APL Assignment No. 3

AYUSH RAJ
EE20B019

February 18, 2022

1 Abstract

This week's Python assignment will focus on the following topics:

- Reading data from files and parsing them
- Analysing the data to extract information
- Study the effect of noise on the fitting process
- Plotting graphs

2 Introduction

For this weeks assignment we are going to use the **Bessel Function** and **Gaussian noise** to study the variation of parameters of a model with changing **Standard Deviation**.

$$f(t) = A * J_2(t) + B * t + n(t)$$

Where, A = 1.05, B = -0.105, J_2 = Bessel function, n(t) = Noise function

We seek to find the relation between A,B and the standard deviation of the Gaussian noise.

3 Problems and Solutions

3.1 Generation Data

Data is generated using the script given with the assignment, **scipy** library is used for getting the **Bessel function**, and t contains 101 equally spaced numbers between 0 and 10, which are fed into the Bassel function and added with noise to generate the data used further.

3.2 Loading data

Using numpy's loadtxt funtion we loaded the data into a numpy matrix, the data had 10 columns and 101 rows, the first column had the values of time in them and the remaing columns had data produced using standard deviation for the noise.

```
data = np.loadtxt("fitting.dat")
t=data[:,0]
y=data[:,1:]
```

3.3 Plotting Data

The plotting of data in this assignment is done using **pylab's** plotting functions, which is nothing but the **Matplotlib' pyplot module**

3.4 Plotting the original function

We plot $f(t)$ without the noise using A = 1.05 and B = -0.105

```
def G(t,A,B):
    return A*jn(2,t)+B*t
g=G(t,A0,B0)
figure(0)
plot(t,g)
plot(t,y)
grid(True)
sigma=logspace(-1,-3,k)
legend(sigma)
labels = append(sigma, "True Value")
for i in range(k):
    labels[i] = "$\sigma" + "_" + str(i + 1) + "$" + "=" + str(round(float(labels[i]), 4))
legend(labels)
xlabel("t"+"$\\rightarrow$")
ylabel("f(t)+n"+"$\\rightarrow$")
title("Data to be fitted to theory")
show()

figure(1)
plot(t,g,label='actual graph')
stdev = std(y[:, 0] - g)
grid(True)
errorbar(t[::5],y[::5,0],stdev,fmt='ro',label='errorbar')
legend()
title("data points for"+"$\sigma$"+"=0.01 along with the exact function")
xlabel("t"+"$\\rightarrow$")
show()
```

Figure 1: Question 3&4

3.5 Plotting Error Bars

We plot the first column of data with error bars using the **errorbar** function with error equal to standard deviation.

We plotted every 5^th data item to make the plot readable, and we also plotted the exact curve to see how much the data diverges.

```
figure(1)
plot(t,g,label='actual graph')
stdev = std(y[:, 0] - g)
grid(True)
errorbar(t[::5],y[::5,0],stdev,fmt='ro',label='errorbar')
legend()
title("data points for"+"$\sigma$"+"=0.01 along with the exact function")
xlabel("t"+"$\\rightarrow$")
show()
```

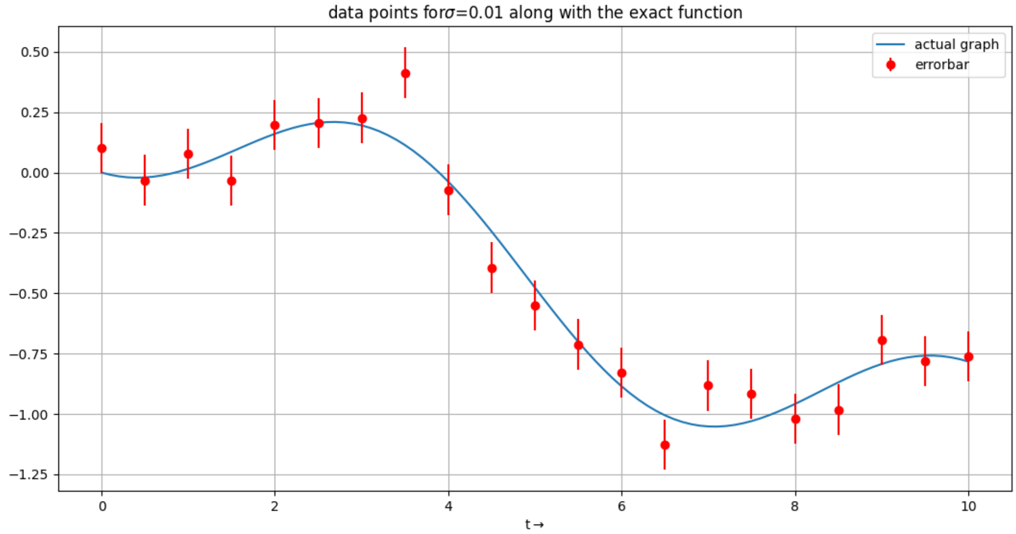

Figure 2: Question 5

3.6 Generation and Verification of M matrix

We generate the **M** matrix with accordance to the following equation:

$$g(t, A, B) = \begin{bmatrix} J_2(t_1) & t_1 \\ \dots & \dots \\ J_2(t_m) & t_m \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix}$$

