PyLab 3 – Nonlinear Fitting Methods I

October 11, 2018 – Ayush Pandhi (1003227457) and Xi Chen (1003146222)

1 Abstract

The radioactive decay emission of an isotope follows an exponential model between intensity and time. An approximation of this relationship is modelled using both a linear and nonlinear curve fit. Both models prove to be a reasonably good fit for the data but the nonlinear model is slightly more accurate as its computed half-life for the gamma emission, 162 $\pm$ 0.8 seconds, is closer to the expected 156 seconds. It is also found that χ^2_{linear} = 2.3 and $\chi^2_{nonlinear}$ = 1.2, thus solidifying the previous findings with a statistical analysis.

2 Introduction

The purpose of this experiment is to determine the half-life of the gamma emission from Ba-137m and accurately fit it to two curve fit models. The theoretically predicted pattern for its decay is an exponential model described by:

$$\mathrm{I(t)} = I_0 0.5^{t/t_{halflife}} \quad or\ equivalently \quad I(t) = I_0 e^{-t/\tau}$$

Where $I(t)$ is the intensity of radiation at time t, I_0 is the initial intensity at time t = 0s, τ is the mean lifetime of the isotope being observed and $t_{halflife}$ is the half-life time of the observed isotope. We aim to model the decay counts by both linear and nonlinear fitting; the linear fitting involves turning the exponential expression into a linear equation in terms of the logarithm of each term. The nonlinear model will directly use the exponential form as above to fit the collected data and thus is expected to be a better fit for the data. Throughout the analysis, background radiation within the laboratory environment is considered and removed as noise in the data reduction process.

3 Methods and Materials

The materials utilized for this experiment were: a Geiger counter, a generator, acid, metastable Barium (Ba-137m), Cesium (Cs-137), and the Jupyter Notebooks software.

4 Experimental Procedure

A Geiger counter was setup to measure the rate of radioactive emissions from the metastable Barium (Ba-137m) material at 20 second intervals for 60 total samples. First, Cesium (Cs-137) is placed inside the generator and then acid is allowed to flow through. Thus, the Cesium (Cs-137) undergoes beta decay into the Barium (Ba-137m); the half-life of this beta decay is known to be 30.1 years. The acid proceeds to wash out the metastable Barium (Ba-137m) and causes decays by gamma emission. The half-life of this emission is known to be 2.6 minutes (156 seconds) at 662 keV. The Geiger counter then measures the rate of this gamma emission and the data is collected into a .txt file. Additionally, the Geiger counter is run to measure the background emission of the laboratory environment and this data is also stored in a .txt file.

5 Results

Before fitting the data to curve fit models, the mean background emission must be removed as noise from the gamma emission to discern its true rate of decay. This is done by computing the mean value from the background emission data file and subtracting it from each individual data point of the gamma emission data set. Using the modified emission data, a linear and non-linear curve fit is generated with scipy's "curve_fit()" module. The linear modelling is based on the function f, where f is defined as a general linear formula:

$$f = ax + b$$

For the curve fit on emission data, the function f is the logarithmic intensity (base 2), the independent variable x is time (in seconds), the slope a is thus the reciprocal of the half-life time, $t_{halflife}$ and b is the logarithmic decay count of the initial intensity (base 2). Therefore, rewriting the intensity of radioactive decay equation to be representative of the linear model specifically applied to our data gives:

$$\log_2(I(t)) = \frac{t}{t_{halflife}} + \log_2(I_0)$$

Where $I(t)$ is the intensity of radiation at time t and I_0 is the initial intensity at time t = 0s. Plotting the linear model fit with estimated parameters, (a = 1.0, b = 0.0), produces a fit that becomes progressively more inaccurate over time.

