

Angular correlation measurements with a segmented clover detector



Sinegugu Happiness Mthembu

Department of Physics

University of Zululand

A thesis submitted for the degree of

Master of science in Physics

2019, February

Supervisor

Prof. S. S. Ntshangase, University of Zululand

Co-Supervisor

Dr. E. A. Lawrie, iThemba LABS

Acknowledgements

”Wenzile okuhle kodwa empilweni yami malibongwe! Malibingwe! igama leNkosi” - Joyous Celebration 22

- Prof. Sifiso Ntshangase, thank you for seeing potential in me and encouraging me to pursue a career in research, this would not have happened without you! Your support is greatly appreciated. Ngiyabonga.
- Dr. Elena Lawrie, thank you for the guidance, patience and your valuable perspectives of understanding physics, I have learned a lot from you, it has been an honor and a pleasure to have a supervisor like you.
- Dr. Tshepo Dinoko, you have been a mentor to me for a few years now, I thank you so much for being there for me, for guiding me and teaching me both physics and life lessons. I appreciate your support and willingness to help at all times.
- To the AFSEG Group; Daphney, Obed, Sive, Jason, Wiseman and Didier, thank you so much for the physics discussions and your readiness to help at all times. Your contributions to this work are highly recognized and appreciated.
- To my colleagues and office mates thank you so much for your support and help.
- To my dear friends thank you so much for making the hard times bearable, for the lunch discussions, the braais, and most importantly for your support. It is a pleasure knowing you all. A special thank you to Sandile

Jongile, for your tireless support from BSc. Honours to now, your motivation and most of all your ability to tell the truth and push me to do my best at all times.

- To Cebo Ngwetsheni, thank you so much for your support my love, and your faith in me. I appreciate having you in my life.
- To NRF/iThemba LABS and the Manus/MatSci program, thank you for the financial support and availing the facility for me to pursue my career in the Science research field.
- To my family, boMvelase, Mnisi wemvula, boMagali, ngiyabonga kakhulu ngothando nokungeseka kwenu, lokhu ngikwenzele nina. MaQwabe ube yinsika kimi ngiyabonga kakhulu Yeyeye. Mvelase baba ngiyabonga ngiyanconcoza ngokungikhuthaza kwakho ngasosonke isikhathi. Samkele, Sanele, Siyanda, Siphokuhle, ngiyabonga kakhulu ngothando lwenu kanye nokungethemba kwenu.

Declaration

I Sinegugu Happiness Mthembu herewith declare that I have produced this thesis making use of the specified references and without the prohibited assistance of third parties. This thesis is submitted for the Master of Science in Physics to the University of Zululand and has not been previously presented in identical or similar form to any other examination board or institution.

Signature..... Date.....

Abstract

Gamma-gamma angular correlation measurements are an important tool to determine (amongst other nuclear properties) the multipolarity and the multipole mixing ratio of a gamma-ray transition. In this work we use for the first time a segmented clover detector placed at close geometry to measure angular correlation functions. Gamma-gamma angular correlation analysis was carried out for the transitions of ^{133}Ba and ^{60}Co . It allowed to determine the angular correlation coefficients, a_2 and a_4 , and to establish the multipole mixing ratios of several gamma-ray transitions. Codes to sort and analyze gamma-gamma angular correlation data were developed and a technique to deduce the mixing ratios was established. It was shown that the iThemba LABS segmented clover detector has an excellent capability for gamma-gamma angular correlations measurements, allowing one to determine multipole mixing ratios with high precision. The analysis of the gamma-gamma cascades of ^{133}Ba proved that with this detector one can determine the $M1+E2$ multipole mixing ratios for gamma-ray transitions with precision similar to that listed in the Nuclear data tables, where the results of many different experiments are combined and averaged. Note that our results were produced from ~ 12 hours of data collection for each sources that were used namely ^{60}Co and ^{133}Ba . The analysis of the $1173 - 1332$ keV cascade in ^{60}Co showed that using the segmented clover one can easily distinguish a stretched dipole from an unstretched quadrupole transition. Moreover, new results on the signs of the multipole mixing ratios

for some transitions in ^{133}Ba were obtained. In addition to measuring the multipole mixing ratios for the transitions in ^{60}Co and ^{133}Ba , an analysis was performed in order to investigate the possibility of a new 437 keV transition in ^{133}Ba but no evidence supporting such a transition was found.

Contents

List of Figures	x
List of Tables	xx
1 Introduction	1
2 Theory	4
2.1 Radioactive decay	4
2.2 Angular momentum selection rules	5
2.3 Weisskopf estimates	9
2.4 Angular distribution of gamma rays	12
2.5 Angular correlations	17
3 Experimental method and data analysis	22
3.1 Semiconductor detectors	22
3.2 Electromagnetic radiation interactions with matter	25
3.2.1 Photo-electric absorption	25
3.2.2 Compton scattering	26
3.2.3 Pair production	28
3.3 The iThemba LABS segmented clover detector	29
3.4 Experimental set-up and data capture	31
3.5 Data analysis	32
3.5.1 Merging the data	32

3.5.2	Gain drift and energy calibration coefficients	33
3.5.3	Crosstalk corrections	35
3.5.4	Building a tree	36
3.5.5	Angular properties of the iThemba LABS segmented clover	37
3.5.6	Singles spectra	42
3.5.7	Coincidence matrices	44
3.5.8	Coincidence gates	48
3.6	Background subtractions	50
3.6.1	Background under the gate	51
3.6.2	Compton scatter peaks	53
3.6.3	Random coincidences	58
3.6.4	Peak fitting	62
3.6.5	Efficiencies	65
3.6.6	Angular correlation intensities	66
3.7	Extracting the a_k coefficients and calculating δ	68
4	Results and discussion	72
4.1	The cascade of 1173 - 1332 keV in ^{60}Co	72
4.2	Gamma-ray cascades in ^{133}Ba	80
4.2.1	The 356 - 81 keV cascade	82
4.2.2	The 53 - 384 keV cascade	88
4.2.3	The 303 - 81 keV cascade	93
4.2.4	The 276 - 161 keV cascade	99
4.2.5	Summary of the angular correlation results for ^{133}Ba	104
5	Searching for a possible new 437 keV transition in ^{133}Ba	106
6	Conclusions	116
A		119

B	133
C	140
D	151
E	154
F	156
G	160
H	165
References	166

List of Figures

2.1	A transition from an initial state, with angular momentum I_i , to a final state with angular momentum I_f is shown. E_i and E_f are the energies of the initial and final states, m_i and m_f are the magnetic quantum numbers for the initial and final state and ψ_i and ψ_f are the wave functions for the initial and final states, respectively. The emitted gamma-ray is characterized by its energy E_γ , angular momentum L_γ , magnetic quantum number m_γ and parity π_γ	7
2.2	Dipole radiation patterns for pure(l,m) multipoles [20].	14
2.3	Quadrupole radiation patterns for pure(l,m) multipoles [20].	15
2.4	A transition from a $I= 1^+$ to $I= 0^+$ state. The initial state has 3 single possible substates of $m = \pm 1$ and $m = 0$	16
2.5	An example of possible transitions for a $0^+ \rightarrow 1^+ \rightarrow 0^+$ cascade.	19
3.1	Band structures for electron energies in different materials, from [33]. . . .	23
3.2	The Compton scattering interaction. Picture taken from [37].	27
3.3	The process of pair production. Picture taken from [39].	28
3.4	The relative importance of the various interaction processes of gamma-rays with matter. Figure taken from [35].	29
3.5	The iThemba LABS segmented clover detector.	30

3.6	Gain drift shown on the left for 20 runs. The centroids of the peaks changed by ~ 12 channels. On the right the gain drift corrections have been implemented. The centroids of the peaks for all the runs remains the same. . . .	34
3.7	A spectrum of double-hit events registered in segments 5 and 7 of crystal B in addback mode, (that is the energy in this spectrum is given by $E = E_5 + E_7$) and (a) before crosstalk correction, (b) after crosstalk correction [45].	36
3.8	The 122 different measuring angles for all 992 segment pairs.	38
3.9	First interaction position in segment 6 of crystal D for the 1332 keV gamma-ray from a ^{60}Co source for events when the gamma-ray was fully absorbed in segment 6; (a) z -position or depth in the crystal, (b) y -position and (c) x -position.	39
3.10	Calculating the angle between the two segments for angular correlation measurements.	40
3.11	Gain-matched direct spectrum of ^{133}Ba for crystal A segment 1.	42
3.12	A portion of the direct spectrum of ^{133}Ba crystal A segment 1 showing the gamma rays of interest.	43
3.13	Gain matched direct spectrum of ^{60}Co decay for crystal A segment 2. . . .	43
3.14	A portion of the direct spectrum of ^{60}Co decay for crystal A segment 2 showing the gamma rays of interest.	44
3.15	A time spectrum for coincidence events between detector 1 and detector 12. The spectrum is incremented when detector 1 registers a gamma-ray of energy $E_{\gamma 1}$ in coincidence with $E_{\gamma 2}$ registered on detector 12, and represents the time difference between the time stamps for the two gamma rays. . . .	45
3.16	The gamma-gamma angular correlation matrix for segment 2 (x-axis) and segment 1 (y-axis) for the ^{133}Ba decay.	46

3.17	A summed gamma-gamma angular correlation matrix corresponding to the first angle $\theta = 16^\circ$, containing 8 angular correlation matrices.	47
3.18	A summed gamma-gamma angular correlation matrix corresponding to $\theta = 16^\circ$	47
3.19	Setting a gate on the gamma-gamma angular correlation matrix. In this case the gate is set on the 303 keV gamma-ray, registered in segment 2 (x -axis), creating a gated spectrum in segment 1 (y -axis), shown in Fig. 3.20. The coincidence peak is indicated by the black circle.	48
3.20	A spectrum gated on the 303 keV gamma-ray. The observed coincidence peaks are 81 and 53 keV transitions in coincidence with the 303 keV gamma-ray. The spectrum illustrates the different peaks produced by different backgrounds. The peaks are labeled in the spectrum according to the type of background that produces them. The spectrum was extracted from the summed angular correlation matrix for angle $\theta = 16^\circ$	49
3.21	A gate set on the 356-keV photopeak and background gates set on the left and on the right of the peak of interest. The gated spectra obtained are shown in Fig. 3.22.	52
3.22	A spectrum gated on the 356-keV photopeak, shown in blue and a sum of the two spectra gated on the background on the left and on the right of the photopeak, shown in red.	52
3.23	An example of two Compton scatter peaks.	53
3.24	The 1274 keV and the 1460 keV Compton scatter peaks produced in the spectra gated on the 303 keV (in navy), 283 keV (red) and 323 keV (green). The green and red spectra are a result of the gates on the left and right of the peak.	54
3.25	An example of transitions that result in a Compton scatter peak under the peak of interest.	55

3.26	(a) A gate set on the 303-keV photopeak in ^{133}Ba , with a gate width of 12 channels. (b) Gate set on the 303-keV photopeak in ^{133}Ba , with a gate width of 20 channels. (c) Gate set on the 303-keV photopeak in ^{133}Ba , with a gate width of 30 channels.	57
3.27	(a) The 160-keV coincidence peak lying on top of a Compton-scatter background. The blue histogram was generated by gating on the 276-keV gamma ray. The red and green histograms were generated by gating on the background under the 276-keV peak (gating on left and right of the 276-keV peak). (b) Same as (a) but the two Compton-scatter background histograms were shifted to align with the coincidence peak. The pink histogram is the sum of the background spectra.	58
3.28	The Compton-scatter corrected spectrum. The gated spectrum and the background-gated spectrum are shown in Fig. 3.27a.	59
3.29	A coincidence spectrum generated by gating on 356 keV in the angular correlation matrix for $\theta = 16^\circ$, shown in navy. The spectrum in red is the normalized direct histogram, which is used to correct for random coincidence events.	60
3.30	The same as in Fig. 3.29, but expanded on the y-axis to better illustrate the random coincidence peaks. The bump at ~ 155 keV is a result of the Compton scattering of the 511 keV annihilation gamma ray. The bumps at 62 and 70 keV are caused by the coincidence summing of the X-rays ($K_\alpha = 31$ keV and $K_\beta = 35$ keV).	61
3.31	A spectrum gated on 356 keV after random coincidences correction.	62
3.32	A fit for the 356-keV gamma ray using a Gaussian and a first order polynomial. p0 is the height of the peak, p1 is the centroid of the peak, p2 is the width of the peak, p3 describes the height of the background.	63

3.33	The 356-keV photopeak. Background regions were selected on the left and on the right of the peak, shown in red.	64
3.34	Angular correlation fits for the 1173 - 1332 keV cascade, using data for: (a) all 122 angles, (b) the angles that correspond to combinations of front segments only.	67
3.35	A cascade of two gamma rays.	68
3.36	An a_2 versus a_4 plot for the possible spin assignments of $I_i = 0, 1, 2, 3, 4$ with varying mixing ratios.	69
3.37	A χ^2 versus $\text{atan}(\delta)$. The χ^2 minimum determines the value of δ . The data is for the 1173 - 1332 keV cascade in ^{60}Co . The 1332-keV transition is assumed to be a pure $E2$, while the 1173 keV transition is assumed to be of $E2(+M3)$ multipolarity. A mixing ratio of $\delta = -0.026^{+0.043}_{-0.041}$ was deduced.	70
3.38	An example of how the error bars on δ are determined from a χ^2 plot.	71
4.1	The level scheme of ^{60}Co populated by the decay of ^{60}Ni	73
4.2	The summed angular correlation matrix of ^{60}Co for the angle $\theta = 16^\circ$. The circle indicates the coincidence of the 1173 keV and 1332 keV gamma rays, where 1173 keV is fully absorbed in the detectors on the x -axis of the matrix, while 1332 keV is fully absorbed in the detectors on the y -axis of the matrix. The arrows indicate Compton scattering ridges.	74
4.3	The strong gamma-ray peak corresponds to the 1332 keV coincidence peak. The pink histogram is the final output after all background subtractions have been done. The small gamma-ray peak corresponds to 1173 keV and it is a result of mostly random coincidences and background under the gate.	75
4.4	The experimental data compared with the theoretical angular correlations functions for different cascades.	77

