INDUSTRIAL PROJECT
Gamma spectra curve fitting and peak identification algorithms for nuclear medicine, in collaboration with LabLogic Systems Ltd

University of York

Luke Kilgour

May 3, 2024

Declaration

This project is a collaborative effort between myself, Barnaby Marshall and LabLogic Systems. However, all work within this report is written by myself, except where otherwise stated. All data provided by LabLogic Systems is not included, as this data is sensitive. Rather the data is stored in text files which can only be accessed by those with permission from LabLogic Systems.

Abstract

This project explores how to create a program that identifies radionuclides based on their gamma-ray spectrometry. It investigates the mathematics behind nuclear decay and derives the decay equation. In doing so, this project provides a structured method in which to analyse gamma-spectra and the reasoning behind doing so. The importance of this analysis has many applications in medicinal imaging and use by LabLogic Systems. Analysis within this report contains many details and methods that are useful to this application.

Contents

Chapter 1

Introduction

1.1 Background

Gamma spectroscopy is a method in which radioactive decay is measured. When a radionuclide (radioactive nuclide, commonly referred to as an isotope), is formed, it is unstable. Due to the radioactive decay of the isotope, it becomes more stable. As a result of radioactive decay, there is an emission of particles, known as alpha and beta decay, and an emission of electromagnetic radiation[2]. This exoergic process can be measured using gamma spectroscopy, which measures the loss of energy from the radionuclide, measured in electron-volts(eV.) $1eV = 1.602177 \cdot 10^{-19}$ J (Joules)[2]. The energies of gamma radiation are measured in keV, as the range of emission on the electromagnetic spectrum is between 10 to 10,000 keV. "Gamma-ray emission is not, strictly speaking, a decay process; it is a de-excitation of the nucleus. being one way in which residual excitation energy is dissipated." [2] Gamma spectroscopy uses a detector to pick up these gamma rays at their respective energies, as there are many different types of detectors. This will be discussed in further detail later. The gamma spectra will be analysed using the algorithm produced in this project, in order to identify individual radionuclides and determine their purities.

1.2 Model Formulation

This project will discuss the creation of an algorithm for LabLogic Systems Ltd, which can identify radioactive isotopes based on their gamma spectroscopy. When gamma spectroscopy is performed on these radioactive isotopes, the resulting keV values can be plotted on a graphical interface to produce Gaussian curves. The software provided by LabLogic Systems, known as ‘Laura’, plots the energy levels (keV) against the counts of the corresponding keV values within a given time.

These Gaussian curves have peak energy values, which can be used to identify any radioactive isotope. The specific radioactive isotope can be identified from the gamma spectroscopy peaks. Peak energy values can be compared to the tables of collated data provided by the Laboratoire National Henri Becquerel (LNHB).[1] Comparing the characteristics of radioactive isotopes from the LNHB [1] and cross-examining those characteristics can identify which radioactive isotopes are present within the sample. In addition, a quantitative purity analysis can be produced for any given gamma spectra. This can be used to decompose a given isotope into its decaying components. In addition, a quantitative purity analysis for any given gamma spectra and decompose a given isotope into its decaying components.

Chapter 2

Peak Identification Methods

2.1 Rolling standard deviation

One approach to identifying peaks within a dataset is to take a moving window to calculate the standard deviation (2.1) of the points within the window.

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{2.1}$$

Given a set of points

$$x_0, x_1, x_3, ..., x_n$$

the moving window if set to the whole dataset will take the standard deviation between points x_0 and x_1 and store this value as s_0. Then we take the standard deviation of points x_0, x_1 and x_2 denoting this as s_1. Comparing s_0 and s_1 against each other, if the standard deviation is considerably larger than the previous standard deviation, then there has been a significant change in the data and a peak can appear.

This acts as a discretized derivative, being able to find areas of significant change within the dataset. Furthermore, creating an average standard deviation for the whole set can show how much change occurs over the whole dataset. A comparison of this against the rolling standard deviation can show the proportional change of the local rolling standard deviation against the total change across the dataset. Then it can be inferred that there has been a significant change in the data, potentially indicating a peak.

$$s_1 >> s_0 \implies peak \tag{2.2}$$

This continues until every point $x_0 ... x_n$ has its standard deviation calculated. Peaks should occur at the points with the largest standard deviation between that point and the rest of the dataset.

This method of calculating peaks essentially takes a standard deviation of an nth value and every value before it in the dataset. It iterates this until all values from the dataset have had their standard deviation taken against all previous values. In short, the algorithm computes the standard deviation of each point and all previous points. This happens until every value in the dataset has been analyzed.

The rolling standard deviation can have its window changed to different intervals. For instance, instead of taking the standard deviation of the 100th value with every value before, it could take the standard deviation with the previous 50 values. This is useful when a large peak is at the beginning of the dataset because values after a large peak will have much smaller standard deviations when compared to the rest of the set.

