

# Measuring AU Using Solar Observation

Anita Bahmanyar      Ayushi Singh      Morten Nyborg Stostad  
Department of Astronomy and Astrophysics, University of Toronto

Written by: Anita Bahmanyar

anita.bahmanyar@mail.utoronto.ca

Student Number: 998909098

April 2, 2014

## Abstract

In this lab our goal is to measure the 1 AU in meters by measuring the earth-sun distance with geometry, circular motion and properties of light. We will measure the rotational velocity of the sun using the Doppler shift between each side of the solar limb. We used two methods of finding the covariance and also taking Fourier Transform of the spectrums for this purpose.

## 1 Introduction

Astronomical distances and scales are large enough that it gets bothersome to use usual distance units such as meters and kilometers. Therefore, astronomers defined a new unit for length measurements called Astronomical Unit(AU), which is defined as the distance between the Sun and the Earth. In this lab we will use simple geometry, circular motion and properties of light to measure this unit. In order to do so, we found the wavelength solution of the Charged Coupled Device(CCD) that related the pixel numbers to wavelengths as explained in section [...], obtained the solar spectra along its diameter and by taking advantage of cross-correlation method as discussed in section [...], we figured out the pixel shift and therefore the Doppler Shift in the solar spectra with respect to the starting point( in order to determine the radial velocity of the Sun. From this, we can compute the radius of the sun by knowing the period of the Sun. Then finally we can compute AU using geometry as will be explained more in

section [...]. The goal of this lab is to learn to find the wavelength solution for CCDs, cross-correlate data sets with each other and find the rotational velocity of the sun in order to find its radius and finally to measure AU using simple geometry as described later.

## 2 Observation and Data Acquisition

### 2.1 Neon and Mercury Lamps

We used HR1 spectrometer which consists of Toshiba  $1 \times 3648$  CCD. Size of each pixel is  $8 \mu \times 200 \mu$ . The wavelength range of this spectrometer is 523-573 nm. We used Neon and Mercury lamps to find the wavelength solution for this spectrometer. We set the integration times such that the pixels are not saturated. Therefore, we used 200 ms and 400 ms integration time for neon lamp and 10 ms and 50 ms integration times for the mercury lamp. We also took data of dark counts with the same exposure times for each lamp.

### 2.2 Solar Observing

We went up to 16th floor of Burton Tower to use the 8-inch refractor telescope for our purpose. We mounted another telescope cap in front of the telescope aperture in order to reduce the amount of light coming into telescope from the Sun so that we do not hurt our eyes. Then we mounted the projection screen as it can be seen in Figure 1 so that we get the Sun on the screen. Next step was to line up the shadow of the finder scope until it became circle as in Figure 2 and then focused the the telescope to get a sharp image(We did this by looking at the sharpness of the sun spots). The next thing we need to consider is the temperature of the spectrometer since the spectral resolution and dispersion of the the spectrograph or in general, the wavelength solution changes with temperature and for this mean we placed the spectrometer inside a box to keep the temperature approximately as it was in the astronomy lab, where we collected Neon and Mercury data to solve for the wavelength solution. The box consisted of a resistor that heats up the box. Then we connected the fibre coupler to collimate [...]

## 3 Analysis

### 3.1 Wavelength Solution

We calibrated the wavelength of the spectrometer since we need to figure out how wavelength is related to the pixel numbers to the wavelength. We used neon and



Figure 1: This figure shows the projection screen mounted on the telescope to get and the image of the Sun on the screen(indirect observing)



Figure 2: This figure shows the shadow of the finder scope as a circle which means it is lined up correctly. We also focused the telescope.

mercury lamps for this purpose. We took 200, 400 and 1000 ms integration times for neon and 10 and 50 ms integration times of data for mercury as well as dark data sets for all of these integration times. We then subtracted dark from the data sets. Figure [...] shows the the intensity vs. pixel number of 10ms mercury and 200ms neon lamp graphs. We then found the peaks of the graph using centroid method as discussed in Appendix section. Then we compared these peaks with a reference peak wavelength for neon and mercury in the spectrometer wavelength range and plotted the wavelengths vs. pixel numbers in order to find the wavelength solution. We used linear least squares to fit the graph. This was done manually in Python as explained in Appendix, so it means we did not use the built-in package of Python. Fig. [...] shows the wavelength solution, which is  $y = 0.014x + 524.61$ , where  $y$  is the wavelength and  $x$  is the pixel number.