We also show the $g(t, A_0, B_0)$ and $M \cdot \begin{bmatrix} A \\ B \end{bmatrix}$ are equal.

```
coulmn1=jn(2,t)
column2=t
M=c_[coulmn1,column2]
p=array([A0,B0])
print(allclose(g,dot(M,p)))
```

3.7 Calculation MSE for various combination of A and B

We calculate the MSE for every possible combination of A and B, where A and B are chosen from a set of 21 equally spaced numbers between 0 and 2 and between -0.2 and 0 respectively.

The error was calculated using the formula:

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

```
figure(2)
A = linspace(0, 2, 21)
B= linspace(−0.2, 0, 21)
Eij=zeros((len(A),len(B)))
fk = y[:, 0]
#calculate E for the ith column of data points
for i in range(len(A)):
    for j in range(len(B)):
        Eij[i, j] = ((fk−G(t,A[i],B[j]))**2).mean(axis=0)
```

3.8 Plotting Contour of Error

We plot the coutour using the contour funciton and we also used in inside the clavel function for labeling it.

```
t1, y1 = meshgrid(A, B)
levels = linspace(0.025, 0.5, 20)
contour_lines= contour(t1, y1, Eij,levels)
clabel(contour_lines, levels)
p=scipy.linalg.lstsq(M,g)
plot(p[0][0], p[0][1], color="red", marker="o")
title("contour plot of Eij")
xlabel("A"+"$\\rightarrow$")
ylabel("B"+"$\\rightarrow$")
show()
```

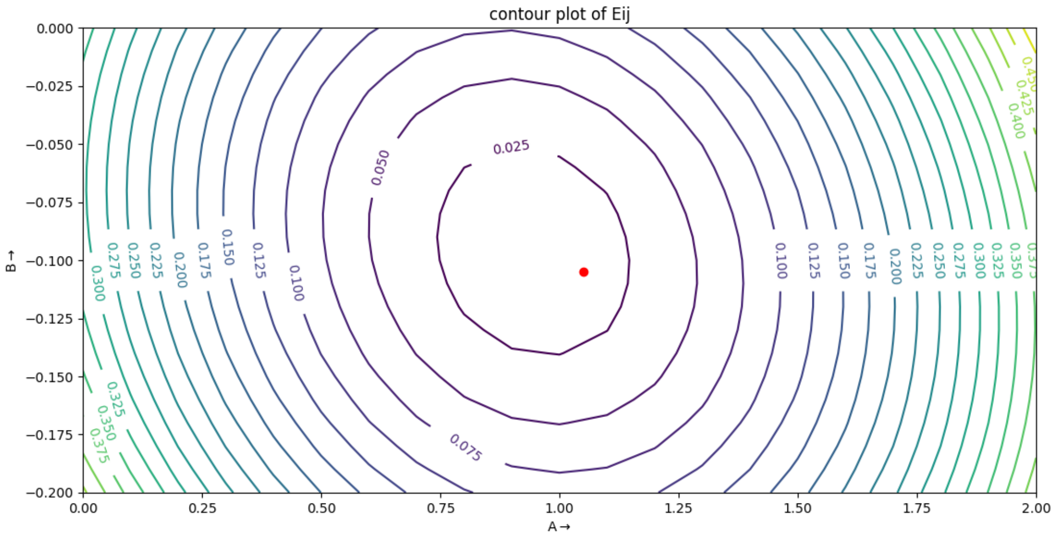

Figure 3: Question 8

3.9 Estimating A and B

A and B are chosed such that they minimize the following expression:

$$M \cdot \begin{bmatrix} A \\ B \end{bmatrix} - b$$

This is done using lstsq function in **scipy linalg module**

```
figure(3)
error_in_A=[]
error_in_B=[]

for i in range(9):
    temp=scipy.linalg.lstsq(M,y[:,i])[0]
    error_in_A.append(abs(temp[0]-A0))
    error_in_B.append(abs(temp[1]-B0))
```

3.10 Plotting Error in A and B vs σ in Linear scale

We use **plot** function for this and plot a slightly faded dashed line connecting the values.

```
plot(sigma, error_in_A, linestyle="--", marker="o", label="error in A")
plot(sigma, error_in_B, linestyle="--", marker="o", label="error in B")
grid(True)
title(" Variation of error with noise")
xlabel("Noise standard deviation"+"$\\rightarrow$",size=15)
ylabel("$MS Error\\rightarrow$",size=15)
legend()
show()
```

