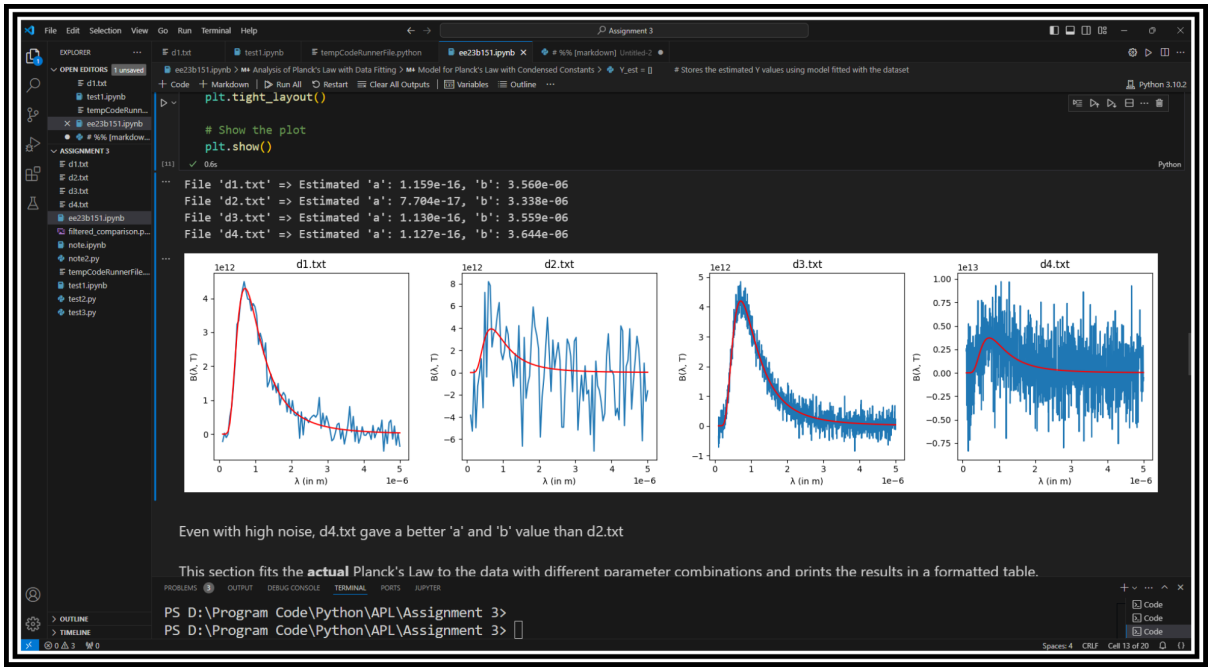

EE2703: Assignment-3

Analysis and Fitting of Planck's Law to Spectral Data

By Siva Sundar (EE23B151)

Date: 16/9/2024

Indian Institute of Technology Madras

Contents

1 Overview

1.1 Abstract

This experiment demonstrates the development of a Python program which implements **Curve-Fitting** method in the **SciPy** library for fitting **Planck's Law** to various different noisy **spectral data**.

1.2 Introduction

In the study of **blackbody radiation**, Planck's Law plays a crucial role in describing the **intensity of radiation** emitted by a blackbody as a **function of wavelength** and **temperature**. Mathematically, Planck's Law is expressed as:

$$B_\lambda(\lambda, T) = \frac{2hc^2}{\lambda^5} \cdot \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1}$$

Here, B_λ represents the **spectral radiance** per unit wavelength, λ is the wavelength of the emitted radiation, T is the absolute temperature of the blackbody, h is **Planck's constant**, c is the speed of light in a vacuum, and k_B is the **Boltzmann constant**.

Understanding and accurately modeling blackbody radiation is essential in various fields such as astrophysics, materials science, and thermal engineering. Accurate determination of parameters such as Planck's constant h and other constants is crucial for practical applications and theoretical studies.

This assignment focuses on analyzing and fitting Planck's Law to spectral data obtained from multiple sources and to play around with curve fitting. The primary objective is to extract data points from multiple files, visualize them, and fit a model based on Planck's law to estimate key parameters.

