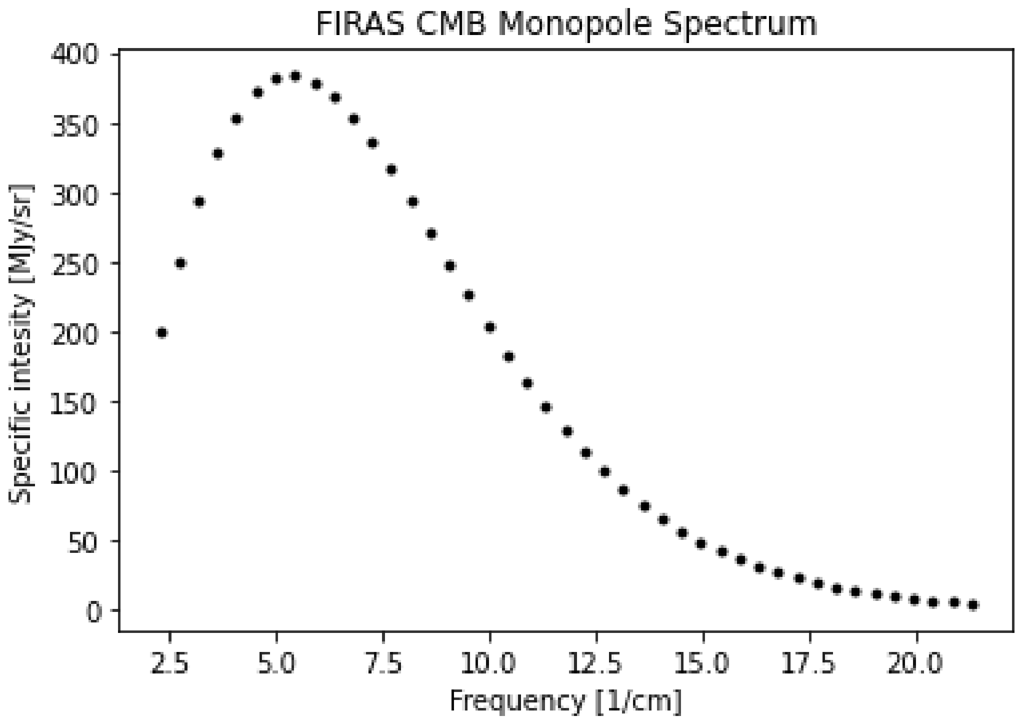
Problem Set 1

Alexis Lohman

September 1st, 2020

1 Downloading and Plotting the FIRAS Data

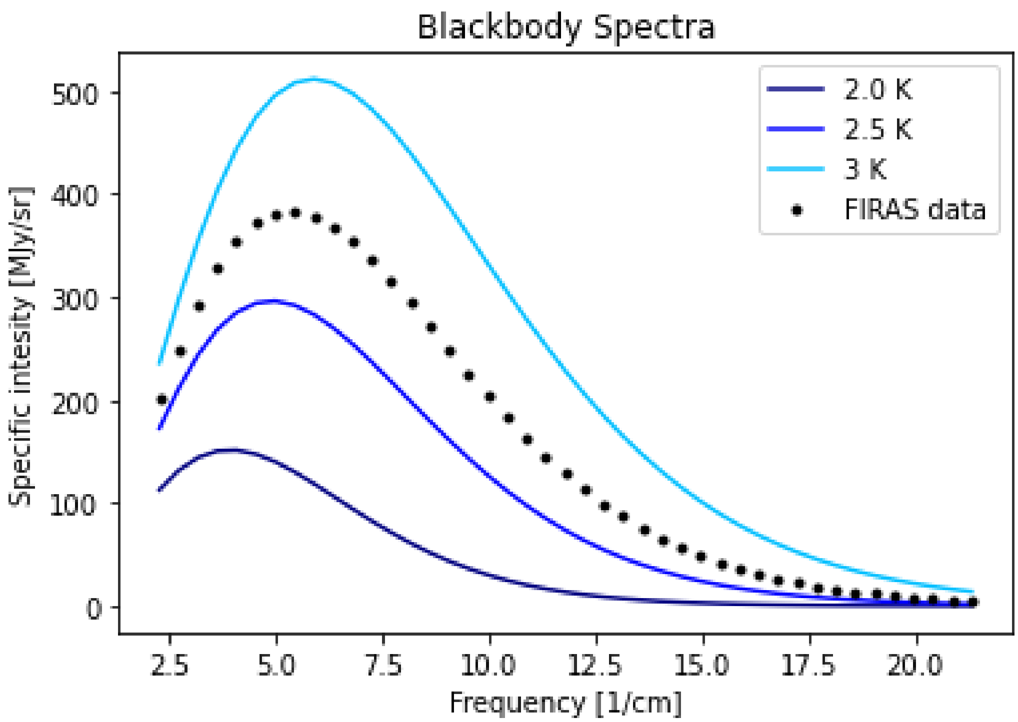
```
Shape of the data array (rows, columns): (43, 5)
```