4.5	Experimentally measured $W(\theta)$ for the 1173 - 1332 keV cascade in ^{60}Co . Only the front segments were considered in the analysis. The data were fitted with Legendre polynomials (red) to extract the A_0 , a_2 and a_4 angular correlation coefficients.	78
4.6	χ^2 versus $\arctan(\delta)$ for the 1173 keV transition in the ^{60}Co decay. A mixing ratio of $\delta = -0.026^{+0.043}_{-0.041}$ was deduced.	79
4.7	The adopted level scheme of ^{133}Cs , populated by the decay of ^{133}Ba . It shows the gamma-rays of interest and their intensities and multipolarities, from [52].	81
4.8	The summed angular correlation matrix for $\theta = 15^\circ$. The red circle indicates the area of interest where the coincidence between the 356 keV and the 81 keV transitions appears. The black lines indicate where the gate was set.	82
4.9	A spectrum (in navy) generated by gating on the 356 keV transition in the summed angular correlation matrix for $\theta = 15^\circ$. The red spectrum was generated by setting gates on the left and on the right regions near the gate and represents the background under the gate. The green spectrum is the normalized direct spectrum used to remove the random coincidence contribution.	83
4.10	The background-subtracted gated on the 356 keV spectrum. It was produced using the summed angular correlation matrix for $\theta = 15^\circ$	84
4.11	A portion of the background-subtracted spectrum gated on 356 keV (shown in Fig. 4.10). The dotted lines were used to test different backgrounds under the 81 keV peak.	84
4.12	Experimental results fitted with an angular correlation function that used (i) the result of this work (green) and (ii) the accepted value in NNDC (red).	85

4.13	An angular correlation fit for the 356–81 keV cascade. Angular correlation coefficients of $a_2 = 0.036$ and $a_4 = -0.005$ were deduced.	86
4.14	The χ^2 versus $\arctan(\theta)$ plot for the experimental angular correlation coefficients of the $\frac{1}{2}^+ \rightarrow \frac{5}{2}^+ \rightarrow \frac{7}{2}^+$ cascade with a range of mixing ratios. The deduced value of δ is $\delta = -0.153_{-0.004}^{+0.017}$	87
4.15	A projection on the x -axis of the summed angular correlation matrix for angle $\theta = 75^\circ$	88
4.16	A projection on the x -axis of the summed angular correlation matrix for angle $\theta = 16^\circ$	88
4.17	The summed correlation matrix for angle $\theta = 16^\circ$. The red circle indicates the 53 - 384 keV coincidence peak. The black lines indicate the 384 keV gate on the y -axis.	89
4.18	A spectrum (in navy) generated by gating on the 384-keV transition in the summed angular correlation matrix for angle $\theta = 18^\circ$. The red spectrum was generated by setting gates on the left and on the right regions near the gate and represents the background under the gate. The green spectrum is the normalized direct spectrum used to remove the random coincidence contribution.	90
4.19	The background-subtracted spectrum gated on 384 keV, from the summed angular correlation matrix for angle $\theta = 16^\circ$	90
4.20	Experimental results were compared with the angular correlation functions that use $\delta = \pm 0.08$ for the 53-keV transition.	91
4.21	The angular correlation fit for the 53 - 384 keV cascade. Angular correlation coefficients of $a_2 = -0.090$ and $a_4 = -0.004$ were deduced.	92
4.22	The χ^2 versus $\arctan(\theta)$ plot for the experimental angular correlation coefficients of the $\frac{1}{2}^+ \rightarrow \frac{3}{2}^+ \rightarrow \frac{7}{2}^+$ cascade. A value of $\delta = +0.079_{-0.065}^{+0.074}$ is extracted for the 53-keV transition.	92

4.23	The summed angular correlation matrix for angle $\theta = 18^\circ$. The 303 keV gate is indicated by black lines and the region of interest for the 303 - 81 keV cascade is shown by a red circle.	93
4.24	As Fig. 4.18, but the gate is set on 303 keV. The coincidence peaks appear on top of a Compton-scatter background.	94
4.25	A portion of the 303 keV gated spectrum (corrected for randoms), with the 81 keV coincidence peak fitted with a Gaussian and a polynomial. The fit parameters are shown in the right corner of the figure, where $p0$ is the height of the peak, $p1$ is the centroid of the peak, $p2$ is the Gaussian parameter σ , and $p3$ is the height of the background above zero.	95
4.26	Experimental angular correlation data for the 303 - 81 keV cascade compared with the theoretical angular correlation functions assuming $\delta = \pm 0.022$	96
4.27	The experimental angular correlation data for the 303 - 81 keV cascade is fitted to extract the angular correlation coefficients. Angular correlation coefficients of $a_2 = -0.037$ and $a_4 = 0.003$ were deduced.	97
4.28	The χ^2 versus $\arctan(\theta)$ plot for the experimental angular correlation coefficients of the $\frac{3}{2}^+ \rightarrow \frac{5}{2}^+ \rightarrow \frac{7}{2}^+$ cascade with a range of mixing ratios. The deduced value of δ is $+0.068 \pm 0.036$	98
4.29	The summed angular correlation matrix for angle $\theta = 18^\circ$, with the 276 keV gate indicated by black lines and the region of interest for the 276 - 161 keV cascade is shown by a red circle.	99
4.30	Spectra generated by gating on 276 keV in the summed angular correlation matrix for $\theta = 18^\circ$. The 53 keV, 81 keV and 161 keV peaks are lying on a Compton-scatter background. The scaled direct spectrum is shown in green and the background-gated spectrum is shown in red. The Compton-scatter peaks shift in the red spectrum.	100

4.31	A portion of the 276 keV gated spectrum after background subtraction. The 161 keV peak is fitted with a Gaussian and a polynomial. The parameters are shown in the right corner of the figure, where $p0$ is the height of the peak, $p1$ is the centroid of the peak, $p2$ is the Gaussian parameter σ , and $p3$ is the height of the background.	101
4.32	Experimental angular correlation data for 276 - 161 keV cascade compared with the theoretical angular correlation functions assuming $\delta = \pm 0.96$	102
4.33	Angular correlation fit for the 276 - 161 keV cascade. Angular correlation coefficients of $a_2 = -0.641$ and $a_4 = -0.092$ were deduced.	102
4.34	The χ^2 versus $\arctan(\theta)$ plot for the experimental angular correlation coefficients of the $\frac{1}{2}^+ \rightarrow \frac{5}{2}^+ \rightarrow \frac{7}{2}^+$ cascade with a range of mixing ratios. The deduced value of the mixing ratio δ for the 161 keV transition is $+1.165^{+0.983}_{-0.504}$	103
5.1	The gamma-gamma angular correlation matrix for angle $\theta = 18^\circ$. The red arrow indicates the 437 keV Compton scattering ridge.	107
5.2	The gamma spectrum for segment 1 of crystal B. The box in red indicates the 437-keV peak. This spectrum can not distinguish between coincidence summing of multiple gamma-rays and a single gamma-ray transition. . . .	108
5.3	Summed spectrum from the direct spectra of the back segments 5, 6, and 8 of crystals A, B, C and D.	109
5.4	A cut along $x = y$ angular correlation matrix for $\theta = 18^\circ$. The projection of this cut on segment 1 is shown in Fig. 5.5 as the navy histogram. . . .	109
5.5	A projection spectra produced by a cut through $x = y$ (navy) of the angular correlation matrix for $\theta = 18^\circ$ and by a cut that is shifted by 20 keV, see Fig. 5.6 (red). The peaks represent Compton-scatter ridges, except the X-rays and the 81 keV peak. The gamma rays that correspond to each ridge are labeled with their energies.	110

5.6	A cut along $x = y$ shifted by 20 keV. The projection of this cut on segment 1 is shown in Fig. 5.5 in red.	111
5.7	Illustration of the Compton scattering of the 384-keV gamma ray in coincidence with the 53-keV transition, contributing to a 437-keV Compton ridge.	112
5.8	Projection of the Compton scattering ridge of the 384 keV gamma-ray (red) in the summed angular correlation matrix for angle $\theta = 18^\circ$. The peak at 28 keV is a Compton-scatter peak, $384 \text{ keV} = 356 \text{ keV} + 28 \text{ keV}$. This spectrum represents events in the 437-keV ridge, produced as illustrated in Fig. 5.7 left. The navy spectrum represents all Compton scatter events in the 437-keV ridge, produced by the 53 - 384 cascade as illustrated in Fig. 5.7 left and right. For more details see text.	113
5.9	Cuts along the 276 keV, 356 keV, 384 keV and 437 keV Compton ridges. The cuts are represented in different colors and labeled with the corresponding gamma-ray energies.	115
5.10	The projection of the cut along the 437-keV ridge in red. The spectrum in navy represents the Compton-scatter events from the 53 - 384 keV, 356 - 81 keV, and 276 - 161 keV cascades that contribute to the 437-keV ridge. The contributions from each cascade was deduced as described in the text, and as shown in Fig. 5.8 (navy) for the 53 - 384 keV cascade.	115

List of Tables

2.1	Different gamma-ray multipolarities depending on L and π . L is angular momentum taken away by the emitted gamma-ray and π is the parity of the gamma-ray.	8
2.2	The angular distribution functions, $W_{L,m}(\theta)$, for $L = 1$ and $L = 2$ and all possible values of m . θ is the angle at which the gamma ray is emitted with respect to the z -axis [20].	13
2.3	List of the first six Legendre polynomials.	18
3.1	Voltage parameters for each of the four crystals of the segmented clover. .	31
3.2	Detector ID assigned to the cores and segments of the detector.	41
4.1	Summarized results and accepted multipole mixing ratios from the NNDC [56].	104

Chapter 1

Introduction

Nuclear structure research is a study of the ground- and excited states of nuclei. The properties of nuclei can be experimentally measured using standard spectroscopic tools such as gamma spectroscopy where the transition energy, parity, multipole order, and the nuclear decay paths are determined. There are different methods to determine the spin and parity of the nuclear states (J^π), the angular momentum (L) of the emitted electromagnetic radiation, and the multipolarity (σL) and the multipole mixing ratio (δ) of a transition. Gamma-gamma angular correlation measurements of nuclear transitions are an important technique for measuring the aforementioned properties of nuclear states. However, the detection system used to perform the measurements determines the accuracy and the precision of the results. The use of angular correlations was first theoretically developed in the 1940's by Hamilton [1]. The first experiments were performed in 1947 by Brady and Deutsh [2]. Most of the angular correlation measurements were done in the 1970's using Na(I) and Ge(Li) detectors with only few measuring angles [3, 4, 5]. Since then the detector technology has advanced quite significantly and the modern detection systems have considerably better efficiency and better energy resolution [6].

The newest generation of gamma-ray spectroscopy detectors are the segmented germanium detectors [7, 8, 9, 10]. The iThemba LABS (Cape Town) research facility is operating

a segmented clover detector consisting of 4 crystals, each one 8-fold segmented [11]. This results in 32 segments, excluding the core contacts of the detector. The 32 segments were used as individual detectors allowing 32×31 segment-segment pairs which corresponds to 992 measuring angles for angular correlations. Since the crystal geometries of the detector are similar, segment-segment pairs that correspond to the same angle were summed together resulting to a reduced number of 122 different measuring angles that can be used for angular correlations.

Angular correlation experiments have not been done previously with a segmented clover at iThemba LABS. Moreover, such measurements have never been done with single segmented clover detectors in close geometry. Modern facilities with gamma-spectroscopy arrays that are constructed of segmented germanium detectors perform angular correlation measurements, however their detectors are placed at a large distance of $\sim 200 - 250$ mm from the source of gamma-rays [12, 13, 14, 15]. They use tracking to determine the interaction position of the gamma-rays. We are using the segmented clover at close distance because it allows: (i) increased efficiency due to the close distance; (ii) opening angle per segment that is similar to the opening angle of a similar detector at ~ 200 mm distance (iii) a large number of detecting angles. Using the segmented clover in close geometry and in a mode where the segments are regarded as individual detectors allows both high efficiency and high granularity. It presents an opportunity to develop a new precise method for gamma-gamma angular correlation measurements for the determination of spins and parities of nuclear states and multipole mixing ratios of transitions.

Generally, to carry out a precise measurement for the mixing ratio of a gamma-ray transition a large number of measuring angles is required. The use of the iThemba LABS segmented clover detector provides such large number of angles. This work focuses on developing an angular correlation analysis method with the iThemba LABS segmented clover

detector. The performance of the iThemba LABS segmented clover detector is evaluated by performing angular correlation measurements on several gamma-gamma cascades in ^{60}Co and ^{133}Ba . The segmented clover detector was placed in at ~ 40 mm from the source. Data were collected for a period ~ 12 hours with each of the ^{60}Co and ^{133}Ba radioactive sources.

Codes were developed to both sort the data and carry out the analysis. As the segments are close to each other there is a high probability that gamma-rays will Compton scatter between the segments. An analysis of the gamma-gamma coincidences was carried out to distinguish the true coincidence of two gamma-rays from the Compton scattering of one gamma-ray. Different background subtraction methods were implemented to separate true coincidence events from background.

