

Lab Report Three: Astronomical Spectroscopy

Jinhao Zhang

Group Partners: Mike Gu, Epsilon Mao

Email: jinny.zhang@mail.utoronto.ca

Utorid: zha12748

Lab Group: D

December 9, 2024

ABSTRACT

Spectroscopy is a fundamental tool in modern astronomy, offering critical information about the physical and chemical properties of celestial bodies. This lab use the Ocean Optics USB650 spectrometer for laboratory analysis, and the LISA spectrograph for telescopic observations. The focus is on calibrating wavelength mapping, spatial distortions, and flat field effects. Wavelength calibration was achieved using hydrogen and neon-argon spectral lines, with Gaussian fitting and chi-squared tests optimizing polynomial models, resulting in average uncertainties of ± 0.64 nm for hydrogen and ± 1.76 nm for neon-argon. Spatial calibration addressed 2D distortions, while detector calibration normalized intensities and corrected for thermal and fixed-pattern noise. The calibration process successfully identified hydrogen in mean-sequence stars, planets, and detected rich and complex source of emission and absorption of elements in Nebulae.

1. Introduction

From determining the elemental compositions of stars to measuring the redshifts of distant galaxies, spectroscopy provides essential tools for probing the universe. This method underpins much of modern astronomy by utilizing slits to isolate incoming light, which is then dispersed into its constituent wavelengths using a diffraction grating or prism [Vanderlinde \(2024c\)](#). The resulting spectrum allows for the precise analysis of the physical and chemical properties of celestial objects. The development and calibration of spectroscopic instruments are therefore crucial for ensuring accurate and meaningful observations.

In this lab, we focus on both laboratory-based and astronomical spectroscopy to explore the capabilities of modern spectrometers and investigate their calibration processes. Specifically, we utilized the USB650 spectrometer and the LISA spectrograph to measure and analyze spectra from controlled sources, such as gas discharge lamps, as well as celestial targets, including stars and planets. A critical step in the experiment is wavelength calibration, which involves mapping the pixel positions from the detector to the physical wavelengths using known spectral lines [AST325/326 \(2024\)](#). Additionally, we address detector noise, a key source of uncertainty in spectroscopic measurements, by examining and correcting for dark current and flat-field effects.

Astronomical spectroscopy is particularly dependent on statistical methods to interpret data accurately. For example, calibration processes involve fitting spectral line data to theoretical models, often requiring polynomial fits to describe relationships between pixel positions and wavelengths. Noise characteristics, such as thermal noise and photon noise, can be modeled using statistical distributions like Gaussian distributions, highlighting the importance of statistical in improving the reliability of observational data.

This report begins by detailing the experimental setup, including the operational principles of the USB650 and LISA spectrometers. The calibration processes for laboratory and astronomical spectra are described, followed by the results obtained from these analyses. Finally, we discuss the implications of the findings for both the calibration of spectrometers and the broader field of astronomical spectroscopy.

2. Data and Observation

The following table gives a detailed description of the data collection process in each lab session.

Date	Personnel	Notes
7/11/2024	M. Yiquan, Z. Jinhao, G. Yanjie	Familiarize the USB650 spectrometer, Collected data for hydrogen lamp and mysterious gas for inspection
13/11/2024	M. Yiquan, Z. Jinhao, G. Yanjie	Collected raw data for moon, Saturn, Neon lamp, Tungsten lamp, and dark data from LISA Telescope
14/11/2024	M. Yiquan, Z. Jinhao, G. Yanjie	Collected data for hydrogen lamp and mysterious gas at 15ms integration time with the USB650 spectrometer

Table 1:: Group D Data Collection Layout for Lab 3

2.1. USB650 Spectrometer Setup and Data Collection

From Figure 1, we can identify the lamp (1), fiber (3), spectrometer(2), and lab laptop (4). To setup the USB650 Spectrometer, we first connect the USB650 Spectrometer to the lab laptop. Next, we open up OceanView software on the laptop to ensure that it can locate the spectrometer. Then, we turn on the gas lamp and point the fiber at the lamp. We need to ensure that the photons are not over saturated on the OceanView software. Finally, we can take the data. Below is our specifications in the OceanView software for the spectrometer and a diagram for the instruments used.

Setting	Value
Integration Time	15 ms
Number of Images	10
Trigger Mode	4
Scans to Average	1
Electric Dark Correction	True
Boxcar width	0

Table 2:: USB650 Spectroscopy Settings

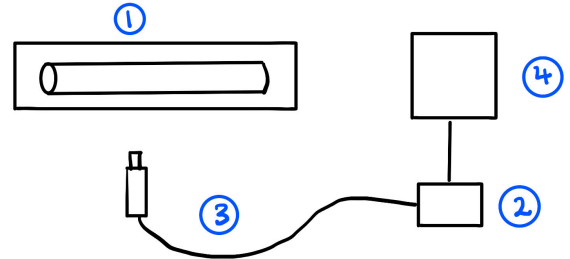


Fig. 1.—: Schematic of Setup

We would repeat the same process for the mysterious gas lamp, ambient data, and dark data with the same specifications. I averaged the data in the 10 images and plotted the spectrum for each. The plots can be found in the Appendix 6.

2.2. LISA Telescope Data Collection

To gather data from the LISA Telescope, we first use TheSky software to locate the stellar source we are interested in. The telescope will point itself towards the source. We would then turn the window so that the

telescope have a clear view of the sky. Next, we use Max DL Pro 6 software to capture the spectral data for inspection. Below is the information on what was gathered that day and the settings of the telescope.

Time	Object	Exposure Time
19:53	Moon	100s
20:13	Saturn	300s
20:15	Neon Lamp	20s
20:26	Tungsten Lamp	10s

Setting	Value
Temperature	-15 °C
Focal Length	4400mm
Aperture Diameter	400mm
Pixel Dimension	6.45 μ m x 6.45 μ m
Binning	1 x 1

Fig. 2.—: Data Collection Log for LISA Telescope

Table 3:: LISA Spectrometer Settings

However, the Saturn data had cloud blocking the sky so we just got a spectrum of the city lights instead. We decided to not use the Saturn data and instead analyze Vega, Orion Nebula, and Jupiter from the master data file in addition to the Moon. The raw spectrums can be found in Appendix 6.

3. Data Reduction

In order to analyze the spectrum from the spectrometers accurately, we have to first perform calibrations. For the USB650 Spectrometer, we implemented spectral calibration. For the LISA spectrometer, we used spectral, spatial, and detector calibration.

Spatial calibration was performed to align the spatial dimension of the 2D detector image. For each row of the detector, we identified the brightest pixel (representing the peak intensity) and plotted its pixel position against the corresponding row number [AST325/326 \(2024\)](#). A linear fit was applied to determine the spatial calibration.

