EE2703 : Applied Programming Lab
Assignment 3

Harisankar K J
EE20B043

18 February 2022

Abstract

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

Tasks

0.0.1   Q1:Generation of Data

Data generated using the generate script given is stored into the 'fitting.dat'
file. the data contains the output of a function(a variation of Besel Function)
along with noise.

0.0.2   Q2:Loading the data into a matrix

The data values in 'fitting.dat' is extracted and stored in a matrix.
Code:

```
yf = np.loadtxt("fitting.dat")
(N, k) = yf.shape
t = yf[:, 0]
```

0.0.3   Q3:Plot of the Data to be Fitted

Matplotlib is used the plot the values.
Code:

```
scl = logspace(-1, -3, 9)
figure(0)
for i in range(1, k):
    plot(t, yf[:, i], label=("sigma = " + str(round(scl[i-1], 4))))
xlabel(r'$t$', size=20)
ylabel(r'$f(t)+n$', size=20)
title(r'Plot of the data to be fitted').
```

0.0.4 Q4:Plotting Original Function with A=1.05 and b=-0.105

The original function is defined and plotted.
Code:

```
def g(tk=t, A=1.05, B=-0.105):
    return A*sp.jn(2, tk)+B*t
y = g()
plot(t, y, label='true')
legend()
grid(True)
show()
```
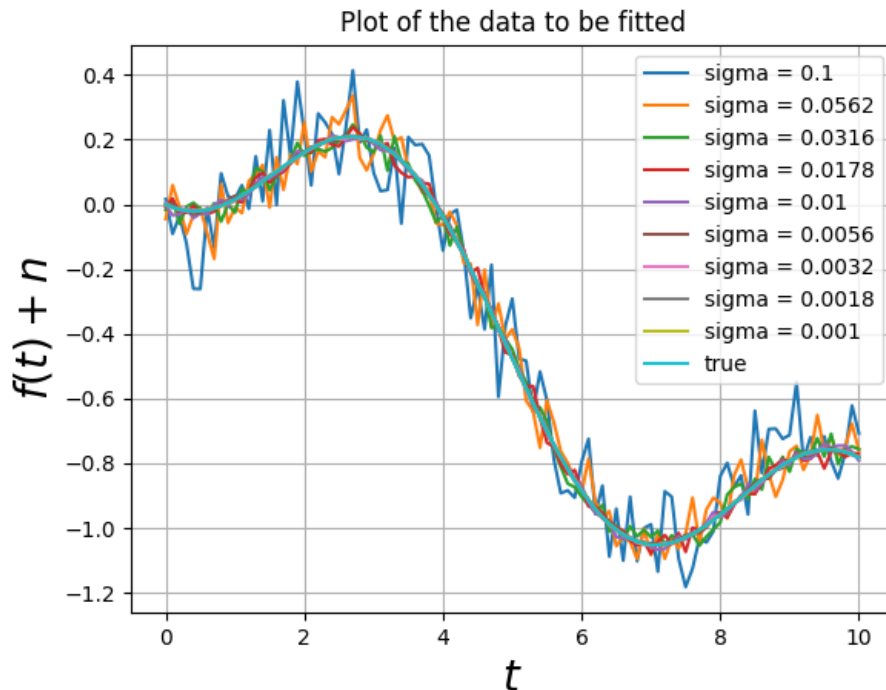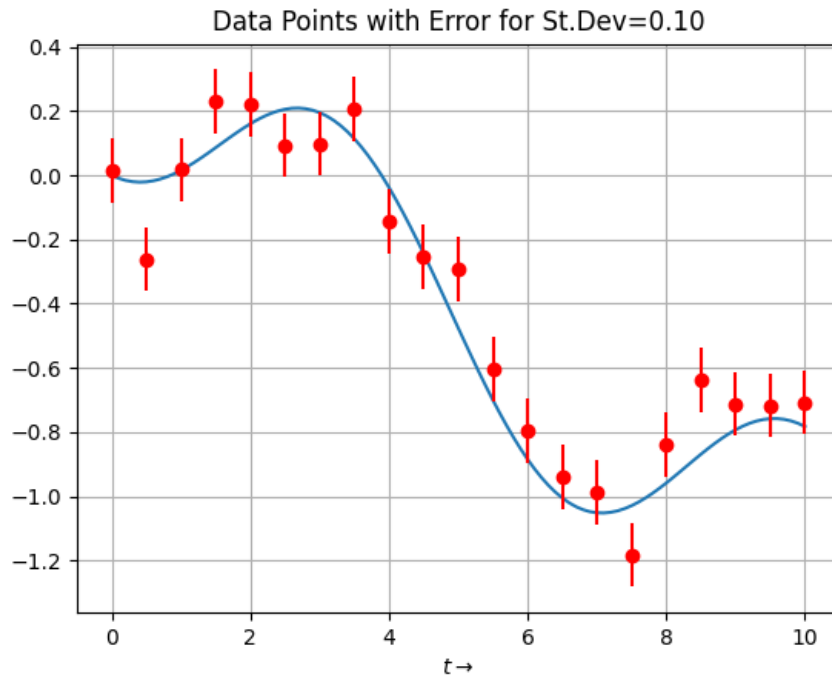

Plot of All the Data Values along with true value.