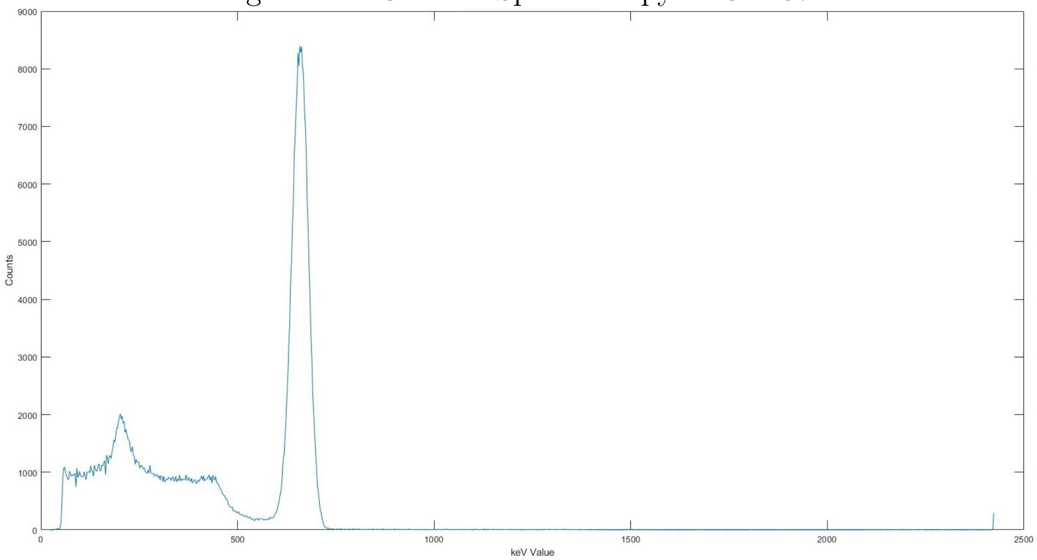
2.1.1 Testing on Cs-137

The graph on Figure 2.1 shows the Gamma Spectroscopy of Caesium-137 (Cs-137). In the spectra you can see a clear and defined peak at approximately 660 keV. This made it suitable for testing the first version of the algorithm I had designed, as it had the greatest and most identifiable peak compared to the other isotopes we had been provided with.

I decided to use an array to store the top 25 standard deviations, aiming to find that one of these 25 standard deviations would contain a standard deviation around 660 keV. This array can be accessed within modern high-level programming languages such as MatLab. When the array is accessed after running the rolling standard deviation, this version takes a rolling standard deviation of the entire set; we can see clearly that five values within the array are around 660 keV.

The data provided to us by LabLogic Systems takes measurements and the same discrete energy values every time, and since it is discrete, sometimes the peaks can be located at ±5 around the expected peak value. To account

Figure 2.1: Gamma Spectroscopy of Cs-137



for this I included boolean statements in later versions of the program that check within ±5 of where a peak is expected to occur. As seen below.

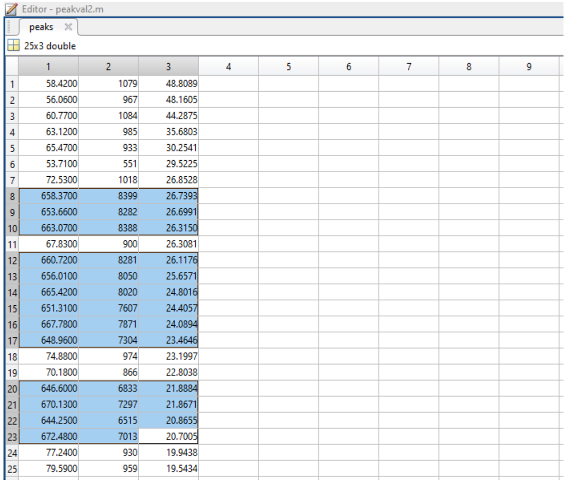
```
cs137=peaks(peaks<665.0 & peaks>655.0);
if isempty(cs137)== false
    disp("sample is cs137")
```

2.1.2 Identifying a radionuclide

One method to identify a radionuclide is to use the collated data from the LNHB[1]. After creating an array containing the suspected peaks of gamma spectra the density of these values can be compared. If these values correspond to the data on the LNHB of where a gamma transition occurs, we can identify the radionuclide.

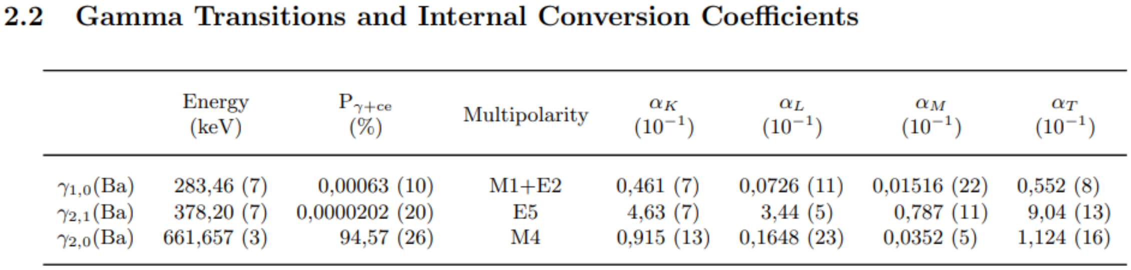
From the LNHB it's know that a peak is expected at 661 keV. Using this fact, Cs-137 can be identified using the gamma spectra by comparing this to the array of the top 25 highest standard deviations, as seen in Figure 2.2. Looking at our array, we can see clearly that around 661 keV, there exists a high density of standard deviations. We know that any spectra that have a high density of standard deviations around 661 keV must be Cs-137. This can be repeated for any radionuclide, using the probabilities of gamma-ray transitions from the LNHB.

Figure 2.2: Array of top 25 highest standard deviations for Cs-137



	1	2	3	4	5	6	7	8	9
1	58.4200	1079	48.8089						
2	56.0600	967	48.1605						
3	60.7700	1084	44.2875						
4	63.1200	985	35.6803						
5	65.4700	933	30.2541						
6	53.7100	551	29.5225						
7	72.5300	1018	26.8528						
8	658.3700	8399	26.7393						
9	653.6600	8282	26.6991						
10	663.0700	8388	26.3150						
11	67.8300	900	26.3081						
12	660.7200	8281	26.1176						
13	656.0100	8050	25.6571						
14	665.4200	8020	24.8016						
15	651.3100	7607	24.4057						
16	667.7800	7871	24.0894						
17	648.9600	7304	23.4646						
18	74.8800	974	23.1997						
19	70.1800	866	22.8038						
20	646.6000	6833	21.8884						
21	670.1300	7297	21.8671						
22	644.2500	6515	20.8655						
23	672.4800	7013	20.7005						
24	77.2400	930	19.9438						
25	79.5900	959	19.5434						