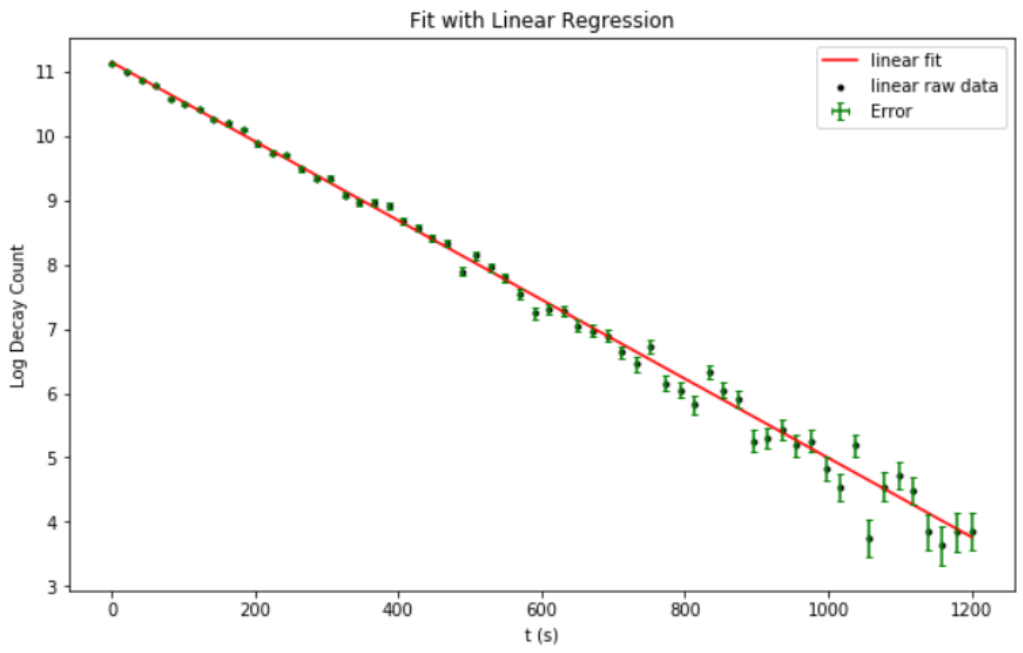


Figure 1: *Linear model using curve_fit() over-plotted against raw emission data points with corresponding errors.*

The nonlinear model is described by the function g, where g is defined as a general exponential formula:

$$g = be^{ax}$$

For the emission data, the function g represents the intensity of the emission. The independent variable, x, is again time (in seconds), a is the negative reciprocal of mean lifetime of the isotope and b is the initial intensity at time t = 0s. For this experiment, the nonlinear fit effectively represents the equation:

$$\mathrm{I(t)} = I_0 e^{-\frac{t}{\tau}}$$

Where τ is the mean lifetime of the isotope being observed. Plotting the nonlinear model fit with estimated parameters, (a = 0.006, b = I_0), produces a fit that is slightly inaccurate early on but becomes increasingly more accurate over time. From visual comparison, the nonlinear fit produces an overall better approximation of the data set.

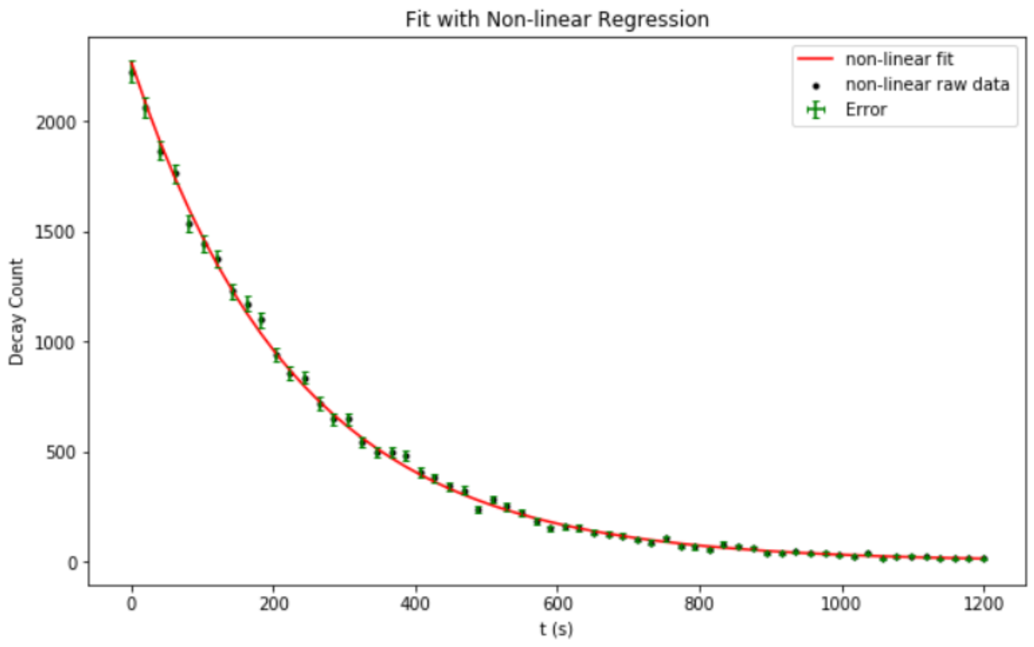


Figure 2: *Nonlinear model using curve_fit() over-plotted against raw emission data points with corresponding errors.*

