Assignment 3

Om Shri Prasath (EE17B113)

February 27, 2019

1 Aim of Assignment

In this assignment we aim to :

- Observe the error of trying to find the least error fit function for a data due to noise.
- Find the relation between the error and the noise.

2 Procedure

The function to be fitted is :

$$f(t) = 1.05J_2(t) - 0.105t$$

The true data is collected from the above function.

2.1 Creating noisy data

To create the noisy data, we add random noise to the function. This random noise (denoted by $n(t)$)) is given by the normally distributed probability function :

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

Where σ is generated in code implementation as :

```
sigma = logspace(-1,-3,9)
```

Thus the resulting noisy data will be of the form :

$$f(t) = 1.05J_2(t) - 0.105t + n(t)$$

Thus for 9 different values of sigma, the noisy data is created and stored in the ***fitting.dat*** file

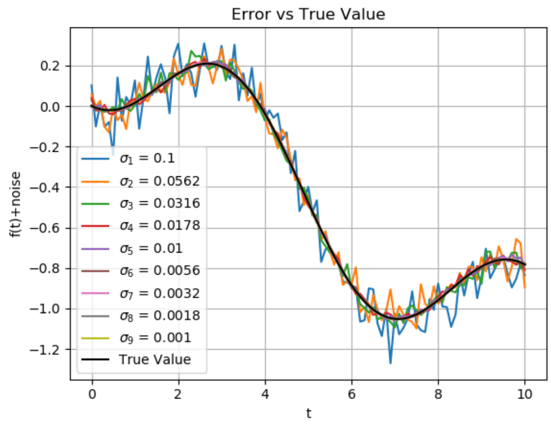

Figure 1: Noisy Data With the True Data

2.2 Analyzing the noisy data

The data is read into a Python code and is plotted using PyPlot. The output result looks as follows:
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 :

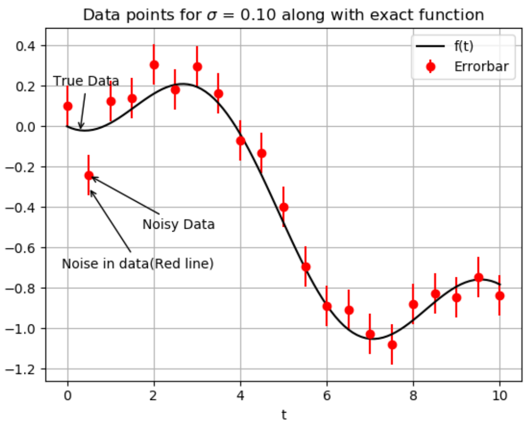

Figure 2: Noisy Data with Errorbar

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

2.3 Finding the best approximation for the noisy data

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

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

To find the coefficients A and B, we first try to find the mean square error between the function and the data for a range of values of A and B , which is given by :

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

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

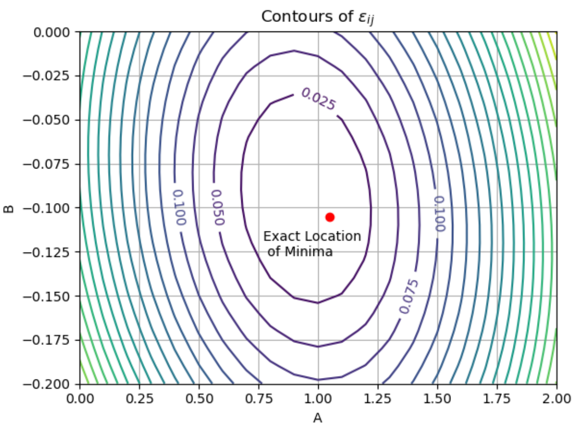

Figure 3: Contour Plot of ϵ_{ij}

We can see the location of the minima to be approximately near the orignal function coefficients Now 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).
The plot for the different noisy data is as follows :

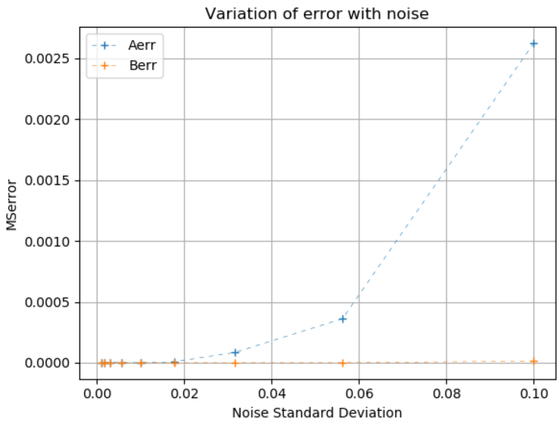

Figure 4: Error vs Standard Deviation

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

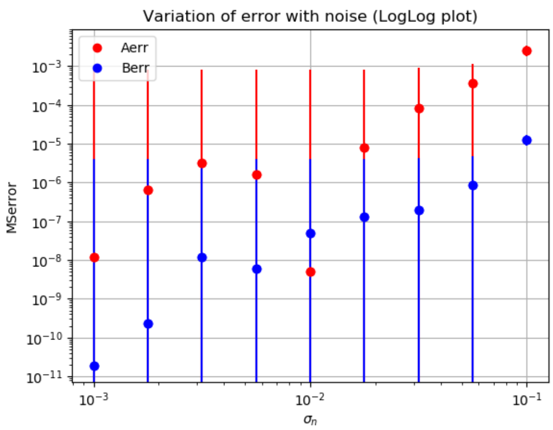

Figure 5: Error vs Standard Deviation loglog Plot

We can see an approximate linear relation between σ and ϵ. This is the required 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 calcuation of the **least error fit** for a given data