Assignment 3: Data Estimation

Arivoli Ramamoorthy EE23B008

September 16, 2024

1 Introduction

A report on the code written for Assignment-3 of EE2703. It explains a python script written to fit blackbody radiation data using Planck's Law.

1. Fitting only temperature (T) while keeping h, c, and k constant.
2. Fitting h, c, and k as scaling factors while fixing temperature (T).
3. Fitting all parameters (T, h, c, and k) simultaneously.

2 Planck's Law

Planck's Law describes the spectral radiance $I(\lambda, T)$ of a blackbody as a function of wavelength (λ) and temperature (T):

$$I = \frac{2hc^2}{\lambda^5 \left(\exp\left(\frac{hc}{\lambda kT}\right) - 1\right)}$$

where:

- h is Planck's constant,
- c is the speed of light in vacuum,
- k is Boltzmann's constant,
- T is the absolute temperature,
- λ is the wavelength.

3 Code Explanation

3.1 Constants and Imports

The script begins by importing necessary libraries such as `numpy`, `matplotlib`, and `scipy.optimize`. It also defines important physical constants (approx):

- Planck's constant: $h = 6.6 \times 10^{-34}$ J·s,
- Speed of light: $c = 3.0 \times 10^{8}$ m/s,
- Boltzmann's constant: $k = 1.38 \times 10^{-23}$ J/K.

3.2 Planck's Law Function

The `planck` function implements Planck's Law. This function will be used in the fitting process to generate theoretical curves based on temperature and other constants.

```
def planck(lam, T, h, c, k):
    return (2*h*c**2) / (lam**5 * (np.exp((h*c)/(lam*k*T)) - 1))
```

3.3 Fitting Procedures

- `fit_temperature`: This function fits the temperature (T) while keeping h, c, and k fixed to their constants. The `curve_fit` function from `scipy.optimize` is used to perform this optimization.
- `fit_constants`: This function fits the scaling factors for h, c, and k while keeping temperature fixed to the previously determined value.
- `fit_all_parameters`: This function fits all parameters (T, h, c, and k) simultaneously without fixing any constants.

The script prints the fitted parameters and their values to the console and generates comparison plots.

4 How to Run the Code

To run the Python script, follow these steps:

1. Run the script from the command line, passing the path to the data file as an argument:

   ```
   python ee23b008.py data.txt
   ```

2. The script will perform the fits, print the results to the console, and display plots comparing the original data and the fitted curves.

5 Outputs (for d1.txt):

Output for d1.txt:
Fitted Temperature (keeping h, c, k constant): 4010.70 K
Temperature Standard Deviation: 14.23 K

Fitted parameters (T fixed, h, c, k fitted):
T = 4010.70 K
h = 6.65e-34 Js
c = 2.95e+08 m/s
k = 1.37e-23 J/K

Fitted parameters (T, h, c, k) with no constants :
T = 4476.80 K
h = 1.23e-33 Js
c = 2.17e+08 m/s
k = 1.68e-23 J/K

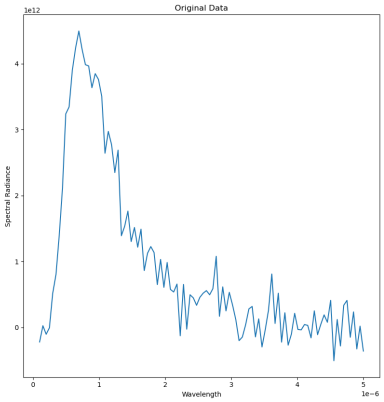

Figure 1: Original data d1.txt

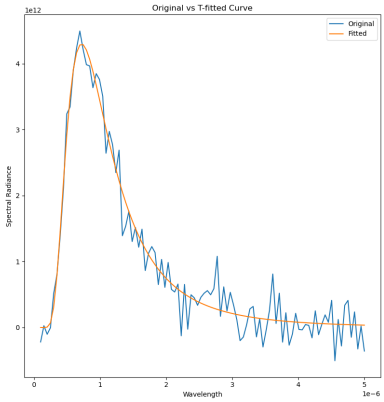

Figure 2: T-fitted

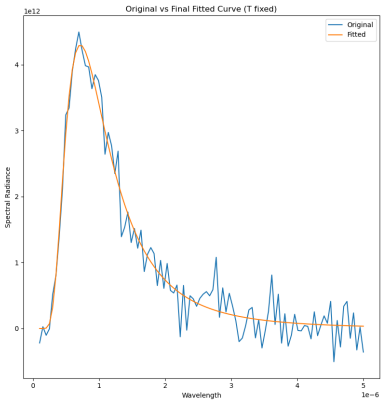

Figure 3: T-fixed

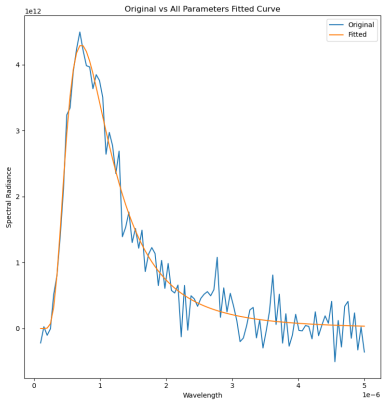

Figure 4: All parameters fitted

6 Conclusion

The script supports the fitting of temperature while keeping constants fixed, fitting constants while fixing temperature, and fitting all parameters simultaneously. And we can see that, when more contants are fixed, the fit is more accurate, and when no constants are fixed, the values are very bad as expected.