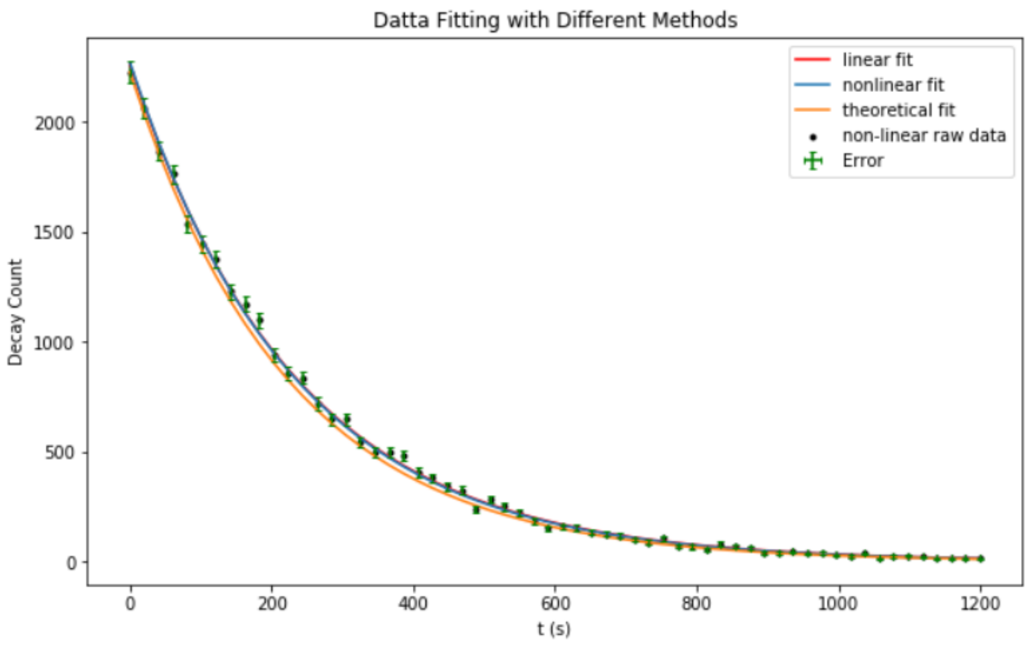
The half-life time for each model can be computed by analyzing the a parameter estimated for each fit:

$$Linear\ Model: t_{halflife} = -1/a_{linear}$$

$$Nonlinear\ Model: t_{halflife} = -\ln(2)/a_{nonlinear}$$

The resulting estimations of half-life time are 162 $\pm$ 0.8 seconds (linear) and 161 $\pm$ 1.5 seconds (nonlinear). The expected half-life time in theory is 156 seconds and so this result further emphasizes that the nonlinear model represents the radioactive decay data better than the linear model.

A further analysis is performed by over-plotting the theoretical curve, which follows the original formula from section 2, alongside both curve fit models and raw data for both a standard y-axis and a logarithmic y-axis scale. It is difficult to definitively conclude that either curve fit model is significantly better than the other from this, however the non-linear fit does seem to be ever so slightly closer to the theoretical fit.



