# Mass Attenuation

Jason Pruitt

November 2019

## 1 Data and Analysis

The collected data is displayed both as raw data in Tables 1 and 2, and as linear fits in Figures 1,2,3, and 4, each considered with the function

$$I = I_0 e^{-\mu x}, \tag{1}$$

where $I$ is the intensity returned by the relation of the unblocked intensity $I_0$, the exponential to the power of thickness $x$, and the absorption coefficient of $\mu$. The intensity of any given measurement was the counts of the measurement over the 4 minute time interval. Seeking to find the absorption coefficient, Equation 1 was linearized to yield

$$lnI = -\mu x + lnI_0, \tag{2}$$

where our fit of Equation 2 to the raw data returned the $\mu$ value, which was compared to an interpolation of the NIST data for both aluminum [1] and lead [2] for the two peaks of the Co-60 at 1.173 MeV and 1.333 MeV. The uncertainty in the counts was simply the square roots of the counts, the uncertainty in the slope or the $\mu$ value was returned by the fit in python, and the uncertainty in each data point was described by

$$\delta lnI = ln\left(\frac{N}{t}\right)\frac{\sqrt{N}}{N}. \tag{3}$$

As displayed in Table 3, the $\mu$ values for aluminum were within 2 $\sigma$ of the accepted values, whereas the measured values for lead were not in as good of agreement with the acccepted. However, with consideration to the equipment and the way the analysis was conducted, we concluded that the software that determined the ROI must not have properly considered the background counts, which lead to the conclusion that the uncertainty reported in Table 3 must be too small. This explains why the lead points do not agree with the accepted within the reported uncertainty.

To answer the question in the lab manual, "Given an initial intensity, what thickness of aluminum and lead will cut that intensity in half?", we solve Equation 1 for the thickness $x$ assuming the intensity is half the initial intensity, giving us

$$x = \frac{ln2}{\mu}. \tag{4}$$

Using the accepted $\mu$ values from interpolating the NIST data in Table 3, we obtain the data shown in Table 4, which uses both the accepted and measured values to calculate the thicknesses of aluminum and lead that would reduce the initial intensity by a half. The uncertainty in thickness calculated with the measured values was found with

$$\delta thick = \sqrt{\left(-\frac{ln2}{\sqrt{\mu}}\right)^2 \delta\mu^2}. \tag{5}$$

| Al Thickness (cm) | Al Peak1 Net Area | Al Peak2 Net Area |
|---|---|---|
| 0.006±.002 | 10500 ± 100 | 10300 ± 100 |
| 1.013±.002 | 9600 ± 200 | 9000 ± 200 |
| 1.613±.002 | 8800 ± 90 | 8600 ± 90 |
| 2.305±.002 | 7590 ± 90 | 7600 ± 90 |
| 2.542±.002 | 7360 ± 90 | 6740 ± 80 |
| 3.617±.002 | 5570 ± 70 | 5870 ± 80 |
| 4.124±.002 | 5240 ± 70 | 5300 ± 70 |
| 4.720±.002 | 4990 ± 70 | 4550 ± 70 |

Table 1: Raw Data for Aluminum

| Pb Thickness (cm) | Pb Peak1 Net Area | Pb Peak2 Net Area |
|---|---|---|
| 0.632± .002 | 7400± 90 | 7124 ± 80 |
| 1.016± .004 | 5500± 70 | 5297 ± 70 |
| 2.751± .006 | 2580± 50 | 2260 ± 50 |
| 3.881± .008 | 1200± 30 | 1320± 40 |
| 4.979± .0010 | 660± 30 | 906 ± 30 |
| 6.002± .0012 | 600± 20 | 477± 20 |
| 6.002± .0012 | 400± 20 | 409± 20 |

Table 2: Raw Data for Lead

| | Measured $\mu$ | Accepted $\mu$ |
|---|---|---|
| Al at 1.173 MeV | 0.173 ± 0.011 | 0.154 |
| Al at 1.333 MeV | 0.160 ± 0.010 | 0.144 |
| Pb at 1.173 MeV | 0.51 ± 0.02 | 0.707 |
| Pb at 1.333 MeV | 0.499 ± 0.017 | 0.639 |

Table 3: Measured absorption coefficients compared to the accepted values for 1.173 MeV and 1.333 MeV

| | Thickness with measured $\mu$ (cm) | Thickness with accepted $\mu$ (cm) |
|---|---|---|
| Al at 1.173 MeV | 3.95 ± 0.018 | 4.5 |
| Al at 1.333 MeV | 4.03 ± 0.017 | 4.8 |
| Pb at 1.173 MeV | 1.35 ± 0.03 | .98 |
| Pb at 1.333 MeV | 1.36 ± 0.018 | 1.03 |

