EE2703 : Applied Programming Lab
Assignment 3 Report

Abhishek Sekar
EE18B067

February 11, 2020

1. Aim of the Assignment

Our aim in this assignment is twofold:
i) Model a function which will fit the given values with the **least possible error**. Plot the respective errors obtained.
ii) Determine the relationship between the errors in the above case and the noise present in the data.

2. A Description of the Assignment

2.1. Generating noisy data

The program **generate data.py** generates the data corresponding to the following function:

$$f(t) = 1.05J_2(t) - 0.105t + n(t) \tag{1}$$

Where the equation represents a function comprising of a
bessel function
a linear function on time
varying amounts of noise (n(t))
Noise corresponds to the random fluctuations in the value due to a multitude of random effects.
Probability of noise here takes the following form:

$$Pr(n(t)|\sigma) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{n(t)^2}{2\sigma^2}\right) \tag{2}$$

The noise has a normal distribution with the standard deviation varying as follows in the program:

$$\sigma = logspace(-1, -3, 9) \tag{3}$$

The data corresponding to the function with different amounts of noise added are then stored in 9 columns of the file **fitting.dat**.

2.2.Plotting the noisy data

This noisy data is then plotted in order to get a visualization of the given problem statement.
A graph encompassing all curves is plotted followed by an errorbar plot which plots the real value of the function, excluding noise alongside data of one of the curves.

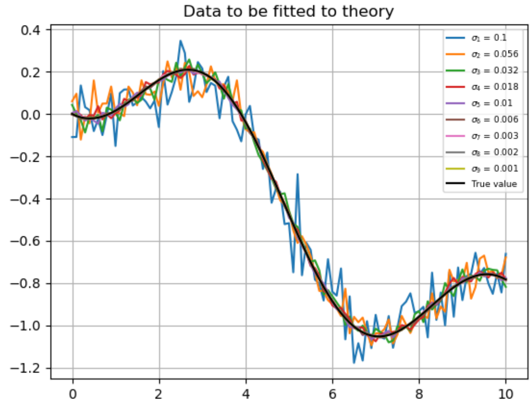

Figure 1: Noisy Data With the True Data

As we can see, the noisiness in the data increases with increasing value of σ. Another view of how the noise affects the data can be seen below :

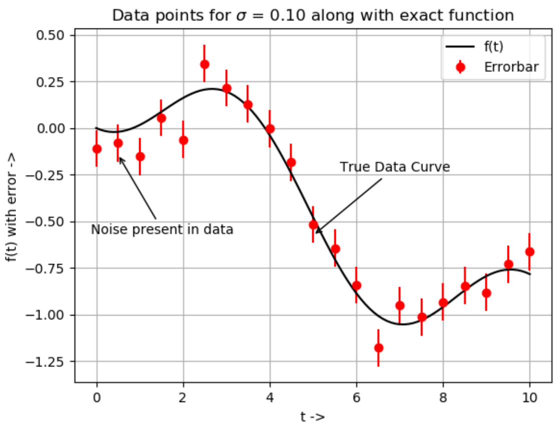

Figure 2: Noisy Data with Errorbar

The red lines indicate the standard deviation of the noisy data from the original data.

2.3.Attempt to create an accurate data model

From the data, we can conclude that the data can be fitted into a function of the form :

$$f(t) = AJ_2(t) + Bt$$

Where A and B are unknown coefficients.

Mean Square Error:

To find the coefficients **A** and **B**, we first try to find the mean square error between the function and the data for 21 values of **A** and **B** in the intervals [0,2] and [-0.2,0] respectively. Expression for mean square error:

$$\epsilon_{ij} = \frac{1}{101}\sum_{k=0}^{101}(f_k - g(t_k, A_i, B_j))^2$$

where ϵ_{ij} is the error for (A_i, B_j) set. The contour plot of the error is shown below :

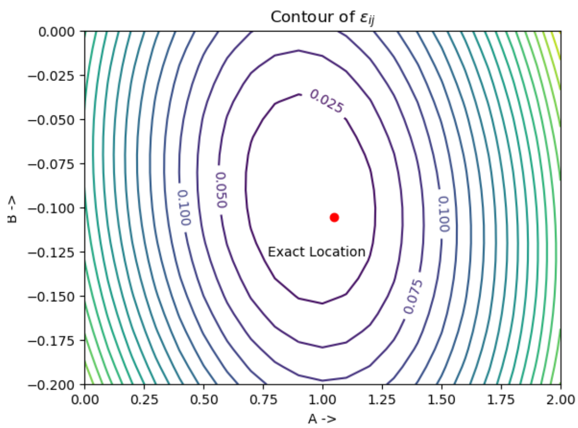

Figure 3: Contour Plot of ϵ_{ij}

We can see that the location of the minima is approximately near the original function coefficients.

Least Squares Method:

Now we attempt to do this with an other method. We solve for the function using the `lstsq` function in Python for the equation $M.p = D$ for p where $M = \begin{bmatrix} J_2(t_1) & t_1 \\ ... & ... \\ J_2(t_m) & t_m \end{bmatrix}$, $p = \begin{bmatrix} A_o \\ B_o \end{bmatrix}$ and D is the column matrix of the noisy data.

Thus, we solve for p and then find the mean square error of the values of A and B found using **lstsq** and the original values (1.05,-0.105).

2.4. Plotting error against noise

The plot for the different noisy data is as follows :

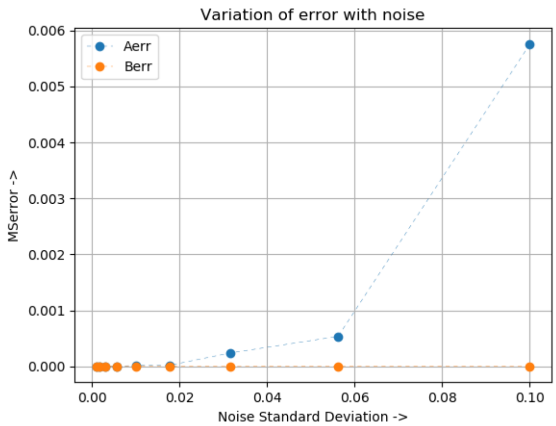

Figure 4: Error vs Standard Deviation

This plot does not give useful information between σ and ϵ, but when we do the **loglog** plot as below :

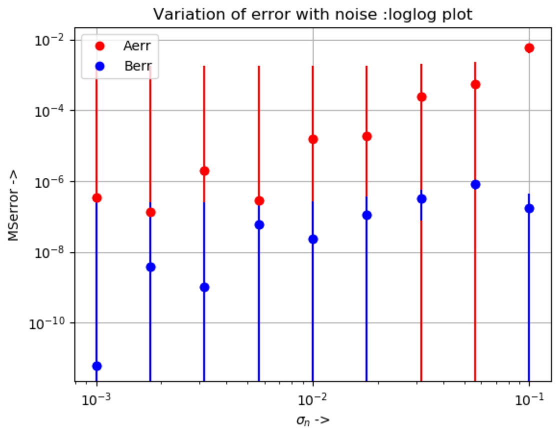

Figure 5: Error vs Standard Deviation loglog Plot

We can see an approximate linear relation between σ and ϵ. This is the expected result.

3.Conclusion

From the above procedure, we were able to determine that the **logarithm of the noise linearly affects** the **logarithm of the error** in the calculation of the **least error fit** for a given data