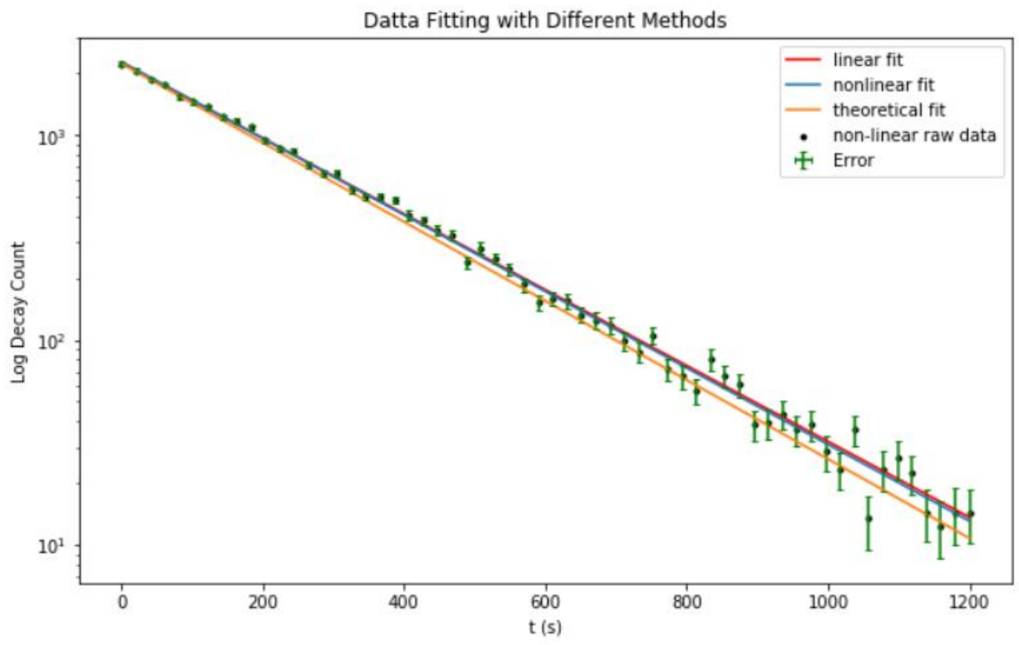


Figure 3: *All models over-plotted against raw emission data with corresponding error; (Top) Standard y-axis, (Bot) Log y-axis.*

Looking at the χ^2 values we see that χ^2_{linear} = 2.3 and $\chi^2_{nonlinear}$ = 1.2, which suggests that both models are reasonably accurate fits to the data. However, $\chi^2_{nonlinear}$ is slightly closer to 1 and thus is statistically a better model even if the difference is small. This agrees with the previous results and it is concluded that the nonlinear fit is a more accurate model of the data.

6 Discussion

As expected the nonlinear model served as a better approximation of the gamma emission data set. This is firstly validated by the computed half-life time of each model: 162 $\pm$ 0.8 seconds (linear) and 161 $\pm$ 1.5 seconds (nonlinear). The expected half-life was 156 seconds which is closer to the nonlinear model's output. Additionally, it can be seen in Figure 3 that the nonlinear approximation is very slightly closer to the theoretical model than the linear approximation. Statistically, the χ^2 values for each respective data set: χ^2_{linear} = 2.3 and $\chi^2_{nonlinear}$ = 1.2 also bolsters the notion that the nonlinear model is overall a better fit for the data. However, it should be noted that both models reasonably represent the data set and the nonlinear model is only slightly more accurate.

The uncertainty in the experiment first arises from the error in experimental values from the Geiger counter which follows counting statistics. Thus the error in decay count is determined according to the rule:

$$\sigma_{source} = \sqrt{N}$$

Where σ_{source} is the uncertainty in the decay count and N is the number of events. Additionally, when correcting for background radiation, the error is propagated as:

$$\sigma_{signal} = \sqrt{\sigma^2_{source} + \sigma^2_{background}} = \sqrt{N_{source} + N_{background}}$$

Furthermore, when dealing with the transformation of the data to a logarithmic scale in the linear model, the uncertainties follow the standard propagation rule for logarithms:

$$\sigma_{\log(y_i)} = \left|\frac{\sigma_{y_i}}{y_i}\right|$$

Finally, to propagate error for the computed half-life values for each respective curve fit model the rule that is followed is:

$$\sigma_{\mathrm{t_{halflife}}} = \sqrt{Var(t_{halflife})} = \frac{\sigma_a}{a^2}$$

Where $Var(t_{halflife})$ is the first diagonal element of the covariance matrix provided by the curve fitting module and corresponds to the variance of the parameter a.