Table 4: Calculated thicknesses using measured and accepted absorption coefficients at 1.173 MeV and 1.333 MeV

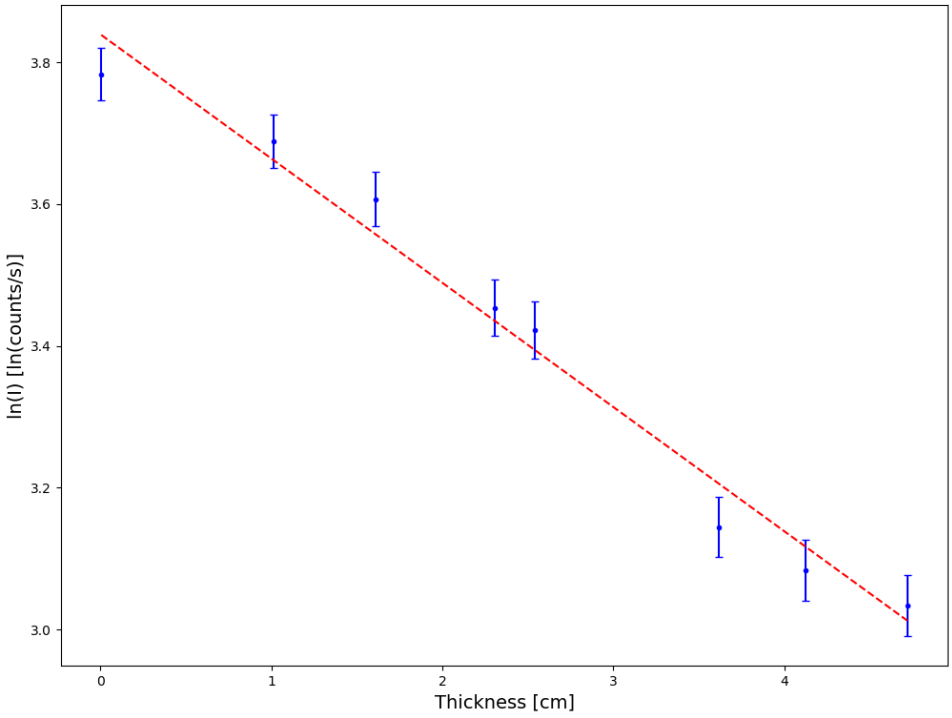


Figure 1: Plot to find $\mu$ value for Al from slope of lnI vs Al thickness at Co-60's 1.173 MeV peak

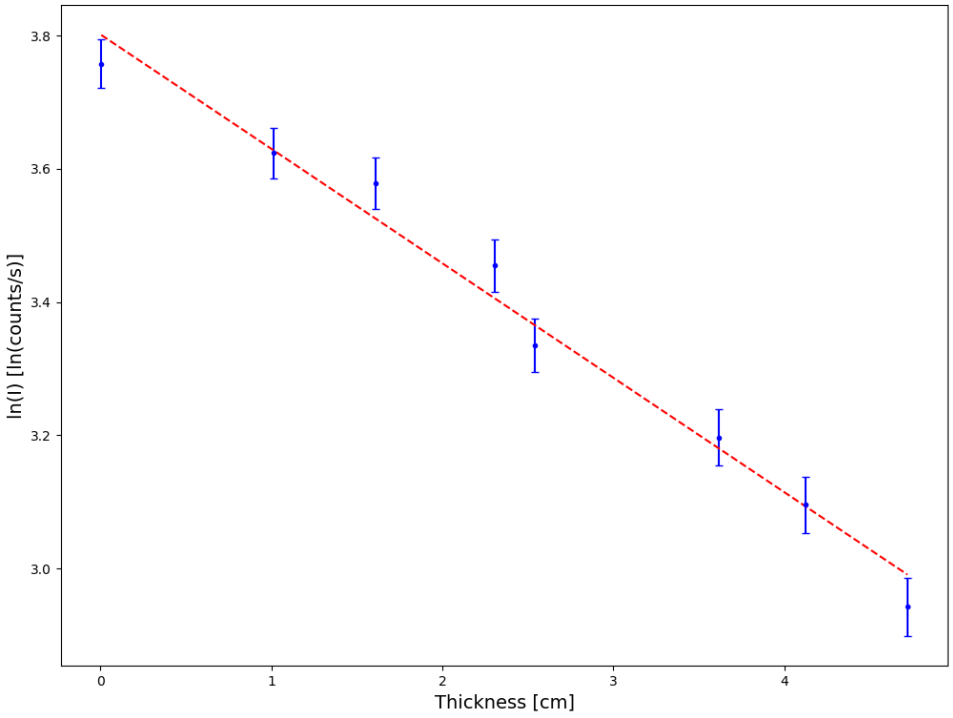


Figure 2: Plot to find $\mu$ value for Al from slope of lnI vs Al thickness at Co-60's 1.333 MeV peak

