Exploring Blackbody Radiation of a Tungsten Light Bulb

By: Carl Ancheta & Joaquin Arcilla

1. Introduction

A blackbody is an object where the electromagnetic radiation emanating from it is only from the thermal motion of its charged particles [1]. Examples of blackbodies are the Sun and heated metal that glows red [2]. Blackbody Radiation caused problems in classical physics as the model did not predict the drop in energy at wavelengths in the Ultraviolet spectrum (dubbed the 'ultraviolet catastrophe'). Quantum theory was able to model the true relationship between the emitted wavelength and energy.

The energy of a blackbody is dependent on the wavelength of radiation emitted as well as the temperature. The peak energy of radiation shifts to lower wavelengths as temperature increases. This corresponds to heating an object up, changing its colour from none to red and eventually white. The relationship between the peak wavelength and temperature is called *Wien's Displacement Law* and is defined as:

$$\lambda_{max} \times T = 0.002898\, mK \qquad (1)$$

The relationship between energy increasing as with temperature is called *Stefan-Boltzmann Law* and is defined as:

$$\frac{P}{A} = I = \varepsilon\sigma T^4 \qquad (2)$$

The experiment being performed uses a tungsten light bulb as the blackbody, and a prism spectrophotometer to measure the light intensity of different wavelengths which are separated by the prism.

2. Procedure

2.1 Equipment

- Prism Spectrophotometer
- Computer with LabVIEW Acquisition Software

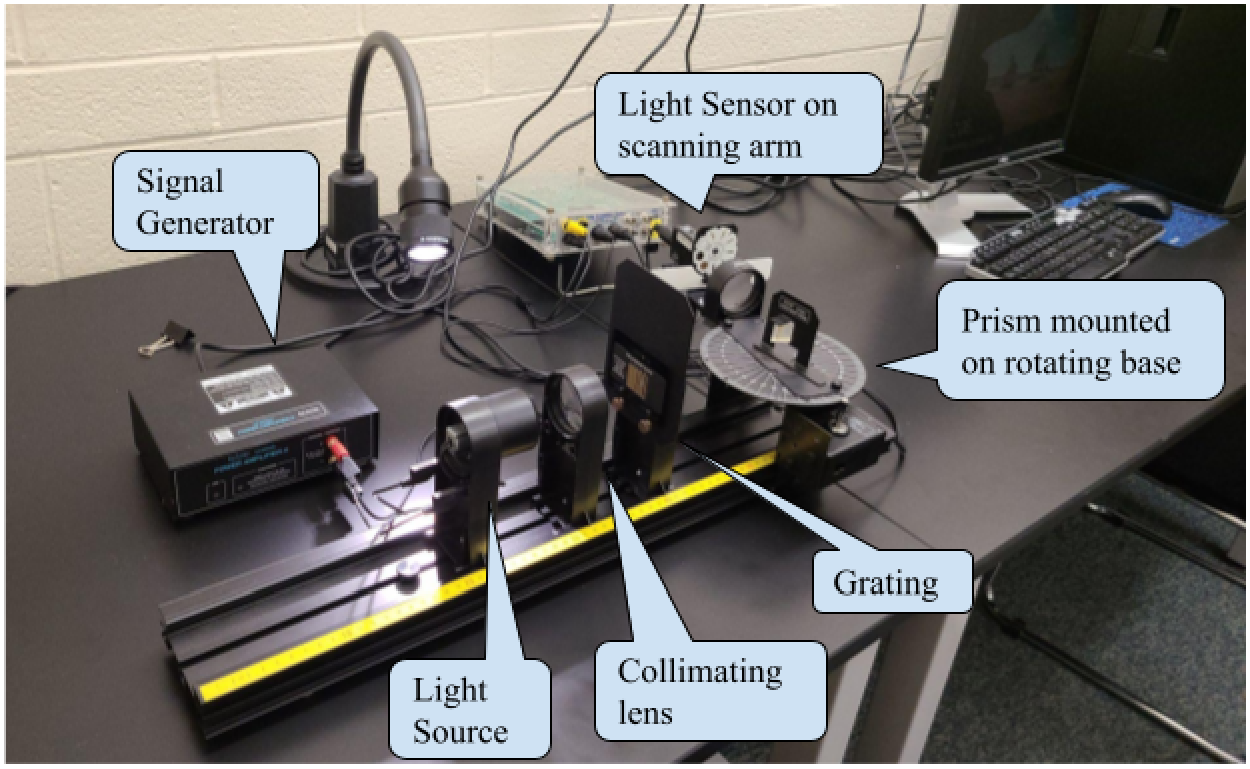

Figure [01]. Prism Spectrophotometer Apparatus with parts labelled

2.2 Experimental Method

2.2.1 Setup

The Prism Spectrophotometer will be set up in the lab. Verify the apex of the prism is aligned with the light source and base of the prism is aligned with the 0° mark on the spectrophotometer. Adjust the position of the collimating lens to be ~10 cm from the collimating slits. Set the collimating slits to Slit #4. Set the light sensor mask on Slit #3.

