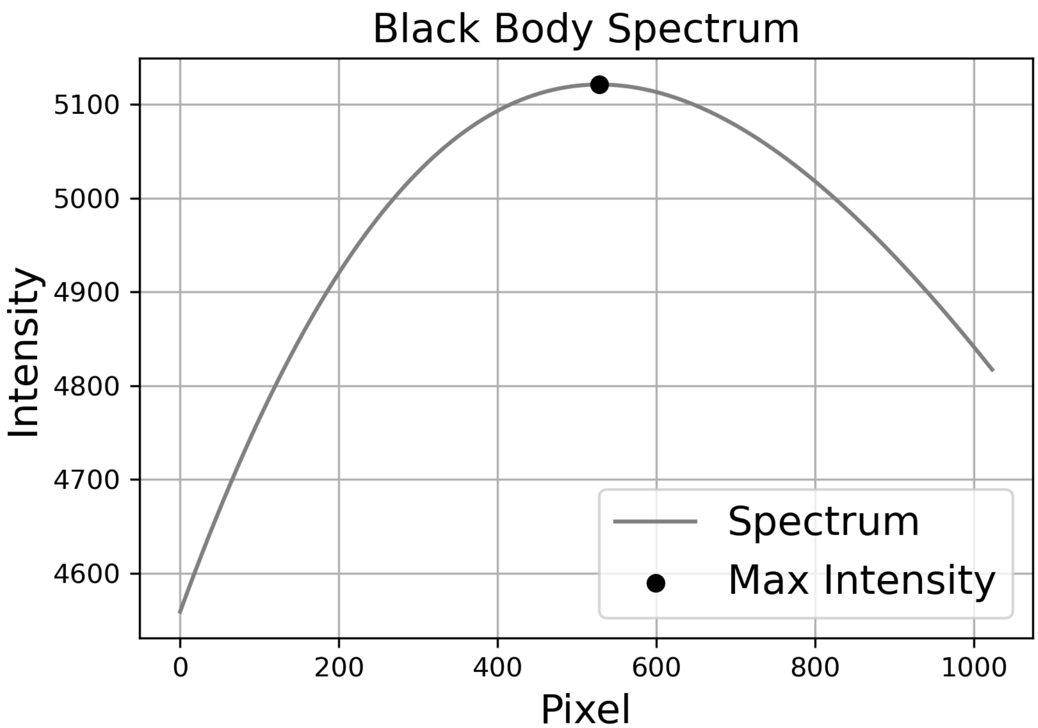

Fig. 2.—: The spectrum of the blackbody to be analysed. This plot only relays relative information about the pixel number and its intensity. We also know the wavelength increases from left to right with the pixel number, so the goal is to create an equation that will translate the relative wavelength information into concrete wavelength values.

In producing a solution we started by mapping the given 33 peak wavelengths to the Neon spectrum. By doing this we showed that specific pixel intensities from the spectograph correlate to actual wavelength measurements.