Figure 2.3: Gamma-ray transition data for Cs-137



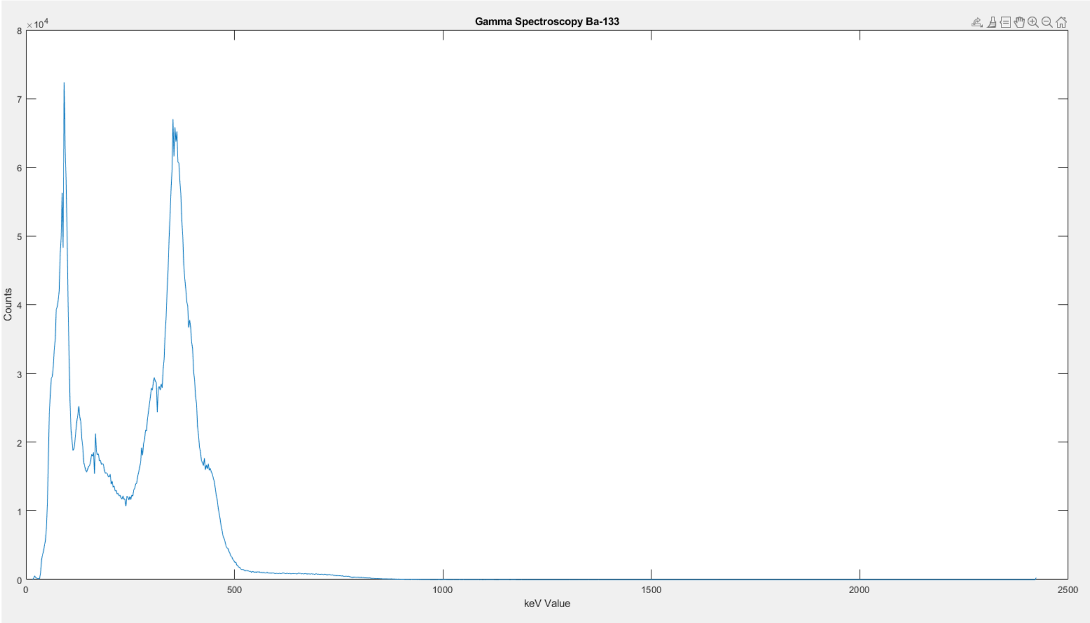
2.2 Gamma Transitions and Internal Conversion Coefficients

	Energy (keV)	$P_{\gamma+ce}$ (%)	Multipolarity	α_K (10^{-1})	α_L (10^{-1})	α_M (10^{-1})	α_T (10^{-1})
$\gamma_{1,0}$(Ba)	283,46 (7)	0,00063 (10)	M1+E2	0,461 (7)	0,0726 (11)	0,01516 (22)	0,552 (8)
$\gamma_{2,1}$(Ba)	378,20 (7)	0,0000202 (20)	E5	4,63 (7)	3,44 (5)	0,787 (11)	9,04 (13)
$\gamma_{2,0}$(Ba)	661,657 (3)	94,57 (26)	M4	0,915 (13)	0,1648 (23)	0,0352 (5)	1,124 (16)

2.1.3 Testing on Barium-133

After testing the rolling standard deviation algorithm on Caesium-137, I decided to test the program on Barium-133. This data had another noticeable peak when plotted. However, the peak of Barium-133 appears much closer in the dataset to another peak. This would test the algorithm's bias towards early onset peaks. Since the program could detect both peaks it was evident that the program was working.

Figure 2.4: Gamma Spectroscopy of Ba-133

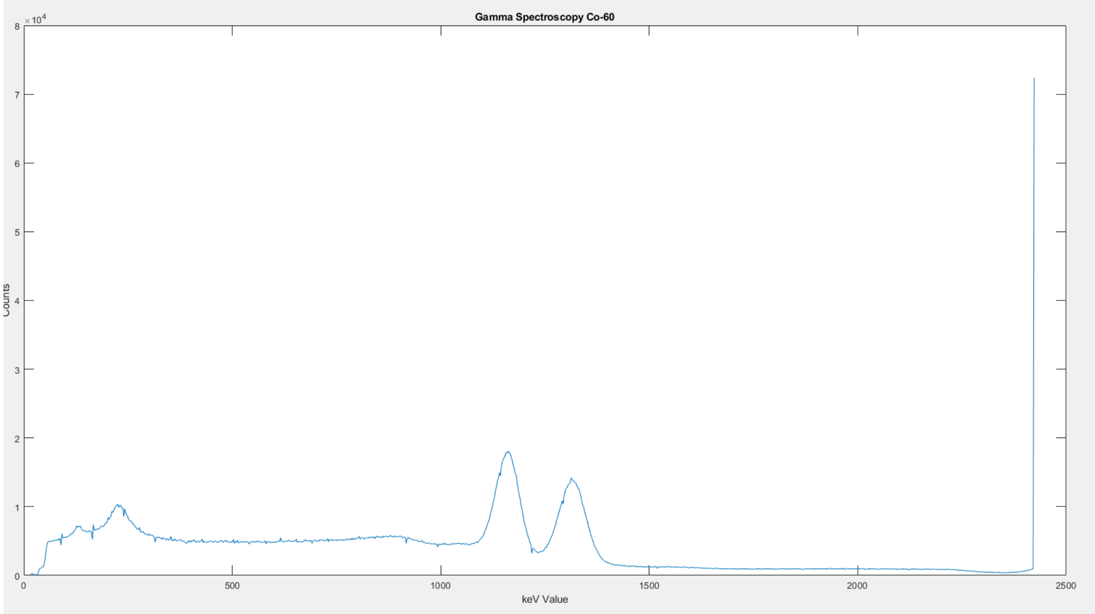


```
ba133=peaks(peaks>88.0 & peaks<92.0);
cs137=peaks(peaks<665.0 & peaks>655.0);
if isempty(cs137)== false
    disp("sample is cs137")
elseif isempty(ba133) == false
    disp("sample is ba133")
```

Upon implementing the boolean statements into the program, which correspond to the expected peaks from the LNHB, the program was be able to search the array of top 25 standard deviations. If there exists a peak within the array, the boolean statements will return a true or false statement which can tell us if it is either Cs-137 or Ba-133.