The strength of this experimental design is that it is performed using highly precise equipment and thus the uncertainty in the data is very low and produces clear results. The background radiation was also computed to correctly remove noise from the emission data. On the other hand, the experiment is not easily replicated as it requires access to a Geiger counter and the specific radioactive isotopes, in addition to the expertise to operate and monitor the experiment. However, for an experiment on radioactive emission, a complex setup cannot be avoided without repercussions in the accuracy of the data set and thus the experimental setup should not undergo changes.

7 Numbered Questions

Question 1:

The nonlinear regression method gave a closer estimate to the expected half-life (156 seconds) by a very slight margin: 162 $\pm$ 0.8 seconds (linear) and 161 $\pm$ 1.5 seconds (nonlinear). The difference is very miniscule on the plots but especially in Figure 3, when the logarithmic y-axis scale is used, it can be seen that the nonlinear model is indeed more closely resembles the theoretical model than the linear by a small amount.

Question 2:

The half-life using linear fit has error $\pm$ 0.8 seconds and $\pm$ 1.5 seconds for that of the nonlinear fit. Unfortunately, the theoretical half-life value did not fall within the error margin of either model. However, both models do still represent the emission data reasonably well overall.

Question 3:

The χ^2 values that were computed for each respective model are: χ^2_{linear} = 2.3 and $\chi^2_{nonlinear}$ = 1.2. This suggests that both models provide a decently good fit for the data as they are both close to 1 but are still incomplete fits. Additionally, this shows that the nonlinear model is a slightly more accurate fit over the linear model, which agrees with previous results.

8 Conclusion

From the experiment we conclude that both the linear and nonlinear curve fitting methods produced a reasonably accurate approximation of the emission data which adheres to an exponential decay relation between intensity and time. The nonlinear fitting proved to be a slightly better fit for the data set as it produced a half-life time that was closer to the expected value (161 $\pm$ 1.5 seconds) than the linear model (162 $\pm$ 0.8 seconds) and had a χ^2 value closer to 1 (χ^2_{linear} = 2.3 and $\chi^2_{nonlinear}$ = 1.2) which suggests it was the more accurate fit.

9 References

[1] Pandhi, A. and Chen, X., University of Toronto, Toronto, ON. "PHY224 Laboratory Notes: PyLab 3 – Nonlinear Fitting Methods I", October 2018.