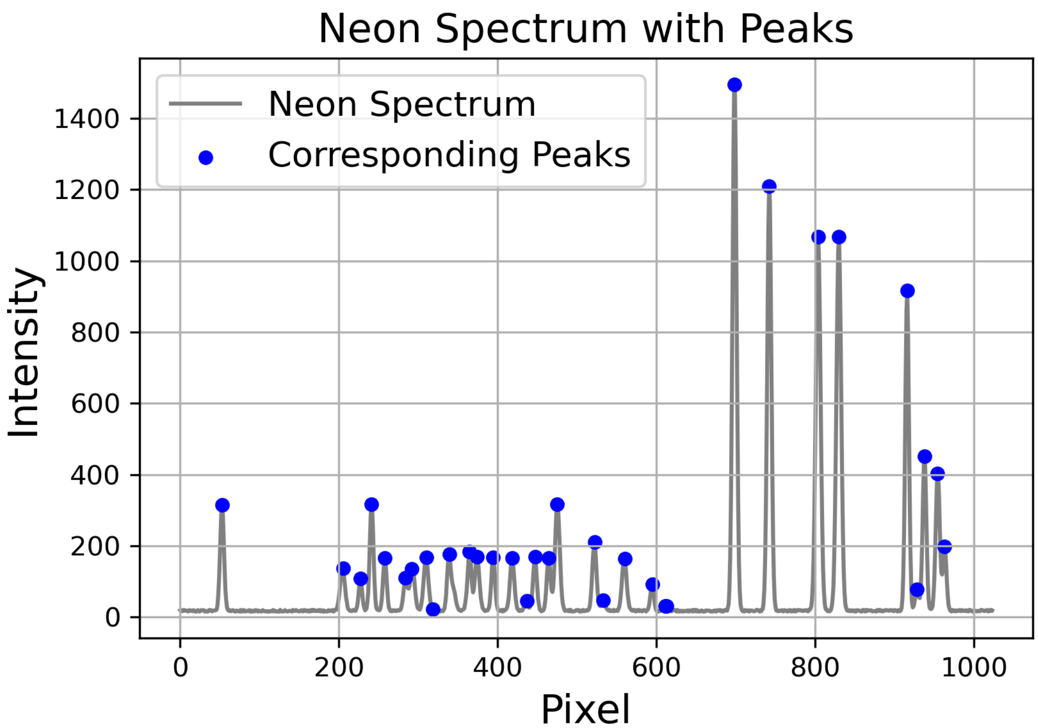

Fig. 3.—: After computationally finding the highest 33 peaks in the neon spectrum, we can now go about assigning these peaks a wavelength from *Figure 4* in ascending order.

By plotting the wavelength of each peak to its corresponding centroid (5.1) we noticed a linear trend. Taking this a step further we then performed a linear regression, $y = mx + b$, to interpolate how any specific pixel number relates to the wavelength of light received by it.

$$m = \frac{N\Sigma(xy) - \Sigma(x)\Sigma(y)}{N\Sigma(x^2) - (\Sigma(x))^2} = 0.237 \pm 0.002$$

N is the number of data points (33 in this case)

$$b = \frac{\Sigma(y) - m\Sigma(x)}{N} = 530 \pm 1$$

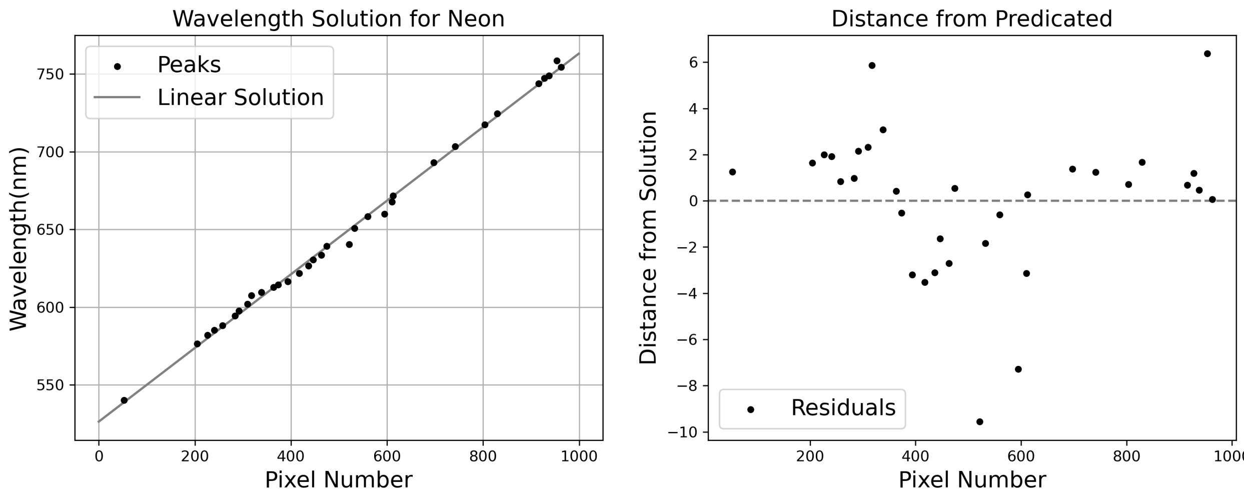

Fig. 4.—: Figure (a): The wavelength solution is the linear regression, $y = mx + b$. With this is mind we are free to choose any pixel number, x, and produce a wavelength value. Figure (b): The difference between the 33 wavelength measurements and the wavelength solution. In general, a randomly distributed residual plot is a sign of a good fit which is not the case here, but the differences here are sufficiently small that the residual plot and the trend it shows are misleading (4)(5.3).