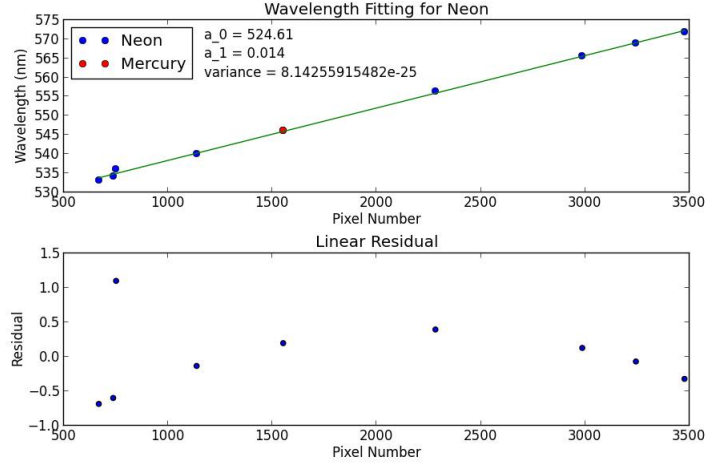


Figure 3: The upper image shows the wavelengths vs. pixel numbers of neon and mercury peaks as compared with their reference spectrum. The wavelength solution is written in the graph. The lower image shows the error in the data and as it is visible it is very small and the variance is also very small that is negligible.

Figurea [...] and [...] show the intensity vs. wavelength for neon 200ms and mercury 10ms.

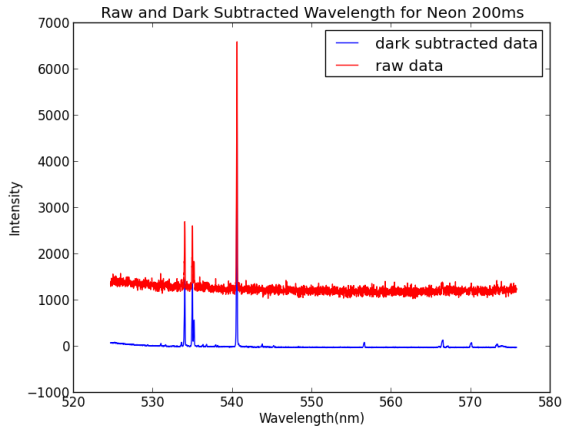


Figure 4: This figure shows the intensity vs. wavelength of neon 200ms. The raw data is very noisy unlike the mercury raw data.

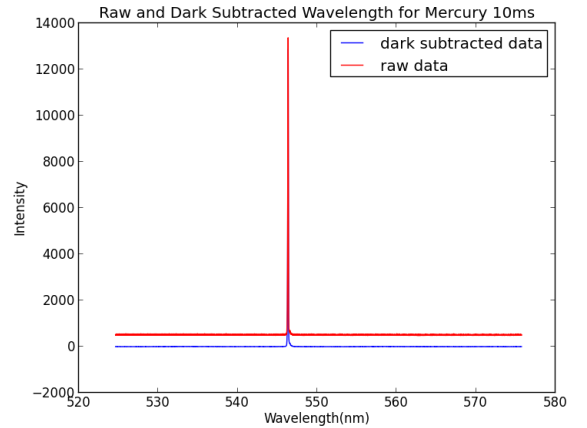


Figure 5: This figure shows the intensity vs. wavelength of mercury 10ms.

### 3.2 Solar Data

After finding the wavelength solution of the spectrometer, we should work with solar data. One of the solar data spectrums is plotted vs. wavelength in Fig. [...].

We collected 170ms and 200ms data sets on March 13, 2014 and three sets of data with 125ms integration time on March 18, 2014. We found the mean of each data file and plotted these mean values vs. time. The times were extracted from the file names as it will be discussed in details in Appendix. Then we considered the limb darkening which is given by equation [...]:

$$I = \frac{2}{5}I_0\left(\frac{3}{5} + \sqrt{\frac{(t - t_0)^2}{\Delta t^2}}\right) \quad (1)$$

$I_0$  is the intensity of the center value,  $\Delta t$  is the difference in the time of the two start and end point data and  $t_0$  is the time of the center data

## 4 Determining Doppler Shift

As we were collecting the solar data, the sun was rotating. We need to account this rotation since it affects the wavelength of the light that we observe. As the sun rotates, one edge is turning toward us, making the light to appear blue-shifted (light has shorter wavelength) and the other edge is turning away from us, making it to appear red-shifted (light has longer wavelength). This is known as Doppler Shift and it is given by equation [...]:

$$\frac{\Delta\lambda}{\lambda} = \frac{\Delta v}{c} = \frac{2v_{rad}}{c} \quad (2)$$

Where  $\Delta v$  is the change in velocity,  $\Delta\lambda$  is the change in wavelength given some reference wavelength,  $\lambda$  and  $c$  is the speed of light in vacuum. The spectrometer we used has a velocity resolution of 33 km/s. Figure [...] shows the spectrums of the two edges of the sun. As it can be seen in the image, even the two furthest spectrums are very close to each other.