1.3 Methodology

The approach for this assignment involves the following key steps:

1. **Data Extraction and Visualization:** Utilize Python to read data from text files and generate plots for visual inspection. This step involves extracting data points from multiple files and visualizing them to understand the underlying trends and noise in the data.
2. **Model Fitting:** Apply curve fitting techniques to match the experimental data with a simplified version of Planck's Law, characterized by condensed constants. This involves using numerical methods to fit the model to the data and assess how well the model represents the observed data.
3. **Parameter Estimation:** Estimate parameters such as the temperature, Planck's constant, speed of light, and Boltzmann constant by fitting the actual Planck's Law model to the data. This step includes optimizing the model to determine the best-fit parameters and evaluating their accuracy.

By integrating libraries such as `matplotlib`, `numpy`, and `scipy.optimize`, this assignment aims to provide a comprehensive understanding of data fitting in the context of physical laws. It demonstrates practical applications of numerical methods and optimization techniques, offering insights into how these tools can be used to analyze and interpret scientific data.

2 Data Extraction and Visualization

2.1 Reading the file

Using `XY_extract` function, we can extract the X and Y coordinates from the files. It takes in file-names as input and returns two lists which contain X and Y coordinates.

```
filenames = ["d1.txt", "d2.txt", "d3.txt", "d4.txt"]
X = []; Y = []    # Contains all the X and Y values of the files

for filename in filenames:
    X_val, Y_val = XY_extract(filename)
    X.append(X_val); Y.append(Y_val)
```

Code Snippet 1: Spectral Data Extraction

2.2 Plotting the data points

The data from each file is plotted using `matplotlib.pyplot` in subplots. Each subplot represents the data from one file, with X values (wavelength in nanometres) on the x-axis and Y values (spectral radiance in $Wsr^{-1}m^{-3}$) on the y-axis.

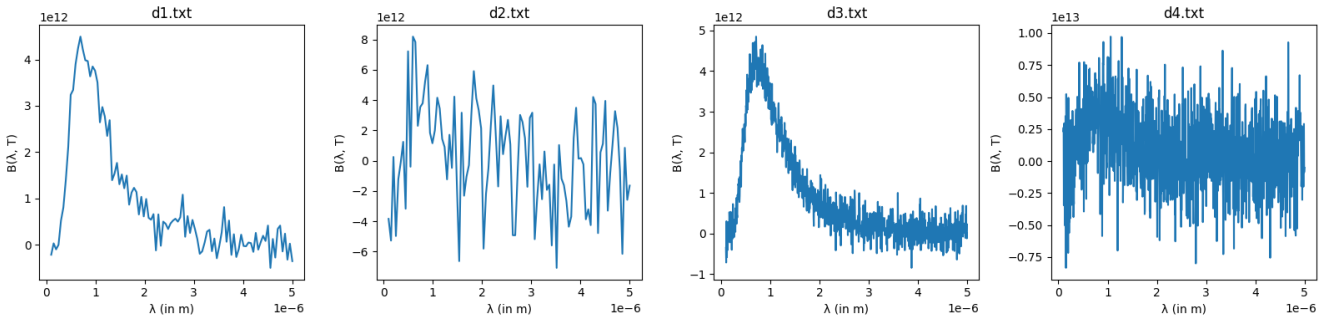

Figure 1: Plotting data from each file

2.3 Observations from plots

We can see the variations in the amount as well as the amplitude of noise in each of the datasets:

- `d1.txt` and `d2.txt` has less amount of noise while `d3.txt` and `d4.txt` has a lot of noise.
- The noise in `d1.txt` and `d3.txt` have similar amplitudes which are lower compared to the amplitudes seen in `d2.txt` and `d4.txt`.

Hence, we can describe the noise present in each file as:

- ⋆ `d1.txt`: Low amount & Low amplitude.
- ⋆ `d2.txt`: Low amount & High amplitude.
- ⋆ `d3.txt`: High amount & Low amplitude.
- ⋆ `d4.txt`: High amount & High amplitude.

