Dataset-1

In this documentation, we will discuss the construction of the M matrix for least squares linear regression and visualize the original noisy data with error bars, along with the overlaid estimated line. We will also add a legend to the plot to label all the components appropriately.

Construction of the M Matrix

The M matrix is constructed for least squares linear regression with two columns: one containing the x-coordinates of the data points, and the other containing a column of 1's. This allows us to fit a linear equation of the form y = mx + c to the data, where:

- y is the dependent variable (in this case, the observed y-values).
- x is the independent variable (the x-coordinates of the data points).
- m is the slope of the fitted line.
- c is the y-intercept of the fitted line.

The M matrix is constructed as follows:

```
M = np.column_stack([xarr, np.ones(len(xarr))])
```

Original Noisy Data with Error Bars

The plot below displays the original noisy data along with error bars. Error bars are plotted for every 25 data points.

```
plt.plot(xarr, yarr, label="Observed plot with noise")
plt.errorbar(xarr[::25], yarr[::25], yerr=std_dev, fmt="o", label="Error Bar")
```

yerr can be calculated as the standard deviation of difference in given y-coordinates and estimatted y-coordinates using least squate fitting.

```
err = np.array([y1 - y2 for y1, y2 in zip(fitted_yarr, yarr)])
std_dev = np.std(err, axis=0)
```

Estimated Line Overlay

Estimated line is drawn by:

```
plt.plot(xarr, fitted_yarr, label=f"Best fitted line: y = {m}x + {c}")
```

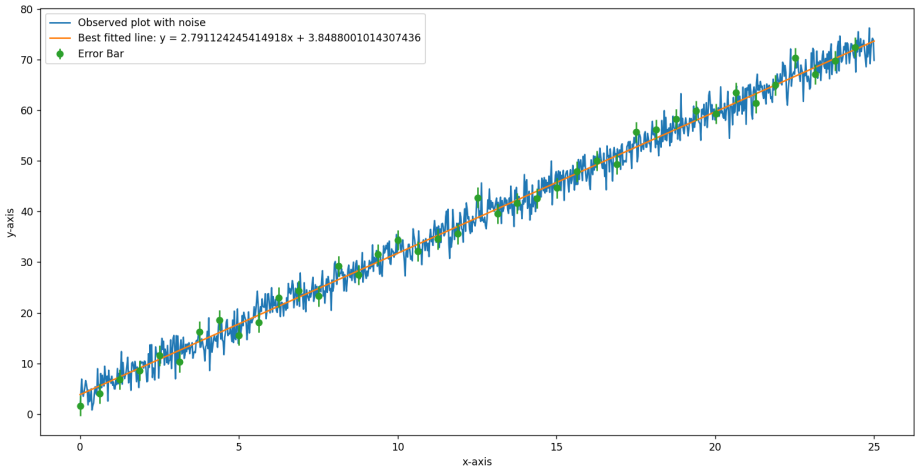
where fitted_arr is the y-coordinates obtained by least square fitting.

The estimated line have approximate

- slope of, `m = 2.7911`
- y-intercept, `c = 3.8488`

Legend

- "Observed plot with noise": Original data with noise.
- "Best fitted line: y = mx + c": Estimated line obtained from least squares fitting.
- "Error Bar": Error bars for 1 in 25 datapoints.



DataSet-2

Estimating Periodicity of Sine Waves

The periodicity of the sine waves was estimated by visually inspecting the dataset and identifying the approximate interval between consecutive peaks or troughs in the plot. This simple yet effective method provides a reasonable estimate of the frequency of a periodic signal.

In this specific case, the dataset represents a time series plot. By observing the plot, it was noted that the wave repeats itself approximately every 2.5 units of time. Therefore, the estimated frequency was calculated as the reciprocal of this period, which is

$$\frac{1}{2.5} = 0.4Hz.$$

Construction of the M Matrix and Least Squares Equation

The M matrix was constructed based on the assumed form of the signal, which is a combination of three sine waves with frequencies (f), (3f), and (5f).

The matrix (M) was constructed as follows:

$$M = \begin{bmatrix} \sin(2\pi f x_1) & \sin(2\pi 3 f x_1) & \sin(2\pi 5 f x_1) \\ \sin(2\pi f x_2) & \sin(2\pi 3 f x_2) & \sin(2\pi 5 f x_2) \\ \vdots & \vdots & \vdots \\ \sin(2\pi f x_n) & \sin(2\pi 3 f x_n) & \sin(2\pi 5 f x_n) \end{bmatrix}$$

where $x_1, x_2, \ldots, x_n$ are the elements of the input array x_{arr}.

The solution is obtained using the NumPy function `np.linalg.lstsq`, which returns the values of (a), (b), and (c) which are the amplitudes of each of three sine terms respectively.

