Hydrogen Emission Spectrum
Lab #5
3/23/2022

Yassin Kortam

PHYS 4C

Prof. Mike Masuda

Table of Contents

Introduction & Purpose

The purpose of this lab is to experimentally measure the emission spectrum of hydrogen. The Rydenberg constant will then be derived experimentally from this data and compared to the expected value. A simple error analysis will be conducted.

Equipment

- A spectrometer
- A hydrogen spectrum tube
- Logger Pro

Procedure

1. Connect the spectrometer via USB to a PC with Logger Pro installed.
2. Turn on the hydrogen spectrum tube.
3. Place the spectrometer optical fiber over the tube and shift it until distinct intensity peaks show on Logger Pro.
4. Save the data as a CSV file.

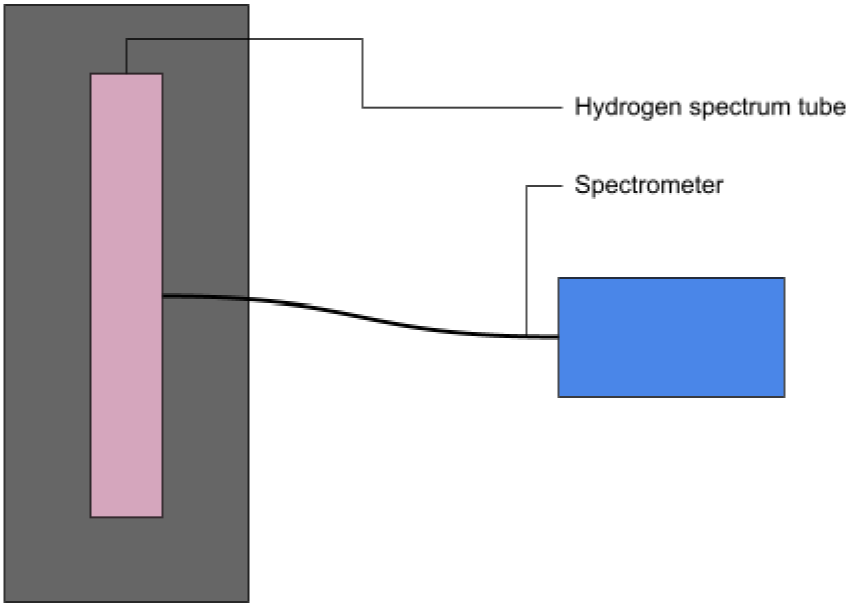

Figure 1: General lab setup

Data & Analysis

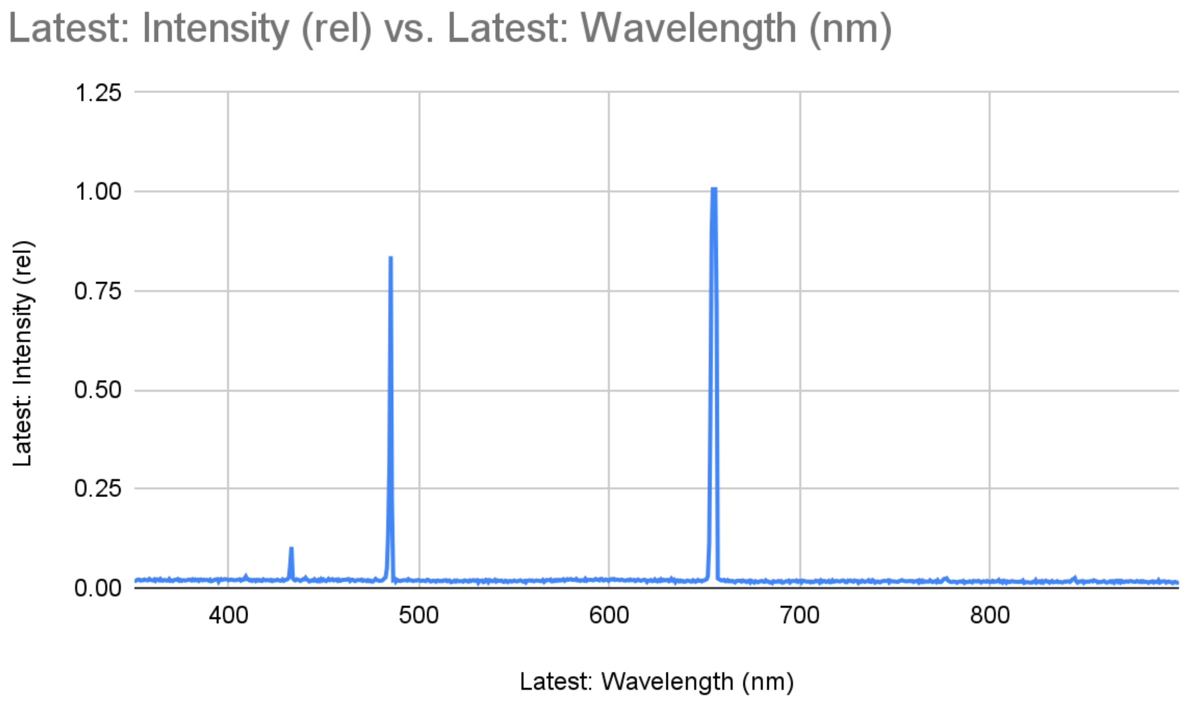

Figure 2: Hydrogen emission spectrum raw data

Peak Wavelength Finder

```python
import pandas as pd

expected_peaks = 4
sample_maxima = 17

raw_data = pd.read_csv('hydrogen.csv')
i = [i for i in raw_data['Latest: Intensity (rel)']]
l = [i for i in raw_data['Latest: Wavelength (nm)']]
wavelengths_of_peaks = []

counter = 0

for k in range(sample_maxima):
   max_intensity = max(i)
   max_intensity_index = i.index(max_intensity)
   wavelength_of_peak = l[max_intensity_index]

   wavelengths_of_peaks.append(wavelength_of_peak)
   i.remove(max_intensity)

wavelengths_of_peaks.sort()

def remove_repeats(dataset, variance):
   for k in range(len(dataset)-1):
       if k>0 and k < len(dataset):
           thisvariance = 100*abs((dataset[k]-dataset[k-1])/dataset[k])
           if thisvariance < variance:
               dataset.remove(dataset[k])
   return dataset

variance = 0
while len(wavelengths_of_peaks) > expected_peaks:
   remove_repeats(wavelengths_of_peaks, variance)
   variance += 1
print(wavelengths_of_peaks)
```

Nf and Ni Finder

```
import random

max_error = 2
expected_ratio = 1.41*10**-1
error = 100

while(error > max_error):
   nf = random.randint(1,8)
   ni = random.randint(1,8)
   ratio = (nf**-2 - ni**-2)
   error = 100*abs((expected_ratio-ratio)/expected_ratio)