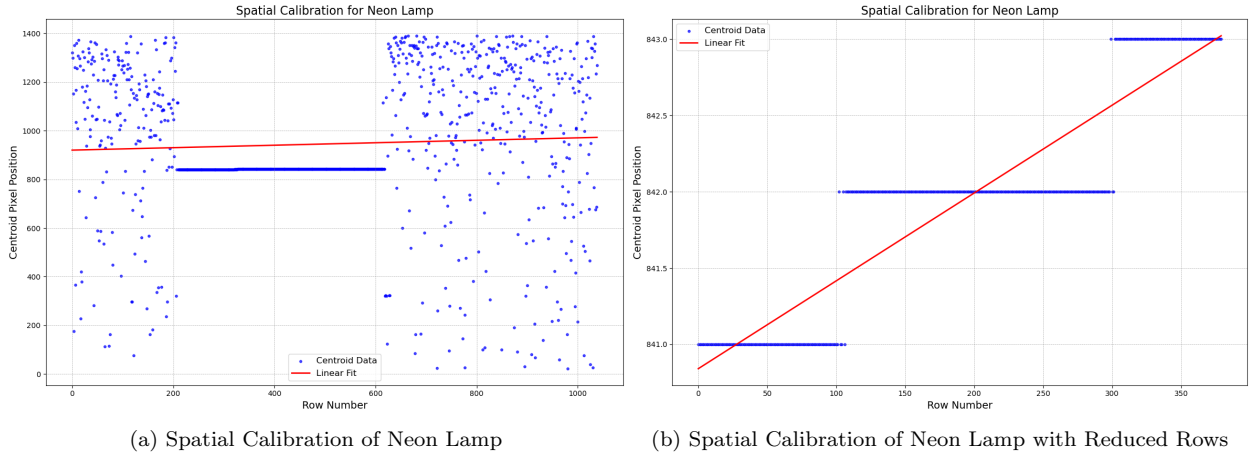


Fig. 3.—: Spatial Calibration Linear Fit for LISA Telescope

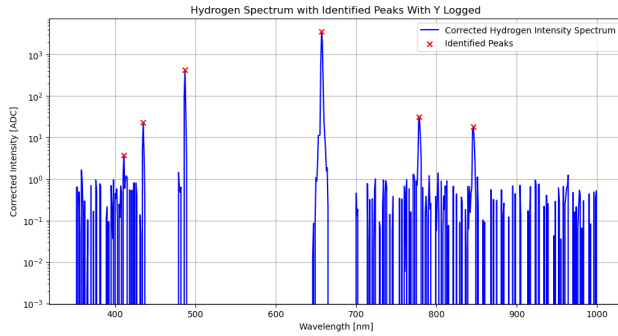
Figure 3a shows that there are random points scattered around the edge of the graph. This is because the camera is larger than the telescope’s detector, so the spectrum captured not going to cover the whole image [AST325/326 \(2024\)](#). Because of this, we limit the rows we are interested in. After doing this, we have

a good linear relationship between the row number and centroid pixel position shown in 3b. This calibration step ensures that variations in spatial alignment across the detector are accounted for, allowing accurate spectral extraction from the 2D image.

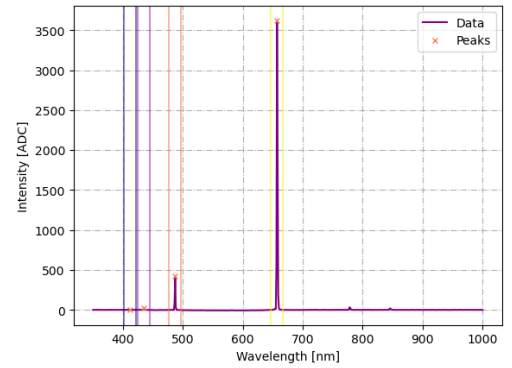
To determine the relationship between pixel positions and wavelength, we performed spectral calibration. We first find the peaks in the data and match the corresponding wavelength of the element’s emission wavelength. To find the peaks, we decided to use centroids. The centroid of a peak was calculated using the formula below:

$$x_{\text{cent}} = \frac{\sum x_i I_i}{\sum I_i} \quad (1)$$

where x_{cent} is the centroid, x_i represents the pixel positions, and I_i denotes the intensity values Nerval (2024). This approach accounts for the distribution of intensity values around the peak, providing sub-pixel precision that is not achievable by merely taking the pixel with the maximum intensity. Below is the centroiding process for the USB650 Spectrometer.



(a) Hydrogen Spectrum with Y Logged

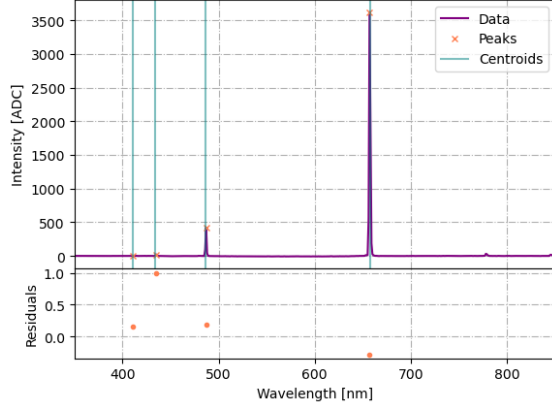


(b) Spectrum of Hydrogen with Peaks and Range Identified

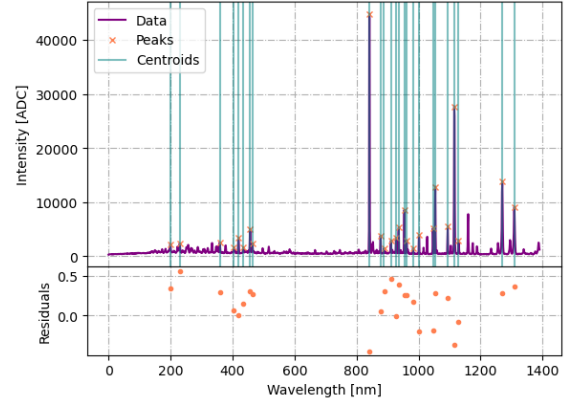
Fig. 4.—: Spectrums of Hydrogen

We would first subtract the ambient data from the hydrogen data to remove dark noise and the intensities of photons from the lights in the room. From Figure 4a, six peaks were identified. However, only the first four peaks was used for the calibration as the other two are Oxygen Emission wavelength which will be discussed further in Section 4. The four peaks are pixel position, 411, 435, 487, and 657 respectively. With the four peaks identified, a range was identified and shown in Figure 4b where the centroid calculation would start and end from and each range ensures that only one peak is in each range.

After determining the centroids of the brightest spectral lines, we mapped the pixel positions to known wavelengths using data from NIST. For the hydrogen lamp, we used wavelength 410.174nm, 434.046nm, 486.136nm, 656.285nm NIST (2024a). For the neon lamp calibration, we found out it was actually a neon argon lamp, so we did three repeated spectral calibration Vanderlinde (2024b). We first started with the nine points and corresponding wavelength given to us in Lecture 12. Then, we would calibrate and plot out the the spectrum again and try to match more wavelength to the centroids. After three repeats, 25 peaks were used in the spectral calibration of the LISA spectrometer data. The process is same as the hydrogen lamp. Below are the plots for the final centroids for both the hydrogen lamp and the neon argon lamp.