The final step was applying the wavelength solution to the blackbody spectrum. After finding the pixel number of the highest intensity through python we used our wavelength solution to find the peak wavelength and temperature (5.4).

$$x_{max} = 528$$
$$\lambda_{peak} = m(x_{max}) + b = 651nm \pm 2nm$$
$$T = k/\lambda_{peak} = 4450K \pm 10K$$

The next goal is applying the same process of finding a wavelength solution but with a couple variations. The first is obtaining the calibration spectrum by extracting it from a .fits file and the second is producing wavelength solutions of different polynomial degrees. The wavelength solution will then be used to find the velocity of a receding gas cloud.

Similarly, we applied the method of creating a wavelength solution to a 2D spectra of OH telluric skylines. The spectra contained a spike near the $1.6\mu m$ wavelength relating to a cluster of FEII gas and using this information we estimated its receding velocity. The method of 2D spectra differed in the way that the data was reduced but after obtaining the spectra the process followed much the same (2.3). Additionally, the wavelength solution was done in three separate fits of varying polynomial degree though ultimately the solution was found to be well approximated with the linear regression (4). The slope and intercept of this solution are:

$$m = 0.377 \pm 0.001 \quad \text{and} \quad b = 1590$$

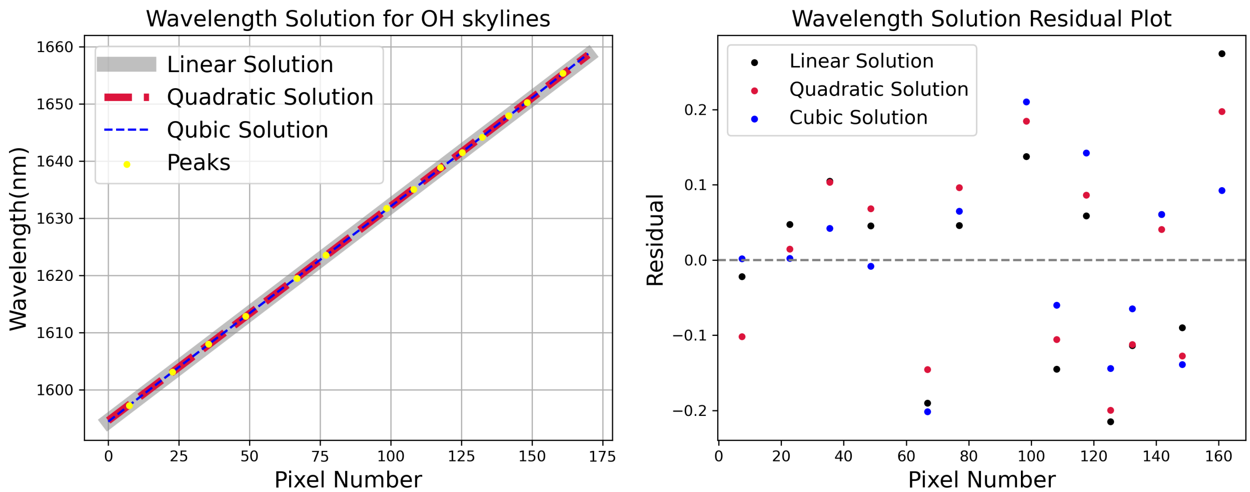

Fig. 5.—: Figure (a): The two additional polynomial fits contribute insignificantly to improving the accuracy and precision of the solution and hence appear as one solution in a scatter plot. Figure (b): The residual plot better confirms the insignificant difference between fit types since all the residuals are under a third of a nanometer wavelength.

We used the wavelength solution to find the wavelength corresponding to the pixel number of the FEII peak. This pixel value was taken to be 133 and so the wavelength and velocity-dopplershift calculations came out (5.4):

$$\lambda_{peak} = m(133) + b = 1.64\mu\text{m}$$

$$v = \frac{\lambda_{obs} - \lambda_{actual}}{\lambda_{actual}} c = 106\text{km}\,\text{s}^{-1} \pm 42\text{km}\,\text{s}^{-1}$$

