

EE2703 Assignment3

Anvith Pabba EE19B070

5th March 2021

1 Abstract

This week's Python Assignment focuses on the following topics:

- Analysing the data to extract information
- Studying the effect of noise on the fitting process
- Plotting the relevant graphs

2 Introduction

Let us model our data in the following form:

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

where A, B are the parameters to be fitted and $J_2(t), t$ are the given functions in variable t . Here,

- $A = 1.05$
- $B = -0.105$
- $n(t)$ is the function of the added noise
- $J_2(t)$ is Bessel's function

3 The Assignment

3.1 : Generating the data

We generate the required data by running the `generate_data.py` script. This gives us a `fitting.dat` file which contains 10 columns and 101 rows. In this file, the first column is time and the remaining columns are the corresponding data.

3.2 : Loading the data

We use the built in `loadtxt` command that unpacks the `fitting.dat` (or any given input file) into a new variable, `data`. This data is loaded column wise.

3.3 : Defining the function

We can define the function $g(t, A, B)$ in python using the following snippet:

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

3.4 : Plotting the curves:

We plot the exact value of the function, along with the 9 other noisy plots that we extracted from the generated data. These curves together make up the 'Figure 0' plot.

3.4.1 : Code Snippet

We can plot them using the following python code:

```
ylabel('$f(t)+noise$',size = 15)
xlabel('$t$',size = 15)
title('Figure 0')
for k in range(9):
    labels = '\u03C3' + subscripts[k] + '=' + str(np.round(sigma[k],3))
    plot(data[0],data[k+1],label = labels)
legend(loc = 'upper right')
plot(data[0],g(data[0],1.05,-0.105),color = 'black', label = 'f(t)')
legend(loc = 'upper right')
grid(True)
show()
```

3.4.2 : Plot of Figure 0

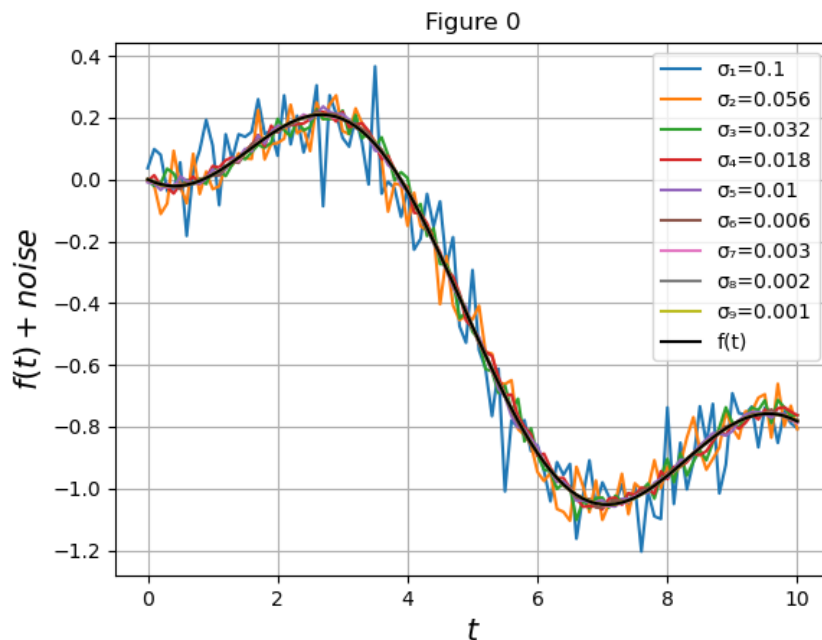


Figure 1: True and noisy plots

3.4.3 : All the different labels

- ylabel, xlabel : these label the y-axis and the x-axis
- title : gives a title to the plot
- labels : gives a label of each plot that is shown in the legend
- color : sets the color of the plot

- `plot()` : is the function used to plot the graph

3.5 : The Errorbars plot

Plotting the Errorbars gives us a visual representation of the error between a noisy plot and an exact plot. The picture below plots the Errorbar at every 5th data point.