(a) Hydrogen Spectrum with Centroids and Residuals



(b) Neon-Argon Spectrum with Centroids and Residuals

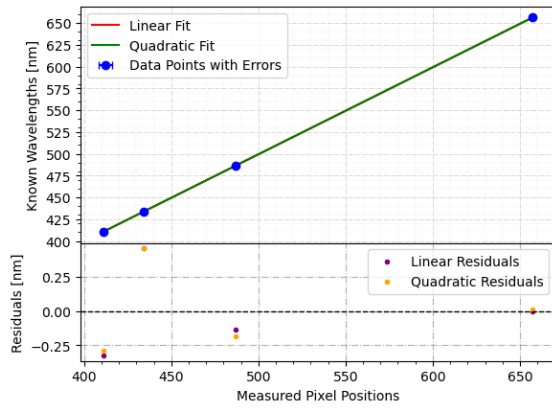
Fig. 5.—: Spectrums with Centroids

To estimate the uncertainties in the centroid positions, we modeled the intensity profiles of each peak as Gaussian distributions. By fitting Gaussians to the data, the uncertainty in the centroid position would just be the standard deviation of the Gaussian. The plots and implementations can be found in the Appendix 6.

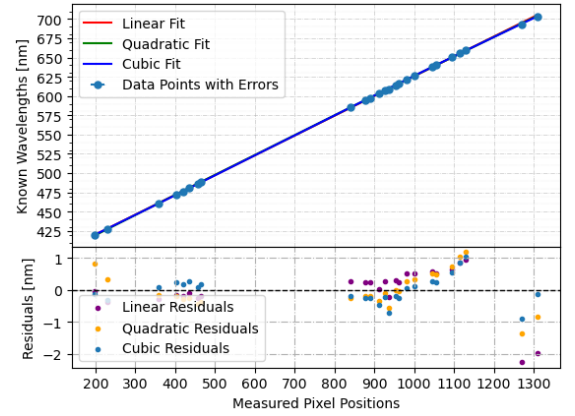
A polynomial fit was then applied to this mapping. The calibration process involved assessing different polynomial degrees using the reduced chi-squared statistic:

$$\chi^2_\nu = \frac{\sum \left(\frac{\text{Observed} - \text{Model}}{\sigma} \right)^2}{\nu} \quad (2)$$

where ν is the number of degrees of freedom, and σ is the uncertainty in the observed values [Vanderlinde \(2024a\)](#). The model with the lowest reduced chi-squared value was selected as the best fit. For the USB650 Spectrometer, we found that a linear fit is the best fit, and for the LISA Telescope, we found that a cubic fit would be the best fit for the calibration. Below are best fit graphs:



(a) Polynomial Fit for Hydrogen Lamp



(b) Polynomial Fit for Neon-Argon Lamp

Fig. 6.—: Polynomial Fits Between Pixel Positions and Wavelengths

The final step in the data reduction was detector calibration, which corrects for non-uniformities and noise in the raw data. Using the formula provided in the lab manual:

$$P_i = \frac{R_i - D_i}{F_i - D_i},$$

where P_i is the calibrated pixel value, R_i is the raw pixel value, F_i is the flat field pixel value, and D_i is the averaged dark frame pixel value [AST325/326 \(2024\)](#).

Dark frames were averaged to minimize noise, while flat fields corrected for pixel-to-pixel sensitivity variations and uneven illumination across the detector. These corrections ensured that the calibrated data represented true variations in the observed signal.

4. Data Analysis

In Figure 6a, we present both the linear and quadratic fits to model the relationship between pixel position and wavelength. The residuals for both fits appear random, indicating that the models effectively describe the data. However, a closer examination reveals that the residuals for the linear fit are consistently smaller than those for the quadratic fit. This suggests that the linear model provides a more accurate and simpler representation of the data. Therefore, we choose the linear fit as the preferred model for our analysis. In addition, we found that the chi-squared values for linear, quadratic, and cubic are 0.010946319066561818, 0.007656646580026887, 9.111911435154445e-25 respectively. We can see that the chi-squared are not decreasing that much, also justifying why linear fit is the best model. This is the same reason for choosing the cubic fit in Figure 6b for the spectral calibration for the LISA Telescope data. After the calibration, we plot out the Hydrogen Spectrum.

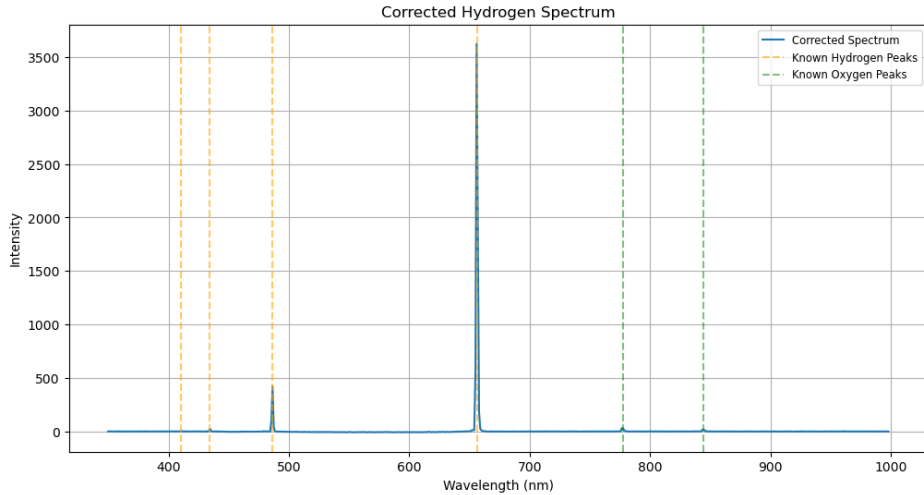


Fig. 7.—: Corrected Hydrogen Spectrum

The known hydrogen peaks align well with our spectrum, with the alpha, beta, and gamma emissions clearly identified. The final hydrogen emission is less visible but becomes apparent after applying a logarithm-

mic scale to the Y-axis. Additionally, the peaks at 777.5 nm and 844.6 nm correspond to oxygen emissions, likely due to oxygen contamination in the hydrogen lamp, potentially from an incomplete seal [NIST \(2024b\)](#). Now we use our calibration on the mysterious gas and find out that it is a neon lamp. The spectrum is shown below.