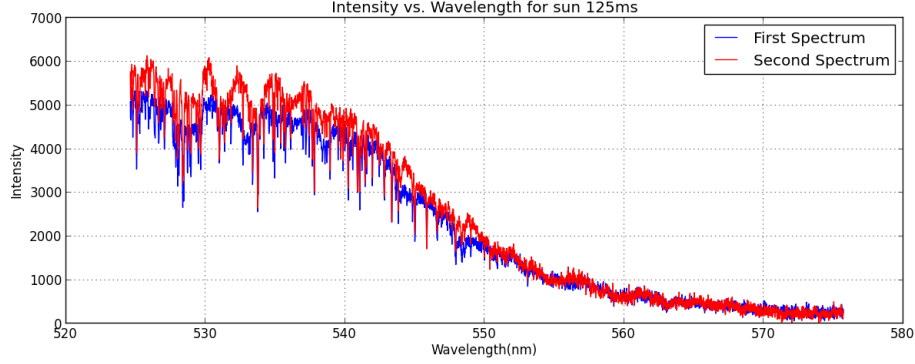


Figure 6: *Top* :Two spectra of the two edges of the sun over-plotted. As it can be seen, the two spectra are very close to each other. *Bottom* : The two spectra are zoomed so the features are better shown.

## 4.1 Cross-Correlation

Cross-correlation is a method used in this lab in order to show how similar the different spectra are with respect to a reference spectrum of the sun. Covariance is a statistical parameter that shows how correlated two variable are. Covariance can be computed by Eq. [...]:

$$s_j^2 = \frac{1}{N-1} \sum (x_i - \bar{x})(y_i - \bar{y}) \quad (3)$$

Where N is the size of the data set,  $\bar{x}$  and  $\bar{y}$  are the mean values of the two data sets. In theory, the maximum value of the covariance shows where the two data sets are most similar. In order to find the shift in the pixel values of the two data sets, we considered an array from -10 to 10 and iterated it by 1 pixel each time, and computed the covariance. Then we plotted covariance vs. the array we formed. Then we should find the centroid of the cross-correlation in order to find the pixel shift. This can be computed by Eq. [...]:

$$\langle j \rangle = \frac{\sum_j j X_j}{\sum_j X_j} \quad (4)$$

## 4.2 Window Function

In applying cross-correlation we shift the spectra relative to one another. In doing so, we beginning and the tail of the spectra would not match and line up with other spectra since our data set is finite. Since our data is discrete (we cannot integrate from  $[-\infty]$  to  $[\infty]$ , causes the energy from the true frequency to "leak" into adjacent frequencies. Leakage is one of the most common digital signal processing errors and it cannot be eliminated completely, it can only be minimized. Therefore, we could use the window functioning method that weights the beginning and end values of the data

set to zero and it also minimizes edge effects and leakage of frequencies and amplitude of the signal. In this lab we used the Hanning window function which is given by Eq. [...] below:

$$w(n) = 0.5(1 - \cos(\frac{2\pi n}{M-1})) \quad (5)$$

It is used for smoothing and it is also known as apodization which means that it "removes the foot", i.e. smoothing discontinuities at the beginning and end of the sampled signal) or tapering function.

## 5 Discussion

Using the methods discussed in section 4, we computed the pixel shift from the two end spectra using the one that starts earlier as the reference spectrum. Then we used the wavelength solution we found for the spectrometer to convert this pixel shift into wavelength shift so that we could use Doppler Shift equation given by Eq. [...] to compute the radial velocity of the Sun. The pixel shift found is -0.511 which corresponds to approximately -0.008 nm shift in the wavelength. The radial velocity then would be -2.30 km/s .

Figures [...] and [...] show the spectrum of the beginning of the sun and the plot of the window function applied to the same data, respectively.

