

EE2703 : Applied Programming Lab
Assignment 3
Fitting Data to Models

Saurav Sachin Kale
EE19B141

March 5, 2021

Aim

The aim of this assignment is to

- Process data with noise
- Perform curve fitting to fit the data to the mathematical model
- Study the effect of noise on curve fitting by analysing error in the calculation of the coefficients in the mathematical expression for varying amounts of noise

Procedure

- On running “generate_data.py” (kindly ensure the python file and fitting.dat are in same directory) a set of data points are generated in the file “fitting.dat” for the equation

$$f(t) = 1.05J_2(t) - 0.105t + n(t) \quad (1)$$

where $n(t)$ is the amount of noise. The noise is given to be normally distributed, i.e., its probability distribution is given by

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

with σ , the standard deviation of noise given by

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

logspace is a python function which returns number spaces evenly in interval on a log scale.

The first column in data points file ”fitting.dat” has the value of t .

- Our mathematical model for this data is:

$$g(t; A, B) = AJ_2(t) + Bt \quad (2)$$

A and B are to be found so as to fit this curve to our data points.

- The true values of A and B are:

$$A = 1.05, B = -0.105$$

- The equation (2) can be interpreted as a matrix product in the following manner:

$$g(t; A, B) = \begin{pmatrix} J_2(t_1) & t_1 \\ \dots & \dots \\ J_2(t_m) & t_m \end{pmatrix} \cdot \begin{pmatrix} A \\ B \end{pmatrix} = M \cdot p \quad (3)$$

- M contains the values of $J_2(t)$ and the corresponding t values. p consists of the coefficients A and B
- Now we can find A and B by taking the mean squared error as follows:

$$\epsilon_{ij} = \frac{1}{101} \sum_{k=0}^{101} (f_k - g(t; A, B))^2 \quad (4)$$

- Plotting a contour plot of ϵ_{ij} vs. A and B and finding those values of A and B for which Error $\rightarrow 0$ gives the estimate of those values of A and B which fit the curve approximately to the data points.
- This estimate is found using the method of least squares. The python function below from the SciPy library is used:

```
scipy.linalg.lstsq(M,p)
```

(uses the method of least squares to find estimates of A and B)