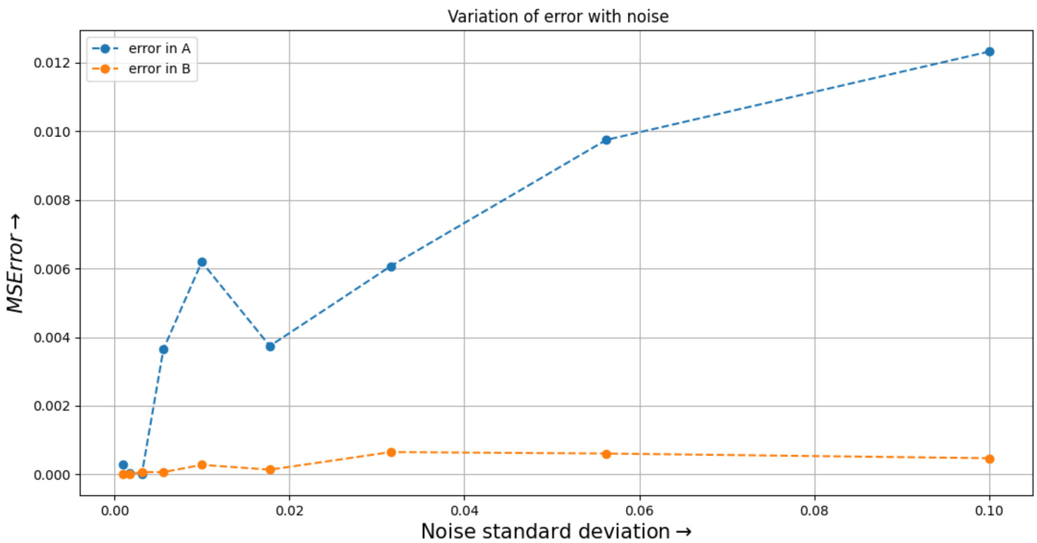

Figure 4: Question 10

3.11 Plotting Error in A and B vs σ in Log scale

We use **errorbar** function for this and plot error bars for the values

```
figure(4)
loglog(sigma, error_in_A,linestyle=" ", marker="o", label="error in A",markerfac
loglog(sigma, error_in_B,linestyle=" ", marker="o", label="error in B",markerfac
stem(sigma, error_in_A, 'b',use_line_collection = True,markerfmt=" ")
stem(sigma, error_in_B, 'r', use_line_collection = True,markerfmt=" ")
grid(True)
title("Variation of error with noise")
xlabel("$\sigma_n\\rightarrow$")
ylabel("$MS Error\\rightarrow$")
legend()
show()
```

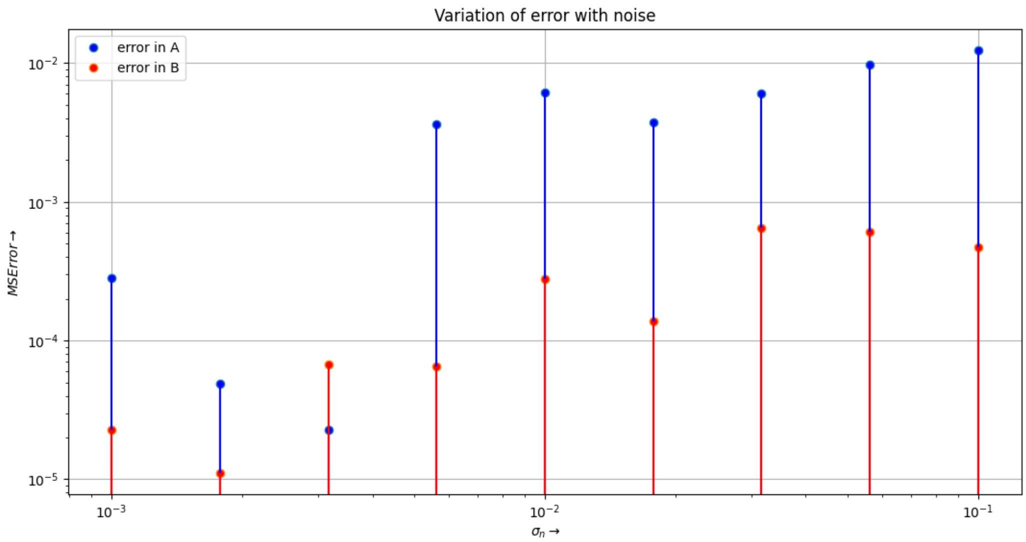

Figure 5: Question 11

4 Conclusion

From the plots we can see that the errors in estimated A and B increases with the standard deviation of the Gaussian noise in the data. We also see that the increase in error in log-log plot is somewhat linear.