The results show that angular correlation measurements with excellent precision can be carried out. In addition it provides new information on the signs of the mixing ratios of several transitions in ^{133}Ba . Precise measurements of the spin and parity of nuclear states and of the transition mixing ratios allow more stringent comparisons with nuclear models and their improvement.

Chapter 2 discusses the theoretical concepts of angular correlations relevant to this study. Chapter 3 discusses the interactions of electromagnetic radiation with matter, details of the set-up and data collection and the developed angular correlation analysis method. The experimental results and a comparison with the internationally accepted values for the mixing ratios are discussed in Chapter 4. Chapter 5 presents an analysis on the 437 keV Compton ridge in the angular correlation matrices. Some conclusions, remarks and plans for future work are summarized in Chapter 6.

Chapter 2

Theory

Two things are of major importance in this study: **(i)** the experimental techniques used to extract useful information from the collected experimental data and **(ii)** the theoretical description of the angular correlations which takes into account the relevant nuclear properties. This chapter explains in detail how the angular correlation function can be determined theoretically based on the angular momenta of the initial and final nuclear states and the mixing ratios of the emitted gamma-rays.

2.1 Radioactive decay

Unstable nuclei are referred to as radionuclides. They normally decay by emitting particles and/or electromagnetic radiation [16]. The type of decay is classified by the type of particle or radiation emitted, for example, a radionuclide that decays by the spontaneous emission of an alpha particle is referred to as an alpha-decaying nucleus, if it decays by emitting electrons or positrons it is referred to as a beta-decaying nucleus. A nucleus that decays by emitting electromagnetic radiation undergoes gamma decay.

The most common types of radioactive decays are the alpha, beta, and gamma decay defined as follows [17];

- Alpha (α) decay : ${}^A_ZX_N \rightarrow {}^{A-4}_{Z-2}Y_{N-2} + {}^4_2He$
- Beta minus (β^-) decay : ${}^A_ZX_N \rightarrow {}^A_{Z+1}Y + e^- + \bar{\nu}_e$
- Beta plus (β^+) decay : ${}^A_ZX_N \rightarrow {}^A_{Z-1}Y + e^+ + \nu_e$
- Electron capture (β) decay : ${}^0_{-1}e + {}^A_ZX_N \rightarrow {}^A_{Z-1}Y + \nu_e$
- Gamma (γ) decay : ${}^A_ZX_N^* \rightarrow {}^A_ZX_N + \gamma,$

where X is the decaying parent nucleus, Y is the daughter nucleus produced by the decay of X , A is the atomic mass of the nucleus, Z is the number of protons, N is the number of neutrons, e^- is an electron, e^+ is a positron, ν is a neutrino, $\bar{\nu}_e$ is an anti-neutrino and γ represents a gamma-ray. It is important to note that the α and β decays result in a change of the decaying nucleus to another nucleus, whereas in a gamma decay the nucleus only de-excites from an excited nuclear state (denoted by the $*$) to a lower-energy state. There are rules, called selection rules, that govern how much energy and angular momentum a gamma-ray takes away during the decay and determine the nature of the gamma decay.

2.2 Angular momentum selection rules

Electromagnetic radiation is produced as a result of a change in the charge and the current distribution in the nucleus over time. Magnetic and electric fields are produced as the current and charge distributions change, respectively. A multipole moment is assigned to the charge and current distributions with specific spatial dependence.

For example the $\frac{1}{r^2}$ electric field is an electric dipole field and is characterized by an electric dipole moment, $\frac{1}{r^3}$ electric field is an electric quadrupole field and is characterized by an electric quadrupole moment and so on. Thus the emitted radiation is created as a result of a change in the field; e.g. a change in the electric dipole field of the nucleus creates electric dipole ($E1$) radiation, a change in the electric quadrupole field of the nucleus creates electric quadrupole ($E2$) radiation.

The multipole expansion of the electromagnetic field is given by [18],

$$\phi(r) = \frac{1}{4\pi\epsilon_0 r} \sum_{i=1}^N q_i + \frac{1}{4\pi\epsilon_0 r^2} \sum_{i=1}^N q_i r_i \cos\theta_i + \frac{1}{4\pi\epsilon_0 r^3} \sum_{i=1}^N \frac{q_i r_i^2}{2} (3\cos^2\theta_i - 1) + \dots \quad (2.1)$$

where q_i is a point charge, r is a vector from the center of a charge distribution to the point of interest, r_i is the vector from the center of the charge distribution to the point charge q_i and θ_i is the angle between the vectors r and r_i .

The factor 2^L -pole is known as the multipole order of the radiation where $L = 1$ is for dipole radiation, $L = 2$ is for quadrupole radiation, $L = 3$ is for octupole radiation and so on. L is the angular momentum taken away by the gamma-ray. The general multipole radiation field is denoted by σL where σ is either E (for electric) or M (for magnetic) radiation. The conservation of the angular momentum and the parity leads to a set of selection rules that govern which multipole radiations are permitted and which are forbidden for certain transitions.

Consider a gamma-ray (γ_1) emitted as the initial nuclear state, with angular momentum I_i and parity π_i , decays to a lower-energy state, with angular momentum I_f and parity π_f . The initial and final nuclear states have specific properties e.g. energy, angular momentum and parity as shown in Fig. 2.1. The projection of the angular momentum, L ,

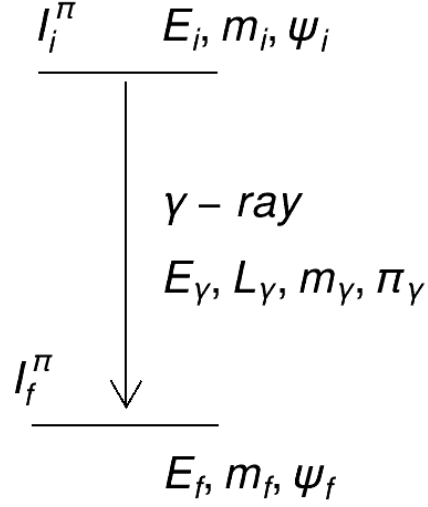


Figure 2.1: A transition from an initial state, with angular momentum I_i , to a final state with angular momentum I_f is shown. E_i and E_f are the energies of the initial and final states, m_i and m_f are the magnetic quantum numbers for the initial and final state and ψ_i and ψ_f are the wave functions for the initial and final states, respectively. The emitted gamma-ray is characterized by its energy E_γ , angular momentum L_γ , magnetic quantum number m_γ and parity π_γ .

with respect to the z-axis is given by the quantum number m which spans $-L \leq m \leq L$. The parity of the gamma-ray is given by;

$$\pi(EL) = (-1)^L \quad (2.2)$$

for electric radiation, and

$$\pi(ML) = (-1)^{L+1} \quad (2.3)$$

for magnetic radiation. It is important to notice that the electric and magnetic radiation with the same value of L have opposite parity. The angular momentum and parity for gamma rays with different multiplicities are listed in Table 2.1.

2.2 Angular momentum selection rules

The selection rules for a gamma-ray transition, shown in Fig. 2.1, are:

$$\begin{aligned}
 E_\gamma &= E_i - E_f \\
 |(I_i - I_f)| &\leq L \leq (I_i + I_f) \\
 \pi_\gamma &= \pi_i \cdot \pi_f \\
 m &= m_i \cdot m_f
 \end{aligned}
 \tag{2.4}$$

It should be noted that monopole radiation, $L = 0$, cannot proceed through emission of

Table 2.1: Different gamma-ray multipolarities depending on L and π . L is angular momentum taken away by the emitted gamma-ray and π is the parity of the gamma-ray.

Type of radiation	L	π
$E0, M0$	--	--
$E1$	1	-1
$M1$	1	+1
$E2$	2	+1
$M2$	2	-1
$E3$	3	-1
$M3$	3	+1

a gamma-ray because the minimum angular momentum that a gamma-ray can take away is $1\hbar$.

As an example showing how the selection rules are applied we consider a gamma-ray transition from a state with $I_i^\pi = 1/2^+$ to a state with $I_f^\pi = 3/2^+$. The characteristics of the emitted gamma-ray are as follows;

- $\pi = \pi_i \cdot \pi_f = +1 \cdot +1 = +1$
- $|(I_i - I_f)| \leq L \leq (I_i + I_f) \longrightarrow |\frac{1}{2} - \frac{3}{2}| \leq L \leq \frac{1}{2} + \frac{3}{2} \longrightarrow 1 \leq L \leq 2$

According to the selection rules there is no parity change between the states therefore the multipolarity of the emitted gamma-ray for $L = 1$ is $M1$ and for $L = 2$ is $E2$. Therefore this is a mixed $M1 + E2$ transition. Mixed multipolarity means that there is a contribution of both the electric quadrupole and the magnetic dipole components in the emitted radiation.

In general, there are three types of transitions that result from the use of the selection rules; (i) pure magnetic or electric transitions, (ii) mixed magnetic and electric transitions and (iii) the $L = 0$ transitions (for example $0^+ \rightarrow 0^+$) which are forbidden for gamma-ray decay and can proceed through internal conversion. The gamma-ray multipolarity can be deduced from the angular distribution of the emitted gamma-rays.

2.3 Weisskopf estimates

Radioactive decay is a probabilistic process, thus all allowed radiation types have a certain probability to take place. The probability for a gamma-ray to be emitted from an excited energy state to a lower-energy state per unit time is given by [19];

$$\lambda(\sigma L) = \frac{8\pi(L+1)}{\hbar L[(2L+1)!!]^2} \left(\frac{E_\gamma}{\hbar c} \right)^{2L+1} \frac{1}{2I_i + 1} | \langle I_f || m_{fi}(\sigma L) || I_i \rangle |^2, \quad (2.5)$$

where, ω is the frequency of the gamma-ray, $\hbar c = 197.3 \text{ MeV}\cdot\text{fm}$ and $(2L+1)!!$ is a double factorial given by $(2L+1)(2L-1)(2L-3)\dots$. The transition probability depends on $m_{fi}(\sigma L)$, which is the matrix element of the multipole operator. σ is either E or M and stands for electric or magnetic transition respectively.

The multipole operator governs the transition between the initial (excited) state and the final (lower-energy) state. Evaluating the matrix element requires the knowledge of the nuclear wave functions of the two states involved in the transition. This transition

then results in a creation of a gamma ray that can be characterized by its energy and multipolarity.

V. Weisskopf derived a general expression for the transition probability assuming that the transition is a result of a single nucleon moving from one single-particle state to another, the spin-dependent factors are equal to one and that the nucleus has a radius of $R = r_0 A^{1/3}$. The solution for the transition probabilities for electric and magnetic radiation is [19];

$$\lambda(EL)_w = \frac{8\pi(L+1)}{L[(2L+1)!!]^2} \frac{e^2}{4\pi\epsilon_0\hbar c} \left(\frac{E_\gamma}{\hbar c}\right)^{2L+1} \left(\frac{3}{L+3}\right)^2 c(R)^{2L}, \quad (2.6)$$

$$\lambda(ML)_w = \frac{8\pi(L+1)}{L[(2L+1)!!]^2} \frac{10e^2}{4\pi\epsilon_0\hbar c} \left(\frac{\hbar}{m_p c}\right)^2 \left(\frac{E}{\hbar c}\right)^{2L+1} \left(\frac{3}{L+2}\right)^2 c(R)^{2L-2}, \quad (2.7)$$

where L is the angular momentum, c is the speed of light, R is the radius of the nucleus, e is the charge of the electron, m_p is the mass of the proton, $\hbar$ is the reduced Planck's constant, ϵ_0 is the permittivity of free space. The constants have the following values, $c = 3 \times 10^8$ m/s, $r_0 = 1.2$ fm, $\frac{e^2}{4\pi\epsilon_0} = 1.44$ MeV·fm, $\hbar c = 197.3$ MeV·fm, $m_p = 938.27231(28)$ MeV/c².

The results from Eq. 2.6 and Eq. 2.7 give estimates of the transition probability which are known as Weisskopf estimates, λ_w . The transition probabilities for magnetic and electric transitions and for $L = 1 - 5$ are:

$$\begin{aligned} \lambda(E1)_w &= 1.0 \times 10^{14} A^{2/3} E^3, & \lambda(M1)_w &= 5.6 \times 10^{13} E^3, \\ \lambda(E2)_w &= 7.3 \times 10^7 A^{4/3} E^5, & \lambda(M2)_w &= 3.5 \times 10^7 A^{2/3} E^5, \\ \lambda(E3)_w &= 3.4 \times 10^1 A^2 E^7, & \lambda(M3)_w &= 1.6 \times 10^1 A^{4/3} E^7, \end{aligned} \quad (2.8)$$

$$\lambda(E4)_w = 1.1 \times 10^{-5} A^{8/3} E^9, \quad \lambda(M4)_w = 4.5 \times 10^{-6} A^2 E^9,$$

where E is in MeV. To make use of the Weisskopf estimates, a relative comparison is made with the measured transition probability as follows;

$$\lambda = \frac{\lambda_{meas}}{\lambda_W} [W.u], \quad (2.9)$$

where, λ_{meas} are the measured transition probabilities, while λ is the transition probability expressed in Weisskopf units ($W.u$). It is evident from Eqs. 2.8 that for the same multipole order, L , the electric transitions are more probable by roughly two orders of magnitude and the transition probability decreases fast with increasing the multipole order L .

