EE2703 : Applied Programming Lab
Assignment 3

Andapally Snehitha
EE20B008

February 18, 2022

Extracting and Visualizing the data

The given data was obtained by running the generate data.py script. The data contained 10 columns : the first column was time, and the other 9 columns were each riddled with different amounts of noise, with standard deviation uniformly sampled from a logarithmic scale. On plotting all the 9 columns, the following graph was observed :

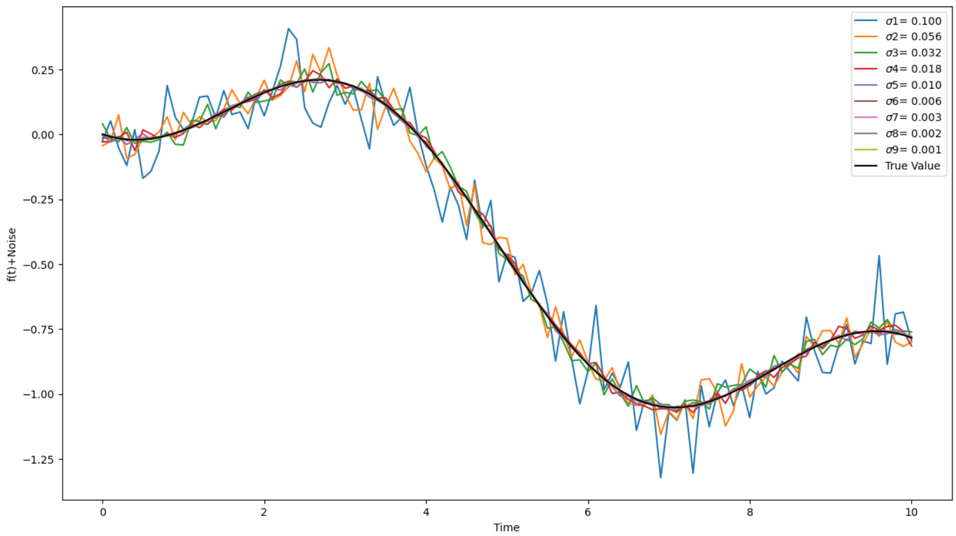

Figure 1: Plot of all data columns

The Function

Since we are aware of the actual function to be extracted from the data columns, the general shape of the function to fit the given data is known. A simple implementation does the required computation :

```
def g(t,A,B):
    return A*sc.jn(2,t)+B*t
```

Visualising noise - The Error bar plot

An error bar is a convenient way of visualising the uncertainty in the reported measurement. The error bars for the first data column are plotted using the error bar() function. The graph obtained by plotting every 5th data point with error bars and the original data is as follows:

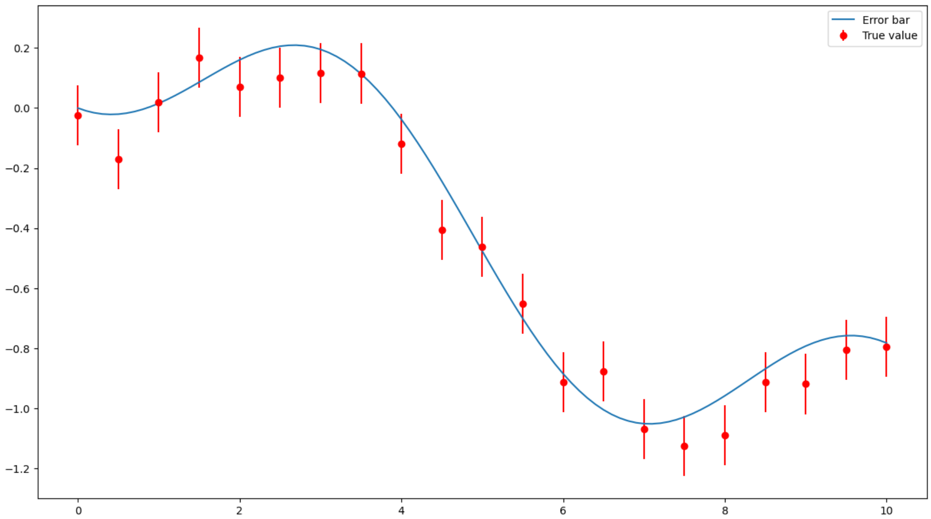

Figure 2: The Error bar Plot

Each point in the data column can be seen to be varying mostly within a width of the true value, except in a few cases. These are the points that most affect the prediction of the function parameters A and B .

The Matrix Equation

We can compare the results obtained using matrix multiplication and the user-defined function using the np.array equal() function. If this returns a *True* value, then they are equivalent.

