EE2703: Applied programming Lab
Assignment 3: Fitting Data to Models

Vishnu Varma V
EE19B059

March 3, 2021

1 Plotting the Extracted Data

The data to plot was obtained by running *generatedata.py* file. The First column of the data contains time and the remaining 9 columns contain the data points. These data points were obtained by adding noise which is normally distributed with sigma uniformly sampled on a logarithmic scale.

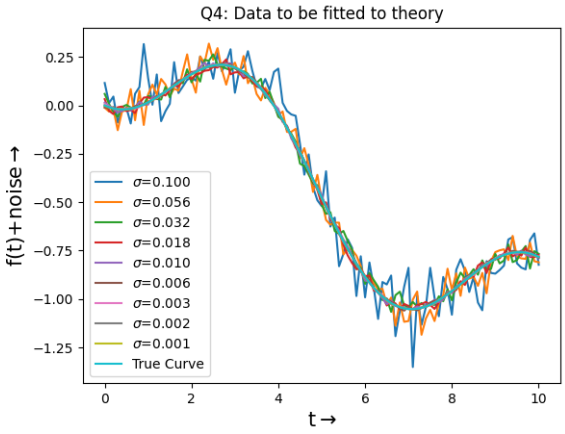

Figure 1: Data to be fitted to theory

2 Function Definition

The true function without any noise is given by

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

this function is realised using the following code which takes t, A and B as input and returns the function value

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

3 Errorbar Plot

Using this plot we can infer how noise affects the true value of the function the following graph shows the actual plot and when compared to the data given in column 1 how noise creates error in the value.

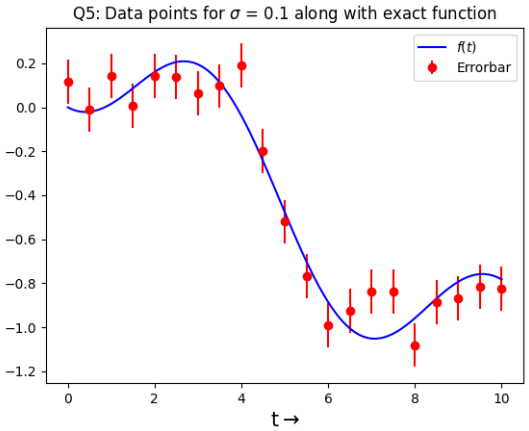

Figure 2: Data points for $\sigma = 0.1$ along with exact function

4 Equivalence of function

By Multiplying M matrix and parameter matrix we get a vector and the values of the function $g(t, A, B)$ gives us a vector, both the vectors obtained by matrix multiplication and function definition are compared and the result is true so the two vectors are equal.
The following code when executed output is True. So the two vectors are equal.
Code to check the equivalence:

```
fn_column = sp.jn(2,time)
M = np.c_[fn_column,time]
#known parameters to verify if both give the same answer
A = 1.05; B = -0.105
C = np.array([A,B])
#g_mul is obtained by multipying M and C
g_mul = np.matmul(M,C)
#g_func is obtained by the function definition
```

```
g_func = np.array(g(time,A,B))
```

5 Error function

To Calculate the mean squared error for each of the data columns we use for loops to parse through the data The following code calcualtes the error for each column of the given data:

```
for k in range(9):
        f = y_col[:,k]
        for i in range(21):
                for j in range(21):
                        e[i][j][k] = np.sum((f -np.array(g(time,A[i],B[j]
```

6 Contour Plot

Contour function of the matplotlib module is used to plot the contour. The mean squared error is plotted and the minima is found. argmin function of numpy module returns the index of minima for the flattened array. unravel function is used to get the loaction of minimum.
The Exact location and the location at which it is minimum is shown in the following graph:

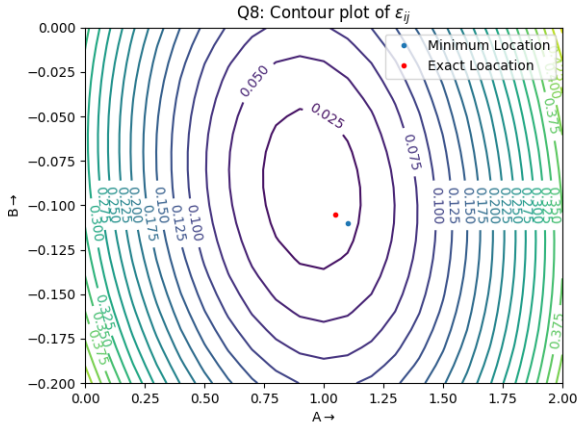

Figure 3: Contour plot of ϵ_{ij}

The Exact Location is A = 1.05 an B = -0.105
From the plot error function has only one minimum and it occurs at A = 1.10 and B = -0.110

7 Error in Estimation of Parameters : Linear Scale

On a linear scale : When σ_n is varied the corresponding vaiation in parameters A and B is shown using the following plot:

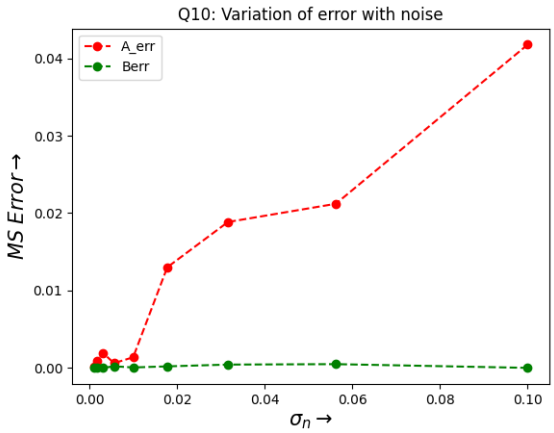

Figure 4: Variation of error with noise Linear Scale

The error estimate is non-linear with noise

8 Error in Estimation of Parameters : Log Scale

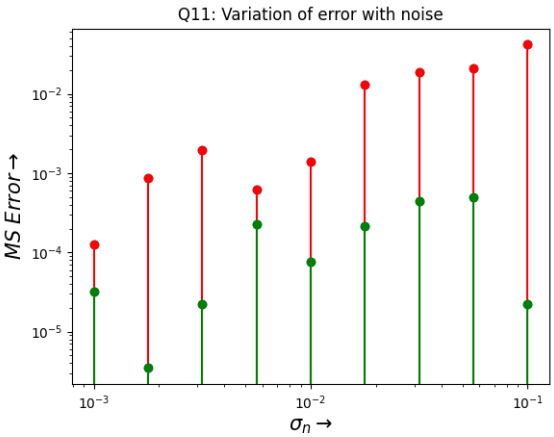

Figure 5: Variation of error with noise Log Scale

On a Log scale : When σ_n is varied the corresponding vaiation in parameters A and B is shown using the above plot:

We observe that the error in the plot varies with σ but not exactly linearly but variation is less drastic compared to the linear scale plot.

9 Conclusion

The data extracted which had variable noise was plotted along with the True curve the variation from the true curve with the data given in column1 is plotted in Errorbar Plot. The vector obtained from multiplication of function vector and parameter vector and the vector with values obtained directly from the function definition is equal so the output when they are compared is True. The contour plot of the mean squared error has only one minimum. Mean squared error vs σ is plotted in both linear and log scale the variations in log scale is comparitively more linear than that in linear scale.