It is possible to have mixed multipolarity transitions characterized by a multipole mixing ratio, δ . These transitions could be in the form of $EL + M(L+1) + E(L+2) + M(L+3) + \dots$ or $ML + E(L+1) + M(L+2) + E(L+3) + \dots$. For the former case most often only the first term is considered because in general $E(L) \gg M(L+1)$, see Eq. 2.8. For the latter case the first two terms could be considered because $M(L)$ and $E(L+1)$ can in some case be of similar order of magnitude.

The general expression for the multipole mixing ratio, δ is given by the ratio of the $L+1$ and L matrix elements:

$$\delta^2 = \frac{\lambda(L+1)}{\lambda(L)}. \quad (2.10)$$

The most often observed mixed transitions are with a multipolarity of $M1 + E2$. In this case, δ is defined as the ratio of the electric quadrupole, $E2$, matrix element and the magnetic dipole, $M1$, matrix element. For a transition decaying from an initial state with spin I_1 to a final state with spin I_2 , δ^2 , which is proportional to the ratio of the intensities of the transitions given by;

$$\delta = \frac{\langle I_2 || E2 || I_1 \rangle}{\langle I_2 || M1 || I_1 \rangle}. \quad (2.11)$$

The contributions of both multipolarities sums to one and is given as;

$$\%M1 = \frac{1}{1 + \delta^2} \times 100\%, \quad (2.12)$$

$$\%E2 = \frac{\delta^2}{1 + \delta^2} \times 100\%. \quad (2.13)$$

The possible values for the mixing ratios are infinite, it spans $(-\infty, \infty)$. A large mixing ratio of $\delta > 1$ denotes that the $E2$ component is more dominant than the $M1$ component.

2.4 Angular distribution of gamma rays

The angular distribution of a gamma ray is the intensity of the gamma-ray emission as a function of the angle θ . The angle θ is defined with respect to a chosen axis, called z-axis. The projection m of the angular momentum of a nuclear state is measured along the same z-axis.

Measuring the angular distribution of the emitted gamma rays can assist in the determination of the multipole orders of the gamma-ray transitions, and of the multipole mixing ratios of mixed transitions. The experimentally observed angular distribution is given by;

$$W(\theta) = \sum_{m_i, m_f} W_{m_i \rightarrow m_f}(\theta) = \sum W_{L,m}(\theta), \quad (2.14)$$

where $W(\theta)$ is given by a sum of all possible angular distribution functions for all possible m -states of the initial and final nuclear states. It can also be described as a sum of the angular distribution functions for each possible gamma-ray multipolarity, for instance $W_{Lm}(\theta)$ is the angular distribution of a gamma ray with angular momentum L and projection m . L and m are calculated from I_i , I_f , m_i and m_f using the selection rules discussed in Eq. The angular distributions, $W_{L,m}(\theta)$, are shown in Figs. 2.2 and 2.3. 2.4.

Table 2.2: The angular distribution functions, $W_{L,m}(\theta)$, for $L = 1$ and $L = 2$ and all possible values of m . θ is the angle at which the gamma ray is emitted with respect to the z -axis [20].

Radiation	$m = 0$	$m = \pm 1$	$m = \pm 2$
$L = 1$ (Dipole)	$\frac{3}{8\pi} \sin^2\theta$	$\frac{3}{8\pi} (\frac{1}{2}(1 + \cos^2\theta))$	—
$L = 2$ (Quadrupole)	$\frac{15}{16\pi} \cos^2\theta \sin^2\theta$	$\frac{5}{16\pi} (4\cos^4(\theta) - 3\cos^2\theta + 1)$	$\frac{5}{16\pi} (1 - \cos^4\theta)$

Consider a dipole transition ($L = 1$) from an initial state of $I_i = 1^+$ to a final state of $I_f = 0^+$, see Fig. 2.4. The initial state has three magnetic sub-states that are $m_i = -1, 0, +1$ and the final state has only one $m_f = 0$ sub-state. Each of the initial-level sub-states is populated with equal probability.

The measured angular distribution function of the gamma-ray transition is given by the sum of the angular distribution functions for the sub-states, see Eq. 2.14. Using Table 2.2, the sum of the angular distributions can be computed as follows;

$$\begin{aligned}
 W(\theta) &= W_{1-1} + W_{10} + W_{1+1} \\
 &= \frac{3}{8\pi} (\frac{1}{2}(1 + \cos^2\theta)) + \frac{3}{8\pi} \sin^2\theta + \frac{3}{8\pi} (\frac{1}{2}(1 + \cos^2\theta)) \\
 &= \frac{3}{4\pi}
 \end{aligned} \tag{2.15}$$

From the solution of Eq. 2.15, one can see that there is no angular dependence meaning that the distribution is isotropic. Therefore, no information about the spins can be deduced. This calls for another technique that introduces an anisotropic distribution as a result of unequal population of the magnetic sub-states. This can be accomplished by using another experimental technique for determining multipolarities and multipole mixing ratios, for instance the angular correlation technique. This method entails observing a sequence of two consecutive gamma-ray emissions, known as a gamma-gamma cascade.

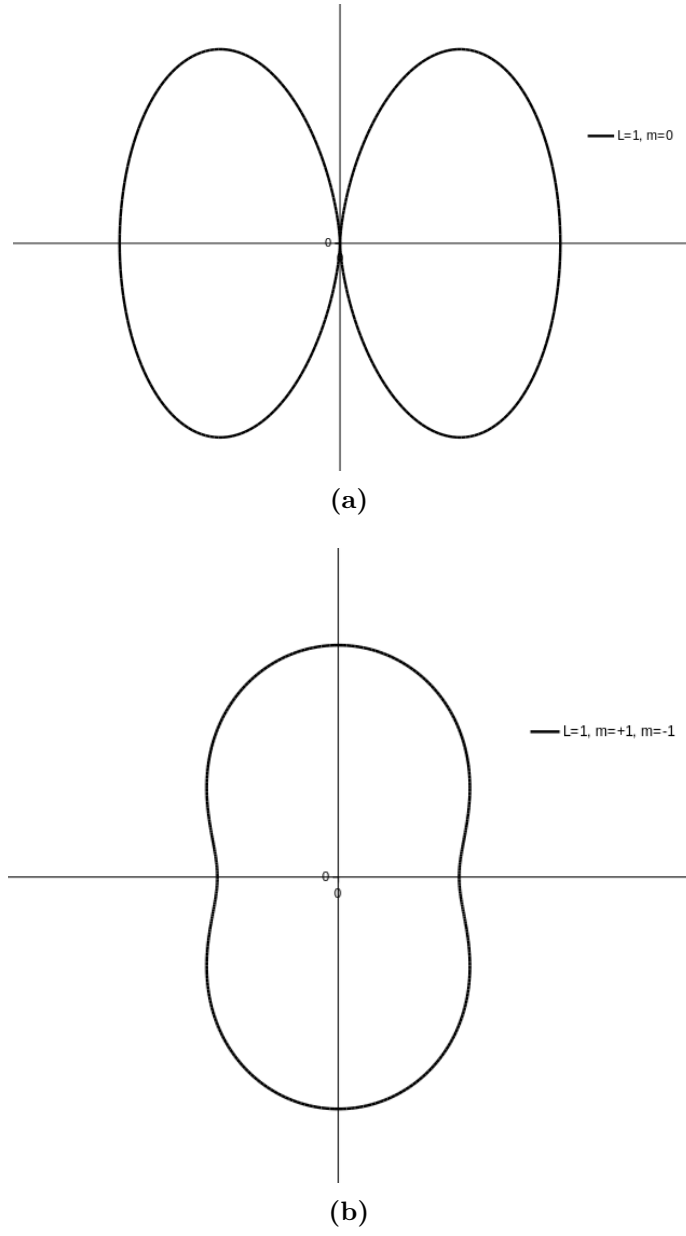
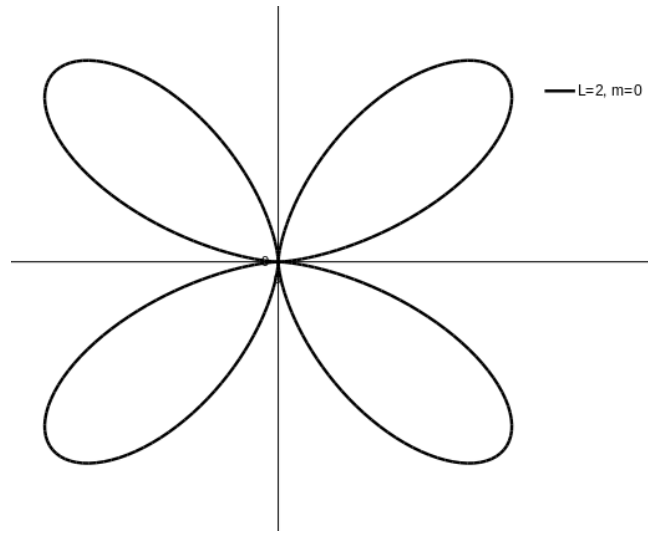
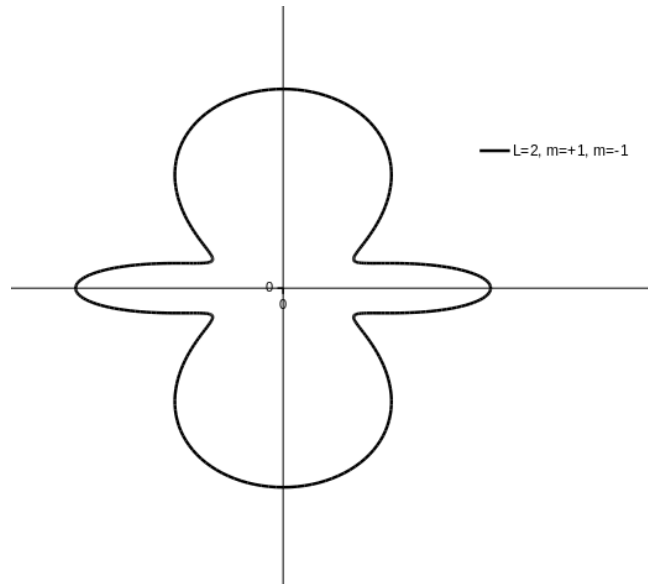


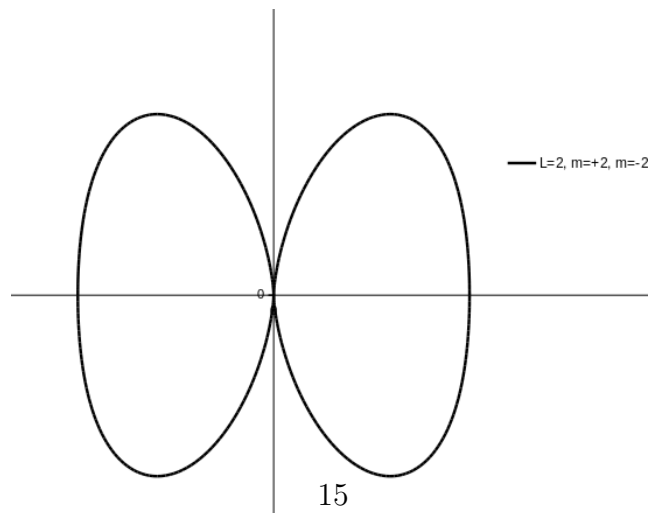
Figure 2.2: Dipole radiation patterns for pure(l,m) multipoles [20].



(a)



(b)



(c)

Figure 2.3: Quadrupole radiation patterns for pure(l,m) multipoles [20].

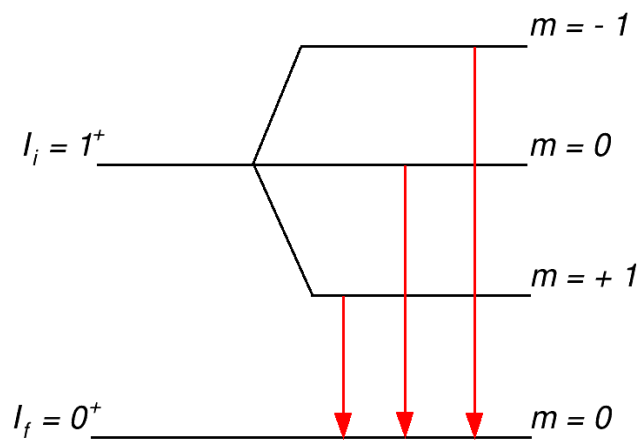


Figure 2.4: A transition from a $I = 1^+$ to $I = 0^+$ state. The initial state has 3 single possible substates of $m = \pm 1$ and $m = 0$.

2.5 Angular correlations

The angular correlation between gamma rays in coincidence is defined as the probability of the second gamma ray being emitted at a certain angle θ relative to the direction of the first gamma ray. The theoretical angular correlation is expressed as follows [21],

$$W(\theta) = \sum_{k=0, k=\text{even}}^{2L} a_k P_k(\cos\theta), \quad (2.16)$$

where $P_k(\cos(\theta))$ are Legendre polynomials which are a set of even and odd functions dependent on the order k . The experimentally observed angular correlation is expressed by [22],

$$W'(\theta) = \sum_k a'_k P_k(\cos\theta). \quad (2.17)$$

the a_k and a'_k angular correlation coefficients are related through the attenuation factors coefficients Q_k , see Eq. 2.18.

$$a_k = \frac{a'_k}{Q_k}. \quad (2.18)$$

Q_k accounts for the fact that the detectors used to measure gamma rays cover a range of angles whereas the theoretical angular correlation function is for gamma rays measured at specific angles. The Q_k factors are in most cases negligible (very close to 1) and for the analysis of this work they were assumed to be 1. However, the determination of the Q_k factors is planned as future work.

The general form and derivation of the angular correlation function is given in [23, 24].

The first six Legendre polynomials are listed in Table 2.3, where x represents $\cos(\theta)$.