2.1.4 Problems arising with Cobalt-60

Figure 2.5: Gamma Spectroscopy of Co-60



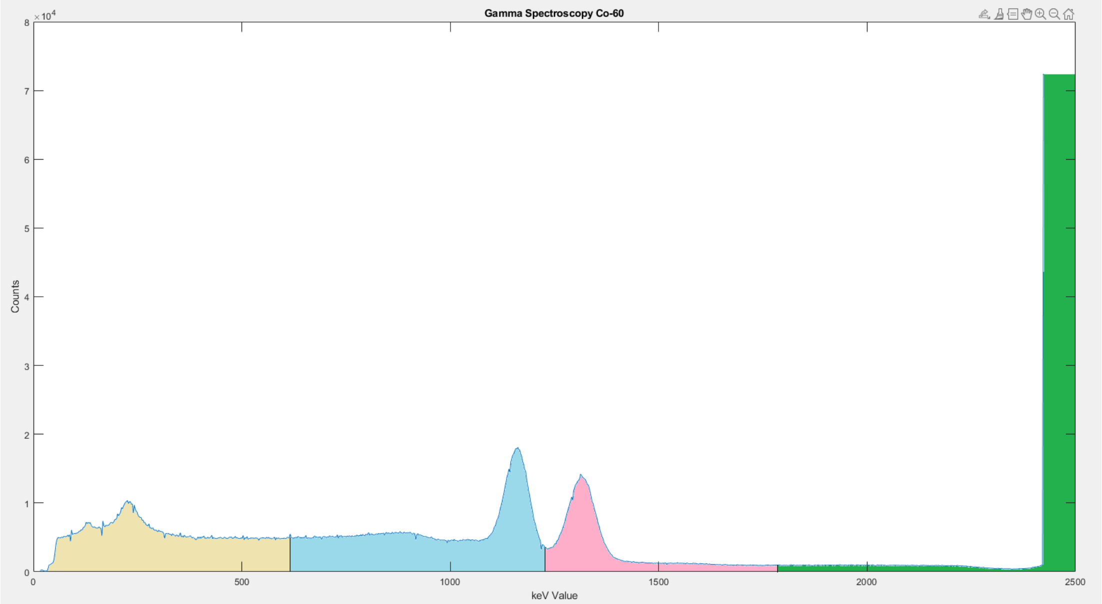
The data provided by LabLogic Systems and the data accessible on the LNHB do not always correlate, as can be seen with Cobalt-60 (Co-60). When we plot Cobalt-60 in MatLab or using the tool Laura, it can be seen that it has a significant peak at the start of the spectra, which is not mentioned in the LNHB materials. (Note this peak is significant locally, as the spectra for Co-60 do not contain as much spiking as Cs-137 or Ba-133.)

The features of these spectra posed a problem to the program I had written as the bias towards initial peaks began to become apparent. The only identifiable feature known to Co-60 was so far into the dataset after a very significant peak, which meant that the program could not detect the peak. This caused me to rethink my approach of taking the rolling standard deviation of the whole set.

Upon further research, found in an article written by the US-EPA (United States Environmental Protection Agency) [3], it can be noted that the first large peak on the gamma spectra, approximately at 1310 keV, is a feature of Co-60. This peak can be used to identify Co-60 and now can be implemented into the program.

The dataset is a power of 2, since it is 1024 values long, which means that I could partition the dataset into four sections of 256 values. Partitioning the data into 256 values and clearing the rolling standard deviation as I go would allow me to search specific regions and the standard deviations of each section would be more reflective of the true local peaks, as opposed to the heavily weighted first peak of Co-60.

Figure 2.6: Gamma Spectroscopy of Co-60 split into 256 value worth sections



From the beginning, Marshall and I were aware of this issue from the outset of employing a rolling standard deviation. We had discussed, methods of mitigating the weighting problem. Problems in dealing with Co-60 became particularly pronounced as the peaks proved to be most challenging to detect in the LabLogic Systems data. This problem should be considered if this approach is to be taken forward by LabLogic Systems.

2.1.5 Double Rolling Standard Deviation

After splitting the 1024-long dataset into two and running the rolling standard deviation on both halves, I found that the issues surrounding Co-60 had been resolved. This meant that instead of splitting the dataset into four, we only needed to split the dataset once. Peaks in both halves of every dataset were now much more apparent. Now that I could identify Co-60 from its gamma spectrometry, it meant that I could move on to testing Na-22.

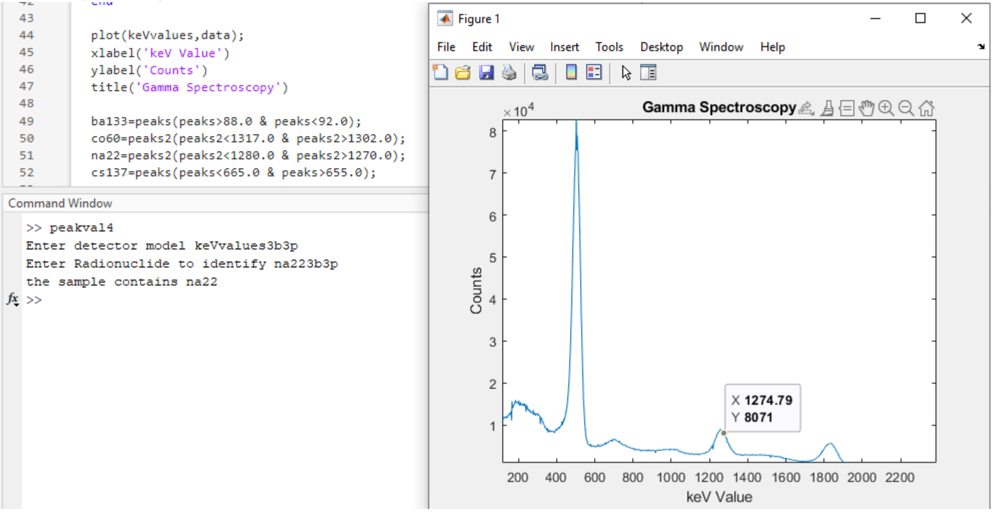
By making this change to the program, Na-22 also worked by using the second window for the rolling standard deviation. By splitting the data into two sections and using the rolling standard deviation on both halves, it improved all gamma spectra analysis. The weighting problem had been solved. The program then worked for every sample provided by LabLogic Systems.