Now, let's see which of these types of data give better results, ie, better values for the parameters in Planck's law.

3 Model Fitting

In this assignment, we have two main objectives: **First**, we aim to fit Planck's Law to the spectral data to identify the best-fitting curve without focusing on parameter estimation. **Second**, we use known values for some parameters to fit Planck's Law to the data and estimate the remaining parameters, such as Planck's constant, the speed of light, temperature, and the Boltzmann constant.

3.1 Finding the Best-Fit Curve

We simplify the Planck's law by condensing the constants which results in the following equation:

$$B(\lambda) = \frac{a}{\lambda^5} \cdot \frac{1}{e^{\left(\frac{b}{\lambda}\right)} - 1}$$

where,

- $a = 2hc^2 = 1.194 \times 10^{-16}$ W.m^2
- $b = \frac{hc}{k_B \cdot T} = 3.6 \times 10^{-6}$ m

Using `curve_fit` method in `SciPy` library, we fit this equation to the datasets which yields the following curves.

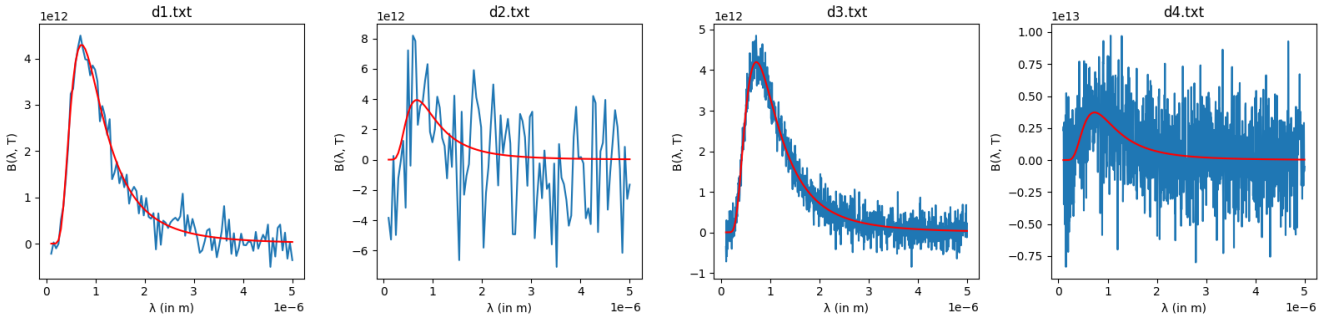

Figure 2: Red curves show fitted model

Observations

We can see that `d2.txt`, which had low amount of noise but with high amplitude, has its a and b values highly deviated from actual values as seen from the table below.

File	Value of a	Value of b
d1.txt	1.159e-16	3.560e-6
d2.txt	7.704e-17	3.338e-6
d3.txt	1.130e-16	3.559e-6
d4.txt	1.127e-16	3.644e-6

Table 1: Estimated values of a and b for different files.

It is suprising to see that `d4.txt`, with high noise of high amplitude, gave a better estimate than `d2.txt`.

One reason that `d4.txt`, despite having a high amplitude of noise, provided better results than `d2.txt` could be attributed to the effect where the noise, when distributed evenly, might average out its impact. This can be explained by the **Central Limit**

Theorem (CLT). The CLT states that the distribution of the sum (or average) of a large number of independent, identically distributed random variables approaches a normal distribution, regardless of the original distribution of the variables.

3.2 Finding the Parameters

The `curve_fitter` function fits Planck's Law to the data X and Y in the following ways, depending on which parameters are known:

1. **Fitting with 3 Known Constants**: The function estimates the fourth parameter by fitting the model while keeping three constants fixed at known values.
2. **Fitting with 2 Known Constants**: The function estimates two parameters by fitting the model with the other two constants fixed. Here, as 'h' is related to 'c' by 'a' and 'k_B' is related to 'T' by 'b', we have **two independent variables**. Hence, fixing any one variable from each pair will also enable us to find the rest of the parameters. ie, If I chose h and c as *given* parameters, we don't have enough information to find T or k_B separately.

