Scintillations of Gamma Rays and the Compton Effect

Jason Pruitt

November 2019

1 Introduction

The groundwork from two famous scientists set up two important effects considered by our experiment. Einstein, through his investigation and conclusions of the Photoelectric Effect [1], described the ejection of electrons from metals through photon energy interactions. Arthur Compton's work described the process in which an incoming x-ray or gamma-ray photon deflects and scatters an electron, imparting energy, dubbed the Compton Effect [2]. In our lab, we sought to compare our measured values of the Compton and Photoelectric interactions of the gamma decaying Caesium-137 nucleus to the existing and known values, despite being unable to directly measure gamma rays.

2 Theory

The governing equations of this experiment all detail the energy of a gamma ray. The form of each of the equations allows us to circumvent the problem of directly measuring gamma rays by considering what their energies are dependent on. The change of direction and change in energy of a photon due to the collision between it and an electron is described by

$$E'_\gamma = \frac{E_\gamma}{1 + (\frac{E_\gamma}{mc^2})(1 - cos(\theta))}, \tag{1}$$

otherwise known as the Compton Effect, where E_{gamma} is the incident energy of the gamma-ray, mc^2 is 511 keV or the electron rest mass energy, and θ is the scattering angle. We then observe the energy of the electron exiting the atom as

$$E_{electron} = E_\gamma - E'_\gamma, \tag{2}$$

where E'_γ is described by Equation 1. Another photon-electron interaction, the Photoelectric effect, is described by

$$E_{electron} = E_\gamma - E_{binding}, \tag{3}$$

although since $E_{binding}$ is many times lower in magnitude than E_{gamma}, $E_{electron}$ is essentially equal to E_γ, meaning that all of the gamma ray's energy is imparted to the electron. We therefore rewrite Equation 3 as

$$E_\gamma = E_{electron}, \tag{4}$$

to describe the energy of the gamma ray in this interaction.

3 Methods

The apparatus displayed in Figure 1 consisted of a scintillator, a photomultiplier tube, and an Multi-Channel Analyzer. The gamma rays analyzed initially pass through the NaI scintillator crystal and undergo either Compton scattering or the Photoelectric effect, causing them to vary in energy. In either process, gamma rays ejected electrons from the atom and produced visible light that then was amplified by the photomultiplier tube it traveled through to produce signals for the Multi-Channel Analyzer to read. The MCA counted each time a particular energy was read, and thus was able to produce a histogram of the energy vs the counts. An oscilloscope trace, shown in Figure 2, was obtained to compare later to the Caesium-137 spectrum.