2 Function to Calculate Planck's Equation

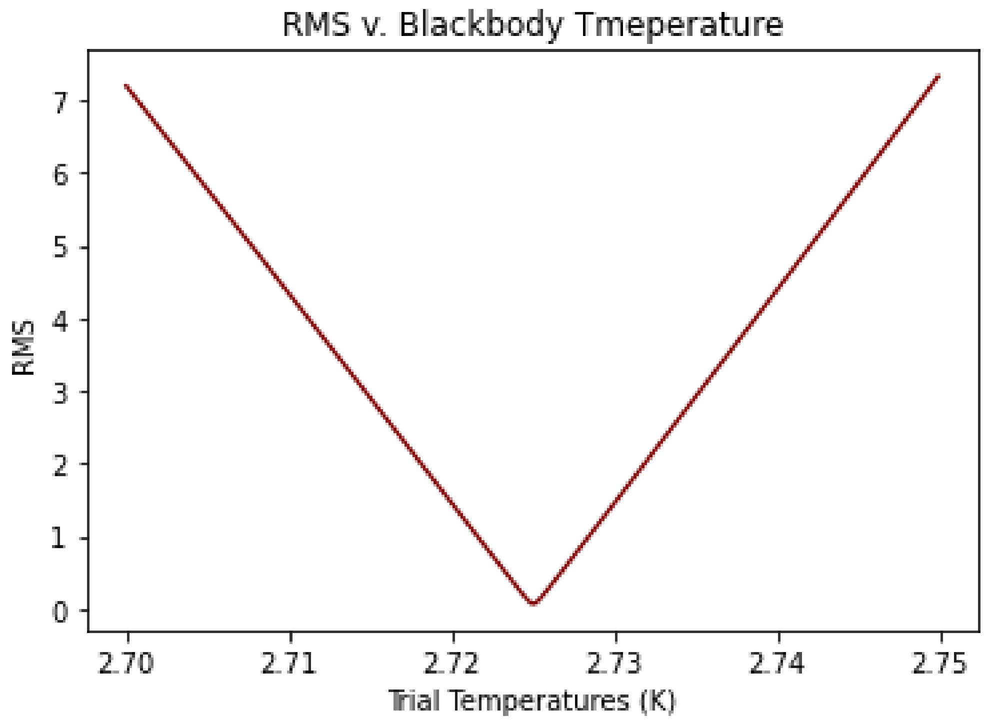
1. Yes, the three spectra look similar in shape and follow the same trend.
2. Yes, they are within the right ballpark to match the FIRAS measurement. The T at 2.5 K seems to fit the best but it truly lies between 3 K and 2.5 K.

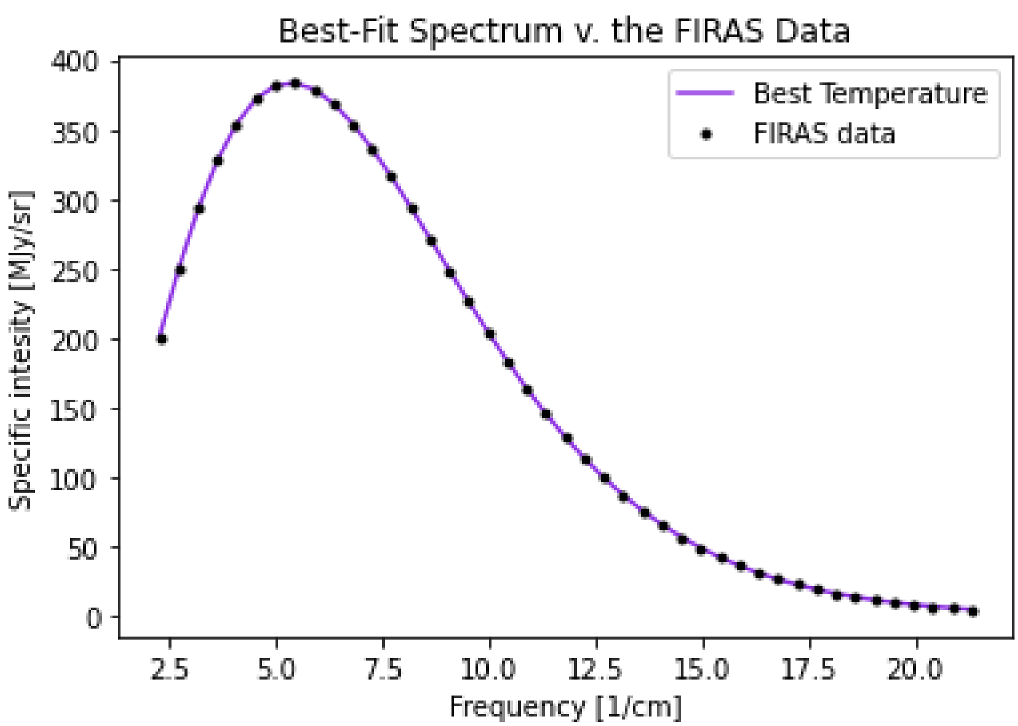
3. My estimated T that the FIRAS measurement uses would be in the range 2.65 - 2.8 K. This is based off my assessment of the graph comparing the three spectra and the FIRAS data.