Error Computation

The following code snippet is used to compute the error e in each data column:

```
E = np.zeros([21,21])
for j in range(len(A1)):
```

```
    for k in range(len(B1)):
        for l in range(len(f[:,0])):
            E[j][k] += (f[l][1]- g(f[l][0], A1[j], B1[k]))**2/101
```

y columns represents the data columns. The mean squared error for the first data column can now be plotted and the minima can be found to obtain the best estimate for A and B. The following code block does the needful:

```
cnt=py.contour(A1,B1,E, levels = 20)
py.clabel(cnt)
```

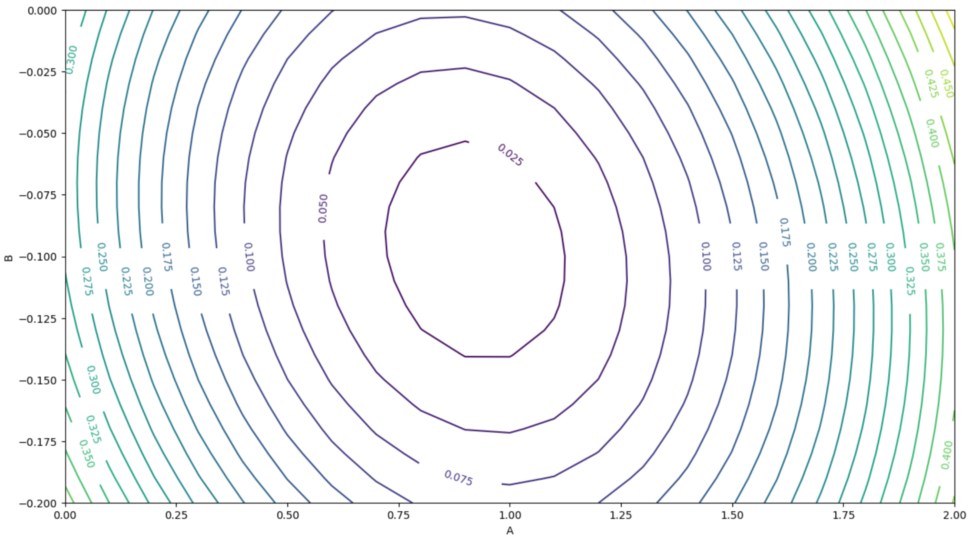

Figure 3: The Contour Plot

From the graph, the error function has one and only one minimum and it occurs at A = 1.10 and B = 0.10.

Error of A and B on a linear Scale

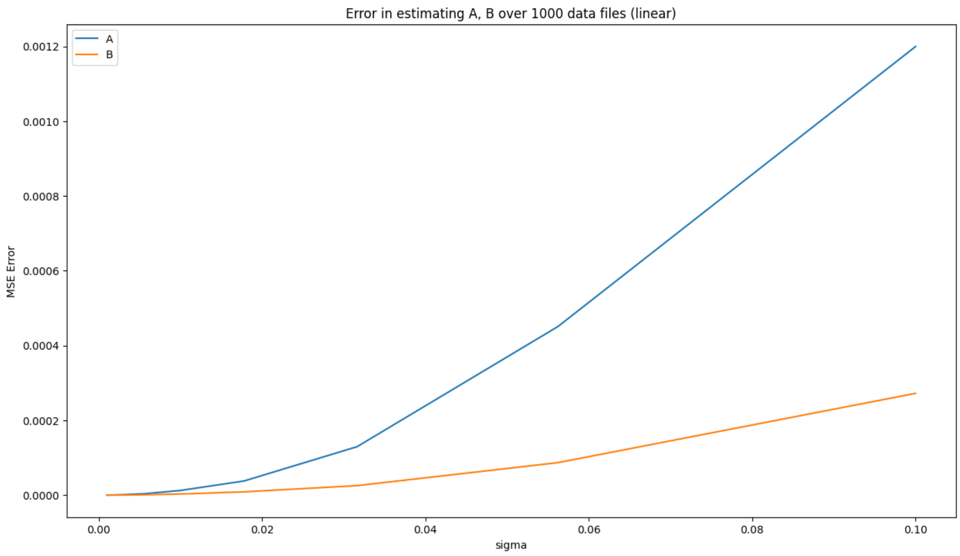

Figure 4: Error of A and B on a linear Scale

Error of A and B on a loglog Scale

Figure 5: Error of A and B on a loglog Scale

Conclusions

We have used the scipy library to estimate coefficents of a linear combination of 2 functions with varying amounts of noise. As the noise in the data used to estimate the linear combination increases, the error of the prediction also increases on a linearly on a loglog scale.