Lastly, we went about finding the surface temperatures of the stars Capella and Elnath. The process followed much the same as finding the temperature of the blackbody using neon calibration only with a more intensive data reduction (5.5). With Wien's law, the temperatures were found to be:

$$T_{Capella} = 7150K \pm 160K$$

$$T_{Elnath} = 5660K \pm 110K$$

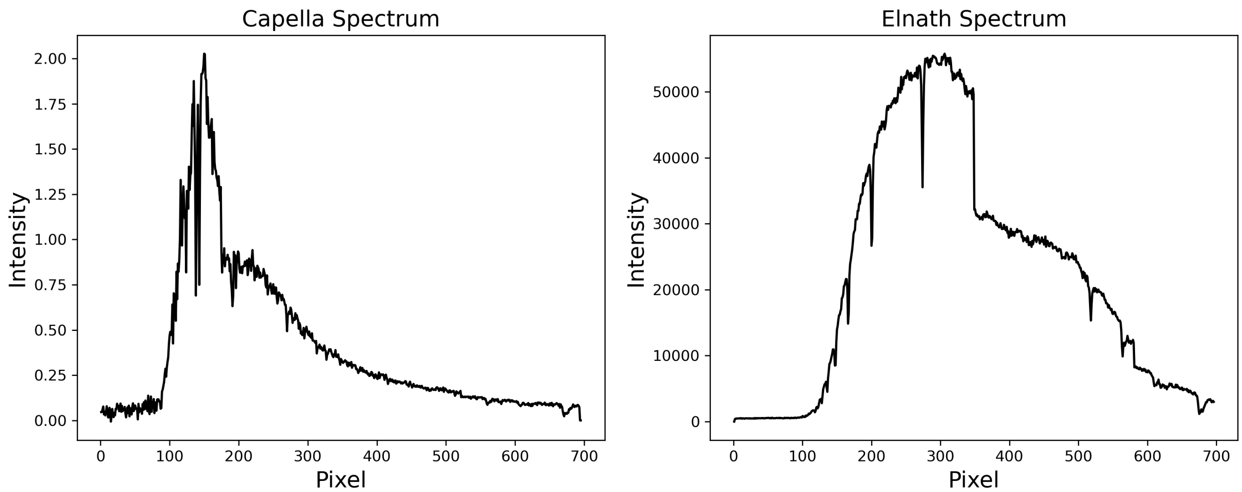

Fig. 6.—: Raw spectra of Capella and Elnath taken from DS9. Like in the case of the blackbody spectrum, the pixel numbers for the peak values of these plots are fed through the a wavelength solution and Wien's law to get a temperature.

4. Discussion & Conclusion

The residuals of the wavelength solution for the neon spectrum show a clear trend but we can argue that the solution is still useful because the residual distances are relatively small compared to the wavelength measurements used to find them. The median value for the residuals in this case is 0.68nm which is acceptable when dealing with wavelengths in the 500nm to 800nm range. Additionally we can argue the validity of using the linear fit out of the collection of polynomial fits for the OH solution because the residuals for all fits are bounded within a third of a nanometer wavelength with solution wavelength values in the 1000nm+ range.

If we assume the blackbody spectrum from our analysis represents light from a main sequence star we can try and classify it. The class represented by stars in the 4500K temperature area are K-class and have an orange colour so it stands to reason that we could be seeing light from this class (5).

The velocity of the FEII gas cloud has a large uncertainty. This is explained due to the compounding nature of the uncertainty propagation used to find it. The uncertainty for the FEII peak wavelength was on the order of e-10 which, when divided by the actual wavelength on the order of e-6, leaves us with an uncertainty in velocity on the order of (e-4)c. This uncertainty is one less order than our velocity and, by the nature of the dopplershift-velocity equation, is extremely sensitive to changes in observed wavelength values. This sensitivity is also likely the cause of the discrepancy between the measured Capella and Elnath temperatures and their commonly accepted temperature values. Since the neon calibration for the spectograph was well fit, the small deviations in observed wavelength are most likely due to noise in the spectrograph which can come from observational issues; for example, atmospheric conditions and light pollution. These examples are brought forward since the Capella and Elnath spectra were taken in downtown Toronto but its always a good idea to reduce potential noise between the telescope and the object you are viewing.

The results of the analysis, while either already well studied or not particularly useful, provide us with a better understanding of the importance of spectroscopy. Its a powerful tool and can be used to predict quantitative measurements like temperature and velocity. In turn, these quantitative measurements can be used to make qualitative conclusions like how the temperature of a main sequence blackbody relates to its class or how the red-shift velocity of a star can bound its location in a binary orbit. For future experiments, the process of wavelength calibration will be a integral step when taking spectroscopic measurements.