print(nf,ni,error)
```

Color	Wavelength	Frequency	Energy
purple	4.09E-07	7.34E+14	4.86E-19
purple	4.31E-07	6.96E+14	4.61E-19
blue	4.82E-07	6.23E+14	4.13E-19
red	6.47E-07	4.63E+14	3.07E-19

Table 1: Note wavelengths are found programmatically

Color	Energy	Ratio	nf	ni	%E
purple	4.86E-19	2.23E-01	2	6	0.35%
purple	4.61E-19	2.12E-01	2	5	0.94%
blue	4.13E-19	1.89E-01	2	4	0.79%
red	3.07E-19	1.41E-01	2	3	1.50%

Table 2: Note nf, ni, and error are found programmatically

Color	Wavelength	Wavenumber	Ratio	Rh	%E
purple	4.09E-07	2.45E+06	2.22E-01	1.10E+07	0.32%
purple	4.31E-07	2.32E+06	2.10E-01	1.11E+07	0.74%
blue	4.82E-07	2.08E+06	1.88E-01	1.11E+07	0.93%
red	6.47E-07	1.54E+06	1.39E-01	1.11E+07	1.40%

Table 3: Rydenberg constant approximation with error

After extracting the peak wavelengths using a Python program, the photon energies E can be calculated:

$$1) \quad f = \frac{c}{\lambda}$$

$$2) \quad E = hf = 2.18 \cdot 10^{-18}\left(\frac{1}{n_f^2} - \frac{1}{n_i^2}\right)$$

The right hand side of equation 2 is in terms of the orbitals of the electrons. To solve for these orbitals, the difference of the inverse square of the orbitals is written in terms of the energy divided by the coefficient of the difference. A python program guesses random integer solutions for the equation that minimize error.

Finally, given the programmatically derived orbital values, the rydberg constant can be derived:

$$3) \quad k = R_H\left(\frac{1}{n_f^2} - \frac{1}{n_i^2}\right)$$

Conclusion

Error was satisfactory, increasing with wavelength and being under 1% for all measurements except for red light. Interestingly, error for the rydberg constant would have been under 0.5% for all entries if the equivalent expression for the guessed orbitals expression was used, and the error in the rydberg constant calculations corresponds almost perfectly to the programmatically derived error while finding the orbitals.

Another source of systematic error is the wavelength finder calculator since it merely approximates where the peaks are in a somewhat crude fashion. A larger time investment in developing the program would have reduced error somewhat.