Open Blackbody Radiation LabVIEW software on the computer and turn on the signal generator. Press the 'Clear Data' button in LabVIEW to reset the data collection. Rotate the scanning arm counterclockwise until it stops, this will be the starting point for each test done. Turn off all lights when taking data to avoid extra light being measured in the data.

2.2.2 Observe Light from the Blackbody Light Source

Set Broad Spectrum Light Sensor Gain to 10x on the light sensor, and press the tare button. Select 80° acquisition in LabVIEW. Press the 'Start Collecting Data' button and set voltage to 6V. The light source will turn on. Observe the spectrum of colours on the Light sensor screen by turning the scanning arm until the spectrum is seen. Press the 'Stop Collecting Data' button when done observing to turn off the light source to avoid overheating.

2.2.3 Determine Initial Angle (θ_{init})

Set the scanning arm at the starting point from 2.2.1. Then, slowly rotate the scanning arm through the spectrum until data collection stops on LabVIEW and the lightbulb turns off. Observe the 2 peaks in intensity. The second smaller peak is at the initial angle, record this angle. Repeat 3-5 times to determine the initial angle for the experiment

2.2.4 Wien's Displacement Law

Switch to a 30° acquisition in LabVIEW. Return scanning arm to the starting point. Collect data like in previous steps using voltages from 10V to 4V, and collect the voltage and current values displayed in LabVIEW for each test. Note: At lower voltages gain may need to be switched to 100x, press the tare button on the light sensor several times when this is changed

2.2.5 Stefan-Boltzmann Law

Stay on 30° acquisition. Collect data like in 2.2.4 using voltages from 10V to 4V. In LabVIEW use the vertical cursors to set limits around the peak intensity area. Press the 'Integrate' button to calculate the area under the curve between the limits, record this number. Then set the cursor limits to the 'tail' area of the peak and press the 'Integrate' button again. Record this area as well. Repeat for at least 7 voltages total.

3. Data

Sample Initial Angle measurement, 80° measurement under 6V:

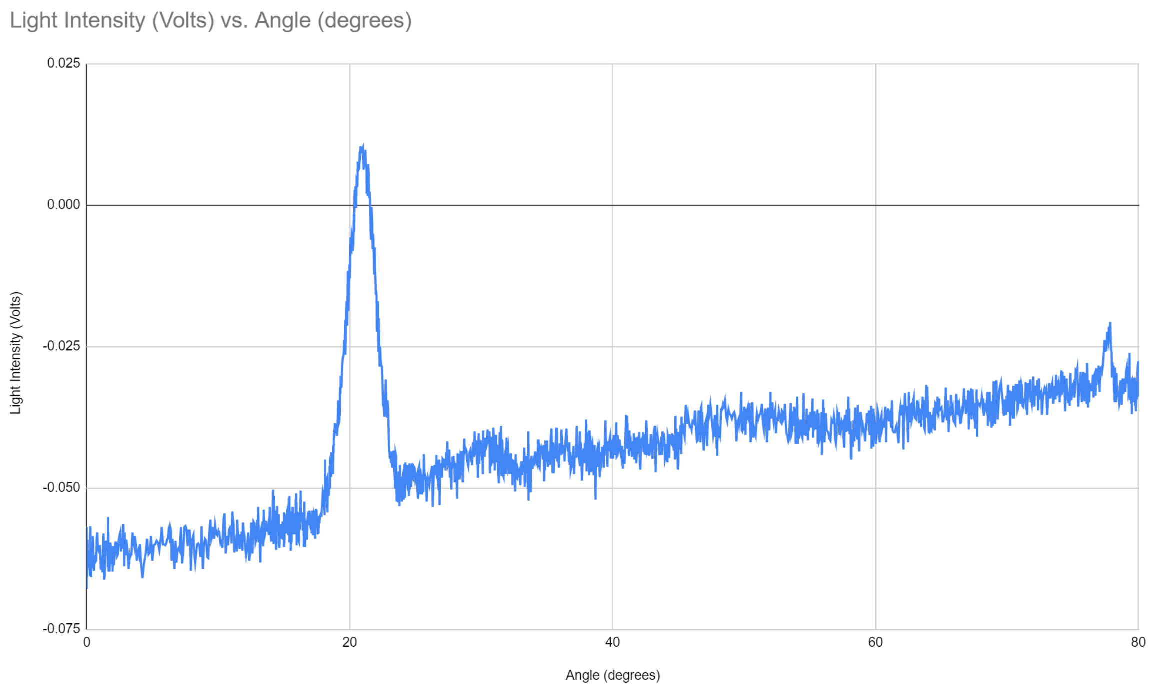

Figure [02]: Light Intensity vs Angle with 6V bulb for Initial Angle Measurement

Sample Wien's Law measurement for 10V:

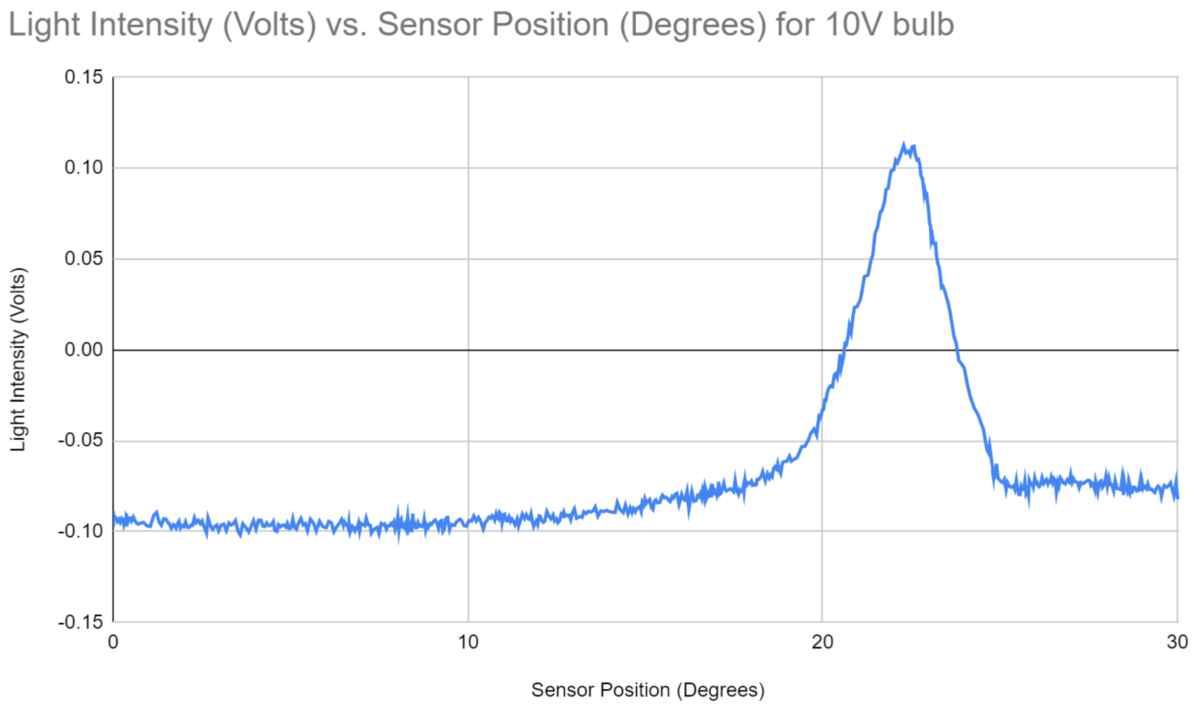

Figure [03]: Light Intensity vs Angle with 10V bulb for Wien Displacement Law

Voltage and Current measurement used to calculate Temperature:

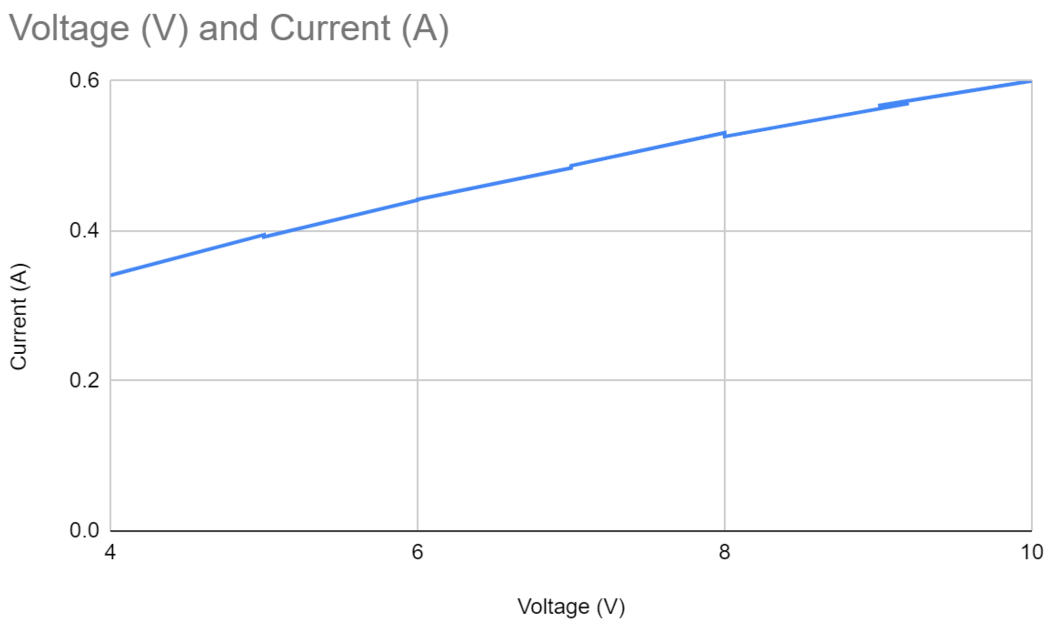

Figure [04]: Voltage vs Current

4. Analysis

4.1 Important Formulae

To calculate the mean average of data from a set of experiments, the following equation was used through Python Code:

$$\bar{x} = \frac{\Sigma x_i/(\Delta x_i)^2}{\Sigma 1/(\Delta x_i)^2} \qquad (\alpha)$$