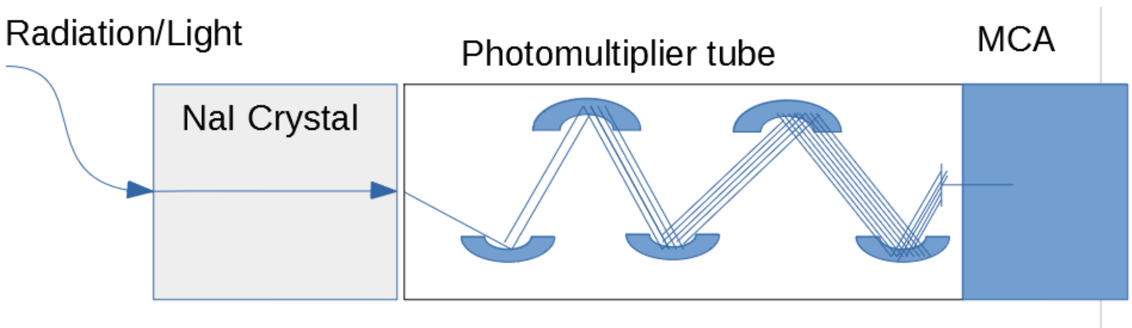

Figure 1: Diagram of the Scintillation Apparatus

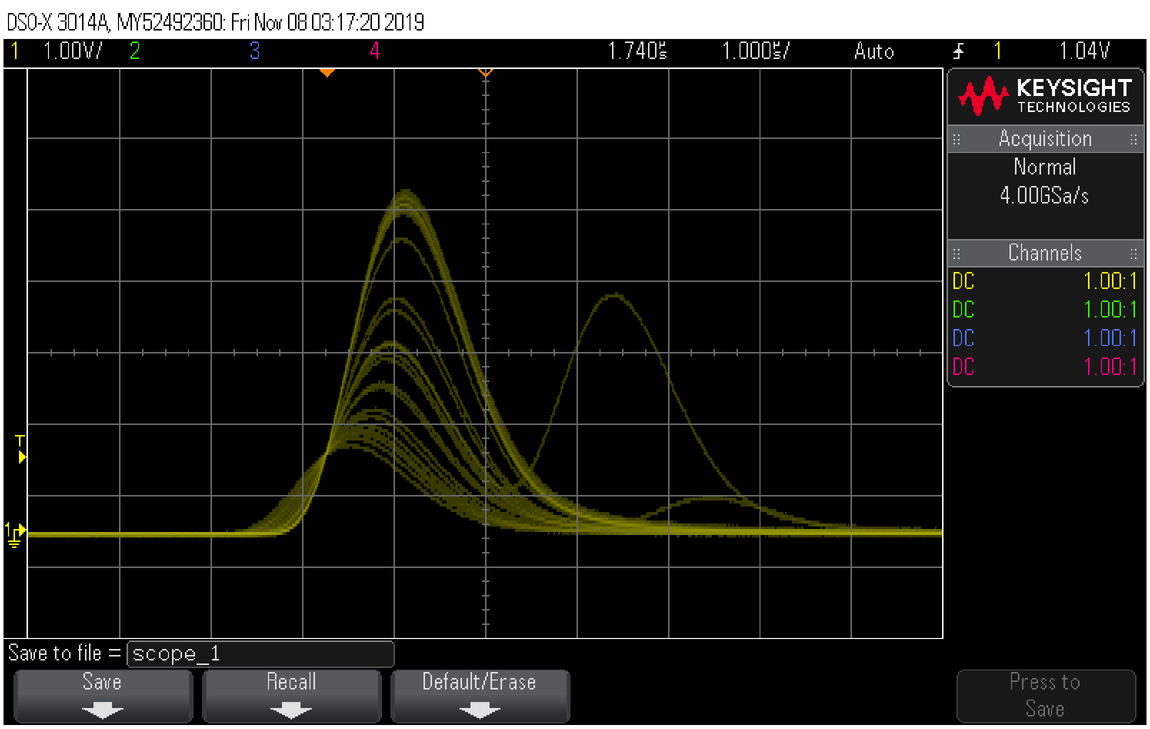

Figure 2: Oscilloscope trace of Cs-137

4 Data and Analysis

Because the MCA only returned the channel in which a particular energy was read at, a relation between energy and channel number needed to be made. Analyzing the spectrums of various other radioactive samples where the energy peaks are known, we produced a calibration curve, shown in Table 1 and Figure 3, of the form

$$(2.11 \times 10^{-7})x^3 + (-2.94 \times 10^{-4})x^2 + (1.51)x + (-17.3), \tag{5}$$

to allow for the conversion from channel number to its corresponding energy. This conversion was then applied to the data received from the Cs-137 spectrum to produce Figure 4, a graph of counts vs energy. In the graph, there are 4 regions of interest. Region 4 is the photopeak, or where the interaction is described by Equation 4, and the entire energy of the gamma ray is considered. Region 3 is the Compton edge, where the energy is described by Equation 1, but θ is 180 degrees, causing E'_γ to be at a minimum. Region 2 is produced by Compton backscattering, or when the gamma ray scatters an electron and travels back into the crystal, creating a photoelectron of energy

$$E_{electron} = E_{Photopeak} - E_{ComptonEdge}. \tag{6}$$

Region 1 is simply the characteristic peak of lead. Table 2 displays the comparison of the measured values to the accepted. Regions 2 and 3 are in similar disagreement with the accepted values, while region 4 is in good agreement with the accepted value, i.e. within 2 σ away.

Sample	Found Channel Number	Known Energies (keV)
Co-57	96.0	122.0
Ba-133	222.0	303.0
Ba-133	269.0	384.0
Na-22	376.0	511.0
Cs-137	483.0	662.0
Co-60	842.0	1173.0
Na-22	912.0	1274.0
Co-60	955.0	1333.0

Table 1: Table of values used for calibration curve.

Region Number	Measured Values (keV)	Accepted Values (keV)
1	21.7 ±.5	
2	204.1±.5	184.35
3	449.7±.5	477.65
4	663.9±.5	662.00

Table 2: Comparison of measured values of energy to accepted for Cs-137.