(a), (b), and (c) are estimated from the Matrix (M) by the equation:

```
a, b, c = np.linalg.lstsq(M, yarr, rcond=False)[0]
```

The equation to find the least square fitted y-coordinates is

```
sqf_y = a * np.sin(2 * np.pi * f * x) + b * np.sin(2 * np.pi * 3 * f * x) + c * np.sin(2 *
```

Here f = 0.4Hz(Which is calculated on the top)

Curve Fitting using `curve_fit`

The `curve_fit` function from the SciPy library was used to perform curve fitting. The chosen model function is defined as `sinfunc(x, a, b, c, f)` which represents the combination of three sine waves.

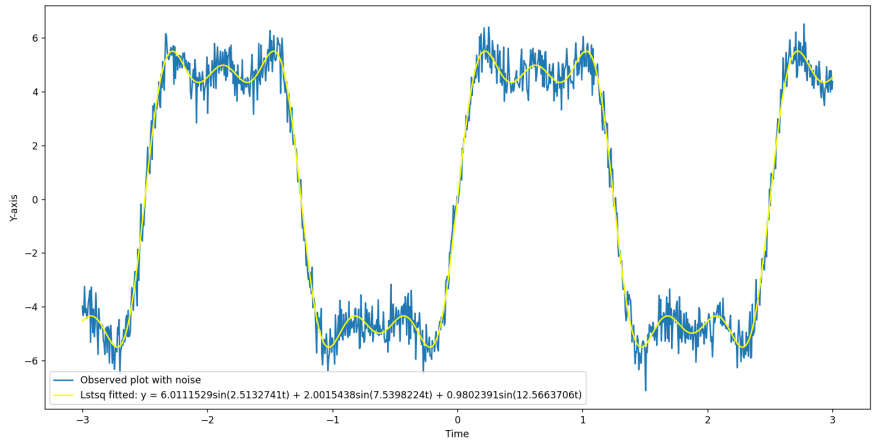
The parameters (a), (b), (c), and (f) are optimized to minimize the difference between the model prediction and the observed data.

The `curve_fit` function returns the optimized parameters and a covariance matrix, which can be used to estimate uncertainties in the parameters.

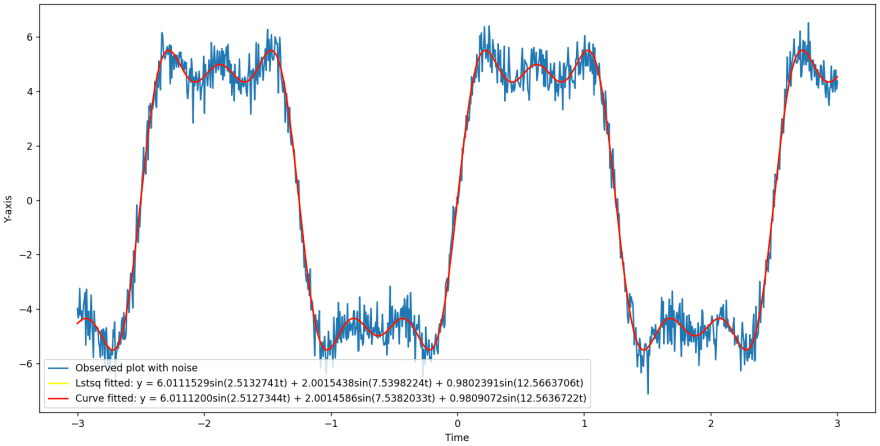
Difference between the results in Least Squares and Curve Fitting

The results obtained in both Least Squares and Curve fitting method are both similar(irrespective of very slight difference).

The plot obtained in case of Least Squares Method is as follows:



The plot obtained by using Curve fitting method is:



We can also able to see the two equations obtained in the above diagram.

Analysis

In one of the outputs,

The equation obtained in case of Lease Square fitting method is:

$y = 6.011152934913329\sin(2.5132741228718345t) + 2.001543820042488\sin(7.5398223686155035t) + 0.9802390885802221\sin(12.566370614359172t)$

The equation obtained in case of Curve fitting method is:

$y = 6.011120024994516\sin(2.5127344300656076t) + 2.0014585623429983\sin(7.538203290196823t) + 0.980907227033235\sin(12.563672150328038t)$

The frequency obtained from Curve fitting method is: 0.3999141052221379 units

We can clearly able to see that the two equations are almost similar and the we can get the repective amplitudes of the sine waves the above equations.