3.5.1 : Code Snippet

```
#plotting the errorbar of noise vs time
xlabel('$t$',size = 15)
title('Q5: Data points for \u03C3=0.100 along with the exact function')
errorbar(data[0][::5],data[1][::5],sigma[0],fmt='ro',label='Errorbar')
#errorbar plot, standard deviation is 0.1 (== sigma[0])
plot(data[0],g(data[0],1.05,-0.105),label = 'f(t)', color= 'black' )
legend(loc = 'upper right')
grid(True)
show()
```

3.5.2 : Plot of Errorbars

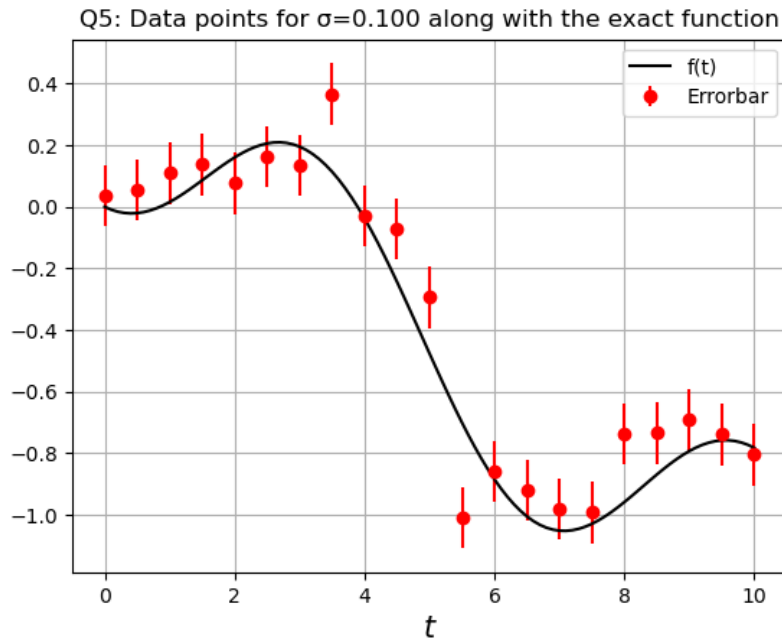


Figure 2: Errorbars plot

3.6 : Creating the required Matrices and checking for equivalence

We generate 2 matrices, M and p , such that :

$$M = \begin{pmatrix} J_2(t_1) & t_1 \\ J_2(t_2) & t_2 \\ \dots & \dots \\ J_m(t_1) & t_m \end{pmatrix} \quad (1)$$

and :

$$p = \begin{pmatrix} A \\ B \end{pmatrix} \quad (2)$$

We then find the dot product of M and p and see if that is equal to the values generated from the function $g(t,A,B)$ that we defined before.

3.6.1 code generating M

```
J = []
for t in data[0]:
    J.append(jn(2,t))
M = c_[J,data[0]]
```

3.6.2 code generating p

```
A0 = 1.05
B0 = -0.105
p = [[A0],[B0]]
```

3.6.3 Checking equivalence of two matrices

To check the equivalence of the 2 obtained matrices, RHS that we get from $g(t,A,B)$ and LHS from $M.p$ (M dot product with p), I have used the `array_equal` command in the numpy dictionary. This command return **True** when both matrices are equal and **False** when both are not. We then print out the output.

3.7 Mean squared error and the contour plot

To check the error between the generated A,B and the exact A,B , we can use the Mean squared error. We calculate the Mean squared error as follows:

$$\varepsilon_{ij} = \sum_{k=0}^{101} (f(k) - g(t_k, A_i, B_j))^2 \quad (3)$$

3.7.1 : code

```
for i in range(21):
    for j in range(21):
        for k in range(len(data[0])):
            e[i][j] += (1/101)*((data[1][k]-g(data[0][k],A[i],B[j]))**2)
```