0.0.5 Q5:Plotting Error-bars

```
figure(1)
plot(t, y, label='true')
errorbar(t[::5], yf[:, 1][::5], scl[0], fmt='ro')
xlabel('$t\\rightarrow$')
title('Data Points with Error for St.Dev=0.10')
```

```
grid(True)
show()
```


Data Points with Error for St.Dev=0.10

Error-bar Plotting

0.0.6 Q6:Comparing Data values of Generated Matrix and True Function Data Matrix

Data value matrix generated and the true function data value matrix compared.

```
y1 = sp.jn(2, t)
M = c_[y1, t]
AB = array((1.05, -0.105))
if allclose(g(), dot(M, AB)):
    print("The values of the matrices match")
else:
    print("matrix values no  similar")
```

0.0.7 Q7:Generating Errors for different 'A' and 'B' and Finding Mean Square Error

```
n = 21
A = linspace(0, 2, n)
B = linspace(-0.2, 0, n)
eps = np.zeros((n, n))
for i in range(n):
    for j in range(n):
        eps[i][j] = mean(square(yf[:, 1]-g(t, A[i], B[j])))
```

0.0.8 Q8:Plotting Contours of Mean Square Error

```
figure(2)
pl = contour(A, B, eps, levels=20)
xlabel('$A\\rightarrow$')
ylabel('$B\\rightarrow$')
title(r'Contour Plot')
clabel(pl)
grid(True)
show()
```
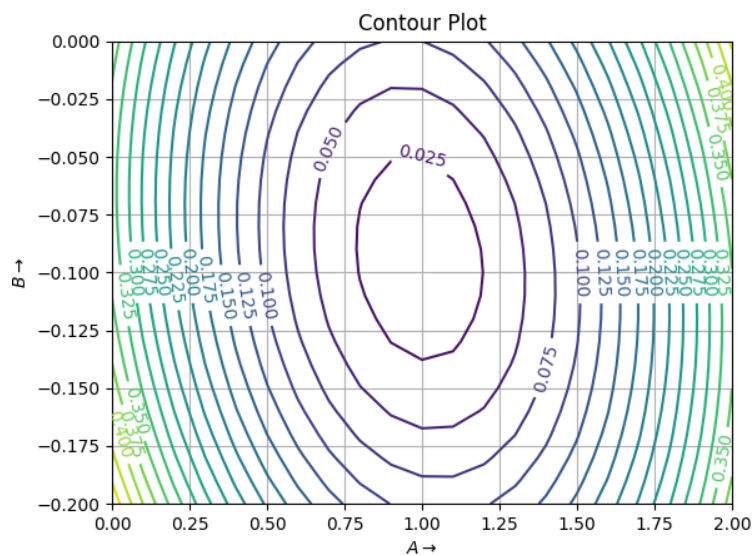

MS Error Contour

0.0.9   Q9:Estimating A and B

The value of A and B as a matrix AB is estimated by using the data matrix
from Q7.
Code:

```
ex = np.zeros((2, 1))
ex = scipy.linalg.lstsq(M, y)[0]
```