To calculate the standard deviation of the mean (the uncertainty of the average) the following equation was used through Python Code:

$$\sigma = \sqrt{\frac{1}{N-1}\Sigma(x_i - \Delta x_i)^2} \qquad (\beta)$$

To calculate the error propagation of addition or subtraction operations (z = x +/- y) with two variables, the following equation was used through Python Code:

$$\Delta z = \sqrt{(\Delta x)^2 + (\Delta y)^2} \qquad (\gamma)$$

To calculate the error propagation of a multiplication of division operation (z = x × y or z $\frac{x}{y}$), the following equation was used through Python Code:

$$\Delta z = z\sqrt{(\frac{\Delta x}{x})^2 + \frac{\Delta y}{y})^2} \qquad (\delta)$$

4.2 Wien's Displacement Law

Wien's Displacement Law is an equation which shows the behaviour of radiation as the temperature of a black body changes. The law states that the peak wavelength multiplied by the temperature of the blackbody will equal a constant:

$$\lambda_{max} \times T = 0.002898\, mK \qquad (1)$$

In order to prove or disprove this law, for multiple different voltages going through the bulb, we measured the intensity of light from the bulb at different angles. Finding the angle at which the light intensity was its highest, we obtained the wavelength at the highest intensity by first changing the measured angle to the actual angle through the initial angle, and then using Snell's law and Cauchy's law to convert the angle into a wavelength:

$$\lambda = \sqrt{\frac{A}{(\sqrt{(\frac{2}{\sqrt{3}}sin\theta + \frac{1}{2})^2 + \frac{3}{4}} - B}} \qquad (3)$$

Where A = 13,900 nm and B = 1.689

To obtain the temperature values, we rearranged the Resistance to Temperature equation and substituted Ohm's Law:

$$T = T0 + \frac{\frac{V/I}{R0} - 1}{a0} \tag{4}$$

For this $T_0 = 20℃$, $R_0 = 1.1\ \Omega$, $\alpha_0 = 0.0045\frac{1}{k}$

α_0 is the thermal coefficient of the tungsten light bulb at 20℃. V and I are different based on each experiment.
With both the wavelengths of the light and the temperature of the bulb, we calculated the experimental values of Wien's law (1).

4.2.1 Initial Angle

In order to calculate the angle of the light, we need a reference angle. We use the point where the light passes through the prism unopposed, which can be found as the second highest peak of the intensity vs angle graph (the first highest peak is the λ in Wien's law). Table 1 contains the angles at 6 trials, at this second highest peak.

Trial #	Measured Angle (Degrees)
Trial 1	77.84948 +/- 0.000005
Trial 2	78.8843 +/- 0.000005
Trial 3	78.97743 +/- 0.000005
Trial 4	78.20132 +/- 0.000005
Trial 5	78.81186 +/- 0.000005
Trial 6	78.85325 +/- 0.000005

Table 1: Measured Angles for Initial Angle Calculation

Using the mean equation (α) and the standard deviation equation (β), the average initial angle was found to be 78.6 +/- 0.5 degrees.

4.2.2 Angle and Wavelength

We conducted 7 experiments with different voltages from 10V to 4V, measuring the intensity vs angles. Each experiment had a peak intensity and a corresponding peak angle. Using these peak angles, we calculated the actual peak angle based on the initial angle from 3.1.1 by subtracting the initial angle from the measured angle: $\theta_{init} - \theta_{measured}$. Then using equation (3) we calculated the peak wavelength. Table 2 contains the information from this process for each experiment, indicated by a voltage value.

Trial # Based on Voltage (V)	Peak Measured Angle (Degrees)	Peak Angle (Measured Angle - Initial Angle	Peak Wavelength (nm)
10	22.196885 +/- 0.000005	56.39938833 +/- 0.000005	1176.147 +/- 0.05
9	22.46076 +/- 0.000005	56.13551333 +/- 0.000005	1360.627 +/- 0.05
8	22.6522 +/- 0.000005	55.94407333 +/- 0.000005	1567.7586 +/- 0.05
7	22.367625 +/- 0.000005	56.22864833 +/- 0.000005	1285.872 +/- 0.05
6	22.352105 +/- 0.000005	56.24416833 +/- 0.000005	1274.588 +/- 0.05
5	22.072610 +/- 0.000005	56.52357333 +/- 0.000005	1112.145 +/- 0.05
4	22.6729 +/- 0.000005	55.92337333 +/- 0.000005	1596.36371258 +/- 0.05

Table 2: Peak Wavelength from Voltages 10V to 4V

4.2.3 Voltage and Current

Using the voltage and current values and equation (4) from each experiment, we calculated the temperature of the light bulb for each experiment. Table 3 contains the voltage and current data, as well as the calculated temperatures. Error propagation was calculated using equation (δ) since equation (4) divides the two variables together.

Voltages (V)	Currents (A)	Temperature (Celsius)
10 +/- 0.5	0.6 +/- 0.10	3164.78114478 +/- 550.68808642
9 +/- 0.5	0.567 +/- 0.10	3004.44765111 +/- 555.55221186
8 +/- 0.5	0.526 +/- 0.10	2870.32837885 +/- 574.4215509

7 +/- 0.5	0.487 +/- 0.10	2701.55891565 +/- 587.33945735
6 +/- 0.5	0.442 +/- 0.10	2540.13346131 +/- 612.43533072
5 +/- 0.5	0.392 +/- 0.10	2374.56606885 +/- 650.63565798
4 +/- 0.5	0.341 +/- 0.10	2167.5162179 +/- 690.9706247

Table 3: Voltage, Current, and their corresponding

4.2.4 Wien's Value Calculations

Using the wavelength and temperature values calculated in the earlier sections, we calculated the experimental Wien values. Error propagation was calculated using equation (δ) since equation (1) multiplies the two variables together. Table 4 contains the Wien's values from the 7 trials.