The parameters a_k are angular correlation coefficients that are dependent on the nuclear spins of the states involved, on the angular momentum carried by the gamma rays and on the mixing ratios. Therefore, the a_k values allow one to determine the spins of the nuclear

Table 2.3: List of the first six Legendre polynomials.

Legendre polynomial	order form
$P_0(x)$	1
$P_1(x)$	x
$P_2(x)$	$\frac{1}{2}(3x^2 - 1)$
$P_3(x)$	$\frac{1}{2}(5x^3 - 3x)$
$P_4(x)$	$\frac{1}{8}(35x^4 - 30x^2 + 3)$
$P_5(x)$	$\frac{1}{8}(63x^5 - 70x^3 + 15x)$

levels and the multipolarity and the mixing ratio of the emitted gamma rays. They are given by,

$$a_k = B_k(\gamma_1)D_k(\gamma_2), \quad (2.19)$$

where $B_k(\gamma_1)$ is dependent only on the first γ ray and $D_k(\gamma_2)$ is dependent only on the second γ ray. $B_k(\gamma_1)$ and $D_k(\gamma_2)$ are given by;

$$B_k(\gamma_1) = \frac{F_k(L_1 L_1 I_1 I_2) + (-1)^{L_1+L'_1} 2\delta_1 F_k(L_1 L'_1 I_1 I_2) + \delta_1^2 F_k(L'_1 L'_1 I_1 I_2)}{1 + \delta_1^2} \quad (2.20)$$

$$D_k(\gamma_2) = \frac{F_k(L_2 L_2 I_2 I_3) + (-1)^{L_2+L'_2} 2\delta_2 F_k(L_2 L'_2 I_2 I_3) + \delta_2^2 F_k(L'_2 L'_2 I_2 I_3)}{1 + \delta_2^2}, \quad (2.21)$$

where γ_1 and γ_2 represent the first and the second gamma ray in the cascade, respectively. L is the angular momentum of the gamma ray, while for mixed multipolarity L' is given by $L' = L + 1$, I represents the angular momentum of the states, for instance I_1 and I_2 are the initial state and final states of the first gamma ray and I_2 and I_3 are the initial and final states of the second gamma ray, while δ is the multipole mixing ratio. The F_k coefficients reflect angular momentum couplings. These coefficients are dependent on the Wigner 6j symbol and on the Racah coefficients [25, 26, 27]. The values of these

coefficients have been calculated in tables for different multiplicities in [21, 28, 29, 30]. For example, for a cascade $4 \rightarrow 2 \rightarrow 0 \Rightarrow F_2(2242) = -0.1707$ and $F_2(2202) = -0.5976$.

There are different conventions for the sign of δ , this is of importance because in Eq. 2.20 and 2.21 δ is not squared as in Eq. 2.11, therefore the sign of δ contributes to the computations of the functions. In this work the Krane-Steffen¹ phase convention was used [31, 32].

As an example consider a nucleus decaying from an initial state of $I_i = 0$ through an intermediate state of $I_m = 1$ to a final state of $I_f = 0$, see Fig. 2.5. During this decay two gamma rays are emitted in coincidence connecting the three states. Assume that we have

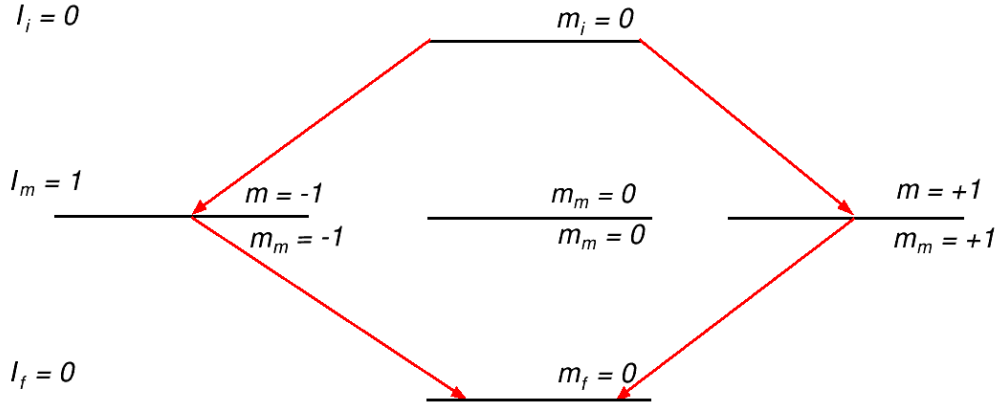


Figure 2.5: An example of possible transitions for a $0^+ \rightarrow 1^+ \rightarrow 0^+$ cascade.

two detectors, Det1 for the detection of the first gamma ray and Det2 for the detection of the second gamma ray. When we choose to detect the first gamma ray in Det1, we are choosing the direction of the z-axis along which the projections of the angular momenta

¹The other typical convention is the Rose-Brink Convention. $\delta(\text{Rose-Brink}) = (-1)^{L-L'} \delta(\text{Krane-Steffen})$

are defined. For $I_m = 1$ the possible values of m_m are $m_m = -1, 0, +1$. The first gamma ray decays from I_i, m_i to I_m, m_m . It therefore has $m = m_i - m_m$ and $|0 - 1| \leq L \leq 0 + 1$ which results in the angular distributions W_{10} and $W_{1\pm 1}$ (see Fig. 2.2) as discussed in the previous section. Since, we now know the direction of the first gamma ray, it is registered in Det1, at $\theta = 0^\circ$. According to Table 2.2 the $I_m = 1, m_m = 0$ sub-state is not populated because $W_{10} = \frac{3}{8\pi} \sin^2(\theta = 0^\circ) = 0$. Only $W_{1\pm 1}$ are non-zero, thus only the sub-states with $I_m = 1$ and $m_m = \pm 1$ are populated. Therefore, only $W_{1\pm 1}$ will contribute to the angular distribution function of the second gamma-ray transition.

Another example is for $I_m = 2$, where the possible values of m_m are $m_m = -2, -1, 0, +1, +2$. Again using Table 2.2 we can conclude that only $W_{2\pm 1}$ states will be populated because $W_{20}=W_{2\pm 2} = 0$ at $\theta = 0^\circ$.

It is important to note that what introduces the dependence on the angle θ is the fact that there is an unequal population of magnetic states in the intermediate state. Therefore, the observation of the second gamma ray with respect to the first will have an anisotropic angular distribution.

Just like each multipole radiation has a characteristic angular distribution, each gamma-gamma cascade has its characteristic angular correlation function too. Angular correlation is measured by placing detectors at different angles and measuring the coincidence intensities of the two detected gamma rays as a function of the angle (θ) between them.

The angle θ is plotted on the x -axis. Generally the most probable transitions observed are pure dipole $E1$ and $M1$, and pure quadrupole $E2$, and mixed $M1 + E2$ dipole plus quadrupole transitions. The correct determination of the multipole mixing ratios can

provide an important means to investigate nuclear states and test various nuclear model predictions.

Chapter 3

Experimental method and data analysis

3.1 Semiconductor detectors

For radiation to be detected it has to interact with the detector material. In semiconductor detectors, information is extracted from the electron-hole pairs formed when the radiation interacts with the detector material. The electron-hole pairs are known as charge carriers. The detection signal is formed by collecting these charge carriers.

Semiconductor detectors are made of semiconductor materials; for example silicon is normally used for charged particle detectors while High Purity Germanium (HPGe) detectors are widely used for gamma-ray spectroscopy. The iThemba LABS segmented clover detector is composed of 4 germanium crystals, therefore, it is a semiconductor detector.

The band structures in solid materials are what classifies them into insulators, semiconductors and conductors. Generally, in metals the conduction band, which is a region where electrons in excited states can migrate to, is not full and is easily accessible. The band gap between the conduction band and the valence band (band where electrons are

bound) gives the amount of energy needed for an electron to be excited from the valence band to the conduction band. The smaller the band gap the higher the material conductivity. Figure 3.1 shows the different band structures for different materials.

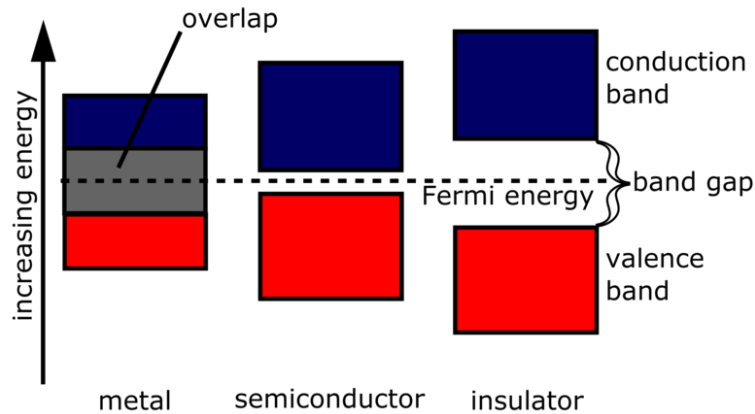


Figure 3.1: Band structures for electron energies in different materials, from [33].

Semiconductors have a band gap of ~ 1 eV. The excitation of a previously bound electron to the conduction band leaves a vacancy in the valence band, and as a result an electron-hole pair is created. If an electric field is introduced to a semiconductor, the electrons and holes migrate in opposite direction.

The addition of impurities to the crystal lattice of the solid material is known as doping. In order to form a p-n junction the material has to be doped. A p-type semiconductor is formed when the impurity introduced is a dopant with less valence electrons than valence holes, known as an acceptor since this dopant creates an acceptor level just above the valence band. If an electron fills the hole, it will bond in a slightly different way (less bound) compared to the other bonds. This allows electrons to participate in covalent

bonds with energies close to the valence band. An example of an acceptor in HPGe detectors is boron [34].

A n-type semiconductor is formed when the impurity introduced is a dopant with more valence electrons than holes, known as a donor. The extra electron is lightly bound and sits very close to the conduction band. This then means that the number of conduction electrons exceeds that of the holes. This determines the conductivity of the material since it is dependent on the flow of electrons. An example of an electron donor in HPGe is lithium. Because radiation damage is more affected by hole trapping, n-type detectors present better performance against neutron damage.

A p-n junction is the boundary at which the n-type and p-type materials join. At this boundary there is a non-conductive layer where the electrons and holes recombine. When a p-n junction is created, a very large electron density difference occurs. The electrons from the donor impurities (n-type) migrate and fill the holes of the acceptor impurities (p-type), by diffusion, generating an electric field. The holes also migrate from the p-type to the n-type material. The diffusion of the charges takes place until a layer is created whereby no free charge carriers are available. This layer is called a depletion layer. The application of a biased potential that makes the n-side of a diode more positive and the p-side more negative increases migration and thus increases the depth of the depletion layer. This is called reverse biasing.

The depletion layer functions as a radiation detector. When radiation enters the depletion layer, charge carriers are created which generates a current flow. The charge carriers are collected and the amount of charge collected is proportional to the energy of the incident radiation. To achieve excellent resolution many charge carriers are needed. This results in a precise energy measurement because it reduces the statistical fluctuations.

3.2 Electromagnetic radiation interactions with matter

For electromagnetic radiation to be detected it should interact with the detector material. When working with gamma-rays and X-rays, there are three significant processes that can take place within the detection material: photo-electric absorption, Compton scattering and pair production. These processes are important to understand because they impact significantly on the shape of the registered spectra. This section aims to discuss the three processes briefly.

3.2.1 Photo-electric absorption

The photo-electric absorption process leads to the complete transfer of the gamma-ray energy to the atomic electron. In this process the gamma ray interacts with an atom by giving off all its energy to an electron (called a photo-electron) that was bound to an atom. The photo-electron is ejected and the atom rearranges itself in order to fill the created hole using an electron from other shells or alternatively it catches a free electron from the medium. Therefore, characteristic X-rays or Auger electrons can be emitted from the atom. The energy of the photo-electron is given by;

$$E_e = E_\gamma - E_{BE}, \quad (3.1)$$

where E_γ is the energy of the incident gamma ray and E_{BE} is the electron binding energy in the atom [34]. The probability for the photo-electric absorption is dependent on the electron binding energy, on the energy of the incident gamma ray and on the atomic number of the detector material, and can be approximated by the expression;

$$\sigma \cong \text{constant} * \frac{Z^n}{(E_\gamma)^3}, \quad (3.2)$$

where n can be 4 or 5 [35]. The probability for photo-electric absorption increases as the atomic number Z increases and decreases fast with increasing gamma-ray energy. It is

important to mention that this probability is higher for more tightly bound electrons, for example for the electrons in the K-shell. They are more likely to be affected than the ones in the L and M shells. Photo-electric absorption is very important in gamma spectroscopy because it produces an event in the full energy peak in the gamma-ray spectrum.

3.2.2 Compton scattering

Compton scattering leads to a partial transfer of the gamma-ray energy to the atomic electron [36]. It is dominant for gamma-ray energies that are in the range of a few hundred keV to about 10 MeV. In this process the gamma ray transfers only part of its energy to the atomic electron. This results in the scattering of both the gamma ray (which changes its direction and energy) and the atomic electron. Figure 3.2 illustrates the process of Compton scattering.

In order to express the relation between the energy lost by the gamma-ray and its scattering angle, the laws of conservation of energy and momentum are applied with the assumption that the electron is free and initially at rest. Figure 3.2 is a schematic illustration of Compton scattering where a gamma ray of energy E_γ scatters off an electron giving it a portion of its energy E_e . This process can occur repeatedly as long as the incident gamma ray has enough energy.