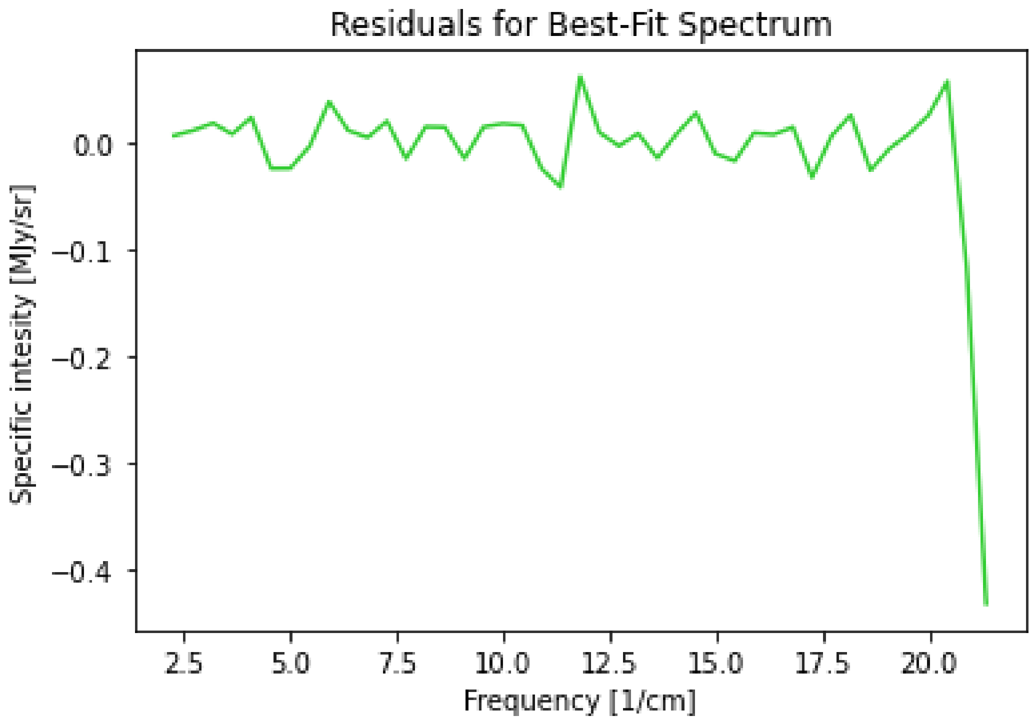


3 Best-Fit Blackbody T for the FIRAS Measurement

```
Best Fit Temperature is 2.725.
```


RMS v. Blackbody Tmeperature
RMS
7
6
5
4
3
2
1
0
2.70
2.71
2.72
2.73
2.74
2.75
Trial Temperatures (K)


Best-Fit Spectrum v. the FIRAS Data
Best Temperature
FIRAS data
Specific intesity [MJy/sr]
400
350
300
250
200
150
100
50
0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Frequency [1/cm]



1. The residual are generally small compared to the FIRAS measurement, it is not until in the upper range of frequency.
2. The blackbody is a good fit of the FIRAS data. There's a low level of discrepancy with the model and the FIRAS measurement.

