Assignment 3: Least Square Fits

Aaron Miller
17322856

October 31, 2019

1 Introduction

The aim of this assignment was to fit experimental data using the least square fit method through python. The experimental dataset used was measurements of radioactive decay, as a function of time (ie: decays per second).

1.1 radioactive decay

Radioactive decay is the process by which unstable atoms lose energy. There are many various forms. For instance, beta where energys lost in the form of an electron and a neutrino being expelled from the atom, and gamma decay where high energy gamma photons are expelled. The process is stochastic when you try to predict when an atom decays, but when we model the behaviour of a collection of a single type of these decaying atoms we find the exponential relationships

$$N(t) = N_0 e^{-\lambda t}, \quad \frac{dN(t)}{dt} = -\lambda N(t) \tag{1.1}$$

where $N(t)$ is the number of atoms present in a sample at a time t, N_0 is the original amount of sample, and λ is the decay constant defined in terms of the half life $t_{1/2}$ which is defined as $N(t_{1/2}) = \frac{N_0}{2}$

$$\lambda = \frac{\ln(2)}{t_{1/2}}.$$

1.2 Least square fitting

All experimental data is subject to error. In order to mitigate this, we strive for a line of 'best fit', $f(x)$ which is our functional guess of the line which fits the data defined in the following sum;

$$S = \sum_{i=1}^{N} \left(f\left(x_i\right)_{p_1,\cdots,p_k} - y_i \right)^2 ,$$

where (x_i, y_i) are the experimental data points we are fitting, S is the sum of squared residuals, and p_i are the fitting parameters defined so that the sum is minimised. In my opinion, this is a very natural definition of the line of best fit.

1.2.1 linear regression

Now for a linear functional guess,

$$x(t) = a + bt, \tag{1.2}$$

after minimising S with respect to a, and b, in order to find the lowest error between our data and our line, we find

$$\begin{array}{c} aN + b\sum_{i=1}^{N} t_i = \sum_{i=1}^{N} x_i \\ a\sum_{i=1}^{N} t_i + b\sum_{i=1}^{N} t_i^2 = \sum_{i=1}^{N} x_i t_i \end{array} .$$

Which yield fit parameters a, and b, defined in terms of the sums as

$$\begin{aligned} a &= \frac{-(\sum_{i=1}^{N} t_i^2)(\sum_{i=1}^{N} x_i) + (\sum_{i=1}^{N} t_i)(\sum_{i=1}^{N} x_i t_i)}{(\sum_{i=1}^{N} t_i)^2 - N(\sum_{i=1}^{N} t_i^2)}, \\ b &= \frac{(\sum_{i=1}^{N} t_i)(\sum_{i=1}^{N} x_i) - N(\sum_{i=1}^{N} x_i t_i)}{(\sum_{i=1}^{N} t_i)^2 - N(\sum_{i=1}^{N} t_i^2)}. \end{aligned} \tag{1.3}$$

2 Discussion and results

2.1 Plot of data

In figure 2.1 we see our experimental data plotted on a linear scale. The data used was

$$\textit{time (in seconds)} = [5, 15, 25, 35, 45, 55, 65, 75, 85, 95, 105, 115],$$

$$\textit{number of decays} = -\frac{dN(t)}{dt} = [32, 17, 21, 7, 8, 6, 5, 3, 4, 1, 5, 1].$$

Now we see that there is an exponential relationship in this dataset, however it is rather noisy. Note the given number of decays is related to Eq.1.1 by $-\frac{dN(t)}{dt} = Decays$, where N is the number of constituents of the sample present.