The reason for splitting the dataset into two was that I had cleared the standard deviation value after computing the first 512 values, meaning that the standard deviations further along in the dataset became more prominent.

2.1.6 Brief Explanation of Program

The final version of my program uses two rolling standard deviations, to create an array of the top 15 standard deviations in each half of the dataset. If within these top 15 standard deviations there exists a high density of keV values around where a peak is expected for a given isotope, the program will identify these spectra as that isotope and return this to the user. Every radionuclide has unique properties where the gamma transitions occur. Using this property, we can identify any given radionuclide. The top standard deviations, which indicate a peak, can be utilised to complete this identification. The outputs of the program and the data from the LNHB are included in Figure 2.7 and Figure 2.8.

Figure 2.7: Gamma Spectroscopy of Na-22, Showing Peak and Result of Running Program

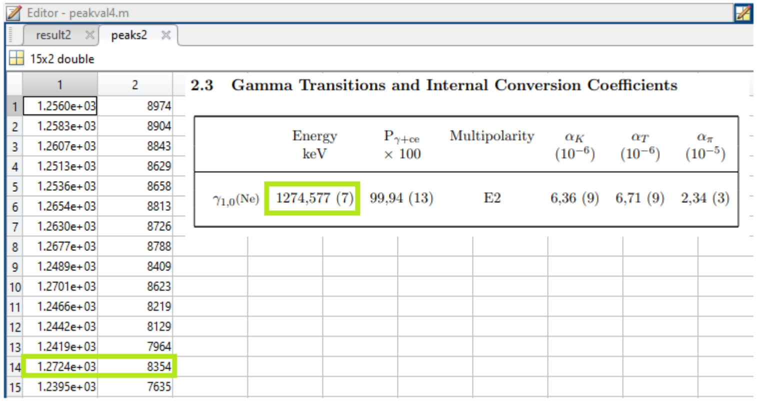


```
    end

    plot(keVvalues,data);
    xlabel('keV Value')
    ylabel('Counts')
    title('Gamma Spectroscopy')

    ba133=peaks(peaks>88.0 & peaks<92.0);
    co60=peaks2(peaks2<1317.0 & peaks2>1302.0);
    na22=peaks2(peaks2<1280.0 & peaks2>1270.0);
    cs137=peaks(peaks<665.0 & peaks>655.0);
```

Command Window

```
>> peakval4
Enter detector model keVvalues3b3p
Enter Radionuclide to identify na223b3p
the sample contains na22
>>
```

Figure 2.8: The Data Behind the Analysis of Na-22 and Data from the LNHB Showing the Transition Occurring at 1274 keV



	1	2
1	1.2560e+03	8974
2	1.2583e+03	8904
3	1.2607e+03	8843
4	1.2513e+03	8629
5	1.2536e+03	8658
6	1.2654e+03	8813
7	1.2630e+03	8726
8	1.2677e+03	8788
9	1.2489e+03	8409
10	1.2701e+03	8623
11	1.2466e+03	8219
12	1.2442e+03	8129
13	1.2419e+03	7964
14	1.2724e+03	8354
15	1.2395e+03	7635

2.3 Gamma Transitions and Internal Conversion Coefficients

	Energy keV	$P_{\gamma+ce}$ × 100	Multipolarity	α_K (10^{-6})	α_T (10^{-6})	α_π (10^{-5})
$\gamma_{1,0}$(Ne)	1274,577 (7)	99,94 (13)	E2	6,36 (9)	6,71 (9)	2,34 (3)

2.2 'Difference Search' Formulated By Barnaby Marshall

2.2.1 Introduction to method

The other collaborator in this project, Barnaby Marshall, has formulated his own approach to the peak identification algorithm. Marshall's algorithm uses the difference of the Gaussian points in the form

$$\frac{\Delta y}{\Delta x} = \frac{y_{n+1} - y_n}{x_{n+1} - x_n} \tag{2.3}$$

where y_{n+1} is the next point in the data set on the y-axis. The y axis in this case is the keV counts. The x-axis is the value of each keV value. Taking an array of these differences, it can be seen that the derivative of the Gaussian points increases over the peaks. Any noticeable deviations in these values will indicate the presence of a peak. This method of peak identification is discussed in Gordon R. Gilmore's book Practical Gamma-ray Spectrometry. [2]

2.2.2 Program Explained

Marshall's program uses a 'difference search' to find significant linear gradients. The density of these significant linear gradients helps to determine the peaks, as around the peaks there will be a greater density of significant linear gradients.

Marshall's program contains several separate functions, which all perform a specified task. Marshall's program initially starts by taking the numerical derivative of all points, as seen in Equation 2.3. Then, by storing these in an array, the program calls the second function.

The second function eliminates any linear gradients which are not significant. To do this, Marshall uses a pre-chosen value that works for most spectra. Marshall hypothesises that there could be a background reduction coefficient that exists, however he has stated that he has not yet found one. The current version of Marshall's program uses the pre-chosen value assigned by Marshall. The second function then eliminates these linear gradients that are insignificant by setting their values to 0.

Marshall's third function then finds the density between all non-zero values and sections with the greatest density of non-zero values that are deemed to be peaks within his program. Marshall currently takes any density greater than 20 in length to mean that there has been a significant change. These are further refined in later functions.

Marshall's function returns a graph which has a highlighted region; this highlighted region is the peak that Marshall's program has determined.