Trial # Based on Voltage (V)	Wien's Value (mK)
10	0.00372225 +/- 0.00064769
9	0.0041 +/- 0.0008
8	0.0045 +/- 0.0009
7	0.0035 +/- 0.0008
6	0.0032 +/- 0.0008
5	0.0026 +/- 0.0007
4	0.0035 +/- 0.001

Table 4: Wien's value of each experiment

Using equation (α) and (β) we calculated the mean value for the Wien's number and the standard deviation, which was: 0.0036 +/- 0.0006 mK. Comparing it to the theoretical value of 0.002898 mK, the experimental data agreed with the theory since the value falls within uncertainty. Therefore our experiment agrees with Wien's Displacement Theory.

4.3 Stefan-Boltzmann Law

The Stefan-Boltzmann Law relates the power, area, intensity and temperature of a black body. The Stefan-Boltzmann Law is:

$$\frac{P}{A} = I = \varepsilon\sigma T^4 \qquad (2)$$

Where $\varepsilon = 1$ for a blackbody, $\sigma = 5.670374419\times10^{-8}\,\frac{W}{m^2K^4}$

Based on equation (2) if one were to graph A vs T^4 should be a linear function.

We measured voltage, current and intensity across 7 experiments from 10V to 4V. In order to get the value of the Intensity I, we used LabVIEW to measure the area under the intensity vs angle graph. Using P = IV, we rearranged the Stefen-Boltzmann equation to be:

$$A = \frac{IV}{I} \qquad (5)$$

Table 5 has the calculated Area values. Error propagation is done by equation (δ)

Trial # based on Voltage	Area (m^2)
10	6 +/- 1
9	6 +/- 1
8	6+/- 1
7	6 +/- 1
6	3.7+/- 0.9
5	3.0+/- 0.8
4	2.0 +/- 0.6

Table 5: Area calculated for each experiment

Plotting Area vs Temperature^4 you get:

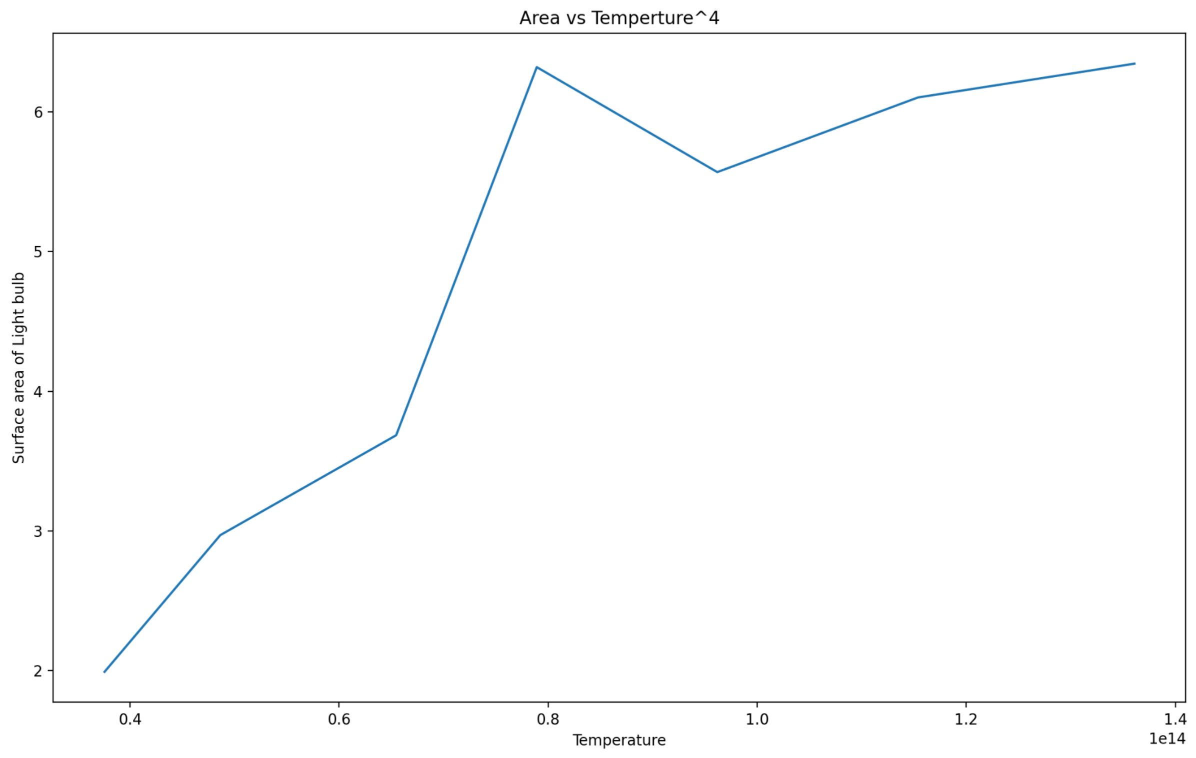

Figure [05]: Area vs Temperature for Stefan-Boltzmann's Law

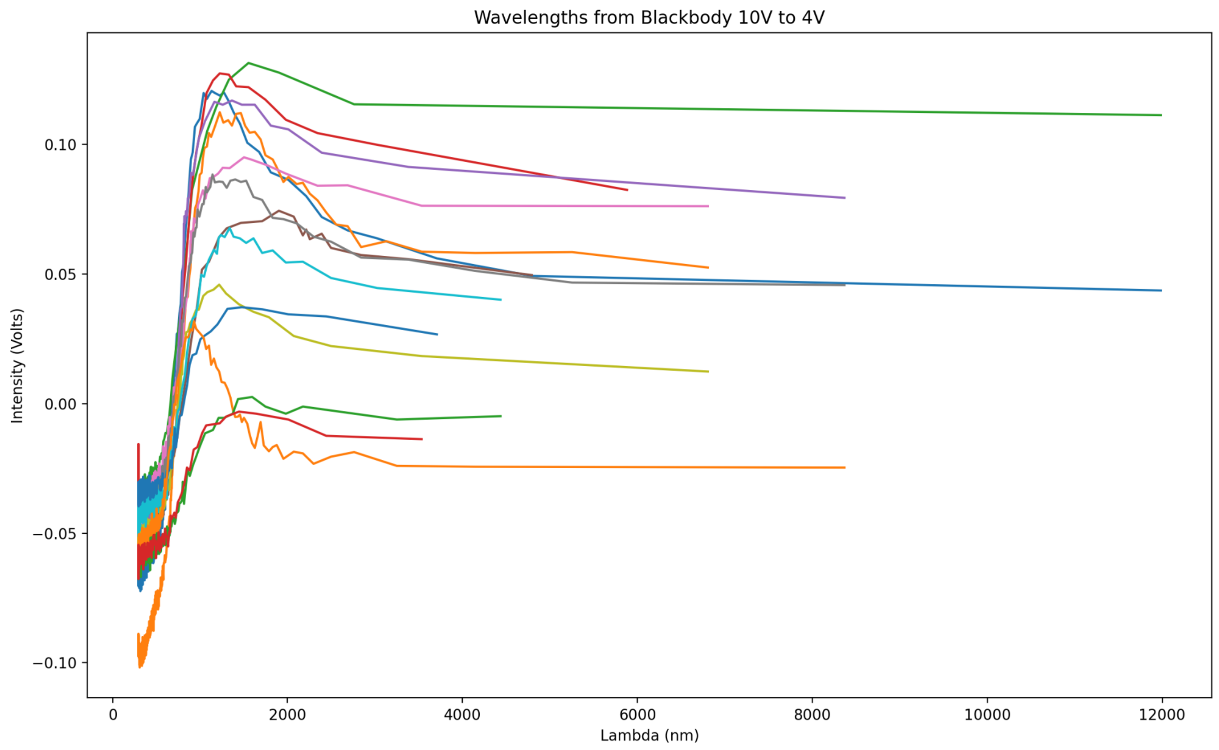

Figure [06]: Intensity vs Wavelength for 7 experiments

4.4 Systematic Uncertainties

No lab station is perfect. This lab had systematic uncertainties due mostly to the fact that we could not prevent energy from being released from other sources. The human body produced radiation that would affect the data. The computer worked on also produced thermal and light radiation that would affect the data. None of this could be avoided during the experiment.

4.5 Discussion

During the experiment, the whole visual spectrum was seen from red to violet on the Light Sensor as the scanning arm was turned. Figure 4 shows the wavelengths vs intensity, showing the energy distribution across the wavelengths and colours.

The calculated values for Wien's Displacement Constant were not consistent with the theoretical values. The values were of the same magnitude. Factors such as other light sources and radiation (e.g. the computer used, any light from the door cracks), would have affected the data collected by the light sensor. It should also be acknowledged that perfect blackbodies do not exist, so the deviation from theoretical values makes sense.

Based on the calculated values for Area in the Stefen-Boltzmann experiment, we did not achieve the same behaviour as the theoretical values. This could be because of a multitude of factors, including the fact that no perfect black body exists, the fact our own bodies give off heat and radiation, and the fact that our room was not completely dark and therefore had light radiation around us.

Visually, through the experiment, the colour of the light from the lightbulb did not seem to change. However, the peak wavelength changes with temperature as Wien's Law states, meaning that the combination of wavelengths that the lightbulb was emitting shifted toward the visible spectrum. Furthermore, the colour of the filament in the light bulb would also change based on temperature, being red at lower temperatures and white at higher temperatures (since more of the intensity is in the visible spectrum, which combines to create white light). These results can be applied to developing new lightbulbs as well; the voltage and current of a newer lightbulb can be chosen based on the temperature they generate, and optimizing for a higher temperature means more visible light is emitted making the light bulb more efficient.