3.7.2 : Contour plot

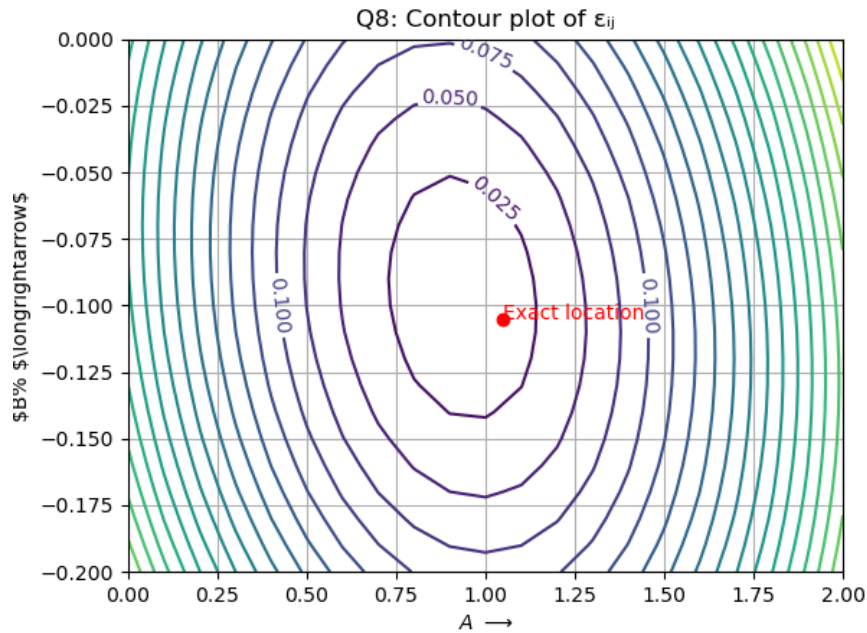


Figure 3: Contour plot of Errors

3.7.3 : Observations

We can see that there exists only 1 minimum (we can also see this through the code below which tells us the number of minimums, and as expected, the number of minimums is one)

3.7.4 : Code to check number of minimums

```
error_min = 100
for i in range(21):
    for j in range(21):
        if e[i][j] < error_min:
            error_min = e[i][j]
```

3.8 Calculating the best fit A and B and plotting the error

We can find the best fit A and B by minimising,

$$|M.(A_{estimate}, B_{estimate}) - (given_data)| \quad (4)$$

To do this, we can use the built in `lstsq()` function from `scipy.linalg` which gives us the direct estimates of A and B.

3.8.1 : Answer to asked question

From the plot below, it is evident that the graph isn't linear with time. So No, the error does NOT linearly vary with noise.

3.8.2 : Plot of error of A and B on the linear scale

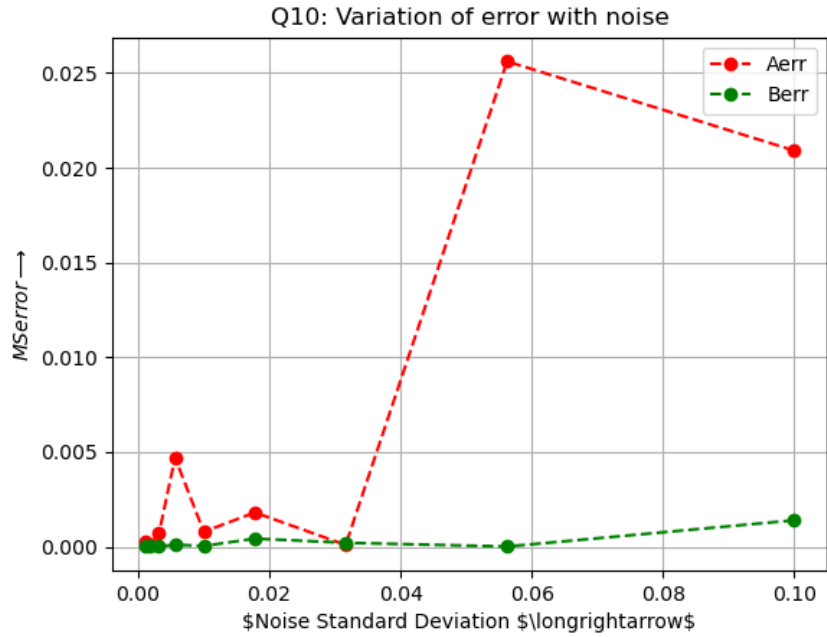


Figure 4: Error of A and B

3.9 Plotting the above error in loglog

We just plot the same data as above, but we change the scale of the x and y axis to the log scale, by using the `xscale` and `yscale` commands

3.9.1 : Answer to asked question

In the graph below, we can see that the error is almost linear. In theory, the loglog plot is linear. So Yes, the errors vary linearly in the loglog plot. As the sigma varies exponentially, this linear plot tells us that the noise also increases exponentially.

3.9.2 : loglog scale plot



Figure 5: Error of A and B in loglog scale

4 Learning

What I've learnt from this Assignment.

- How to extract data using loadtxt
- How to use lstsq to get the best estimates
- How to plot various types of graphs and learnt about the various labels that can be manipulated

5 Conclusion

- The error plot of A and B varies linearly in the loglog scale
- There is only 1 minima in the contour graph

-THE END-
EE19B070
Anvith Pabba
9381040535