2.2.3 Marshall's Comments

After discussing with Marshall, I asked him one strength and one weakness of his program. Marshall stated that his program struggled with peaks that are interconnected, as seen in Co-60, this is something that Marshall said he would like to work on in the future. Marshall asserts that one strength of his program was that it could identify peaks well and display this visually.

2.2.4 Brief Comparison

Marshall and I both have developed working programs which utilise different methods to identify peaks. My program uses the LNHB [1] data primarily to find the peaks and return the radionuclide which the gamma spectra most resembles. Marshall's program uses numerical analysis to find the peaks accurately and display this in a digestible format.

Chapter 3

Nuclear Decay

3.1 Background

Gordon Gilmore asserts in his book Practical Gamma-ray Spectrometry[2], that 'Radionuclides are, by definition, unstable and decay by one, or more, of the decay modes.' This decay process has been considered when calculating the purity of a given radioactive isotope. On the LNHB website [1], information and data relating to the decay process of radionuclides is available. Which also contains the probabilities of each decay process occurring. In future versions of the algorithm, produced by myself and Barnaby Marshall, the program will be able to utilise our peak identification to detect the transition that has occurred and what the radionuclide we were observing has decayed into.

3.2 Half-life Decay

Half-life is the term used to describe the length of time for one-half of a radionuclide sample to decay[2]. When a radionuclide decays into another radionuclide, we reference the initial radionuclide as the parent. The radionuclides produced during and after the decay are referred to as the daughter radionuclides. It is important to note that the half-lives of the parent and daughter will differ, for this reason the half-life will be denoted as $\boldsymbol{t_{\frac{1}{2}}}$ **parent** and $\boldsymbol{t_{\frac{1}{2}}}$ **daughter** for their respective half-lives.

3.2.1 The Mathematics of Decay

Importantly, we must consider the mathematics of the decay process. Gilmore formulated the following decay equations in his book 'Practical Gamma-ray Spectrometry'[2]. "Radioactive decay is a first-order process." [2] To formulate a decay equation are needed the number of atoms present in the sample here denoted as $\boldsymbol{N}$ in Equation 3.1.

$$\boldsymbol{A = \frac{-dN}{dt} = \lambda N} \tag{3.1}$$

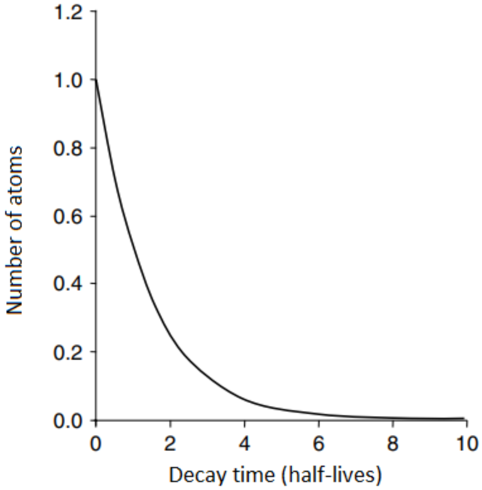
The rate in Equation 3.1 is known as the activity, which is denoted as $\boldsymbol{A}$ in the equation. Lambda, the decay constant, has units reciprocal time $[\boldsymbol{T}]^{\boldsymbol{-1}}$. Using the decay constant, we can formulate an equation to express the half-life of the particular radionuclide, denoted as $\boldsymbol{t_{\frac{1}{2}}}$ in Equation 3.2.

$$\boldsymbol{t_{\frac{1}{2}} = \frac{\ln 2}{\lambda}} \tag{3.2}$$

Solving the first order equation we obtain Equation 3.3. In Equation 3.3, $\boldsymbol{N_0}$ represents the number of atoms at time 0, and $\boldsymbol{N_t}$ represents the number of atoms at time t.

$$\boldsymbol{N_t = N_0 e^{-\lambda t}} \tag{3.3}$$

Figure 3.1: Equation 4.3 plotted



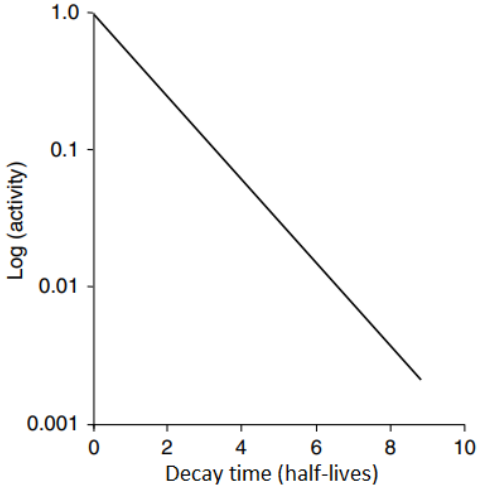
Since $\boldsymbol{N}$ is proportional to activity $\boldsymbol{A}$ an expression for activity can be derived, expressed as an exponential function.

$$\boldsymbol{A_t = A_0 e^{-\lambda t}} \tag{3.4}$$

Taking logarithms of the activity, Equation 3.5 is obtained.

$$\boldsymbol{\ln A_t = \ln A_0 - \lambda t} \tag{3.5}$$

Figure 3.2: Equation 4.5 plotted



3.3 Impact of half-life on gamma spectra for Cs-137

The half-life of Cs-137 is related closely to the gamma spectrometry of Cs-137. Gordon Gilmore's book 'Practical Gamma-ray Spectrometry' [2] explicitly states, 'Normally, the 661.7 keV gamma-ray appears with the half-life of the Cs-137.' 661.7 keV is a peak of Cs-137 and as such, the half-life of Cs-137 should be considered when analysing the spectrometry. When Cs-137 decays it eventually reaches secular equilibrium. Secular equilibrium is a state that occurs when the half-life of the parent nuclide is significantly longer than the daughter nuclide, as seen in Equation 3.6.

$$t_{\frac{1}{2}} \text{ parent} >> t_{\frac{1}{2}} \text{ daughter} \tag{3.6}$$

When Cs-137 decays it will decay into Ba-137 with 100% probability. During this process, it will reach Secular equilibrium. The half-life of Ba-137 is significantly shorter than the half-life of Cs-137, implying that even past the half-life of Cs-137 we will still be able to see peaks relating to Cs-137 at 661.7 keV. Gamma spectra on Ba-137 however, would be needed to confirm this. Further research would be beneficial to LabLogic Systems as it could extend the usage of Cs-137, on the othe it could mean that Ba-137 could be misidentified as Cs-137 as Cs-137 could still remain in samples of Ba-137 and could go undetected through the identification method produced by myself and Marshall.