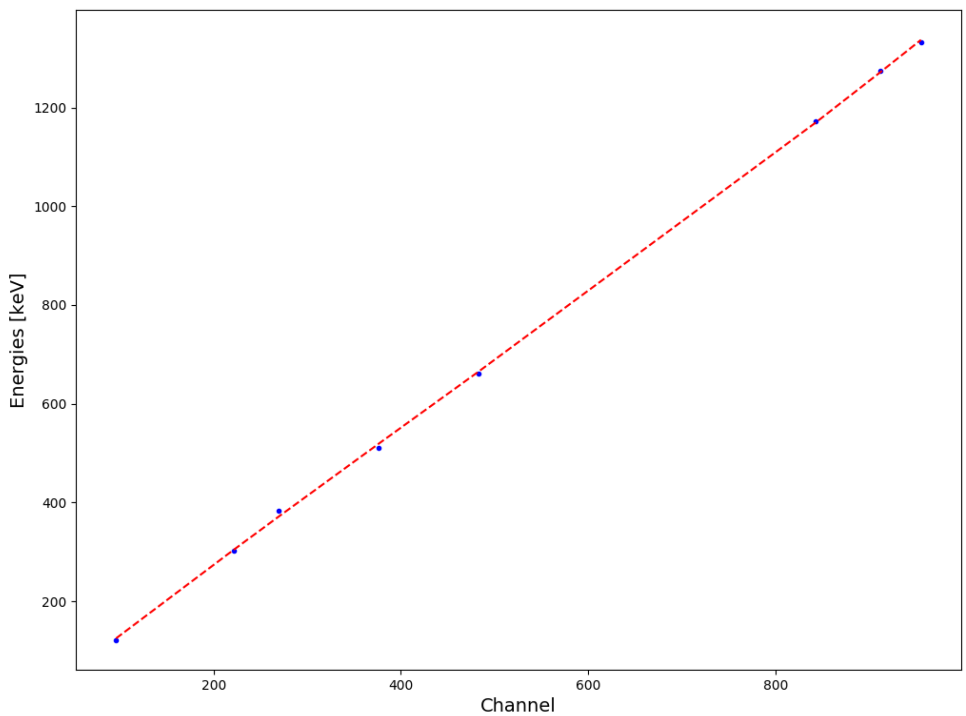

Figure 3: Relationship used to convert channel number to energy in keV.

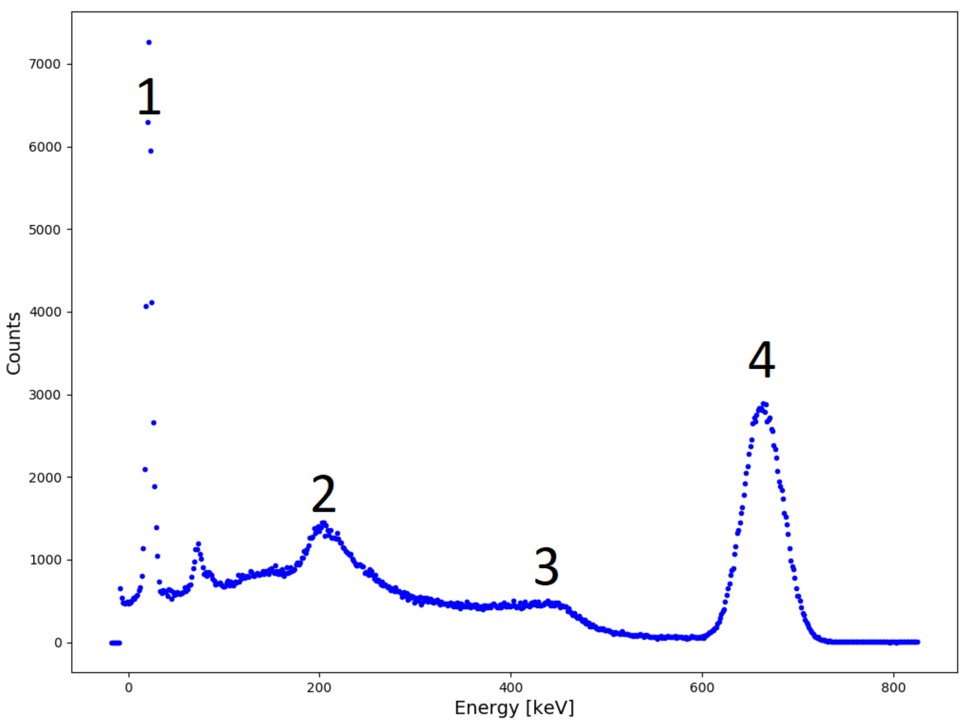

Figure 4: Cs-137 Spectrum with 4 regions of interest: 1 being the characteristic peak of lead, 2 the backscattering peak, 3 the Compton Edge, and 4 the photopeak.