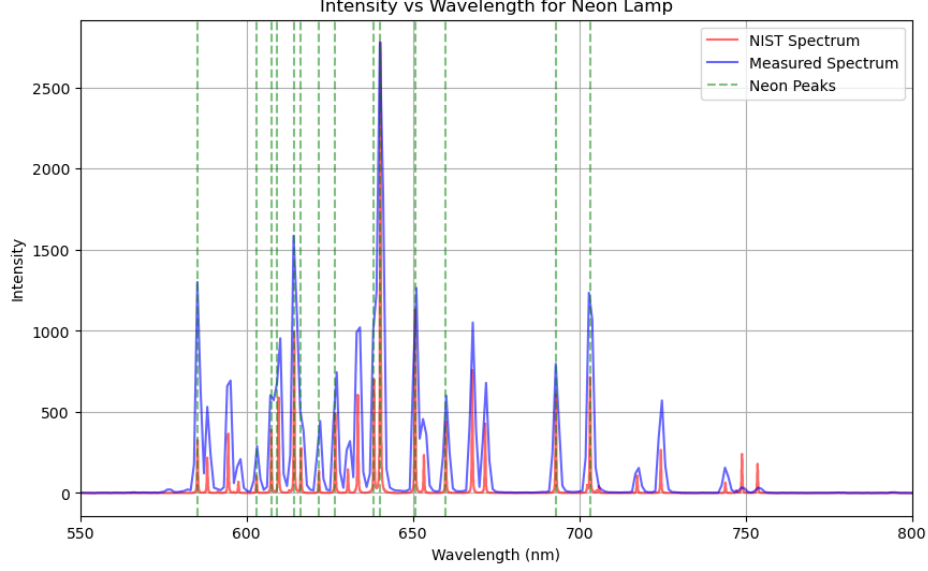


Fig. 8.—: Corrected Neon Spectrum

Our spectrum aligns closely with the NIST Neon data, accurately matching the strong neon emission lines listed by NIST [AtomTrace \(2024\)](#).

Now we move on to the LISA Telescope spectrums. After performing the spatial, spectral, and detector calibration, we have the following graphs for a single spectrum and the 2D spectrum of the Neon-Argon lamp.

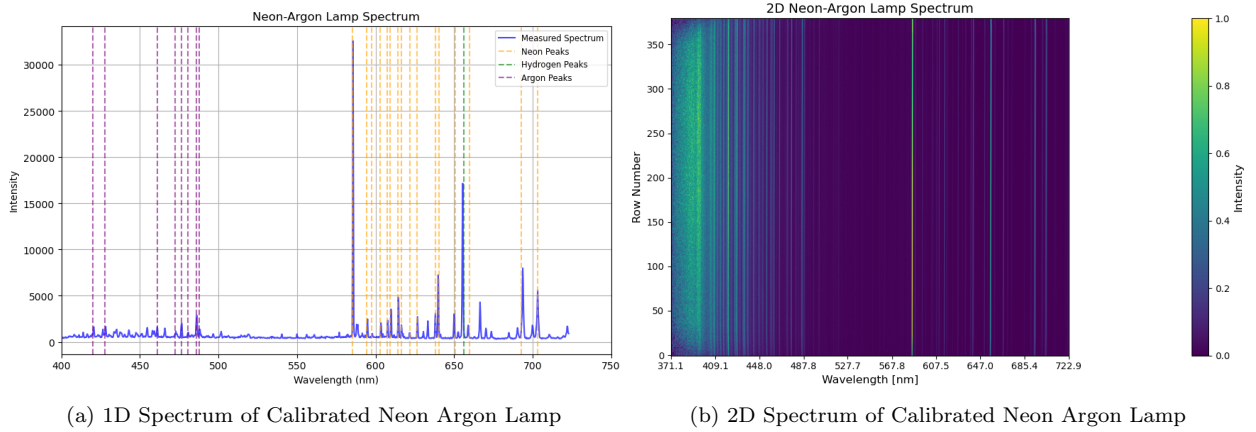


Fig. 9.—: Spectrums of Calibrated Neon Argon Lamp

The calibrated 1D and 2D spectra (Figure 9) show excellent alignment with the known Neon and Argon emission lines, as well as the Hydrogen Alpha line. The strong emission lines from Neon match precisely with the NIST Neon data, confirming the accuracy of the calibration. The alignment of these lines in both spectra validates the successful application of the spatial, spectral, and detector calibrations. Now I calibrate the Moon and Vega for analysis.