Chapter 4

Algorithm

4.1 Code Walkthrough

In this section, I will discuss the program I wrote in MATLAB, making comments where necessary and providing the inputs needed to run the program.

The first section of the program takes the input of text files. These text files contain the data provided by LabLogic systems and are then imported into arrays. These arrays are labelled according to which radionuclide the data belongs to. Additionally, these arrays contain the name of the detector model used to collect the data. For example, cs1373b3p can be decomposed to be Cs-137 3x3 planar. Since MATLAB does not allow 'x' in array names adjacent to numbers, instead of 'x', the program uses 'b'. Arrays containing 2b2p are short for 2x2 planar, and 2b2w is short for 2x2 well.

```
keVvalues3b3p = importdata("keVvalues3x3.txt");
cs1373b3p=importdata("cs137-3x3.txt");
ba1333b3p=importdata("ba133-3x3.txt");
na223b3p=importdata("na22-3x3.txt");
co603b3p=importdata("co60-3x3.txt");

keVvalues2b2p = importdata("keVvalues2b2p.txt");
ba1332b2p = importdata("ba1332b2p.txt");
cs1372b2p = importdata("cs1372b2p.txt");
co602b2p = importdata("co602b2p.txt");

keVvalues2b2w = importdata("keVvalues2b2w.txt");
ba1332b2w = importdata("ba133well.txt");
cs1372b2w = importdata("cs137well.txt");
```

```
co602b2w = importdata("co60well.txt");
na223b2w = importdata("na22-2b2w.txt");
```

The next section of the code asks the user to select the detector type used, whether 2x2 planar or 3x3 planar. The user is then asked which isotope they would like to analyse (the user could alternatively import a different text file.) If they would like to analyse a spectra without knowing its identity. Finally, the variable q takes the size of the count array, known as data.

```
keVvalues = input("Enter detector model ") ;
data = input("Enter Radionuclide to identify ");
q=(size(data,2));
```

The following code is the code for the rolling standard deviation of the data. This section takes the standard deviation iteratively of the first point in the array data, then the first and second points, followed by the points x_0,x_1 and x_2. It then stores these in an array which I have labelled result and result2. The first for loop repeats over half of the array data and the second for loop repeats over the second half. The reason for doing this process twice is that for certain isotopes, such as co60, the program will create higher standard deviations weighted towards the beginning of the loop. Clearing the loop allows the rolling standard deviation to more accurately identify peaks in the second half of the dataset.
The variable 'sd' takes the standard deviation of the current point x_i and 'average' takes the average of all points before and including x_i.

```
result =[];
sd=0.0;
sum=0.0;
average=0.0;
for i = 1:q/2
    sum = sum + data(i);
    average = sum/(i);
    sd = sqrt((data(1,i)-average)^2)/(i);
    result(end+1)=(sd);
end

result2 =[];
sd=0.0;
sum=0.0;
average=0.0;
for i = ((q/2)+1):q
```

```
    sum = sum + data(i);
    average = sum/(i);
    sd = sqrt((data(1,i)-average)^2)/(i);
    result2(end+1)=(sd);
end
```

Variables ‘I’ and ‘J’ are the top 15 highest standard deviations from the loop that created result and result2 respectively. Finding the positions of these standard deviations, relative to the corresponding keV values, is important to return the peak and label the isotope at the end of the program.

```
I=maxk(result,15);
J=maxk(result2,15);
```

The loop in the following code writes the keV values and their corresponding counts to a 2d array. This means that the 2d array contains the top highest standard deviations from earlier, which are the peaks of the gamma spectra. These keV values and counts are then written to peaks and peaks2 respectively.

```
peaks=[];
peaks2=[];
for p = 1:15
    linearIndices1 = find(result==I(1,p));
    linearIndices2 = find(result2==J(1,p));
    peaks=[peaks;keVvalues(1,linearIndices1),data(1,linearIndices1)];
    peaks2=[peaks2;keVvalues(1,(linearIndices2+(q/2))),data(1,(linearIndices2+512))];
end
```

The following code plots the gamma spectra. Similar to the program Laura, the gamma spectra are displayed in a window when the program has finished running. Axes are labelled with appropriate labels relating to the counts and keV values.

```
plot(keVvalues,data);
xlabel(’keV Value’)
ylabel(’Counts’)
title(’Gamma Spectroscopy’)
```

Below are the criteria for identifying which radionuclide has been analysed. This data is taken directly from the LNHB[1]. The criteria for Cs-137 states that, if there exists a keV value in peaks between 655 and 665, then it must be Cs-137. This is because only Cs-137 would have a peak between 655 and 665.

```
ba133=peaks(peaks>88.0 & peaks<92.0);
co60=peaks2(peaks2<1317.0 & peaks2>1302.0);
na22=peaks2(peaks2<1280.0 & peaks2>1270.0);
cs137=peaks(peaks<665.0 & peaks>655.0);
```

The final section of code below is a collection of selective statements, which determine what to return to the user. If the criteria for Ba-133 is satisfied, then the variable ‘ba133’ will not be empty, then the program will return a statement telling the user that the sample must contain Ba-133.

```
if isempty(cs137)== false
    disp("the sample contains cs137")
elseif (isempty(ba133) == false && length(ba133)>=length(co60))
    disp("the sample contains ba133")
elseif (isempty(co60) == false && length(co60)>length(ba133))
    disp("the sample contains co60")
elseif isempty(na22) == false
    disp("the sample contains na22")
else
    disp("error")
end
```