5 Conclusions

To conclude, there exists discrepancies between our measured values of Caesium-137 energies and the accepted. While the energy of the photopeak is in good agreement with the accepted values, the energies of the Compton edge and the Compton backscattering peak are not. However, since these regions are in similar disagreement, this suggests the possibility of a bias for the non-photopeak energies. On top of this, longer NaI crystals could provide for better measurements in that the increased length would allow for more interactions.

References

[1] Einstein: Concerning an heuristic point of view toward the emission and transformation of light (translated to english. https://people.isy.liu.se/jalar/kurser/QF/references/Einstein1905b.pdf.

[2] The discovery of the compton effect. https://www.aps.org/programs/outreach/history/historicsites/

6 Appendix

Appendix: Code

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from scipy.optimize import curve_fit
from scipy.signal import find_peaks

with open('compton.csv','r') as csv:
    header = csv.readline()
    line = csv.readline()
    channel = []
    energies = []
    samples = []

    while line:
        line = line.split(',')
        channel.append(float(line[0]))
        energies.append(float(line[1]))
        line = csv.readline()
```

```
    channel = np.sort(np.array(channel))
    energies = np.sort(np.array(energies))

df1 = pd.DataFrame({
        'channel' : channel,
        'energies' : energies,
        })

print(df1.to_latex())
## Calibration Curve
def calibration(x,A,B,C,D):
    return(A*x**3 + B*x**2 + C*x + D)
popt,pcov = curve_fit(calibration,channel,energies)
perr = np.sqrt(np.diag(pcov))
slopeErr = perr[0]
yfit = calibration(channel, *popt)

plt.figure(figsize = (12,9))
plt.plot(channel,energies,'b.')
plt.plot(channel,yfit,'r--')
plt.xlabel('Channel',fontsize = 14)
plt.ylabel('Energies [keV]',fontsize = 14)
# plt.savefig('calibrationCurve.png')
# plt.show()
plt.close()


with open('cs137data.txt','r') as cs:
    counter = 0
    csChannel = []
    csCounts = []
    for line in cs:
        if 'Channel' in line:
            counter +=1
            continue
        if counter > 0:
            csChannel.append(float(line.split('\t')[0]))
            csCounts.append(float(line.split('\t')[1]))
    csChannel = np.array(csChannel[:600])
    csCounts = np.array(csCounts[:600])

csEnergies = calibration(csChannel, *popt)

# Find every peak higher than 2000
peaksHi, propertiesHi  = find_peaks(csCounts,height = [2000])
```

```
#
peaksMid, propertiesMid = find_peaks(csCounts,height = [1000,1900])

# Find compton edge
peaksLow, propertiesLow = find_peaks(csCounts,height = [300,1000])

# Display heights
# print(propertiesHi["peak_heights"])
# print(propertiesMid["peak_heights"])
# print(propertiesLow["peak_heights"])

# Index of photopeak, or number 4
ind1 = np.where(csCounts == 7270)
ind2 = np.where(csCounts == 1449)
ind3 = np.where(csCounts == 474)
# Hits that value twice so take the later one
ind3 = ind3[0][1]
ind4 = np.where(csCounts == 2888)

plt.figure(figsize = (12,9))
plt.plot(csEnergies,csCounts,'b.')
plt.axvline(x = csEnergies[ind1],ls = '--')
plt.axvline(x = csEnergies[ind2],ls = '--')
plt.axvline(x = csEnergies[ind3],ls = '--')
plt.axvline(x = csEnergies[ind4],ls = '--')
plt.xlabel('Energy [keV]',fontsize = 14)
plt.ylabel('Counts',fontsize = 14)
plt.savefig('csCurvePeaks.png')
# plt.show()
# plt.close()

eGamma = 662
restEnergy = 511

comptonEdgeTheoryEnergy = eGamma - eGamma/(1 + (2*eGamma/restEnergy))
# print('Compton Edge Theory: ',comptonEdgeTheoryEnergy)
# print('Exp for 1: ',csEnergies[ind1])
# print('Exp for 2: ',csEnergies[ind2], '\tTheory for 2: ', eGamma -
    comptonEdgeTheoryEnergy)
# print('Exp for 3: ',csEnergies[ind3], '\tTheory for 3: ',
    comptonEdgeTheoryEnergy)
# print('Exp for 4: ',csEnergies[ind4], '\tTheory for 4: ', eGamma)

df2 = pd.DataFrame({
    'Measured Values' : [csEnergies[ind1],csEnergies[ind2],csEnergies[
        ind3],csEnergies[ind4]],
```

```
    'Accepted Values' : [0,eGamma-comptonEdgeTheoryEnergy,
        comptonEdgeTheoryEnergy,eGamma]
    })

print(df2.to_latex())
```