4 Raw Data and Graphs

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
```

[2]:
```
# Blackbody spectrum function
def blackbody(freq, temp):
    h = 6.62607004e-34;     #kg*m^2/s
    c = 2.99792458e8;       #m/s
    k = 1.38064852e-23;     #kg*m^2/k*s^2

    spec = (2*h*freq**3/c**2)/(np.exp(h*freq/k/temp)-1);

    return spec;
```

```
#Load and plot FIRAS Data
datafile = 'Downloads/firas_monopole_spec_v1.txt';
data = np.loadtxt(datafile)
print('Shape of the data array (rows, columns):');
print(data.shape)
f = data[:,0];
s = data[:,1];
plt.figure(1);
plt.clf();
plt.plot(f,s,'k.');
plt.xlabel('Frequency [1/cm]');
plt.ylabel('Specific intesity [MJy/sr]');
plt.title('FIRAS CMB Monopole Spectrum');
plt.show()
plt.savefig('figure_1.png');

#Convert FIRAS freq from observer units of 1/cm to SI units of Hz.
c = 2.99792458e8;           #m/s
wavel_SI = (1)/(100*f);     #1/m*f
freq_SI = c/wavel_SI;       #1/s

#Calculate blackbody spectrum for T = 2, 2.5, & 3 K (bb_x)
bb_2 = blackbody(freq_SI, 2.0);
bb_25 = blackbody(freq_SI, 2.5);
bb_3 = blackbody(freq_SI, 3.0);

#Convert to MJy/sr (bb_MJy_x), where 1 Jy = 10e-26 and M = 10e6 so MJy =␣
 ↪10e-26*10e6 = 10e-20
bb_MJy_2 = bb_2/(1e-20);
bb_MJy_25 = bb_25/(1e-20);
bb_MJy_3 = bb_3/(1e-20);

#Plot bbspectrum in MJy/sr
plt.figure(2);
plt.clf();
plt.plot(f,bb_MJy_2, 'navy');
plt.plot(f,bb_MJy_25, 'blue');
plt.plot(f,bb_MJy_3,'deepskyblue');
plt.plot(f,s,'k.');
plt.xlabel('Frequency [1/cm]');
plt.ylabel('Specific intesity [MJy/sr]');
plt.title('Blackbody Spectra');
plt.legend(['2.0 K','2.5 K','3 K', 'FIRAS data']);
```

```
plt.show()
plt.savefig('figure_2.png');

#Find best-fit bb spectrum to FIRAS Data
trial_t = np.linspace(2.7,2.75, 500, False);
bb = np.array([blackbody(freq_SI, t) for t in trial_t]);
residuals = s - bb*1e20

def rms(f,t,orig):
    #residual error spectrum = data spectrum - model spectrum
    # rms = sqrt(sum(y_i^2)/n)
    #y_i = res.error val at given freq i, n is total number of data␣
 ↪points

    n = len(f)
    rmse = []

    for x in t:
        test_spec = blackbody(f,x) *1e20
        rms = orig - test_spec
        rmse.append(np.sqrt(np.sum(rms**2)/n))

    return rmse

rms = rms(freq_SI,trial_t,s)

plt.figure(3);
plt.clf();
plt.plot(trial_t, rms, 'maroon');
plt.xlabel('Trial Temperatures (K)');
plt.ylabel('RMS');
plt.title('RMS v. Blackbody Tmeperature');
plt.show()
plt.savefig('figure_3.png');

index = np.where(rms==np.amin(rms))[0][0]
print('Best Fit Temperature is {}'.format(trial_t[index]))

#Plotting Best-Fit Spectrum v. the FIRAS Data
BestT = 2.725
BestFit = blackbody(freq_SI,BestT)*1e20

plt.figure(4);
plt.clf();
```

```python
plt.plot(f, BestFit, 'blueviolet');
plt.plot(f, s, 'k.');
plt.xlabel('Frequency [1/cm]');
plt.ylabel('Specific intesity [MJy/sr]');
plt.title('Best-Fit Spectrum v. the FIRAS Data');
plt.legend(['Best Temperature', 'FIRAS data']);
plt.show()
plt.savefig('figure_4.png');

#Calculate residuals for best-fit spectrum and plotting it
residuals = s - BestFit
plt.figure(5);
plt.clf();
plt.plot(f, residuals, 'limegreen');
plt.xlabel('Frequency [1/cm]');
plt.ylabel('Specific intesity [MJy/sr]');
plt.title('Residuals for Best-Fit Spectrum');
plt.show()
plt.savefig('figure_5.png');
```