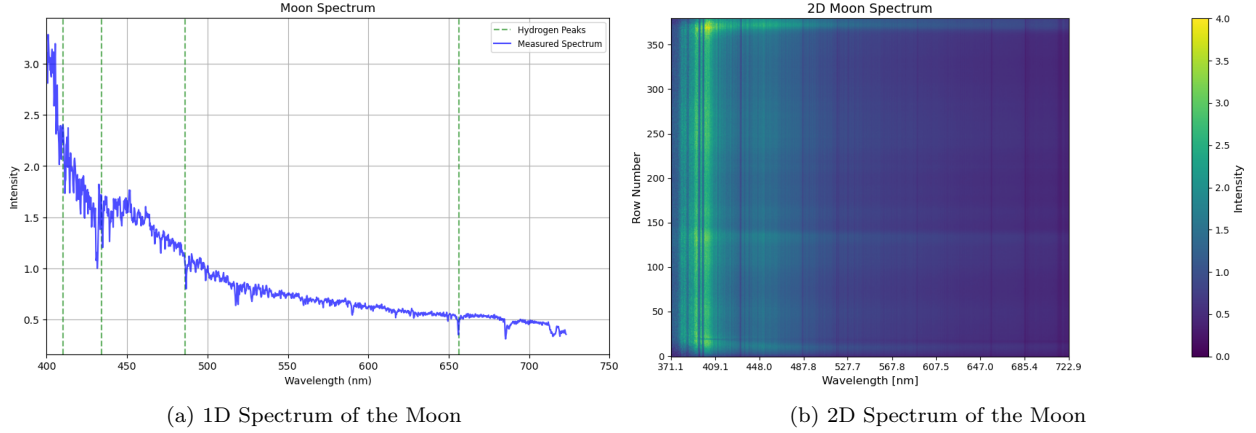


Fig. 10.—: Spectrums of the Moon

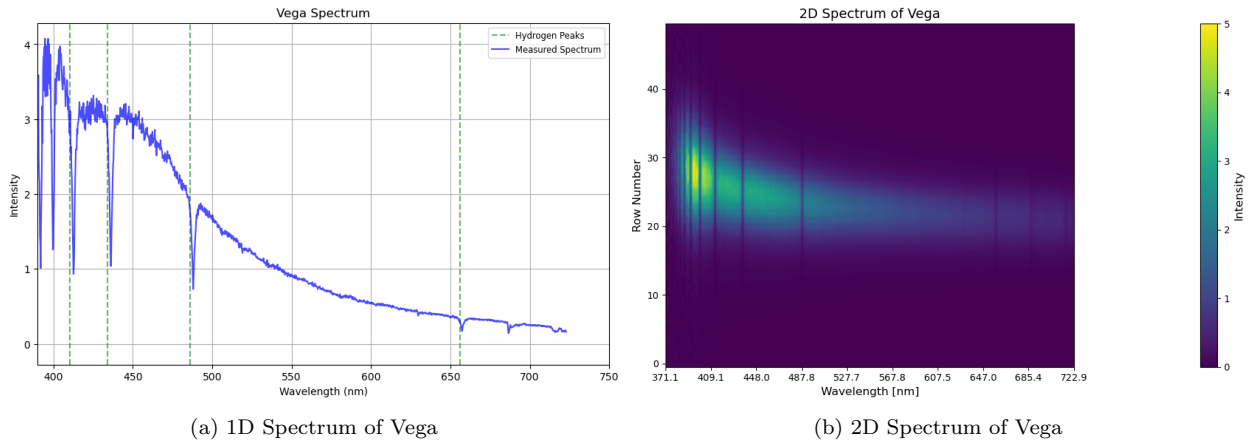


Fig. 11.—: Spectrums of Vega

The 1D and 2D spectra of the Moon (Figure 10) and Vega (Figure 11) clearly show the presence of all four prominent hydrogen absorption lines from the Balmer series.

For the Moon, these lines are reflected from sunlight, as the Moon does not emit its own light but rather reflects the spectrum of the Sun, which prominently includes the Balmer absorption lines due to hydrogen in the Sun’s photosphere [Vanderlinde \(2024a\)](#). In the case of Vega, a hot A-type star, these lines are intrinsic and result from the strong presence of hydrogen in its stellar atmosphere, where hydrogen atoms absorb

light at specific wavelengths corresponding to transitions to higher energy levels.

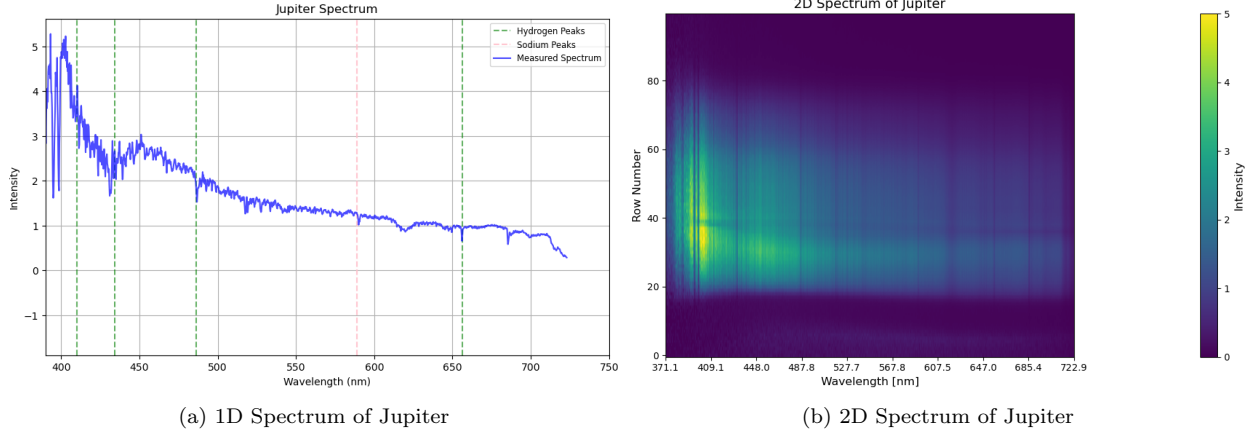


Fig. 12.—: Spectrums of Jupiter

The spectra of Jupiter (Figure 12) show the four hydrogen Balmer absorption lines like the Moon and Vega, reflecting sunlight scattered by its atmosphere. Additionally, the sodium emission line (Na I doublet) likely originates from city light pollution, as light pollution from city lights being scattered in Earth's atmosphere during the observation [LampTech \(2024\)](#).

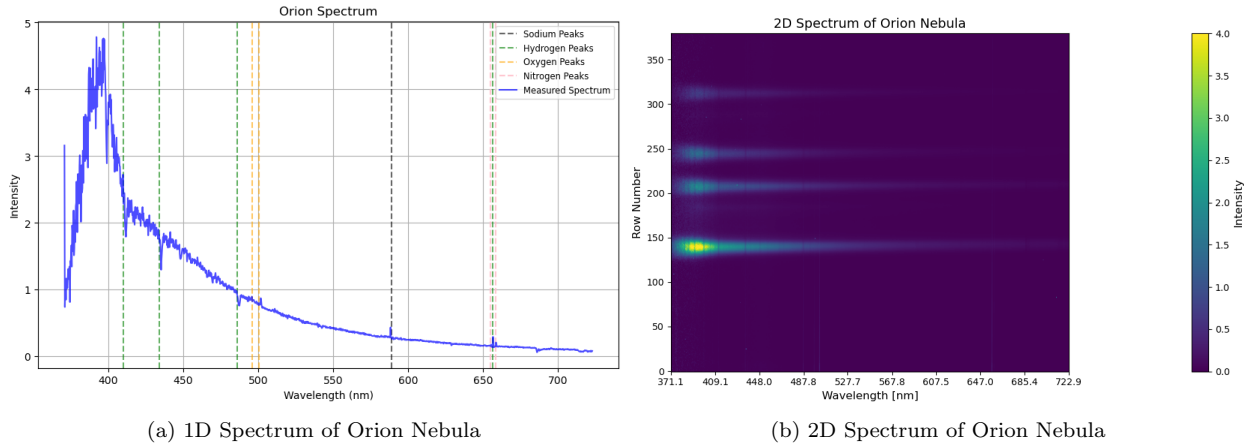


Fig. 13.—: Spectrums of Orion Nebula

The spectra of the Orion Nebula (Figure 13) reveal a fascinating mix of emission and absorption features, highlighting its unique physical conditions. Unlike the Moon and Vega, the Orion Nebula emits Hydrogen Alpha at 656.3 nm while absorbing the other three Balmer lines. This pattern occurs because the nebula contains both hot ionized gas emitting Hydrogen Alpha and cooler regions that absorb higher-energy Balmer transitions.

Additionally, the Orion Nebula shows strong emission lines of oxygen (notably O III at 495.9 nm and

500.7 nm), nitrogen (N II at 654.8 nm and 658.4 nm), and sodium (Na I at 589.0 nm and 589.6 nm) [Fowler \(1960\)](#). These emission lines arise from the ionization and recombination processes within the nebula, driven by high-energy ultraviolet radiation from nearby massive stars, making the Orion Nebula a rich and complex source of emission and absorption features.

5. Discussion & Conclusion

In this experiment, spectra were collected using both in-laboratory and telescopic spectrometers. Centroid estimation techniques were applied, utilizing the center of mass to identify peaks, while Gaussian fitting was employed to estimate the uncertainties. Polynomial fitting was also carried out, with the optimal polynomial degree determined via a chi-squared test. Uncertainty propagation was used to calibrate the spectrum, spatial, and detector characteristics, and the quality of the calibration was assessed by residual analysis and comparison with real spectra.