10 Appendices

The full code used for this lab can be found below as well as on the author's Github as "PyLab 3.py": https://github.com/AyushPandhi/Pandhi_Ayush_PHY224/tree/master/PyLab%203.

```
#PyLab 3: Nonlinear Fitting Methods I
#Author: Ayush Pandhi (1003227457)
#Date: 10/11/2018

#Importing required modules
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
import pylab

#Defining the linear model function
def f(x, a, b):
    return a*x + b

#Defining the exponential model function
def g(x, a, b):
    return b*(np.exp(a*x))

#Loading the radioactive decay data file
sample = np.loadtxt('RadioactiveDecay_TuesdayOct2_2018_decay.txt', skiprows=2, usecols=(0,))
count = np.loadtxt('RadioactiveDecay_TuesdayOct2_2018_decay.txt', skiprows=2, usecols=(1,))

#Loading the background radiation data file
sample_bg = np.loadtxt('RadioactiveDecay_TuesdayOct2_2018_background.txt', skiprows=2, usecols=(0,))
count_bg = np.loadtxt('RadioactiveDecay_TuesdayOct2_2018_background.txt', skiprows=2, usecols=(1,))

#Getting the mean background radiation
mean_count_bg = np.mean(count_bg)

#Adjusting decay data count by subtracting mean background count
count = count - mean_count_bg
```

```
#Defining N, t and dt
N = 60
dt = 20
t = np.linspace(0, N*dt, N)

#Getting count rate and the standard deviation for each
rate = count/dt
sigma = (count**0.5)
sigmabg = (count_bg**0.5)
sigma_signal = ((sigma**2) + (sigmabg)**2)**0.5

#Natural log of the decay count and the error propgation
logy = np.log2(count)
logsigma = np.absolute(sigma_signal/count)

#Linear regression
p_opt_1, p_cov_1 = curve_fit(f, t, logy, (1,0), logsigma, True)
lin_output = f(t, p_opt_1[0], p_opt_1[1])

#Plots of linear regression
plt.figure(figsize=(10,6))
plt.scatter(t, logy, label = 'linear raw data', marker='.', color='k')
plt.plot(t, lin_output, 'r-', label = 'linear fit')
plt.title('Fit with Linear Regression')
plt.xlabel('t (s)')
plt.ylabel('Log Decay Count')
plt.errorbar(t, logy, xerr=0, yerr=logsigma, linestyle='none', ecolor='g', label='Error', capsize=2)
plt.legend()
plt.show()

#Non-linear regression
p_opt_2, p_cov_2 = curve_fit(g, t, count, (0.006, count[0]), sigma_signal, True)
nonlin_output = g(t, p_opt_2[0], p_opt_2[1])

#Plots of non-linear regression
plt.figure(figsize=(10,6))
plt.scatter(t, count, label = 'non-linear raw data', marker='.', color='k')
plt.plot(t, nonlin_output, 'r-', label = 'non-linear fit')
plt.title('Fit with Non-linear Regression')
plt.xlabel('t (s)')
plt.ylabel('Decay Count')
plt.errorbar(t, count, xerr=0, yerr=sigma_signal, linestyle='none', ecolor='g', label='Error', capsize=2)
plt.legend()
plt.show()

#Computing half-life for linear regression method
lin_halflife = -(1/p_opt_1[0])

#Computing half-life for non-linear regression method
nonlin_halflife = -0.69314718056*(1/p_opt_2[0])

print('The half life with linear regression: ', lin_halflife, 's')
print('The half life with non-linear regression: ', nonlin_halflife, 's')
print('The theoretical half life is: 156 s')

#Overplotting linear fit, non-linear fit and theorethical
plt.figure(figsize=(10,6))

#Raw data
plt.scatter(t, count, label = 'non-linear raw data', marker='.', color='k')

#Linear fit
lin_fit = 2**(p_opt_1[0]*t + p_opt_1[1])
plt.plot(t, lin_fit, 'r-', label='linear fit')

#Nonlinear fit
nonlin_fit = nonlin_output
plt.plot(t, nonlin_fit, label='nonlinear fit')

#Theoretical fit
theo_fit = (count[0])*(0.5**(t / 156)) #half life is 2.6min = 156 sec
plt.plot(t, theo_fit, label='theoretical fit')
plt.title('Datta Fitting with Different Methods')
plt.xlabel('t (s)')
plt.ylabel('Decay Count')
plt.errorbar(t, count, xerr=0, yerr=sigma_signal, linestyle='none', ecolor='g', label='Error', capsize=2)
plt.legend()
plt.show()

#Producing the same plot but with a log y axis
plt.figure(figsize=(10,6))

#Raw data
plt.scatter(t, count, label = 'non-linear raw data', marker='.', color='k')

#Linear fit
lin_fit = 2**(p_opt_1[0]*t + p_opt_1[1])
```

```
plt.plot(t, lin_fit, 'r-', label='linear fit')

#Nonlinear fit
nonlin_fit = nonlin_output
plt.plot(t, nonlin_fit, label='nonlinear fit')

#Theoretical fit
theo_fit = (count[0])*(0.5**(t / 156)) #half life is 2.6min = 156 sec
plt.plot(t, theo_fit, label='theoretical fit')
plt.title('Datta Fitting with Different Methods')
plt.yscale('log')
plt.xlabel('t (s)')
plt.ylabel('Log Decay Count')
plt.errorbar(t, count, xerr=0, yerr=sigma_signal, linestyle='none', ecolor='g', label='Error', capsize=2)
plt.legend()
plt.show()

#Calculating standard deviation
sigma_a_1 = np.sqrt(p_cov_1[0,0])/((p_opt_1[0])**2) #linear regression
sigma_a_2 = np.sqrt(p_cov_2[0,0])/((p_opt_2[0])**2) #nonlinear regression
print('Standard Deviation (Linear Model): ', sigma_a_1)
print('Standard Deviation (Non-linear Model): ', sigma_a_2)

#Calculating chi squared
chi_sq = (1/65)*np.sum(((logy - lin_output)/logsigma)**2)
chi_sq2 = (1/65)*np.sum(((count - nonlin_output)/sigma_signal)**2)
print('Chi Squared Reduced (linear): ', chi_sq)
print('Chi Squared Reduced (nonlinear): ', chi_sq2)
```