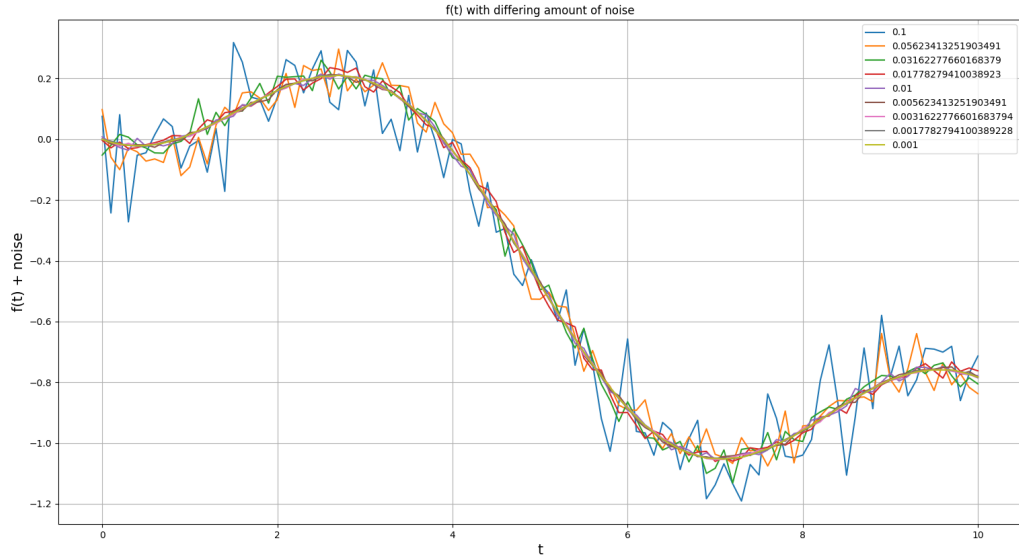


Figure 1: Plot of generated data to fit

Results

1. The plot of the data to be fitted with true curve highlighted:

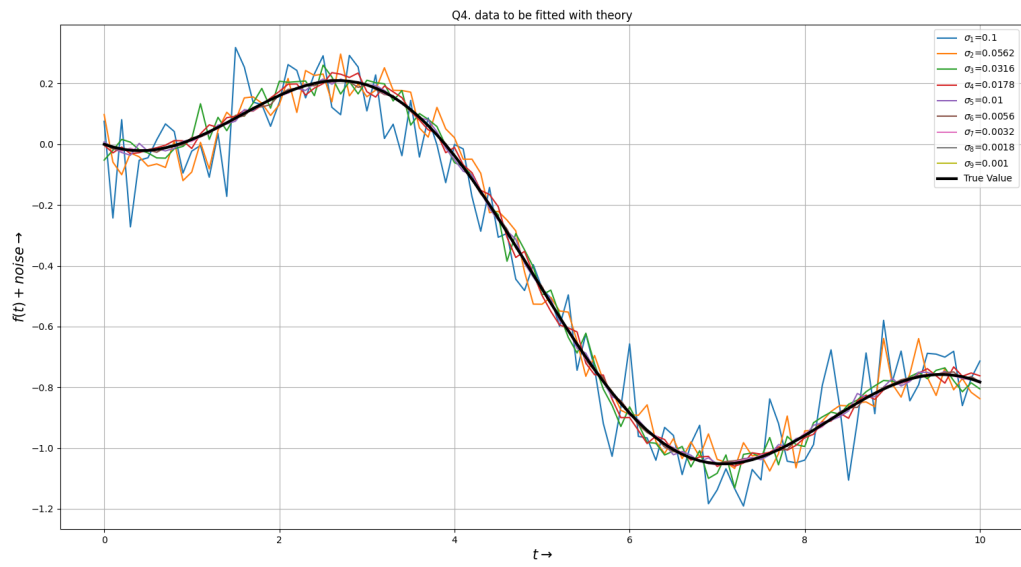


Figure 2: Plot of generated data and the true curve

2. Errorbar plot shows the deviation from the true value for noise whose standard deviation is 0.1

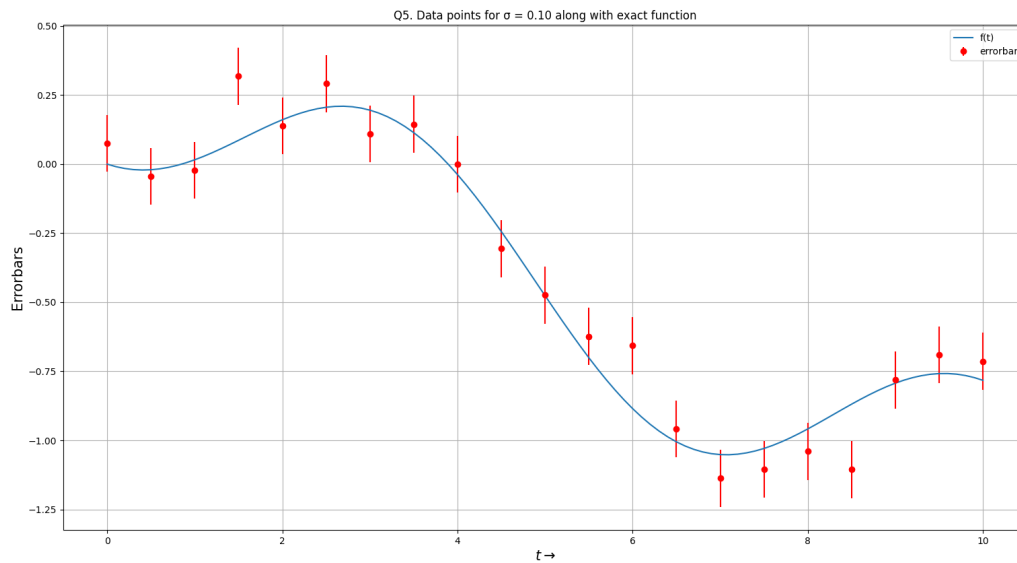


Figure 3: Errorbar plot

3. Finding the product of $M \cdot p$ and verifying it is equal to G . Furthermore, $Mp - G$ is a zero vector, as the output from the python program states:

```
Verifying Mp - G = 0
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0.]
```