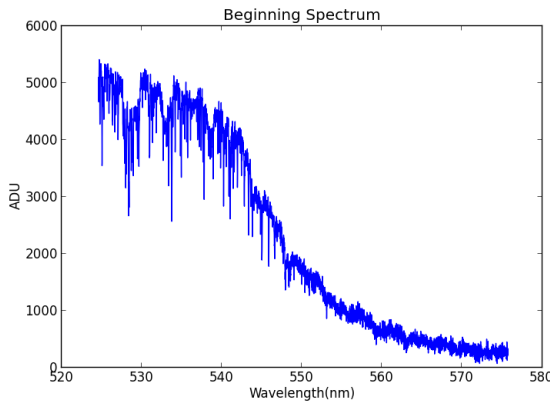


Figure 7: This figure shows the intensity vs. wavelength of beginning of the solar spectrum. This is dark subtracted and divided by flat.

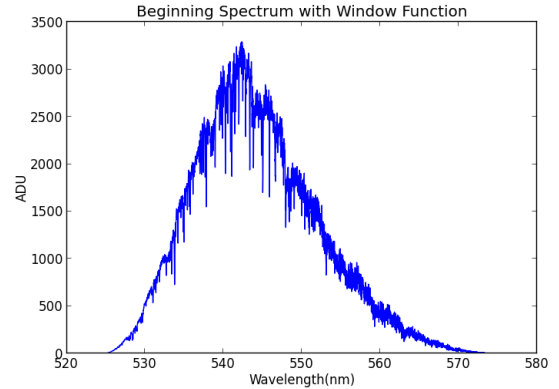


Figure 8: This figure shows the beginning spectrum of the sun which hanning window function is applied on.

After applying the window function to the spectra, I recalculated the pixel shift and it was 0.508, so there is a small change in the pixel shift when window function

was applied.

Using simple geometry, we can finally find the distance  $D$ , which is the distance between the Sun and the Earth. Fig. [...] shows the sketch of the geometry and the math is shown in Eq. [...].

$$D = \frac{R_{sun}}{\tan(\theta/2)} \quad (6)$$

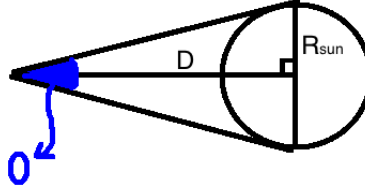


Figure 9: This is a sketch of the simple geometry for calculating the AU using the computed values of angular size of the sun and its radius.

## 6 Conclusion

## 7 References

1-<http://www.aseq-instruments.com/HR1.html> 2-The lecture slides

## 8 Appendix

We first discuss how we found the peaks of the spectrum of neon and mercury using centroid method. Below is the code I used to find the centroids. It goes through all of the pixels and checks the intensity value of each pixel with its neighbour pixel intensity value. If it is higher than both its left and right pixels it points as a peak.

```
for i in range(0,3652,1):
    if values[i+1]>values[i]:
        i+=1
    if values[i+1]<values[i]:
        if values[i]>70:
            Pixellist.append(i)
            centroidintensity = 0.5*(values[i]+values[i+1])
```



```
centroidintensitylist.append(centroidintensity)
```

Here is how fitting wavelength vs. pixel values work using linear squares.

```
ma =np.array([[np.sum(Pixels**2),np.sum(Pixels)], [np.sum(Pixels),len(Pixels)]])
mc =np.array([[np.sum(Pixels*Wavelengths)], [np.sum(Wavelengths)]]) #matrices
mai = np.linalg.inv(ma)
md = np.dot(mai,mc)
```

Next, we will discuss how the times of the solar observations were extracted from the file names.

```
times=[]
for k in range(0,66,1):
    names=np.loadtxt('sun170ms-filenames.txt',dtype='str')
    minutes=names[k][17:19]      # taking minute parts of the file name
    minutes=float(minutes)*60    # converting str to int and converting to seconds
    seconds=names[k][19:21]      # taking second parts of the file name
    seconds=float(seconds)       # converting str to int
    milliseconds=names[k][21:24] # taking mili second parts of the file name
    milliseconds=float(milliseconds)/1000 #converting str to int and to seconds
    time=minutes+seconds+milliseconds
    times.append(time)
t_center=times[5]+(times[59]-times[5])
Delta_t=(times[59]-times[5])/2
I_o=7550 #central intensity
I=I_o*((2./5)+((3./5)*(np.sqrt(1-(((times-t_center)**2)/(Delta_t**2))))))
```