Assignment 3

Kathir Pagalavan EE20B056

February 16, 2022

1 Loading data and graphing

The generate_data.py script generates fitting.dat file which contains nine columns each being a noised bessel function, each having a uniformly sampled standard deviation value of noise from a logarithmic scale.

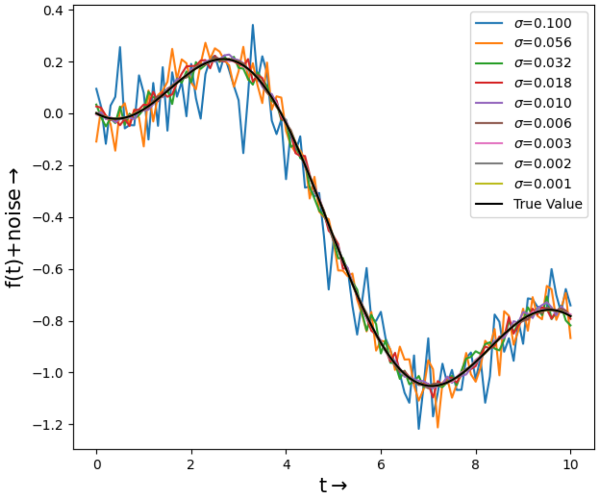

Figure 1: Plot of 'fitting.dat'

2 Errorbar plot

Plot of errorbars of data in first column with every 5th data item for readability and original data. Each point in the data column is varying mostly within a σ width of the true value.

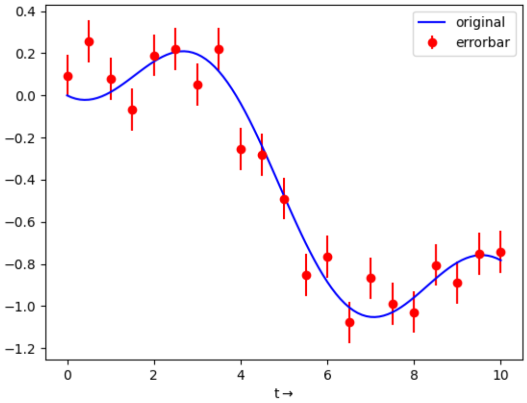

Figure 2: The errorbar plot

3 Error calculation

The error is calculated and stored in the array e.

$$\epsilon_{ij} = \frac{1}{101}\sum_{k=0}^{101}(f_k - g(t_k, A_i, B_j))^2) \tag{1}$$

Root mean squared error for the first data column is plotted and minima is identified to get the best estimate of A and B.

```
pylab.contour(A,B,e[:,:,0])
a = np.unravel_index(np.argmin(e[:,:,0]),e[:,:,0].shape)
pylab.plot(A[a[0]],B[a[1]],'o',markersize=3)
```

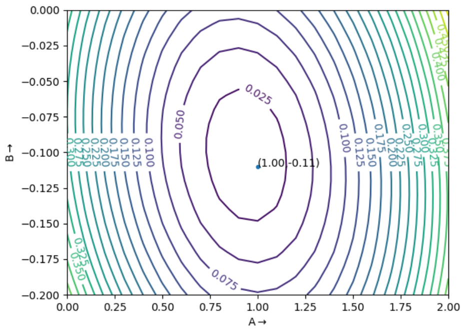

Figure 3: The contour plot

The np.argmin() function returns the index of minima for the flattened array, and the np.unravel index() function is used to get the location of the minimum in the original array.

As we can see in the graph, the error function has one minimum and it occurs at A = 1.10 and B = -0.11

4 Error in estimation of A and B

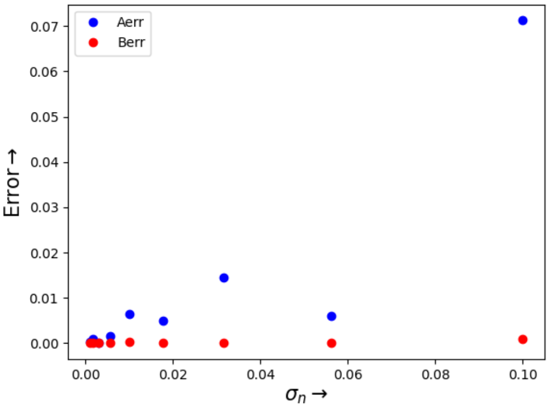

Figure 4: Error vs σ

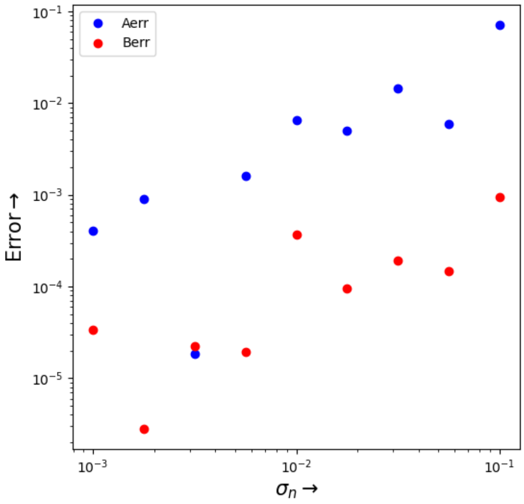

Figure 5: Error vs σ(log scale)

Error vs Noise for the first plot was non linear. The graph becomes approximately linear in log scale.

5 Conclusion

The data with some noise is extracted and the best possible estimate for the underlying model parameters is found by minimizing the mean squared error. We can see that the error is approximately linear with σ in the log scale.