DataSet-3_1

In this program I have initialised the constants as follows(all in SI units):

h = 6.62607015 $\times$ 10^(-34)

c = 299792458

k = 1.380649 $\times$ 10^(-23)

Where h - plack's constant, c - speed of light in vacuum, k - Boltzman Constant

Starting Conditions

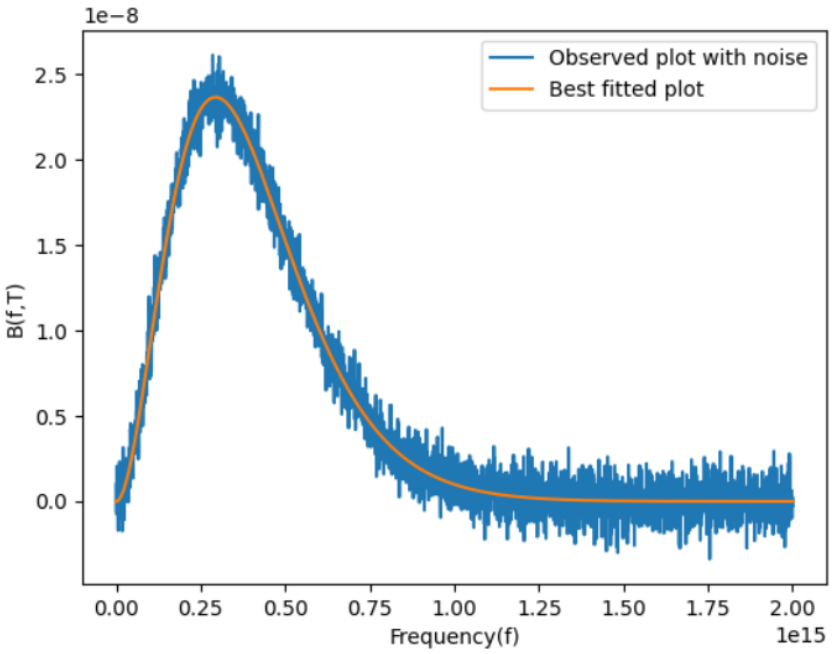
- The starting condition for temperature is finalised by trial and error method
- I have stated with a value less than 10 and I have increased it step by step
- Initially the program will be giving overflow warning
- At some point of starting codition of T the intepreter won't show any overflow warning.
- In this way I have gave the starting condition of **p0 = [200]**

Result

The temperature obtained as output is:

T = 4997.342002631031 Kelvin

The plot obtained is as follows:



DataSet-3_2

Starting Conditions

The starting condition I have given is

p0 = [6.62607 x 10^(-34), 299792458, 1.380649 x 10^(-23), 4997]

which means(All in SI units),

h = 6.62607 x 10^(-34)

c = 299792458

k = 1.380649 x 10^(-23)

T = 4997

- Here I have given the starting condition for the temperature as 4997k as a result obtained in the program dataset-3_1

Result

The final value of the constants obtained is as follows:

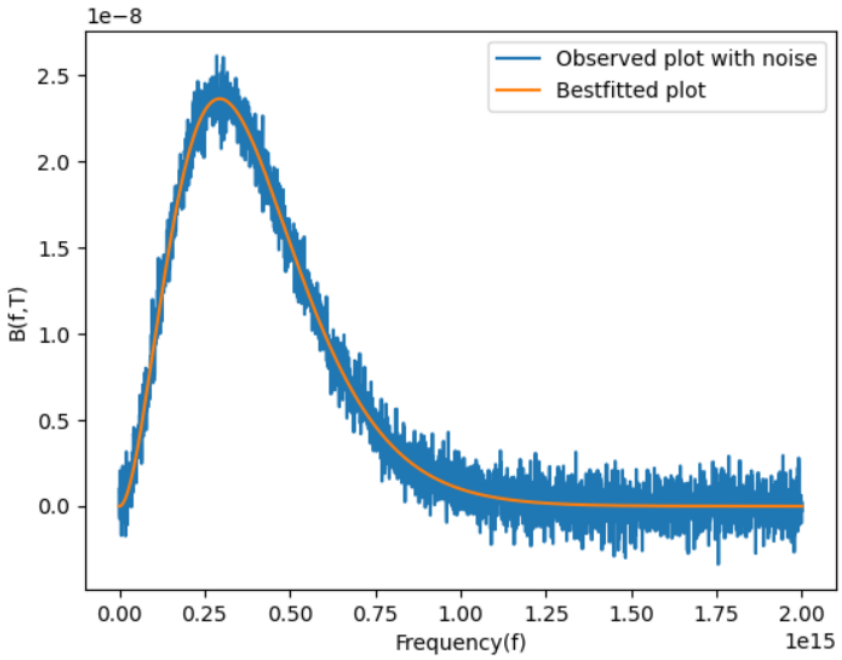
Planks constant(h) : 6.71032690689009e-34 Js

Speed of light in vacuum(c) : 302245768.00576794 m/s

Boltzmann constant(k) : 1.3898849250895194e-23 J/k

Temperature(T) : 5031.945973456874 k

The plot obtained is as follows:



Analysis

- The final values obtained as results are slightly greater than the actual values
- These values can be improved by providing the starting conditions more precisely