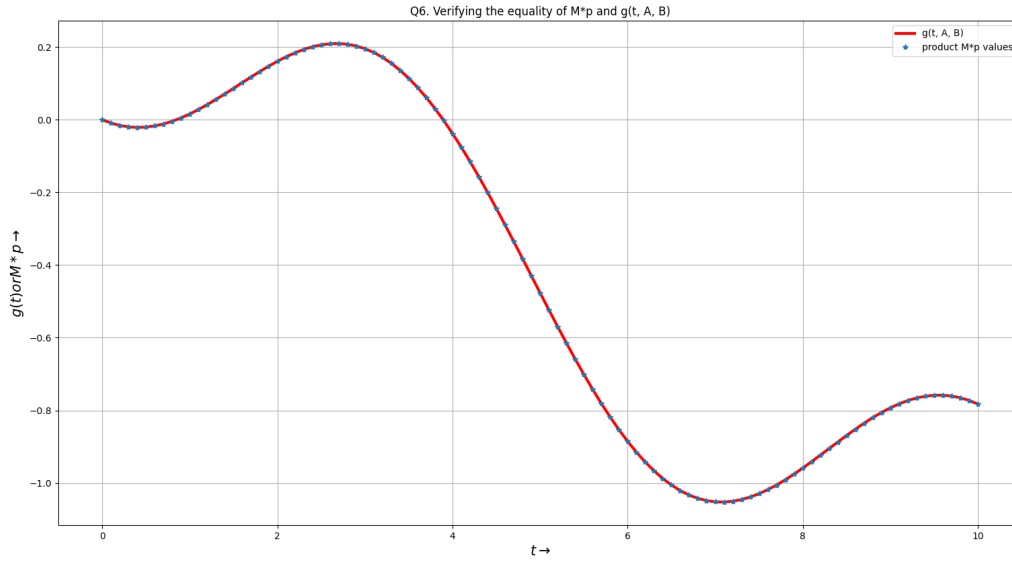


Figure 4: Verifying $Mp = G$ via a plot vs t

4. The error ϵ_{ij} was calculated by two for loops, one for subscript i and j . This implements equation (4) as follows:

```
E = np.zeros((len(A0), len(B0)))
f = y[:, i]
for i in range(len(A0)):
    for j in range(len(B0)):
        E[i, j] = ((f - g(x, A0[i], B0[j]))**2).mean(axis=0)
```

As k goes from 0 to 101, we take the mean of all the values in the numpy array, therefore the summation and division by 101 as in equation (4) is implemented.

5. Contour plot of the mean squared error vs. A and B

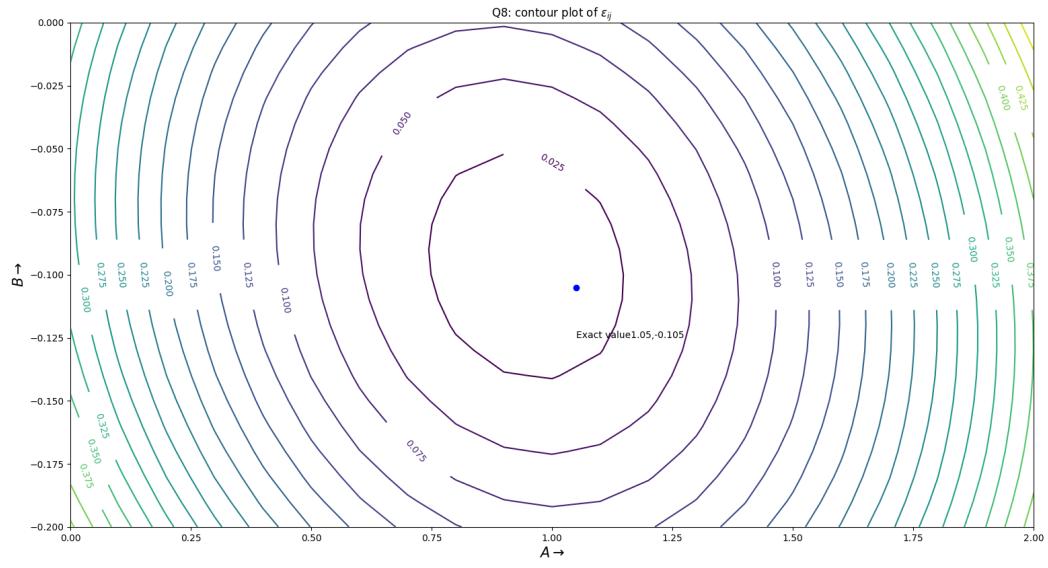


Figure 5: Contour Plot of Mean Squared Error vs A and B

the contours appear converge near the exact value $(1.05, -0.105)$, and has only one minima.