5. Bibliography

Mark J. Pecaut, Eric E. Mamajek. 2013, ArXiv e-prints, arXiv:1307.2657

P. Rousselot1, C. Lidman, J.-G. Cuby, G. Moreels, and G. Monnet. 2000, Astronomy and Astrophysics, v.354, 1134-1150

Michael William's word document

Lab 2 Manual including googledrive, Dae-Sik Moon

Appendix A: Error Propagation

Uncertainty for wavelength values obtained from the wavelength solutions follow the same formula because all wavelength solutions used were of the linear fit. We assumed the uncertainty on the detector to be ± half a pixel width so the error to be propagated looked like:

$$\lambda \pm \sigma = (m \pm \sigma_m)(\text{pixel number} \pm 1/2) + (b \pm \sigma_b)$$

The error propagation for this equation is broken down into two formulas for sums and products:

$$\text{Sums: } \sigma_z = \sqrt{\sigma_x^2 + \sigma_y^2}$$

$$\text{Products: } \sigma_z = \sqrt{\frac{\sigma_x}{x^2}^2 + \frac{\sigma_y}{y^2}^2}\, z$$

The calculations for the temperatures and velocities involved known values and the uncertainty of the wavelength so the propagation was straight forward:

$$\sigma_T = T\,\frac{\sigma}{\lambda}$$

$$\sigma_v = c\,\frac{\sigma}{\lambda_{actual}}$$

Appendix B: Significant Code

5.1. Centroiding

Centroids are weighted sums that return a pixel value that represents the centre of mass of a peak. They are a better representation of the peak position in the context of creating a regression.

```
#Imports
import numpy as np
import scipy as sp

#Data
PIntensity = np.loadtxt('Spectra.dat', dtype = float)
calib_data = np.loadtxt('Calibration.dat', dtype = float)
Pixel = np.arange(0,1024,1) #Range of pixel numbers for spectograph
peaks = sp.signal.find_peaks(calib_data)[0]

#Centroiding
def centroid(x,I):
    return np.sum(x*I)/np.sum(I)
range_val = 5 #Best bound for peak base width by pixel
centroids = [centroid(Pixel[peaks[i]-range_val:peaks[i]+range_val], PIntensity[peaks[i]-range_val:peaks

x = np.array(centroids)
```

5.2. Polynomial Fitting

Scipy curve_fit was used to do all polynomial fitting. The Uncertainties for the polynomial parameters were found by taking the square root of the diagonal of the covariance matrix.

```
#Imports
import numpy as np
import scipy as sp

#Data
PIntensity = np.loadtxt('Spectra.dat', dtype = float)
calib_data = np.loadtxt('Calibration.dat', dtype = float)
peak_wavelengths = np.loadtxt('Peak_wavelength.dat', dtype = float)
Pixel = np.arange(0,1024,1) #Range of pixel numbers for spectograph
peaks = sp.signal.find_peaks(calib_data)[0]
peak_wavelengths = np.loadtxt('Peak_wavelength.dat', dtype = float)
centroids = np.array[] #Contains centroid values found from the data

#Linear Fit
def P1(x,a,b):
```

```
    return a*x+b
popt1, pcov1 = sp.optimize.curve_fit(P1,centroids,peak_wavelengths)
m1 = popt1[0]
b1 = popt1[1]

#Quadratic Fit
def P2(x,a,b,c):
    return a*x**2 + b*x + c
popt2,pcov2 = sp.optimise.curve_fit(Ps,centroids,peak_wavelengths)
k2 = popt2[0]
m2 = popt2[1]
b2 = popt2[2]

#Uncertainties
unc_slope1,unc_inter1 = np.sqrt(np.diag(pcov1))
unc_k2,unc_slope2,unc_inter2 = np.sqrt(np.diag(pcov2))
```