The polynomial fittings for hydrogen and neon-argon calibrations exhibited a linear and cubic relation, respectively, as validated by the chi-squared test. Residuals for both calibrations were small and random, within ± 1 nanometer, indicating a good fit between the calibration model and the true values. However, in the neon-argon calibration, the residuals between pixels 800 and 1200 exhibited a less random distribution and higher uncertainty, likely due to overlapping neon peaks during data collection. This overlap made it challenging to accurately separate the peaks, affecting the precision of the centroid estimation.

For uncertainty analysis, the general formula for uncertainty propagation was used to calculate the error in pixel units. The average wavelength uncertainty was determined to be ± 0.64 nanometers for the hydrogen calibration and ± 1.79 nanometers for the neon-argon calibration. The larger uncertainty in the neon-argon calibration can be attributed to the higher degree of polynomial fitting, which caused more distortion of the data points and greater deviation from the original values. Additionally, for the 2-D calibration, the uncertainty introduced by intensity normalization was considered, although intensity uncertainties had a minimal impact on this study, as the primary focus was on wavelength data.

This experiment provided valuable experience in spectrometer calibration and spectral analysis, reinforcing theoretical concepts learned in astrophysics. A notable source of uncertainty in our measurements was heat, particularly from the lamps. As the lamp heated up during measurements, thermal fluctuations could have induced instability in the spectra, thereby impacting both the centroid estimation and the uncertainty calculations. To mitigate this issue in future experiments, it would be beneficial to extend the data collection period, allowing the lamp ample time to cool between measurements. This adjustment would help minimize thermal effects, resulting in more accurate and reliable spectral data. In addition to the thermal effects, another potential source of error in this experiment could be the ambient light present in the room during data collection. Conducting the experiment in a non-dark environment introduces additional light contamination, which could interfere with the spectrometer’s measurements. Ambient light may introduce unwanted signals, particularly in the lower wavelength ranges, and affect the calibration of spectral features. This unwanted data could lead to inaccuracies in centroid estimation, as the spectrometer may incorrectly interpret the extra light as part of the spectrum. To mitigate this issue in future experiments, it would be advisable to conduct the measurements in a controlled, darkened environment to reduce the impact of external light sources and ensure that only the intended spectral signals are captured.

6. Bibliography

REFERENCES

- AST325/326. 2024, AST325/326 Lab Manual 3, accessed 4-November-2024
- AtomTrace. 2024, Nitrogen Element Database, Accessed 4-November-2024, <https://www.atomtrace.com/elements-database/element/10#NIST>
- Fowler, W. A. 1960, THE EMISSION-LINE SPECTRUM OF THE ORION NEBULA, Accessed 4-November-2024, <https://adsabs.harvard.edu/pdf/1960ApJ...132...18F>
- LampTech. 2024, SOX Lamp Technical Information, Accessed 4-November-2024, <http://lamptech.co.uk/SOX.htm>
- Nerval, S. 2024, Python Image Converting Code for AST325/326, provided as part of the course materials for AST325/326, University of Toronto
- NIST. 2024a, Hydrogen Table, Accessed 4-November-2024, <https://www.physics.nist.gov/PhysRefData/Handbook/Tables/hydrogentable2.htm>
- . 2024b, Oxygen Table, Accessed 4-November-2024, <https://www.physics.nist.gov/PhysRefData/Handbook/Tables/oxygentable2.htm>
- Vanderlinde, K. 2024a, lecture Notes 11, University of Toronto, accessed 4-November-2024
- . 2024b, lecture Notes 12, University of Toronto, accessed 4-November-2024
- . 2024c, lecture Notes 9, University of Toronto, accessed 4-November-2024