6. The variation of error in approximation with the standard deviation of the noise

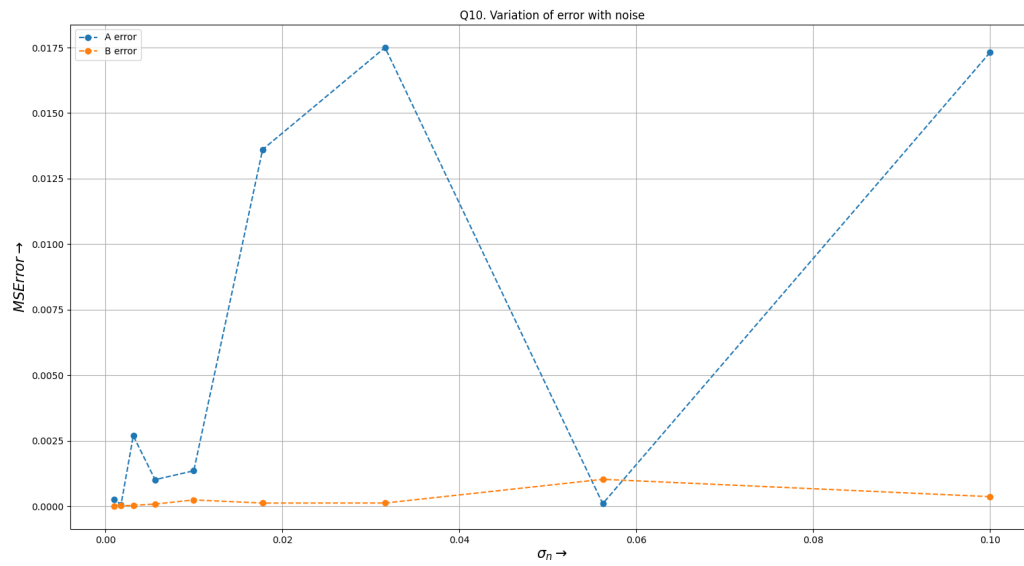


Figure 6: Plot for Error in estimation of A and B vs standard deviation in noise

As seen in the plot the error in estimation of A increases suddenly with standard deviation of noise, while the error in estimation of B remains nearly constant.

7. The loglog plot of variation of error in approximation with the standard deviation of the noise

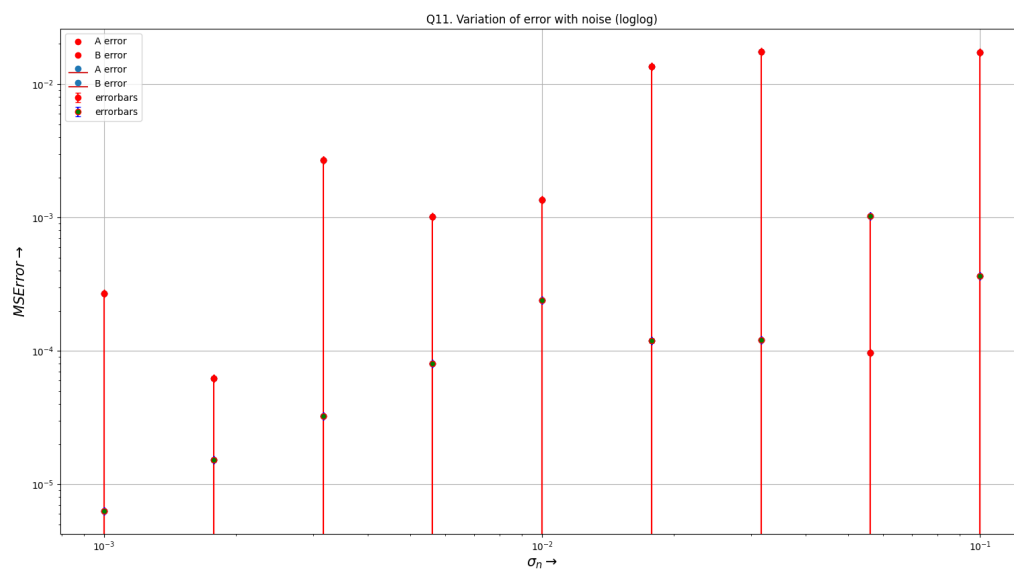


Figure 7: log-log Plot for Error in estimation of A and B vs standard deviation in noise

As seen in the graph, both error in estimation of A and B show linear dependency on standard deviation of noise.

Conclusions

By the minimization of Mean Squared Error in the data points using the method of least squares gives a very close estimation of A and B . The mean square error is within 2% when the standard deviation is 0.1.

The minima of the Mean Squared Error ϵ_{ij} in the contour plot of equation (4) provides the estimates of A and B for the data.

We observe that there is only one minima in the contour plot. This indicates that least squares method has only one optimal solution.

By experimenting with noise of varying standard deviations, we obtained the following results about A and B :

- B varies very little with the standard deviation of noise, while A varies significantly.
- Error in estimating A and B has linear variation with standard deviation of noise in the loglog scale.