The energy transferred to the electron by the incident gamma ray depends on the scattering angle θ . This energy can be in a range of zero to a significant fraction of the gamma-ray energy E_γ . The expression of the energy $E_{\gamma'}$ after scattering through angle θ is given by:

$$E_{\gamma'} = \frac{E_\gamma}{1 + \frac{E_\gamma}{m_0 c^2} (1 - \cos(\theta))}, \quad (3.3)$$

where $m_0c^2 = 0.511$ MeV is the rest mass of the electron. The probability for this process to occur largely depends on the atomic number, Z , of the detector material and on the energy of the gamma ray. The energy transferred to the electron is given by:

$$E_e = E_\gamma - E_{\gamma'}, \quad (3.4)$$

where E_e is the energy transferred to the electron. There are different scenarios that can

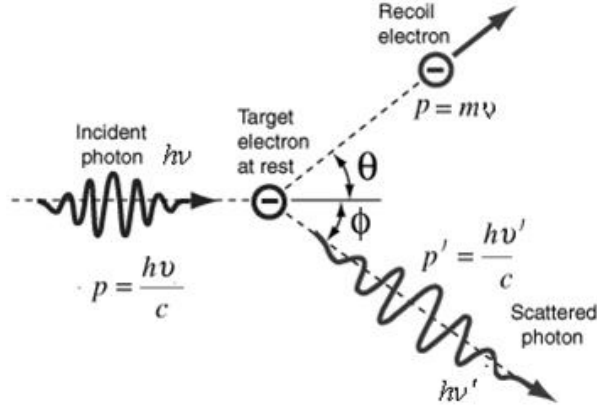


Figure 3.2: The Compton scattering interaction. Picture taken from [37].

take place depending on this interaction. For example, the gamma ray can back scatter at $\phi = 180^\circ$ transferring maximum energy to the electron, as a result a Compton edge is produced in the gamma-ray spectrum. It is also possible that the gamma ray slightly grazes the electron which results in the gamma-ray carrying approximately the same energy as it was carrying before the interaction. The gamma ray can escape the detector with a portion of the initial incident energy, this process introduces Compton background. This background can be decreased by using suppressors that are made of materials with a higher atomic number Z such as Bismuth germanate (BGO).

3.2.3 Pair production

For gamma rays with energy that is more than twice the rest mass of an electron ($1.022 \text{ MeV}/c^2$), pair production can take place [38]. Pair production takes place in the Coulomb field of the nuclei whereby a gamma ray creates an electron-positron pair. This process is dominant for gamma-ray energies greater than 5 MeV. The excess energy above the 1.022 MeV is shared by the electron and positron as kinetic energy.

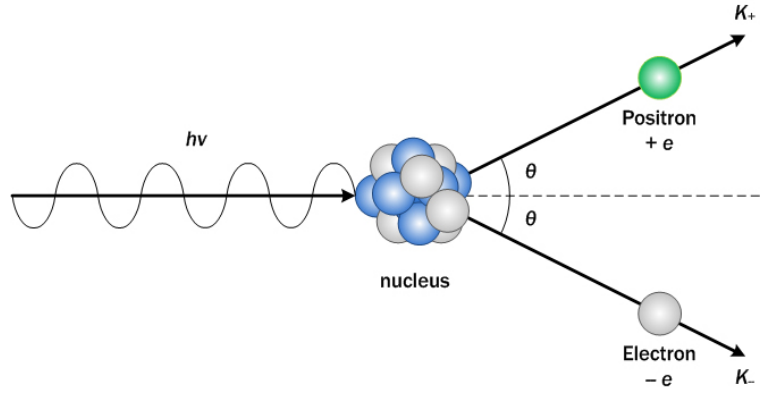


Figure 3.3: The process of pair production. Picture taken from [39].

The positron annihilates by interacting with an electron in the detector material which in turn creates two annihilation gamma rays with energy of 511 keV in opposite directions. The fate of these annihilation gamma rays could be;

- One or both the gamma rays could escape the detector creating single and double escape peaks with energy equal to the energy of the incident gamma ray minus 511 keV or 1.022 MeV, respectively.
- Both gamma rays could be absorbed in the detector material contributing to the full energy peak.

- One or both the gamma rays may Compton scatter out of the detector, contributing to events in the Compton background.

The probability for pair production largely depends on the energy of the incident gamma-ray and varies as the square of the atomic number Z .

Figure 3.4 shows which process is most dominant depending on the incident gamma-ray energy and the atomic number Z of material.

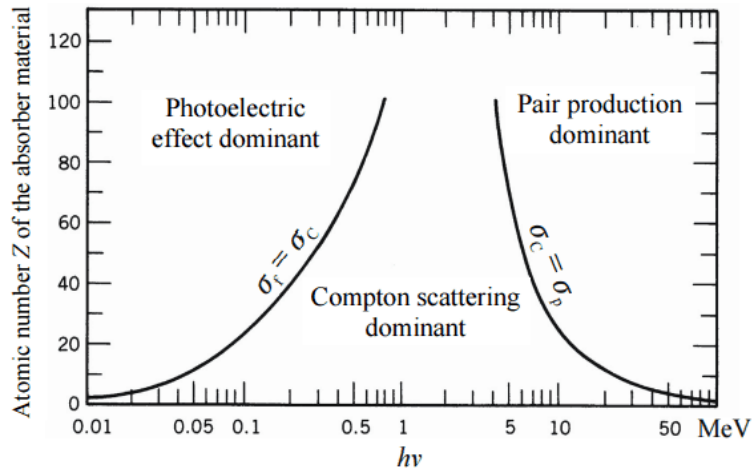


Figure 3.4: The relative importance of the various interaction processes of gamma-rays with matter. Figure taken from [35].

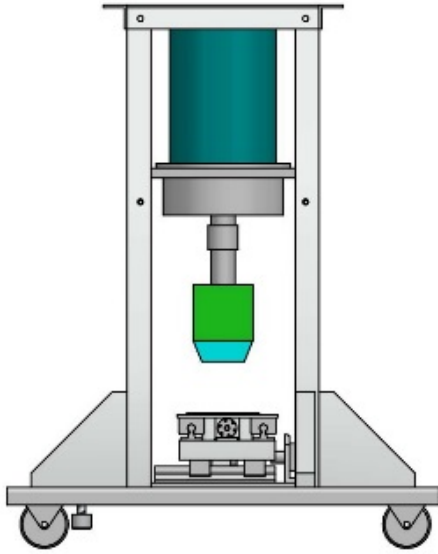
3.3 The iThemba LABS segmented clover detector

The iThemba LABS segmented clover detector is of TIGRESS (TRIUMF-ISAC gamma-ray escape-suppressed spectrometer) type [11]. It is composed of four segmented coaxial HPGe crystals named crystals A, B, C and D with segments 1 to 8 per crystals. The dimension of each crystal is $60 \times 90 \text{ mm}$ before tapering. The longitudinal segmentation is at 35 mm. The detector has 36 charge preamplifier cards (PSC823) in total; 32 for the segments and 4 for the core contacts. The four crystals are held in vacuum inside

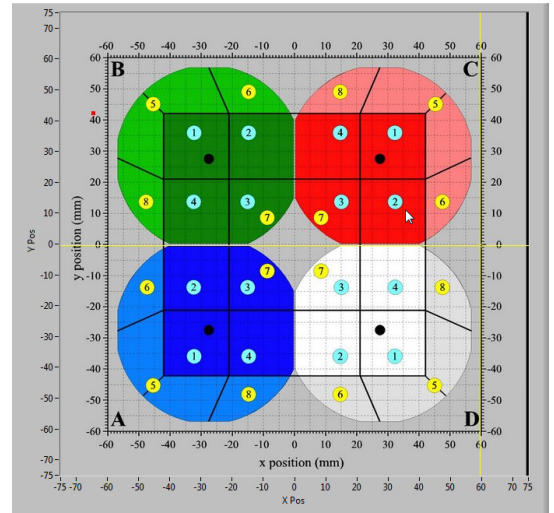
3.3 The iThemba LABS segmented clover detector

a cryostat and at cryogenic temperature. The dewar contains about 3 liters of liquid nitrogen which is used for cooling the crystals.

Figure 3.5a shows a schematic diagram of the iThemba LABS segmented clover detector mounted above a scanning table, with the face of the detector parallel to the scanning table. A radioactive source can be placed on the scanning table and moved to a desired (x, y) position with a very good precision of ~ 0.2 mm. Figure 3.5b illustrates the labeling used for the four crystals and their segments (1 – 8).



(a) Schematic diagram of the iThemba LABS segmented clover, mounted above a scanning table.



(b) The segmentation of the crystals A, B, C, and D. Each crystal has four front segments (labels in blue) and 4 back segments (labels in yellow).

Figure 3.5: The iThemba LABS segmented clover detector.

All four crystals have operating bias voltages applied on the cores which are higher than the depletion voltages. The depletion voltages are equal to the voltage needed to create a depletion layer that covers the whole crystal. The depletion and operational voltages for the segmented clover are listed in Table 3.1.

Table 3.1: Voltage parameters for each of the four crystals of the segmented clover.

Crystal	Depletion Voltage	Operating High Voltage
A	2000 V	+3000 V
B	3500 V	+4000 V
C	1750 V	+4000 V
D	2250 V	+3500 V

The four core contacts are fitted with cold Field Effect Transistors (FETs) and the segments are fitted with warm FETs. The output signals are read at 100 MHz sampling frequency by four XIA Pixie-16 modules [40]. The energy deposited by the gamma-ray radiation is calculated by making use of a trapezoidal filter [40] with parameters: rise time of $5.8\ \mu\text{s}$, flat top of $1.0\ \mu\text{s}$, peak sample of $6.4\ \mu\text{s}$ and peak separation of $13.1\ \mu\text{s}$.

3.4 Experimental set-up and data capture

Data were collected with the segmented clover detector using two gamma-ray sources, ^{60}Co and ^{133}Ba . Each source was placed at close geometry. The energy resolution of the detector was $\approx 2.35\ \text{keV}$ for the cores and $\approx 4\ \text{keV}$ for the segments at $1.332\ \text{keV}$. The absolute efficiency of the detector was measured with a ^{60}Co source. At $1.3\ \text{MeV}$ it is

- in addback mode: 2.9%
- in crystal mode: 2.2%
- in single segment mode: 1.0%,

where single-segment mode refers to using the segments as individual detectors, single-crystal modes to four core crystals as individual detectors, and addback mode when the

crystals are used as a single detector. The distance from each source measurement to the front face of the detector was 42 mm. At that distance the opening angle of the segmented clover was approximately $\frac{1}{8}$ of 4π .

The collected data were stored in binary file format and in histograms. Four binary files were written as an output by the digital data acquisition system (DDAS) and each file contained data detected by one crystal. The XIA electronic modules 0, 1, 2, and 3 were collecting data for crystals A, B, C and D, respectively. Each module core was connected at channel 0 on the XIA Pixie-16 modules and the 8 segments were connected to channels 4 to 11. Each channel was set to a local trigger state, which means that each channel collected data if its signal passed a set threshold. Thus the detector segments were used as independent detectors. The Maximum Integrated Data Acquisition System (MIDAS) was used [41].

3.5 Data analysis

In this section, a detailed explanation of the data analysis methods used to extract information from the collected raw data is given. As mentioned before, after data collection four binary files were saved on disk. These files contained data collected by the 36 output channels of the segmented clover detector.

3.5.1 Merging the data

A ROOT/C++ code was written to merge the collected data into one binary file that contains data from all four crystals, the code is attached in Appendix A. The code was run within a ROOT [42, 43] environment. The initial four binary files from the four modules were opened and read. Each file had information on the registered gamma-ray events, which consisted of the energy of the gamma-ray detected, the time stamp, the

crystal number, the segment number and other information written in binary format. The merging program then read the events, checked the time stamps of each event and created a new binary file in which the events from the 4 crystals are written in chronological order. This merged binary file is then used to create root trees.

3.5.2 Gain drift and energy calibration coefficients

Since data used in this work was collected from well known gamma-ray sources (^{133}Ba and ^{60}Co) the energy calibration was done using their gamma rays and also the ^{40}K peak of 1461 keV, which was part of the room background.

The collected data were sorted into spectra, that is in a counts versus channels format. Generally, the channel number at the centroid of a peak in a spectrum corresponds to the energy of the peak. The area under the peak corresponds to the number of times that this particular gamma ray is detected. For any new data collection procedure, it is a protocol to perform gain matching and energy calibration of the detection system. In this work we performed both of these and then carried out gain drift corrections as well.

Each data set (from ^{133}Ba and ^{60}Co sources) was collected for a period ~ 12 hours. It was observed that the output signals, particularly from the segments, drifted from their initial positions. The drifts were strongly correlated with the liquid nitrogen filling cycles. As a result gain drift corrections were needed. An example of the gain drift in segment 7 of crystal B are shown in Fig. 3.6.