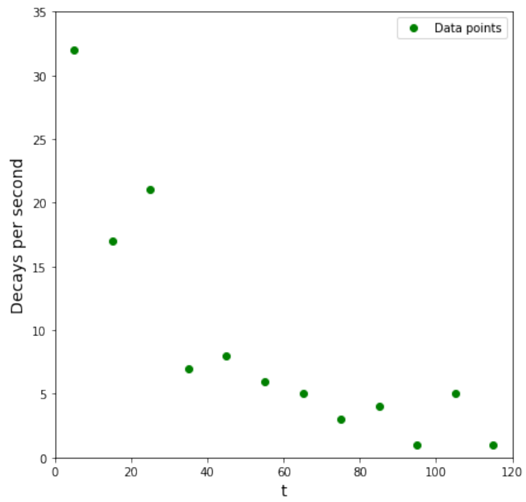

Figure 2.1: Lin-lin plot of experimental data

2.2 Plot of data using a semi-log scale

Now in order to explore this relationship, I have now plotted this data on a linear-log scale, (Fig.2.2). We see that there is an evident negative linear relationship, in the lin-log plot, which implies our function is some negative exponential function. This is congruent with theory as evident from Eq.1.1 which suggests we are plotting

$$\ln(-\frac{dN(t)}{dt}) = \ln(N_0\lambda) - \lambda t,$$

against time t. Thus the slope of this line is $-\lambda$, and the y-intercept is $\ln(\lambda N_0)$. By eye the slope seems to be ≈ -0.03, and a y-intercept of approximately 3.5.

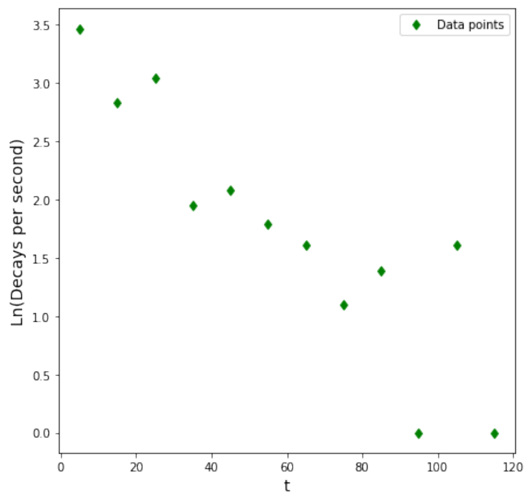

Figure 2.2: Lin-log plot of experimental data

2.3 Implementing linear regression to the data

In order to appropriately fit this data, I needed to first decide which plot to fit. Fig.2.2 is linear and I can apply LSF to it with a linear functional guess equivalent to Eq.1.2. So in order to find the LSF all I needed to do was evaluate the sums in Eq.1.3 to find the two fit parameters (a,b). Below is a snippet of the code used to evaluate these parameters.

```
def parameters(decays, time):

        N=len(decays)
        logdecays=np.log(decays)
        s1=sum(time)
        s2=sum((time)**2)
        s3=sum((logdecays))
        s4=sum((logdecays*time))

        a = (-s2 * s3 + s1 * s4) / (s1**2 - N * s2)
        b = (s1 * s3 - N * s4) / (s1**2 - N * s2)

        return a , b
```

I found

$$(a, b) = (3.3409, 0.0267)$$

, which agree with my estimates. Thus we have an equation of a line.

$$\ln(-\frac{dN(t)}{dt}) = 3.3409 - 0.0267t.$$

These values correspond physically to
$\lambda = 0.0267s^{-1}$, and $ln(\lambda N_0) = a \implies N_0 = e^a/\lambda = 1057$. Fig.2.3 shows this line illustrated on a lin-log graph, and it seems like an accurate fit for the data.

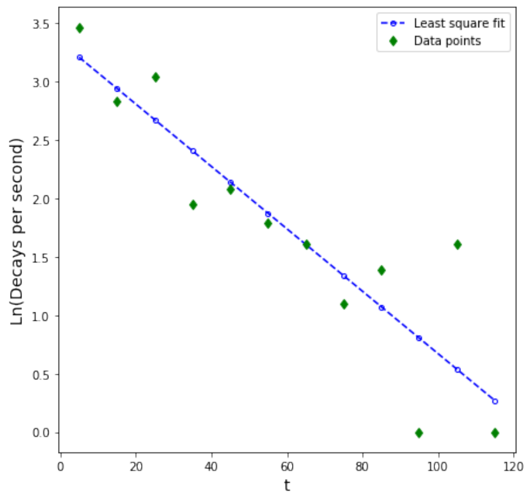

Figure 2.3: LSF of lin-log graph

2.4 Using fit parameters to to plot the decay function

Now in this section we transform back to our original linear scale. As the LSF was preformed on a lin-log plot, in order to translate back to a lin-lin, we take the LSF equation of a line and put it in an exponential function. Essentially applying the inverse of the log function. We return to Eq.1.1, in the form

$$-\frac{dN(t)}{dt} = e^{a+bt}. \tag{2.1}$$

The figure below illustrates this transformed fit.