4.2 Inputs And Results

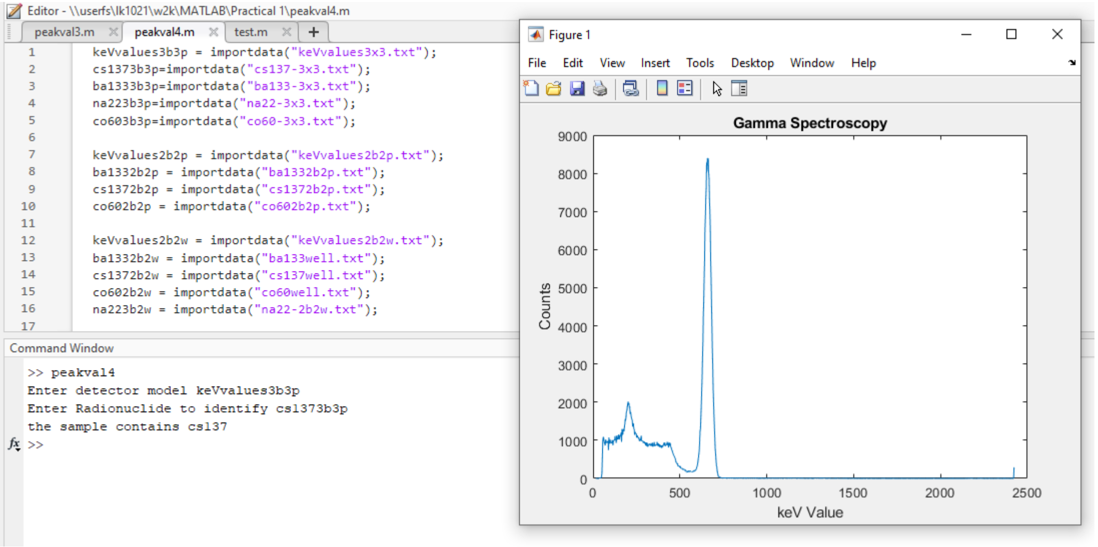
To run this program, only two inputs are needed, the first input is the detector type. The detector types are listed below.

```
Input           Detector
keVvalues3b3p - 3x3 planar
keVvalues2b2p - 2x2 planar
keVvalues2b2w - 2x2 well
```

The user will be prompted to input the radionuclide they want to identify, this could be a text file of their own or one of the text files that I have imported into the program. This text file should contain all the counts of the energy values.

As can be seen in Figure 4.1, the program will then output the data plotted against the corresponding keV values. Additionally, the program will return the radionuclide that it identifies.

Figure 4.1: Outputs Of The Program



```
keVvalues3b3p = importdata("keVvalues3x3.txt");
cs1373b3p=importdata("cs137-3x3.txt");
ba1333b3p=importdata("ba133-3x3.txt");
na223b3p=importdata("na22-3x3.txt");
co603b3p=importdata("co60-3x3.txt");

keVvalues2b2p = importdata("keVvalues2b2p.txt");
ba1332b2p = importdata("ba1332b2p.txt");
cs1372b2p = importdata("cs1372b2p.txt");
co602b2p = importdata("co602b2p.txt");

keVvalues2b2w = importdata("keVvalues2b2w.txt");
ba1332b2w = importdata("ba133well.txt");
cs1372b2w = importdata("cs137well.txt");
co602b2w = importdata("co60well.txt");
na223b2w = importdata("na22-2b2w.txt");
```

Command Window

```
>> peakval4
Enter detector model keVvalues3b3p
Enter Radionuclide to identify cs1373b3p
the sample contains cs137
>>
```

If the program is run in a modern high-level programming language, the arrays will be accessible and will arrays contain the peaks of individual spectra. Information about the standard deviations can also be accessed within the corresponding arrays that I have created. An example of this can be seen in Figure 2.8.

Chapter 5

Future Work And Conclusion

5.1 Future Work

5.1.1 Radionuclide Purity

One area of interest, which Marshall and I were not able to program, was the purity analysis. To calculate the purity, I would need to fit curves to the gamma spectra to accurately measure the inconsistency of the curves to the discrete data. Unfortunately, this was not achieved because the algorithm to identify the radionuclides and to find their peaks. Gilmore's book 'Practical Gamma-ray Spectrometry' [2] gives methods which would help to create such a program, and would be a vital source to use if the chance arises to create a purity identification program.

5.1.2 Further Study

Additionally, it would be interesting to have had the opportunity to analyse more radionuclides. The four radionuclides provided by LabLogic Systems were informative to my understanding of gamma-ray spectrometry, each different in properties and difficulty in identifying. I would like the opportunity to, in future work, work with more gamma spectra, this would allow me to learn more about spectrometry and help me to improve my program so that it would work on all radionuclides.

5.2 Conclusion

Although this report has only detailed a fairly basic introduction to Gamma-ray Spectrometry and peak identification methods, it has described in detail the methods by which I created my program and the reasoning behind each step. In future, I would like the opportunity to further develop this program to include more functionality, such as purity analysis. This report has successfully provided the framework for which anyone who reads this paper can understand the mathematics of peak identification and radionuclide identification.

5.2.1 Acknowledgements

Thank you to Kasia Rejzner for her feedback and suggestions on how to improve this project and take it further. Thank you also to Barnaby Marshall for his partnership and physics knowledge during this project.

Bibliography

[1] Laboratoire national henri becquerel, http://www.lnhb.fr/en/.

[2] Gordon R. Gilmore. *Practical Gamma-ray Spectrometry 2nd Edition.* John Wiley Sons, Ltd, 2008.

[3] Unknown. *EPA Facts about Cobalt-60.* https://semspub.epa.gov/work/HQ/174482.pdfm.