To derive gain drift corrections coefficients the data were split into bunches of 50 million events. Each bunch of data was sorted into root files which contained direct histograms of uncalibrated spectra for each segment and each core. A ROOT/C++ code

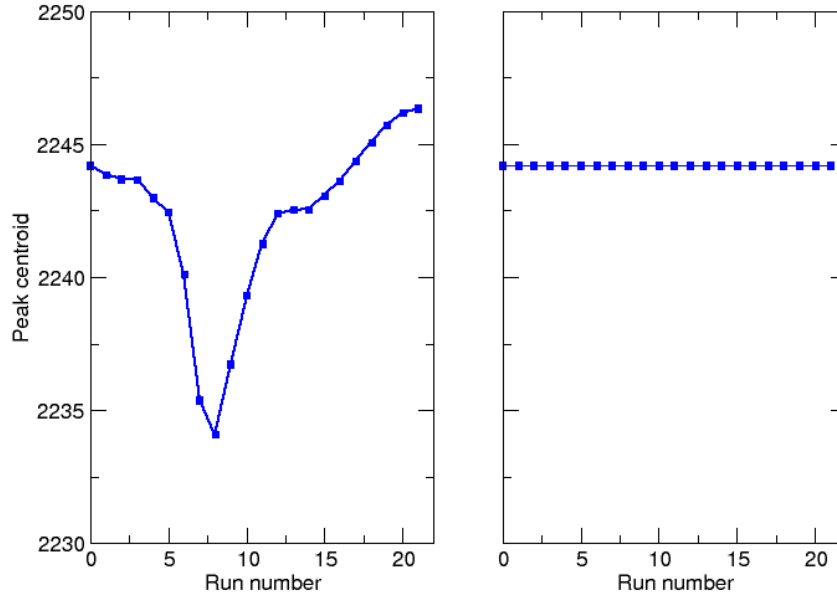


Figure 3.6: Gain drift shown on the left for 20 runs. The centroids of the peaks changed by ~ 12 channels. On the right the gain drift corrections have been implemented. The centroids of the peaks for all the runs remains the same.

was written to plot and fit the peaks in the histograms and extract the centroids of two peaks of interest, one of low channel number (low energy) and another at high channel number (high energy). See Appendix B for the code. The centroids were then used to calculate the gain drift coefficients. These coefficients were later used (when the ROOT trees were sorted) to shift all the peaks to their initial non-drifted channels. This was done for all 36 signal outputs and for all data.

To perform the gain matching and the energy calibration a correspondence between the channels and energy is needed. Since the ^{133}Ba and ^{60}Co are standard calibration sources their gamma-ray energies and intensities are well known. The channel numbers x of the

^{133}Ba and ^{40}K peaks were measured and the a and b parameters from a straight line fit $E = ax + b$ to the data were calculated. The coefficients a and b are the gain matching and offset parameters, respectively. These parameters were deduced for all segments and cores and used later when a ROOT tree was sorted.

3.5.3 Crosstalk corrections

Since the clover detector that was used is segmented there is an electronic coupling between the channels of the segments which is called crosstalk. There are two types of crosstalk; proportional crosstalk and derivative crosstalk. The former is very important for the present study because it affects significantly the energy reading on the segments by producing energy shifts. To correct for the proportional crosstalk our team has measured it and has extracted a matrix of 32×32 crosstalk coefficients, and worked out a technique for correction [44, 45]. A ROOT/C++ code was written, which read the crosstalk coefficients matrix and corrected the data on an event-by-event basis.

Figure 3.7 illustrates the effect of proportional crosstalk. Before proportional crosstalk corrections, the energy peaks are wide and have wrong energies. After the correction the 1173 keV, 1332 keV peaks of ^{60}Co and the 1408 keV peak of ^{152}Eu are narrow and have correct energies.

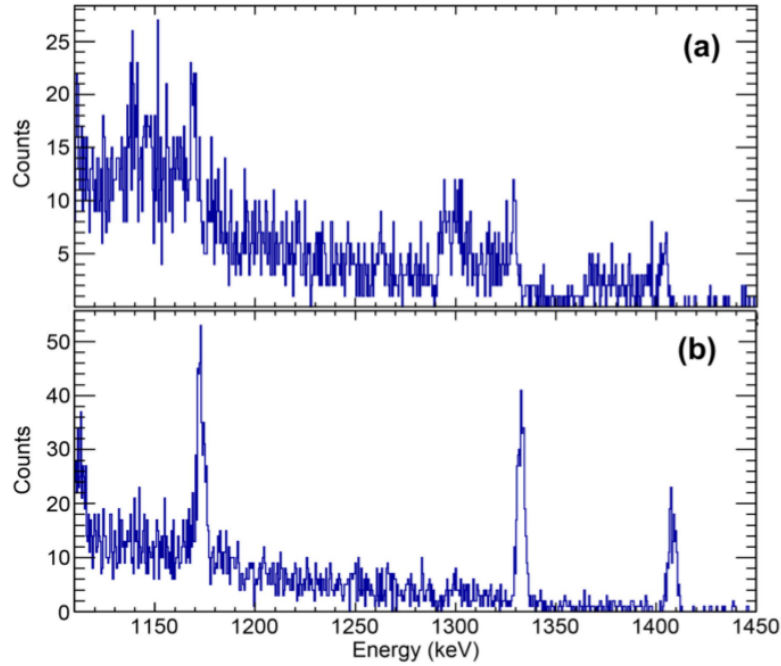


Figure 3.7: A spectrum of double-hit events registered in segments 5 and 7 of crystal B in addback mode, (that is the energy in this spectrum is given by $E = E_5 + E_7$) and **(a)** before crosstalk correction, **(b)** after crosstalk correction [45].

3.5.4 Building a tree

A new C++/ROOT code that builds a tree from raw data was written, see Appendix C. This is a most important step in the analysis because this code converts the XIA raw data from binary format to a ROOT format, which can be used for gamma-spectroscopy analysis. The code took as input the merged binary data file and five text files generated from the steps mentioned in Sections 3.5.2 and 3.5.3. The files are the gain and offset files for the gain matching and energy calibration, the gain and offset files for the gain drift corrections, and the inverted matrix for the proportional cross talk correction. A gamma-ray event was inserted in the tree if its energy was greater than 20 keV. Furthermore, all gamma-ray events which had time stamps within a chosen coincidence time window were placed in the tree as one coincident event. The time window for this analysis was set to

$t = 1 \mu\text{s}$.

The tree contained energy branches and time stamp branches for each core and segment. The tree was later used to sort direct spectra and angular correlation matrices, as well as other information, such as hit pattern, etc.

3.5.5 Angular properties of the iThemba LABS segmented clover

The clover has a total of 32 segments, 16 are front segments and 16 are back segments, see Fig. 3.5a. There are 992 (32×31) segment pairs that produce 992 potentially different angles. In fact the number of different angles is reduced as the result of the geometry of the clover detector, because it consists of four tapered crystals with the same geometry and segmentation. For example, the angle between segment 1 and segment 2 of crystal A is equal to the angle between segment 1 and segment 2 of crystals B, and that of crystals C and D. In addition segments 2 and 4 are cut in the same way, therefore the angle between segments 2 and 3 is equal to the angle between segment 4, and 3 and the angle between segments 2 and 1 is equal to the angle between segments 4 and 1. As a result of such considerations there are only 122 distinctly different angles produced by the 992 segment combinations. They are shown in Fig. 3.8. These 122 angles can be used to measure the angular correlation function and deduce the a_2 and a_4 angular correlation coefficients. The angles range from 0° to 90° .

Since the back segments are placed behind the front segments, low energy gamma rays ($E_\gamma < 250 \text{ keV}$) can not be observed in the back segments as they cannot penetrate through the front segments. For the ^{133}Ba data all gamma rays have $E_\gamma < 400 \text{ keV}$, thus the back segments have a much lower counting rate.

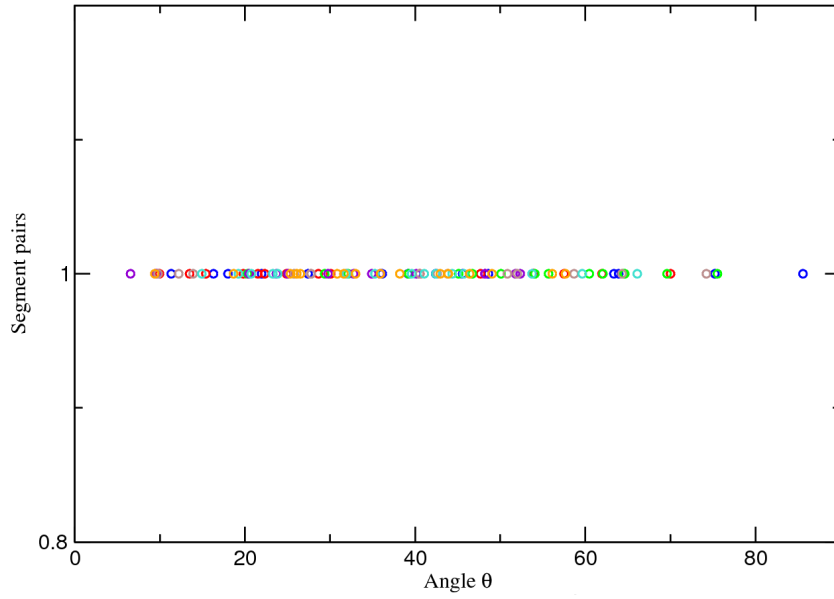


Figure 3.8: The 122 different measuring angles for all 992 segment pairs.

To calculate the angles at which the angular correlations can be measured, GEANT4 Monte Carlo simulations [46] were developed for our detector [47]. These simulations were run for the ^{60}Co source and used to determine the average gamma-ray interaction depth in the detector volume. GEANT4 is a simulation toolkit based on C++, that simulates the movement and interaction of particles through matter [48]. To set up GEANT4 simulations the specific geometry of the iThemba LABS segmented clover detector was used. The position and the distance of the source to the detector was given as input. To calculate the angles the coordinates of the first interaction for the gamma rays that were fully absorbed in the segment of interest were used. Examples are shown in Fig. 3.9. Similar work was done by Schumaker at TRIUMF, the simulations were performed with the sources placed at distances of 11 mm and 14.5 mm from the detectors[49].

The average interaction position for each segment, and for a gamma ray of interest were calculated as follows;

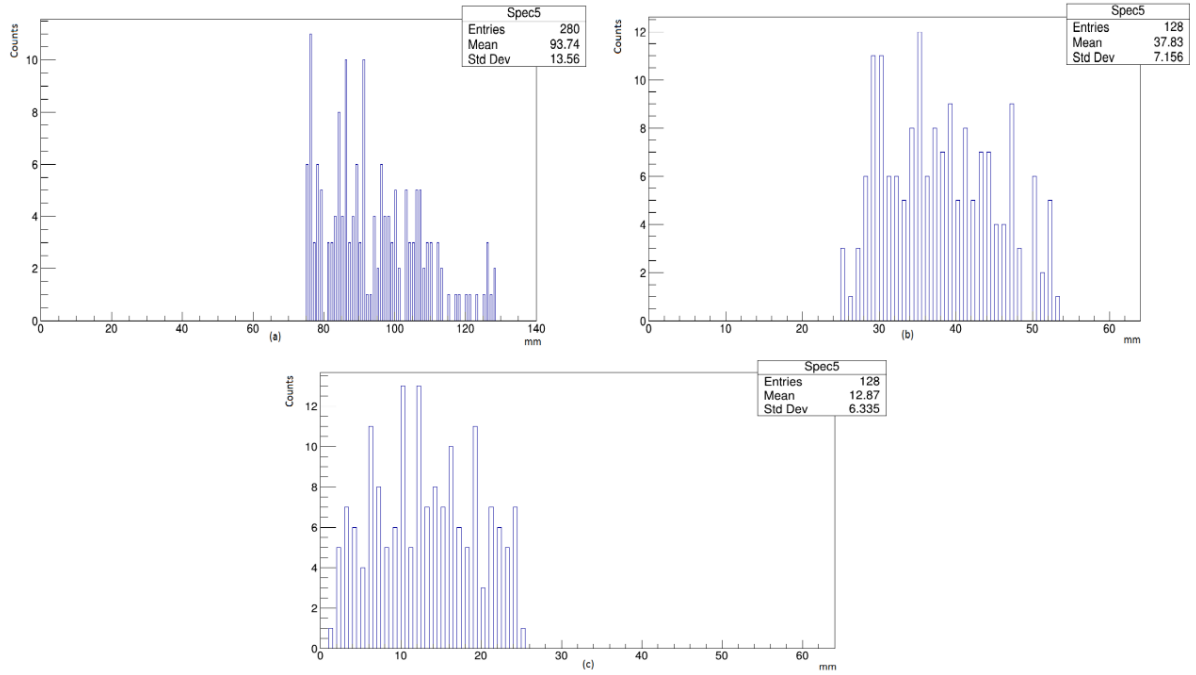


Figure 3.9: First interaction position in segment 6 of crystal D for the 1332 keV gamma-ray from a ^{60}Co source for events when the gamma-ray was fully absorbed in segment 6; (a) z -position or depth in the crystal, (b) y -position and (c) x -position.

- Run GEANT4 simulation with the gamma-ray energy and the distance from the source to the detector as input parameters.
- extract the first interaction coordinates (x_o, y_o, z_o) for the incident gamma ray on each segment for events when the gamma ray of interest was fully absorbed and save them in a histogram.
- using the saved histograms determine the average interaction coordinate (x, y, z) for each segment.
- Use the average interaction coordinates to calculate the angle between the segment pairs.

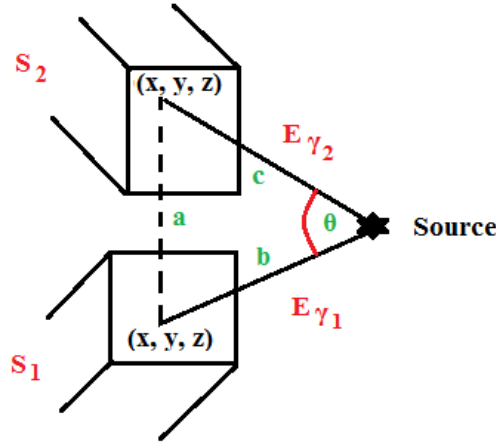


Figure 3.10: Calculating the angle between the two segments for angular correlation measurements.