5. Conclusion

In this lab, the properties of blackbodies were able to be recreated using a tungsten light bulb. Wien's Displacement Law was followed as the peak wavelength increased with the temperature of the light bulb. However, the Stefen-Boltzmann's Law was not followed by the experimental data. The values calculated were not consistent with theoretical values due to various factors, as well as the fact that a perfect blackbody does not exist.

6. References

[1] R. Harris, *Modern Physics.* Addison Wesley, 2008.
[2] R. M. Serbanescu and N. Krasnopolskaia, "Blackbody Radiation," September, 2017.

Appendix 1: Wien’s Displacement Law code

```
#Imports
import numpy as np
import matplotlib.pyplot as plt
from pylab import loadtxt

#Uncertainties
def uncertainty_sum(dx, dy):
    return np.sqrt(dx**2 + dy**2)

def uncertainty_prod(x, dx, y, dy, z):
    return np.sqrt((dx/x)**2 + (dy/y)**2) * z


def import_data(filename):
    data=loadtxt(filename, usecols=(0,1), skiprows=2, unpack=True)

    return data[0], data[1]

def mean_same_uncern(list):
    return sum(list)/len(list)

def standard_deviation(value, list):
    return np.sqrt((1/(len(list)-1))*sum((value - list)**2))

def mean_diff_uncern(list, uncern_list):
    return sum(list/uncern_list**2) / sum(1/uncern_list**2)

#Initial Angle
def index_of_max(list):
    max_value = list[0]
    for i in range(len(list)):
        if list[i] > max_value:
            max_value = list[i]
            index = i;

    return index
```

```
def initial_angle(lists):
	#Only go from 40 to 80 to not hit the first peak
	check_angles = []
	start_index = 0
	for i in range(len(lists[0])):
		if lists[0][i] > 40:
			if not check_angles:
				start_index = i
			check_angles.append(lists[0][i])
	check_intesities = lists[1][start_index:]
		#shorten intesities accordingly

	#Get highest intesity
	max_index = index_of_max(check_intesities)
	initial_angle = check_angles[max_index]
	#print(initial_angle)

	#plt.plot(check_angles,check_intesities)
	#plt.show()

	return initial_angle

def initial_angles(arrayOfLists):
	initials = []
	for i in arrayOfLists:
		initials.append(initial_angle(i))

	return initials

t1_80 = import_data("Blackbody\\InitialAngle\\t1_80.txt")
t2_80 = import_data("Blackbody\\InitialAngle\\t2_80.txt")
t3_80 = import_data("Blackbody\\InitialAngle\\t3_80.txt")
t4_80 = import_data("Blackbody\\InitialAngle\\t4_80.txt")
t5_80 = import_data("Blackbody\\InitialAngle\\t5_80.txt")
t6_80 = import_data("Blackbody\\InitialAngle\\t6_80.txt")

initials = initial_angles([t1_80,
                                  t2_80,
                                  t3_80,
                                  t4_80,
                                  t5_80,
                                  t6_80])

initial_angle_avg = mean_same_uncern(initials)
std_devation_intial_angle = standard_deviation(initial_angle_avg,
     initials)
uncertainty_initial_angle = 0.000005
print(f'Initial Angles: {initials}')
print(f'Initial Angle is: {initial_angle_avg} +/-
{std_devation_intial_angle}')

trials = [1, 2, 3, 4, 5, 6]
print(trials)
#plt.plot(trials, initials)
#plt.errorbar(trials, initials, yerr=uncertainty_initial_angle, fmt =
     '-o')
#plt.show()

#Wien's Displacement Law
def lamda_from_angle(angle):
     A = 13900
     B = 1.689
     return
np.sqrt(A/(np.sqrt(((2/np.sqrt(3))*np.sin(np.deg2rad(angle)) +
0.5)**2
     + 0.75) - B))

def get_theta_peak(lists):
     theta_peaks = []
     for data in lists:
          max_index = index_of_max(data[1])
          theta_peaks.append(data[0][max_index])

     return compress_data_avg(theta_peaks)

def compress_data_avg(thetas): #cause we took 2 of each voltage
     i = 0
     new_thetas = []
     while(i < len(thetas)):
          new_thetas.append((thetas[i] + thetas[i+1])/2)
          i += 2
     return new_thetas

def temperatureOfBulb(volt,amp):
     a0 = 4.5e-3
     t0 = 20
     r0 = 1.1
     return t0 + ((volt/amp)/r0 -1)/a0

def uncern_temperature(v, uncern_v, c, uncern_c, z):
     return uncertainty_prod(v, uncern_v, c, uncern_c, z)

def wiens_law(temps, lamdas):
     return temps * (lamdas / 1e+9)

def uncern_wiens_law(temps, un_temps, lamdas, un_lamdas, wiens):
     return uncertainty_prod(temps, un_temps, lamdas, un_lamdas,
wiens)

wien10_1 = import_data("Blackbody\\Wien\\10_1.txt")
wien10_2 = import_data("Blackbody\\Wien\\10_2.txt")
wien9_1 = import_data("Blackbody\\Wien\\9_1.txt")
wien9_2 = import_data("Blackbody\\Wien\\9_2.txt")
wien8_1 = import_data("Blackbody\\Wien\\8_1.txt")
wien8_2 = import_data("Blackbody\\Wien\\8_2.txt")
wien7_1 = import_data("Blackbody\\Wien\\7_1.txt")
wien7_2 = import_data("Blackbody\\Wien\\7_2.txt")
wien6_1 = import_data("Blackbody\\Wien\\6_1.txt")
wien6_2 = import_data("Blackbody\\Wien\\6_2.txt")
wien5_1 = import_data("Blackbody\\Wien\\5_1.txt")
wien5_2 = import_data("Blackbody\\Wien\\5_2.txt")
wien4_1 = import_data("Blackbody\\Wien\\4_1.txt")
wien4_2 = import_data("Blackbody\\Wien\\4_2.txt")
voltages = np.array([10, 9, 8, 7, 6, 5, 4])
currents = np.array([0.6, 0.567, 0.526, 0.487, 0.442, 0.392, 0.341])
uncern_angles_wien = np.array([0.05] * 7)
uncertainty_voltages = np.array([0.5] * 7)
uncertainty_currents = np.array([0.10] * 7)

theta_peaks = get_theta_peak([wien10_1,
                                        wien10_2,
                                        wien9_1,
                                        wien9_2,
                                        wien8_1,
                                        wien8_2,
                                        wien7_1,
                                        wien7_2,
                                        wien6_1,
                                        wien6_2,
                                        wien5_1,
                                        wien5_2,
                                        wien4_1,
                                        wien4_1])
theta_actual = initial_angle_avg - theta_peaks
lamdas = lamda_from_angle(theta_actual) #in nm
```

```
#uncertainty of lamda
uncerntainty_lamdas = uncern_angles_wien
print(f'Peak Measured Angles: {theta_peaks}')
print(f'Peak Actual Measured Angles: {theta_actual}')
print(f'Wavelength values: {lamdas}')
print(f'Uncertainties: {uncerntainty_lamdas}')
     #uncertainty is only propgated by one variable


#Temperature

temperatures = temperatureOfBulb(voltages, currents)
print(f'Temperature of Bulb: {temperatures}')

#uncertainty of temperature
uncerntainty_temperature = uncern_temperature(voltages,
uncertainty_voltages, currents, uncertainty_currents, temperatures)
print(f'Uncern Temp: {uncerntainty_temperature}')

#Wien's law calculation
wien_values = temperatures * (lamdas / 1e+9)
wien_uncern = uncern_wiens_law(temperatures,
uncerntainty_temperature, lamdas, uncerntainty_lamdas, wien_values)
avg_wien_value = mean_diff_uncern(wien_values, wien_uncern)
avg_wien_uncern = mean_same_uncern(wien_uncern)
std_dev_wien = standard_deviation(avg_wien_value, wien_values)

print(f'Wien Values: {wien_values}')
print(f'Wien Uncertainty: {wien_uncern}')

print(f'Avg Wien Value is: {avg_wien_value} +/- {std_dev_wien}')
```