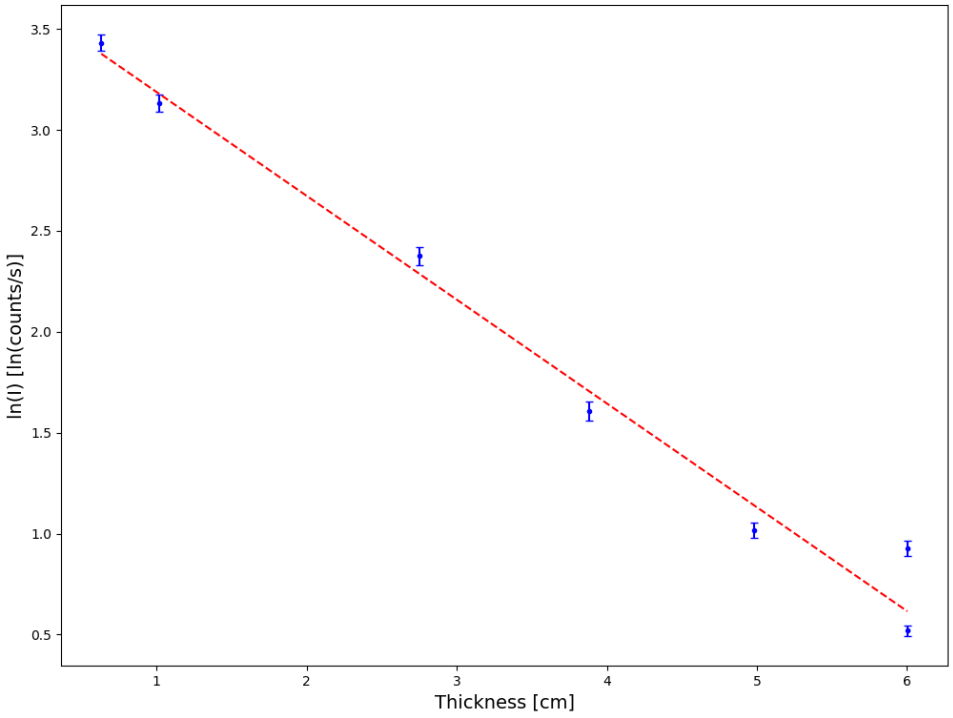


Figure 3: Plot to find $\mu$ value for Pb from slope of lnI vs Pb thickness at Co-60's 1.173 MeV peak