0.0.10   Q10:Error of A and B in Linear Scale

The error in estimate is found out in linear scale and its variation with
noise standard deviation is plotted.
Code:

```
fit = np.zeros((k-1, 2))
for i in range(k-1):
    fit[i] = scipy.linalg.lstsq(M, yf[:, i+1])[0]
Ae = np.zeros((k-1, 1))
Be = np.zeros((k-1, 1))
for i in range(k-1):
    Ae[i] = square(fit[i][0]-ex[0])
    Be[i] = square(fit[i][1]-ex[1])
figure(3)
plot(scl, Ae, label='A')
plot(scl, Be, label='B')
xlabel('Noise standard deviation')
ylabel('MS error')
title('Error in Estimate')
legend()
grid(True)
show()
```
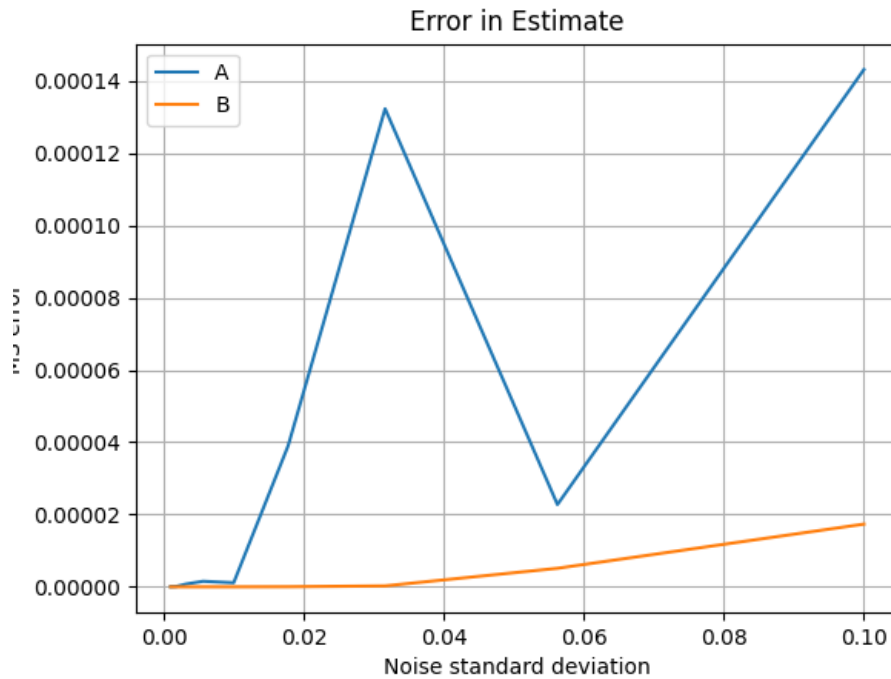
Error on linear scale.

0.0.11   Q11:Error of 'A' and 'B' on log-log Scale

Re-plotting the error in A and B in log-log scale.
code:

```
figure(4)
loglog(scl, Ae[:, 0], 'ro', label='A error(in logscale)')
loglog(scl, Be[:, 0], 'go', label='B error(in logscale)')
errorbar(scl, Ae[:, 0], std(Ae[:, 0]), fmt='ro')
errorbar(scl, Be[:, 0], std(Be[:, 0]), fmt='go')
xlabel('Noise Standard Deviation')
ylabel('MS Error (in logscale)')
title('Error in Estimate (in logscale)')
legend()
grid(True)
show()
```
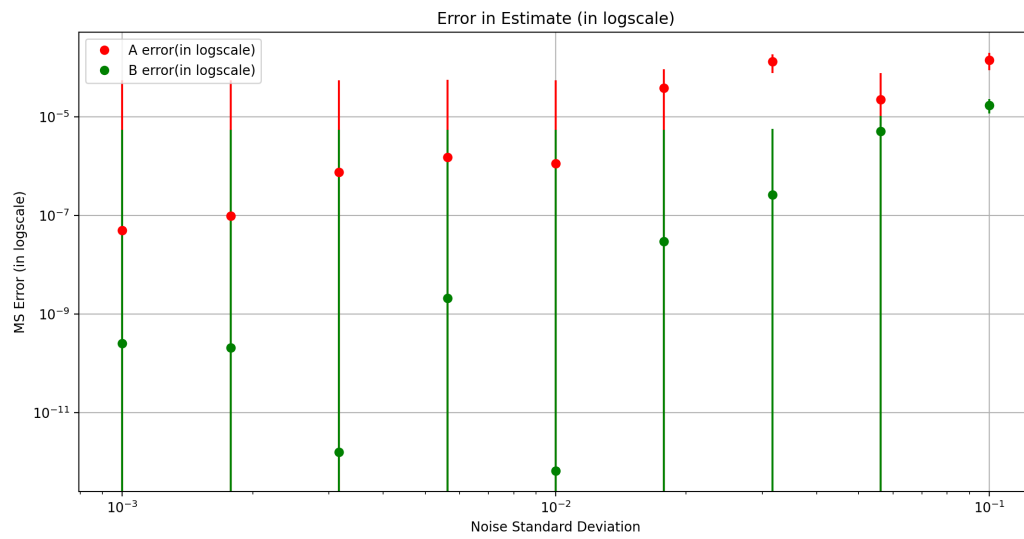
Error on log-log Scale.

Conclusion

We used SciPy to analyse and estimate the coefficients of a function with noise values in the generated output of the function. We also understood that as noise vales in the data increases, the value of error in estimation of coefficients increases.