Appendix 2: Stephen-Boltzmann's Law code

```
#Imports
import numpy as np
import matplotlib.pyplot as plt
from pylab import loadtxt

#Uncertainties
def uncertainty_sum(dx, dy):
     return np.sqrt(dx**2 + dy**2)

def uncertainty_prod(x, dx, y, dy, z):
    return np.sqrt((dx/x)**2 + (dy/y)**2) * z


def import_data(filename):
    data=loadtxt(filename, usecols=(0,1), skiprows=2, unpack=True)

    return data[0], data[1]

def mean_same_uncern(list):
    return sum(list)/len(list)

def standard_deviation(value, list):
    return np.sqrt((1/(len(list)-1))*sum((value - list)**2))

def mean_diff_uncern(list, uncern_list):
    return sum(list/uncern_list**2) / sum(1/uncern_list**2)

def temperatureOfBulb(volt,amp):
    a0 = 4.5e-3
    t0 = 20 + 273
    r0 = 1.1
    return t0 + ((volt/amp)/r0 -1)/a0

def uncern_temperature(v, uncern_v, c, uncern_c, z):
    return uncertainty_prod(v, uncern_v, c, uncern_c, z)

def areafromintensity(voltage, current, intensity):
    power = voltage * current
    return power / intensity


#Temperature
voltages = np.array([10, 9, 8, 7, 6, 5, 4])
uncertainty_voltages = np.array([0.5] * 7)
currents = np.array([0.604, 0.567, 0.528, 0.486, 0.437, 0.393,
0.336])
uncertainty_currents = np.array([0.10] * 7)

temperatures = temperatureOfBulb(voltages, currents)**4
uncern_temps = uncern_temperature(voltages, uncertainty_voltages,
currents, uncertainty_currents, temperatures)

#Area
intensity = [0.9518, 0.836, 0.7584, 0.5382, 0.7114, 0.6614, 0.6747]
area = areafromintensity(voltages, currents, intensity)
```

```
uncern_area = uncertainty_prod(voltages, uncertainty_voltages,
currents, uncertainty_currents, area)


print(area)
print(uncern_area)
plt.plot(temperatures, area)
plt.xlabel("Temperature")
plt.ylabel("Surface area of Light bulb")
plt.title("Area vs Temperture^4")
plt.show()
```