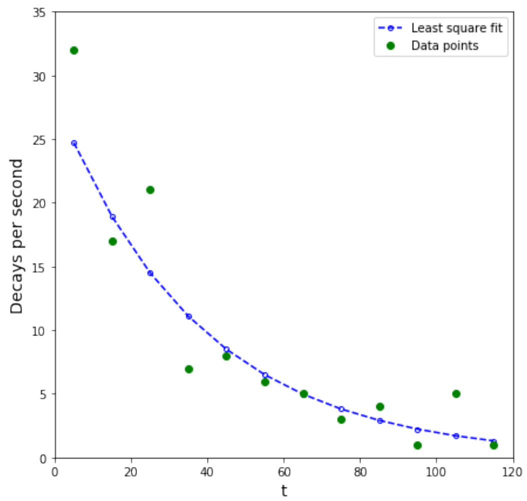

Figure 2.4: Lin-lin graph of transformed LSF

2.5 Non linear least square fit

For the last part, I imported the following library,

```
import scipy.optimize as optimization
```

which contains the relevant non linear LSF function. I applied the non linear LSF to the data in the method described in the supplied question. I fit the data points to the function

$$f(t) = e^{A+Bt}$$

which is equivalent to Eq.2.1. I found the values of the fitting parameters (A, B) to be

$$(A, B) = (3.5657, -0.0325).$$

Note $(A, B) \equiv (a, b)$ from the previous section. The relative error with respect to the second set of parameters of $\Delta(A, B) = (6\%, 18\%)$. Figure 2.5 illustrates this discrepancy. From inspecting this figure that the non-linear fitting hugs the data better as time approaches zero. This discrepancy arrises from the way the fitting is calculated. In the sum of squared residuals used to find the fit parameters (a, b), $f(x_i)$ was a linear functional, whereas for the fit parameters (A,B) it was an exponential.

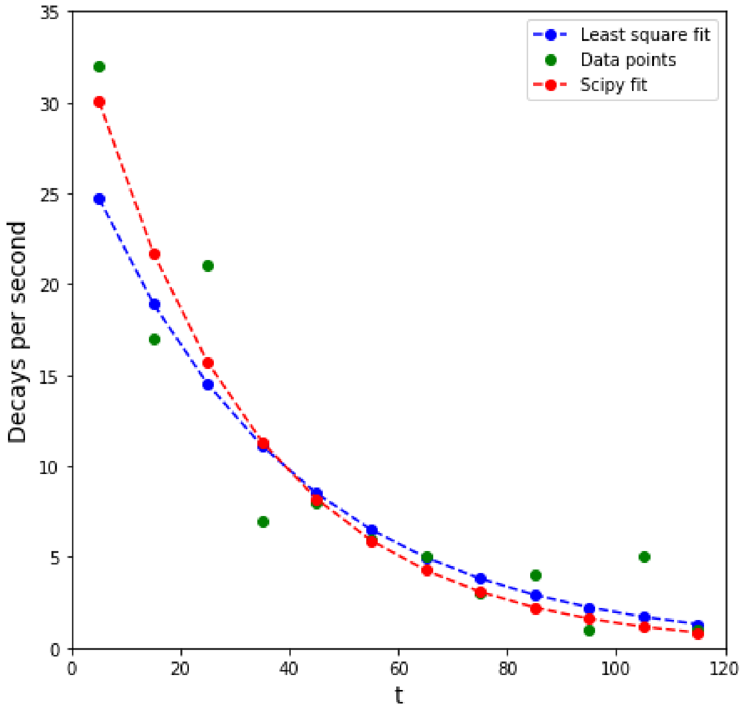

Figure 2.5: Lin-lin graph of the data, and fittings