Appendix

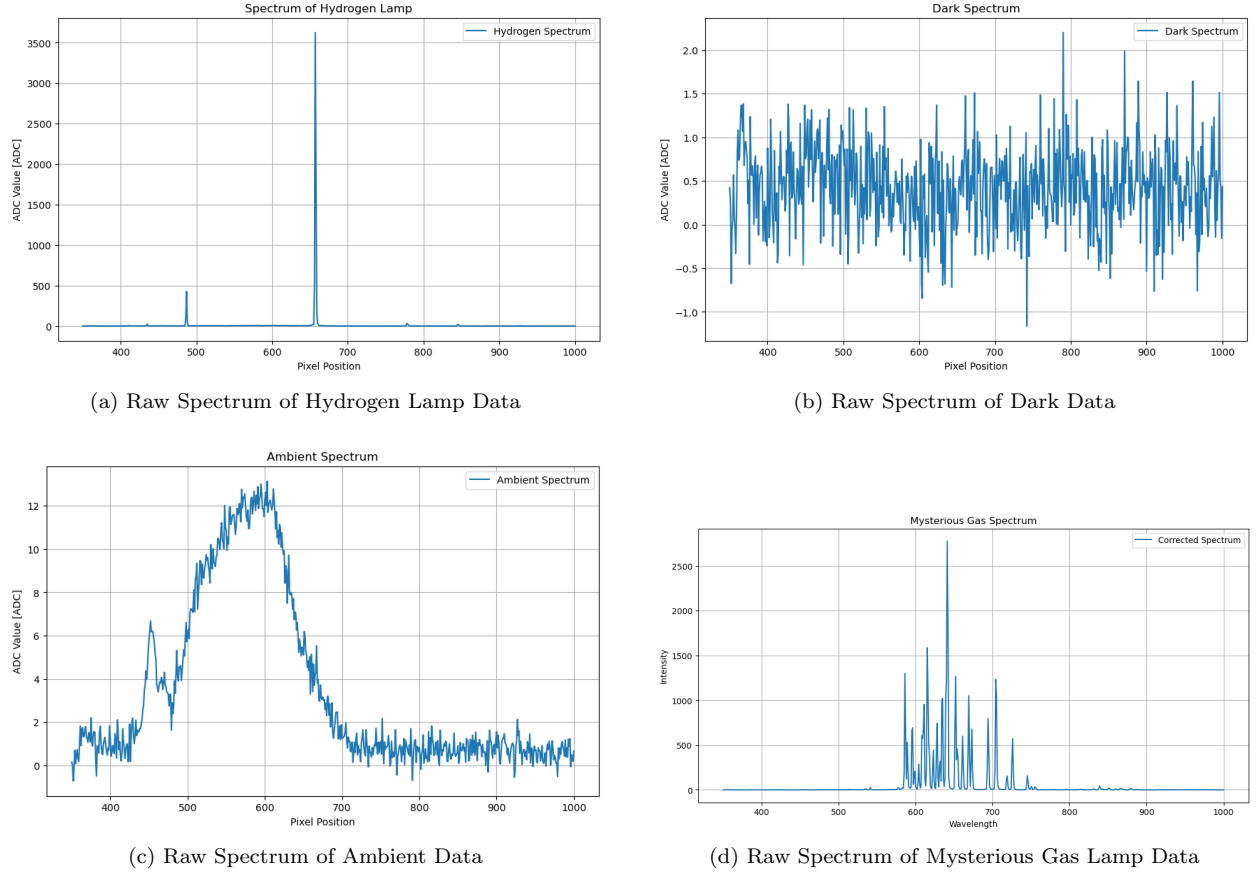


Fig. 14.—: Raw Spectrum of Different Data Source

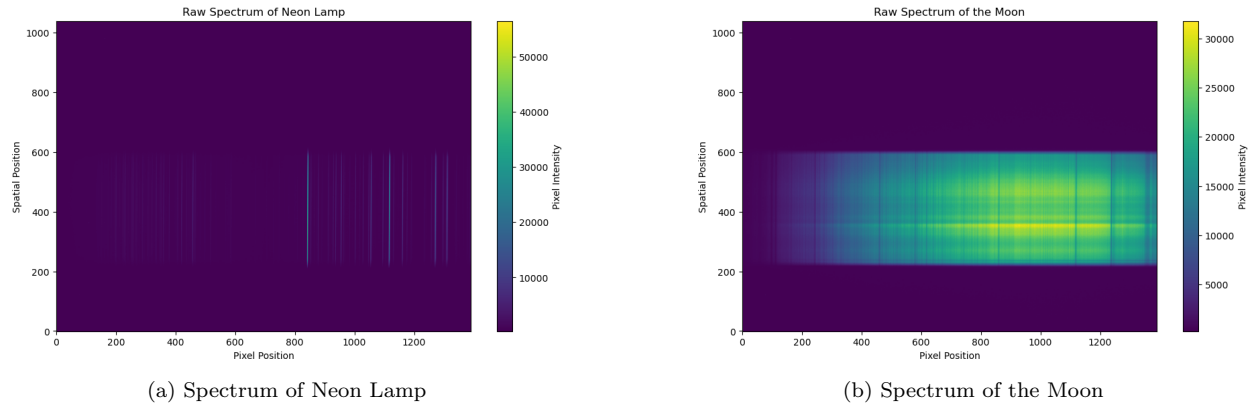


Fig. 15.—: Raw Spectrums from the LISA Telescope

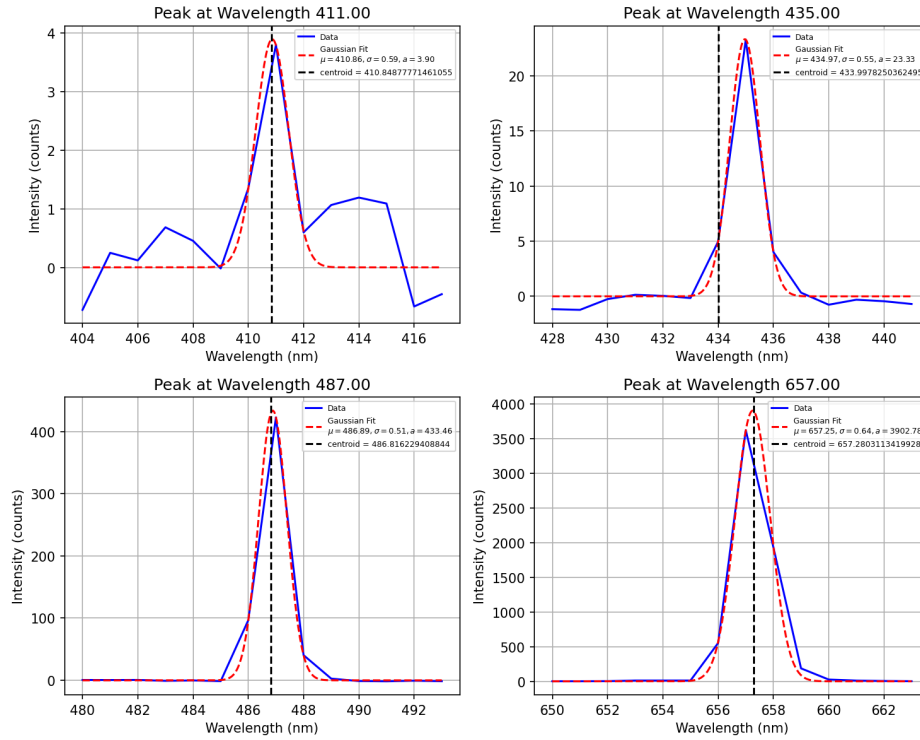


Fig. 16.—: Example Gaussian Approximation Plot

```
# Chi-Squared Function Implementation
def chi2_fit(x: list, y: list, maxdeg: int, sigma = None):
    # return a list of chi2 to seek the best deg of fitting
    #     chi2_list = []
    #     for i in range(1,maxdeg+1):
    #         m, coe = poly_fit(x, y, i)
    #         sigma = np.std(m)
    #         chi2 = np.sum((y - m) ** 2 / sigma ** 2)
    #         chi2_list.append(chi2)
    #     return chi2_list
    chi2_list = []
    reduced_chi2_list = []
    for degree in range(1, maxdeg + 1):
        # Polynomial fit
        coefs = np.polyfit(x, y, degree)
        model = np.polyval(coefs, x)

        # Default sigma: equal weighting
        if sigma is None:
            sigma = np.ones_like(y)
```