For each approach, the function reports the estimated values of the unknown parameters, which helps in better understanding the data.

Observations

Condition	**d1.txt**	**d2.txt**	**d3.txt**	**d4.txt**
Given h, c, k_B	$T = 4020.40$	$T = 3957.28$	$T = 4001.10$	$T = 3905.21$
Given c, k_B, T	$h = 6.583 \times 10^{-34}$	$h = 6.687 \times 10^{-34}$	$h = 6.621 \times 10^{-34}$	$h = 6.825 \times 10^{-34}$
Given h, k_B, T	$c = 2.974 \times 10^{8}$	$c = 3.009 \times 10^{8}$	$c = 2.995 \times 10^{8}$	$c = 3.115 \times 10^{8}$
Given h, c, T	$k_B = 1.388 \times 10^{-23}$	$k_B = 1.366 \times 10^{-23}$	$k_B = 1.381 \times 10^{-23}$	$k_B = 1.348 \times 10^{-23}$
Given c, k_B	$h = 6.440 \times 10^{-34}$	$h = 4.279 \times 10^{-34}$	$h = 6.280 \times 10^{-34}$	$h = 6.260 \times 10^{-34}$
	$T = 3929.80$	$T = 2785.03$	$T = 3833.27$	$T = 3732.11$
Given c, T	$h = 6.440 \times 10^{-34}$	$h = 4.279 \times 10^{-34}$	$h = 6.280 \times 10^{-34}$	$h = 6.260 \times 10^{-34}$
	$k_B = 1.357 \times 10^{-23}$	$k_B = 9.615 \times 10^{-24}$	$k_B = 1.323 \times 10^{-23}$	$k_B = 1.289 \times 10^{-23}$
Given h, T	$c = 2.957 \times 10^{8}$	$c = 2.412 \times 10^{8}$	$c = 2.921 \times 10^{8}$	$c = 2.916 \times 10^{8}$
	$k_B = 1.376 \times 10^{-23}$	$k_B = 1.197 \times 10^{-23}$	$k_B = 1.359 \times 10^{-23}$	$k_B = 1.326 \times 10^{-23}$
Given h, k_B	$c = 2.957 \times 10^{8}$	$c = 2.412 \times 10^{8}$	$c = 2.921 \times 10^{8}$	$c = 2.916 \times 10^{8}$
	$T = 3986.29$	$T = 3465.99$	$T = 3937.58$	$T = 3839.79$

Table 2: Estimated values for the given conditions across the four files.

The file `d2.txt` performs well in the first four cases because providing three known constants allows for more accurate estimation of the parameters a and b , resulting in a better fit to the dataset. However, in the last four cases, since only two constants are given, the condensed constants **remain unchanged**. As a result, the curve-fitting function produces the same fit as before, leading to less accurate results in these cases.

4 Conclusion

This experiment demonstrates the use of Python's `SciPy` library for curve fitting to model Planck's Law and estimate physical constants from noisy spectral data. Key findings include:

- **Noise and Accuracy:** The amount and distribution of noise significantly affect the accuracy of parameter estimation. Surprisingly, `d4.txt`, with high but evenly distributed noise, produced better results than `d2.txt`, which had high amplitude but lower noise, showing the importance of noise distribution.
- **Known Parameters:** The more constants provided, the better the curve fit. When three constants were known, the results were more accurate. With only two known constants, the estimates were less reliable due to fewer constraints on the model.
- **Practical Impact:** Accurate parameter estimation from noisy data has wide applications in fields like astrophysics and thermal engineering. The experiment highlights the importance of managing noise and selecting the right number of fixed parameters for reliable results.

In summary, this experiment showcases effective data analysis using curve fitting, emphasizing the importance of noise handling and parameter selection in achieving accurate results.

CLICK HERE to see the jupyter notebook which implements curve fitting. All the data files are **ASSUMED** to be in the **SAME** directory as the notebook.

5 References

[1] Wikipedia: Planck's Law

[2] Stack Exchange: Why is Gaussian noise called so?

[3] Youtube: Central Limit Theorem