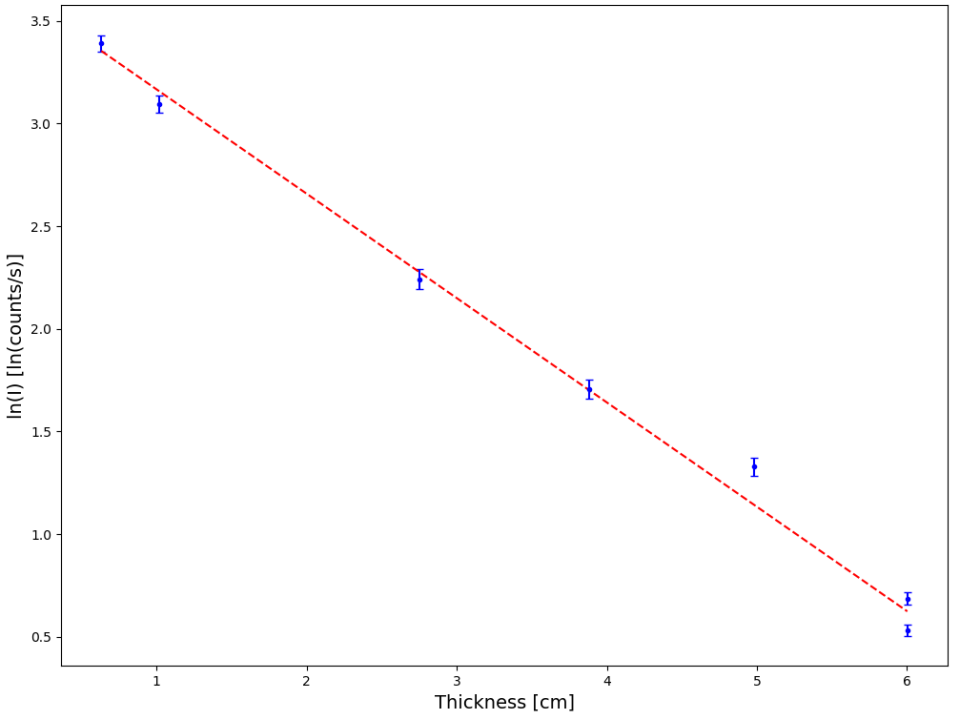


Figure 4: Plot to find $\mu$ value for Pb from slope of lnI vs Pb thickness at Co-60's 1.333 MeV peak

# References

[1] Absorption coefficient for aluminum. https://physics.nist.gov/PhysRefData/XrayMassCoef/ElemTa

[2] Absorption coefficient for lead. https://physics.nist.gov/PhysRefData/XrayMassCoef/ElemTab/z82

# 2 Appendix

Appendix: Code

```
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
from scipy import interpolate
from scipy.optimize import curve_fit

## Notes for Jason:
# For Co-60
# peak1 is at 1173 keV
# peak2 is at 1333 keV

df0 = pd.read_csv('massNotes.csv',index_col = None)

df1 = pd.read_csv('Al_Log.csv',index_col = None)
al_thick = df1[df1.columns[0]].to_list()
al_p1na = df1[df1.columns[1]].to_list()
al_p2na = df1[df1.columns[2]].to_list()

df2 = pd.read_csv('Pb_Log.csv',index_col = None)
pb_thick = df2[df2.columns[0]].to_list()
pb_p1na = df2[df2.columns[1]].to_list()
pb_p2na = df2[df2.columns[2]].to_list()

df3 = pd.read_csv('nist_al_coeff.txt',index_col = None,delimiter = ' ')
al_energy = df3[df3.columns[0]].to_list()
al_muRho = df3[df3.columns[2]].to_list()

df4 = pd.read_csv('nist_pb_coeff.txt',index_col = None,delimiter = ' ')
pb_energy = df4[df4.columns[0]].to_list()
pb_muRho = df4[df4.columns[2]].to_list()

def linear_fit(x,A,B):
    return(A*x+B)
```

```
def theory_val(en,mu):
    peak1 = 1.173 # MeV
    peak2 = 1.333 # MeV
    # print('energy',en)
    # print('muval',mu)

    f = interpolate.interp1d(en,mu)
    xnew = np.linspace(min(en),max(en),10000)
    ynew = f(xnew)
    muTheory1 = f(peak1)
    muTheory2 = f(peak2)

    # plt.figure()
    # plt.loglog(en,mu,'b.')
    # plt.loglog(xnew,ynew,'r-')
    # plt.axvline(peak1)
    # plt.axvline(peak2)
    # plt.show()

    return(muTheory1,muTheory2)

def log_plot(thick,N,fit_function,name):

    time = 4*60 # seconds
    log_intensity = np.log(np.array(N)/time)

    nErr = log_intensity*np.sqrt(np.array(N))/N
    popt,pcov = curve_fit(fit_function,thick,log_intensity,sigma = nErr
        )
    perr = np.sqrt(np.diag(pcov))
    slopeErr = perr[0]
    # p_weight = np.poly1d(popt)
    xtheory = np.linspace(min(thick),max(thick))
    yfit = fit_function(xtheory,*popt)

    plt.figure(figsize = (12,9))
    plt.errorbar(thick,log_intensity,yerr = nErr,fmt = 'b.',capsize =
        3)
    plt.plot(xtheory,yfit,'r--')
    plt.xlabel('Thickness [cm]',fontsize = 14)
    plt.ylabel('ln(I) [ln(counts/s)]',fontsize = 14)
    # plt.show()
    plt.savefig(f'{name}.png')
    plt.close()
    return(popt[0],slopeErr)
```

```
almuExp1,alUnc1 = log_plot(al_thick,al_p1na,linear_fit,'AlPeak1')
almuExp2,alUnc2 = log_plot(al_thick,al_p2na,linear_fit,'AlPeak2')
almuTh1,almuTh2 = theory_val(al_energy, al_muRho)
pbmuExp1,pbUnc1 = log_plot(pb_thick,pb_p1na,linear_fit,'PbPeak1')
pbmuExp2,pbUnc2 = log_plot(pb_thick,pb_p2na,linear_fit,'PbPeak2')
pbmuTh1,pbmuTh2 = theory_val(pb_energy, pb_muRho)
```