[illegible]

```
hspace=0.0)

# Add subplots for the fit and residuals
ax0 = fig.add_subplot(inner_grid[0]) # Top subplot (fit)
ax1 = fig.add_subplot(inner_grid[1], sharex=ax0) # Bottom subplot (residuals)
ax0.grid(ls='-.', linewidth=0.5, alpha=0.7) # Major gridlines
ax0.yaxis.set_major_locator(MultipleLocator(25)) # Major y-ticks every 50
ax0.xaxis.set_major_locator(MultipleLocator(50)) # Major x-ticks every 50
ax0.yaxis.set_minor_locator(MultipleLocator(10)) # Minor y-ticks every 10
ax0.xaxis.set_minor_locator(MultipleLocator(10)) # Minor x-ticks every 10
ax0.grid(which='minor', ls=':', linewidth=0.3, alpha=0.5) # Minor gridlines

# Plot data points and linear fit on the first axis
ax0.errorbar(centroids, known_peaks, xerr=x_uncertainties, fmt='o', color='blue',
             label='Data Points with Errors', capsize=3)
ax0.plot(centroids, fitted, color='red', label='Linear Fit')
ax0.plot(centroids, y_fit_quadratic, color='green', label='Quadratic Fit')

# Customize axis labels and grid
ax0.set_ylabel('Known Wavelengths [nm]')
plt.setp(ax0.get_xticklabels(), visible=False)
ax0.legend(loc='best', fontsize=10)
ax0.grid(ls='-.')

# Plot residuals on the second axis
ax1.scatter(centroids, residuals, color='purple', label='Linear Residuals', marker='.')
ax1.scatter(centroids, quad_residuals, color='orange', label='Quadratic Residuals', marker='.')
ax1.axhline(0, color='black', linestyle='--', linewidth=1)
ax1.set_xlabel('Measured Pixel Positions')
ax1.set_ylabel('Residuals [nm]')
ax1.grid(ls='-.')
ax1.legend()

# Show plot
plt.show()

# Print linear fit parameters
print(f"Linear Fit Parameters: Slope = {popt_linear[0]}, Intercept = {popt_linear[1]}")

# Uncertainty Sample Code
# find errors

peak_uncertainties = []
gaussian_peaks = []
def gaussian_fit(x, mu, sigma, a):
    return a * np.exp(-(x - mu)**2 / (2 * sigma**2))
```

```
wavelengths = np.linspace(350,1000,651)
intensities = intensity
# Peak regions for Gaussian fitting

fit_width = 7 # Number of points to either side of the peak for fitting
fig, axes = plt.subplots(2, 2, figsize=(10, 8), dpi = 150)
axes = axes.flatten() # Flatten axes array for easier indexing
for i in range(len(peaks)):
    # Extract data around the peak
    peak_index = peaks[i]
    start = int(max(0, peak_index - fit_width))
    end = int(min(len(wavelengths), peak_index + fit_width))

    x_data = np.array(wavelengths[start:end])
    y_data = np.array(intensities[start:end])

    # Initial parameters guesses
    print(peak_index)
    mu_guess = wavelengths[peak_index]
    sigma_guess = 2
    a_guess = intensities[peak_index]

    p0 = [mu_guess, sigma_guess, a_guess]

    try: # Fit Gaussian to the data
        popt, pcov = curve_fit(gaussian_fit, x_data, y_data, p0=p0)
        mu_fit, sigma_fit, a_fit = popt
        peak_uncertainties.append(sigma_fit)
        gaussian_peaks.append(mu_fit)
        x_fine = np.linspace(x_data[0], x_data[-1], 1000)
        y_fit = gaussian_fit(x_fine, mu_fit, sigma_fit, a_fit)
        ax = axes[i]
        ax.plot(x_data, y_data, 'b-', label='Data')
        ax.plot(x_fine, y_fit, 'r--', label=f'Gaussian Fit\n $\mu$ ={mu_fit:.2f},  $\sigma$ ={sigma_fit:.2f}, a={a_fit:.2f}')
        ax.set_title(f'Peak at Wavelength {mu_guess:.2f}')
        ax.set_xlabel('Wavelength (nm)')
        ax.set_ylabel('Intensity (counts)')
        ax.legend(loc = 'upper right')
        ax.grid(True)
        ax.axvline(centroids[i], color = 'black', linestyle = '--', label = f'centroid = {centroids[i]:.2f}')
        ax.legend(loc = 'best', fontsize = 6)

peak_uncertainties = np.array(peak_uncertainties)
gaussian_peaks = np.array(gaussian_peaks)
```



```
# Centroiding Example Code
```

```
fig = plt.figure()

gs = gridspec.GridSpec(2, 1, height_ratios=[3, 1], hspace = 0.0)
ax0 = plt.subplot(gs[0])
ax1 = plt.subplot(gs[1], sharex = ax0)

ax0.plot(wavelength, intensity, color = 'purple', label = 'Data')
ax0.plot(wavelength[peaks], intensity[peaks], color = 'coral', ls = '', marker = 'x',
        ms = 5, label = 'Peaks')

[ax0.axvline(c, color = 'teal', alpha = 0.5, label = 'Centroids') if c == centroids[0]
 else ax0.axvline(c, color = 'teal', alpha = 0.5) for c in centroids]

ax1.plot(wavelength[peaks], wavelength[peaks] - centroids, ls = '', marker = '.', color = 'coral')

ax1.set_xlabel('Wavelength [nm]')
ax0.set_ylabel('Intensity [ADC]')
ax1.set_ylabel('Residuals')

plt.setp(ax0.get_xticklabels(), visible = False)

ax0.legend(loc = 'best', fontsize = 10)
ax0.grid(ls = '-.')
ax1.grid(ls = '-.')
plt.xlim((350, 850))
plt.show()

# Spatial Calibration
# 2. Spatial Calibration
def find_brightest_centroid(row):
    """Find the centroid of the brightest spot in the given row."""
    max_pixel = np.argmax(row)
    return max_pixel

centroids = []
for i, row in enumerate(neon_data):
    centroids.append((i, find_brightest_centroid(row)))

centroids = np.array(centroids)
row_numbers = centroids[:, 0]
centroid_pixels = centroids[:, 1]

# Fit a linear relation between row numbers and centroids
def linear_model(x, m, b):
```

```
    return m * x + b

popt, _ = curve_fit(linear_model, row_numbers, centroid_pixels)
spatial_fit = linear_model(row_numbers, *popt)

plt.figure(figsize=(12, 8))
plt.scatter(row_numbers, centroid_pixels, s=10, c='blue', label="Centroid Data", alpha=0.7)
plt.plot(row_numbers, spatial_fit, color="red", linewidth=2, label="Linear Fit")
plt.title("Spatial Calibration for Neon Lamp", fontsize=16)
plt.xlabel("Row Number", fontsize=14)
plt.ylim(840, 850)
plt.ylabel("Centroid Pixel Position", fontsize=14)
plt.legend(fontsize=12)
plt.grid(color='gray', linestyle='--', linewidth=0.5, alpha=0.7)
plt.tight_layout()

plt.show()

print(f"Spatial Calibration Fit Parameters: Slope = {popt[0]}, Intercept = {popt[1]}")

# Spectral Calibration
cubic_func = [7.689e-09, 2.378e-05, 0.2753, 366.2]

calibrated_wavelength = []
checker1 = False
for i in calibrated_data:
    wl = (cubic_func[0]*i*i*i - cubic_func[1]*i*i + cubic_func[2] * i + 366.2)
    calibrated_wavelength.append(wl)
calibrated_wavelength = np.array(calibrated_wavelength)

# Detector Calibration
moon_file = "2024-11-11_Moon_10s.fit" # Replace with your file path
moon_data = fits.getdata(moon_file)[220:600]
dark_files = ["Darks_night-0001_10s.fit",
              "Darks_night-0002_10s.fit",
              "Darks_night-0003_10s.fit",
              "Darks_night-0004_10s.fit",
              "Darks_night-0005_10s.fit",
              "Darks_night-0006_10s.fit"
              ]
dark_images = [fits.getdata(dark) for dark in dark_files]
averaged_dark = np.mean(dark_images, axis=0)[220:600]
R_D = moon_data - averaged_dark
F_D = averaged_tungsten - averaged_dark
P2 = np.divide(R_D, F_D, out=np.zeros_like(R_D), where=F_D != 0)
```