5.3. Residuals

The residuals are the collection of distances from a particular point to the linear fit it defines. They are useful for checking the 'goodness' of a fit.

```
###Fits###

#Linear Fit
def P1(x,a,b):
    return a*x+b
popt1, pcov1 = sp.optimize.curve_fit(P1,x,y)
#Quadratic Fit
def P2(x,a,b,c):
    return(a*x**2)+b*x+c
popt2, pcov2 = sp.optimize.curve_fit(P2,x,y)
#Cubic Fit
def P3(x,a,b,c,d):
    return(a*x**3)+(b*x**2)+c*x+d
popt3, pcov3 = sp.optimize.curve_fit(P3,x,y)

###Residuals###
mean = (np.sum(x)/33)

#Linear
Ypred1 = popt1[0]*x+popt1[1]
residuals1 = y - Ypred1
R2_1= 1 - (np.sum((residuals1)**2))/(np.sum((y-mean)**2))
#Quadratic
```

```
Ypred2 = popt2[0]*x**2+popt2[1]*x+popt2[2]
residuals2 = y - Ypred2
R2_2= 1 - (np.sum((residuals2)**2))/(np.sum((y-mean)**2))
#Cubic
Ypred3 = popt3[0]*x**3+popt3[1]*x**2+popt3[2]*x+popt3[3]
residuals3 = y - Ypred3
R2_3= 1 - (np.sum((residuals3)**2))/(np.sum((y-mean)**2))
```

5.4. Temperature, Velocity and Associated Uncertainties

After obtaining the wavelength solution we use it do do physics. In this case we can convert wavelengths into a temperature or use a wavelength and compare it to the expected to find a dopplershift.

```
#Error
def multierror(x,ux,y,uy,z):
  return (((ux/x)**2+(uy/y)**2)**.5)*z

def adderror(ux, uy):
  return np.sqrt(((ux)**2)+(uy)**2)

#Temp
max_y = max(PIntensity)  #Solve for Maximum Point
max_x = Pixel[PIntensity.argmax()]

peakwave = (m*max_x+b)
k = 2.897771955e-3 #Boltzman Constant with meters
Temp = k/peakwave

unc_mx = multierror(m,unc_slope,528,0.5,m*528)
unc_peakwave = adderror(unc_mx,unc_inter)
unc_Temp = Temp*(unc_peakwave/peakwave)

#Velocity
lambda_observed = P1(133,m,b) * 10**-9 #Unit conversion into m
lambda_rest = 1.6439981e-06 #Given
c = 299792458

velocity = ((lambda_observed - lambda_rest)/lambda_rest)*c

#Uncertainty
unc_detector = 1/2
unc_mx = multierror(popt1[0],unc_slope,133,unc_detector,popt1[0]*133)
unc_wavelength = (adderror(unc_mx,unc_inter))*10**-9
unc_velocity = (unc_wavelength/lambda_rest)*c
unc_velocitykm = unc_velocity/1000
```

5.5. .fit file data reduction

After receiving a raw spectograph we need to reduce it with darks and flats allowing us reduce the noise when we take a spectrum. In this code we do reduction on Capella's spectrograph.

```
import scipy as sp
from astropy.io import fits

dark1_10 = fits.getdata('/Users/danielchui/Lab2/Fits/Dark-0001_10s.fit')
dark2_10 = fits.getdata('/Users/danielchui/Lab2/Fits/Dark-0002_10s.fit')
dark3_10 = fits.getdata('/Users/danielchui/Lab2/Fits/Dark-0003_10s.fit')

dark1_1 = fits.getdata('/Users/danielchui/Lab2/Fits/Dark-0001_1s.fit')
dark2_1 = fits.getdata('/Users/danielchui/Lab2/Fits/Dark-0002_1s.fit')
dark3_1 = fits.getdata('/Users/danielchui/Lab2/Fits/Dark-0003_1s.fit')

flat1_1 = fits.getdata('/Users/danielchui/Lab2/Fits/Flat-1s_01.fit')
flat2_1 = fits.getdata('/Users/danielchui/Lab2/Fits/Flat-1s_02.fit')
flat3_1 = fits.getdata('/Users/danielchui/Lab2/Fits/Flat-1s_03.fit')

capella = fits.getdata('/Users/danielchui/Lab2/Fits/Capella-10s.fit')

meddark_10 = sp.median([dark1_10,dark2_10,dark3_10],axis = 0)
meddark_1 = sp.median([dark1_1,dark2_1,dark3_1],axis = 0)
medflat_1 = sp.median([flat1_1,flat2_1,flat3_1],axis = 0)

reduceddarkimage = capella - meddark_10
reducedflatimage = medflat_1 - meddark_1

reducedimage = reduceddarkimage / reducedflatimage

outfile = "Reduced_Capella_whose_spectra_is_to_be_taken.fits"
```