The cosine rule [50] was used to calculate the angle between two segment pairs.

$$\cos(\theta) = \frac{b^2 + c^2 - a^2}{2bc}, \quad (3.5)$$

where a , b , c are shown in Fig. 3.10. The average interaction depth inside the detector material depends on the energy of the incident gamma ray. Therefore, the effective angle of each segment pair was calculated as a function of the energy. Each source was placed at $z = -42$ mm.

It is worth mentioning that initially, the angles were estimated by taking a center point in the (x, y) plane as the average interaction position and extracting the depth depending on the gamma-ray energy. However, since the geometry of the iThemba LABS segmented clover detector is complicated due to tapering and the presence of a central core contact that is not composed of germanium material, GEANT4 simulations are needed for accurate determination of the angles of the detector.

As the four germanium crystals are segmented in the same way, there are more than one segment pair (usually 4, or 8, or 16) that corresponds to the same angle. Thus, all angular correlation matrices that correspond to the same angle were summed, which increased statistics significantly. To make the analysis simpler, the segments and the cores were assigned an identity number (detector ID) which is an integer between 0 and 35. For example: detector 0 represents the core of crystal A and detector 1 represents crystal A segment 1 and so on until detector 35 represents crystal D segment 8. Table 3.2 lists the detector ID's.

Table 3.2: Detector ID assigned to the cores and segments of the detector.

Crystal	Segment	ID	Crystal	Segment	ID
A	0	0	C	0	18
-	1	1	-	1	19
-	2	2	-	2	20
-	3	3	-	3	21
-	4	4	-	4	22
-	5	5	-	5	23
-	6	6	-	6	24
-	7	7	-	7	25
-	8	8	-	8	26
B	0	9	D	0	27
-	1	10	-	1	28
-	2	11	-	2	29
-	3	12	-	3	30
-	4	13	-	4	31
-	5	14	-	5	32
-	6	15	-	6	33
-	7	16	-	7	34
-	8	17	-	8	35

3.5.6 Singles spectra

A ROOT/C++ code was written to build singles spectra from the ROOT trees that contain the sorted, gain-drift corrected, gain matched and crosstalk corrected data. The singles spectra are 1-dimensional histograms that contain the number of counts per channel, where the channels are mapped to the gamma-ray energy. In our case the channel was equal to the gamma-ray energy in keV. Singles spectra were sorted for each and every segment and core. These spectra were used to extract the gamma-ray intensities. Throughout this thesis the singles spectra are referred to as the direct spectra.

Typical direct spectra for the two sources ^{60}Co and ^{133}Ba are shown in Figs. 3.11 – 3.14.

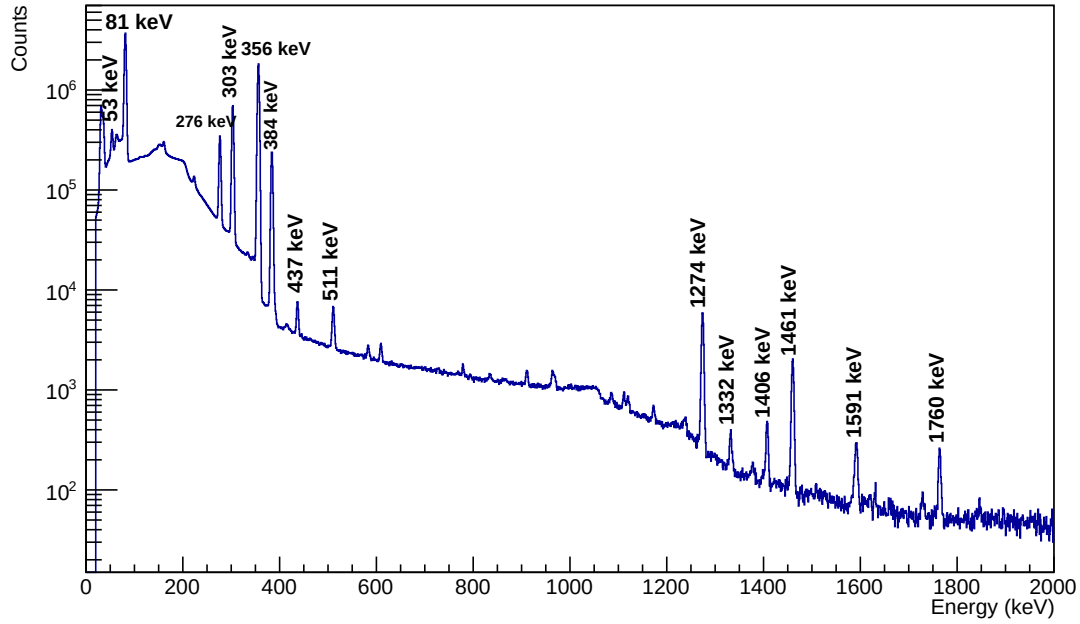


Figure 3.11: Gain-matched direct spectrum of ^{133}Ba for crystal A segment 1.

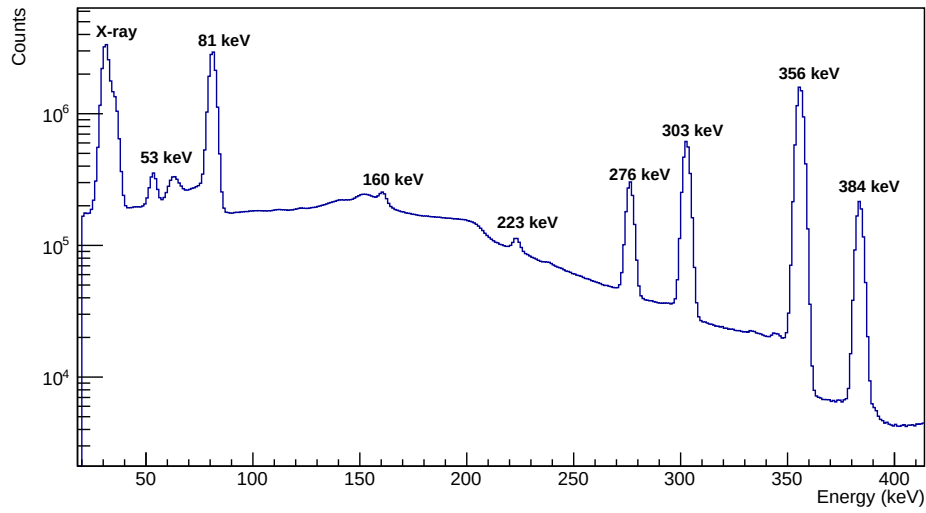


Figure 3.12: A portion of the direct spectrum of ^{133}Ba crystal A segment 1 showing the gamma rays of interest.

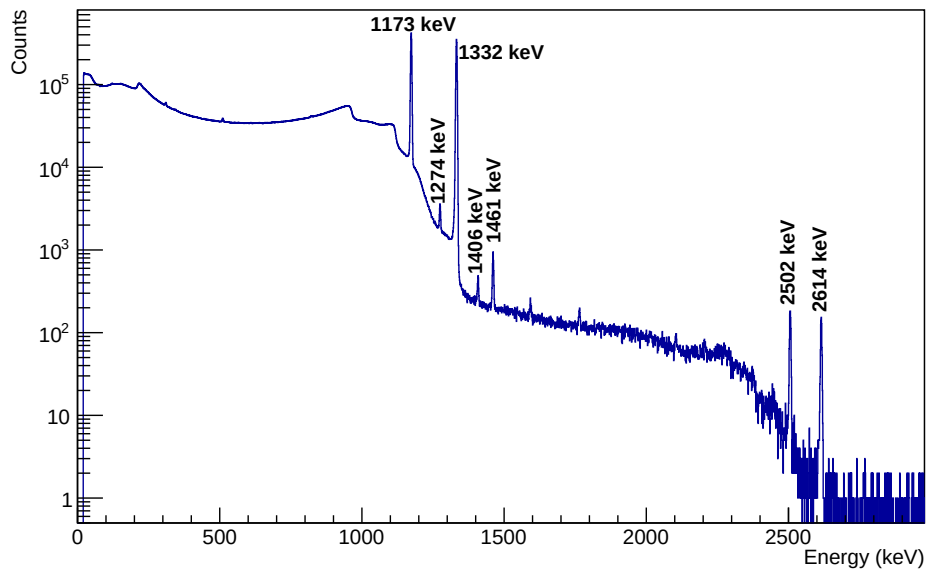


Figure 3.13: Gain matched direct spectrum of ^{60}Co decay for crystal A segment 2.

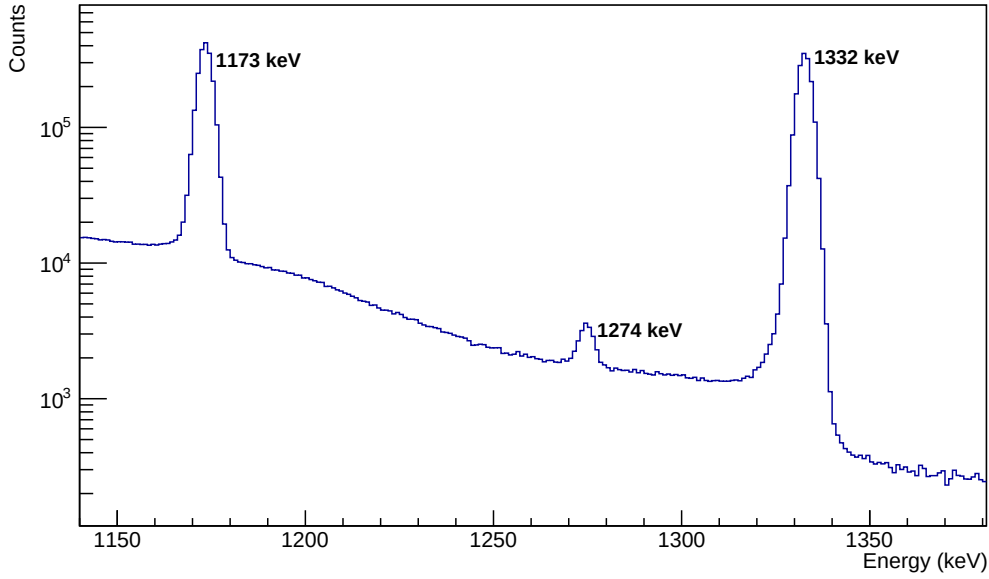


Figure 3.14: A portion of the direct spectrum of ^{60}Co decay for crystal A segment 2 showing the gamma rays of interest.

3.5.7 Coincidence matrices

In order to perform gamma-gamma angular correlation analysis, 2-dimensional gamma-gamma matrices were needed. The x - and y -axes of the angular correlation matrix store the energy entries for events when two gamma rays are detected in coincidence with each other on two different segments. For instance, if the segment s_i registers energy E_1 in coincidence with segment s_j which registers energies E_2 , the angular correlation matrix s_i-s_j will be incremented at channel (E_1, E_2) . Each matrix contained all coincidence data for the specific segment combination and for all gamma rays. Angular correlation matrices, corresponding to all $32 \times 31 = 992$ segments combinations were needed.

A C++/ROOT code was written to generate these 992 angular correlation matrices, see Appendix D. This code also sorted a time spectrum, that presents the time difference

between the gamma-ray energies detected in coincidence. The area of the peak in the time spectrum corresponds to the number of coincidence events. For instance, Fig. 3.15 plots on the x-axis the time difference between the time stamps (in $\times 10$ timeclocks) of the detected gamma-rays in segments 1 of crystal A and segment 3 of crystal B.

An example of a single gamma-gamma angular correlation matrix for detectors 2 and 1 is

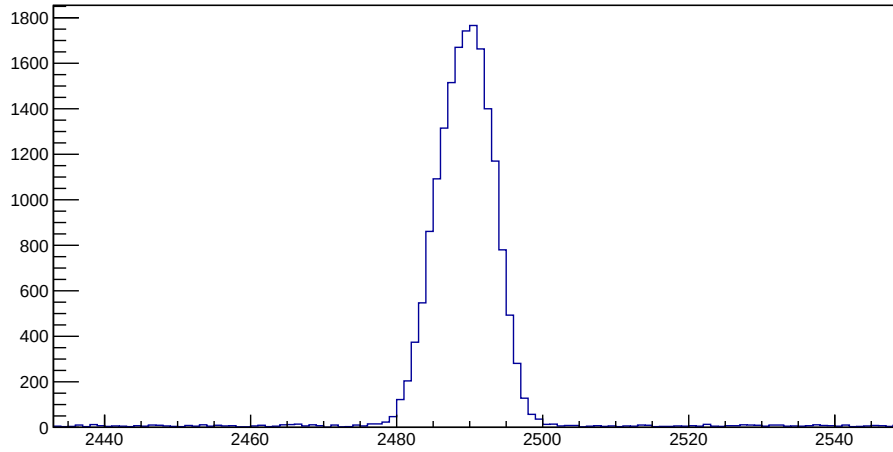


Figure 3.15: A time spectrum for coincidence events between detector 1 and detector 12. The spectrum is incremented when detector 1 registers a gamma-ray of energy $E_{\gamma 1}$ in coincidence with $E_{\gamma 2}$ registered on detector 12, and represents the time difference between the time stamps for the two gamma rays.

shown in Fig. 3.16. The gamma rays detected in detector 2 are placed on the x -axis while those in detector 1 on the y -axis. The diagonal lines are a result of Compton scattering.

Out of the 992 angular correlation matrices some were added together as they corresponded to the same angle. For instance the matrices for the detector IDs of (1, 2), (10, 11), (19, 20) and (28, 29) corresponded to the same angle as described in section 3.5.5. Therefore, all the matrices that corresponded to the same angles were summed

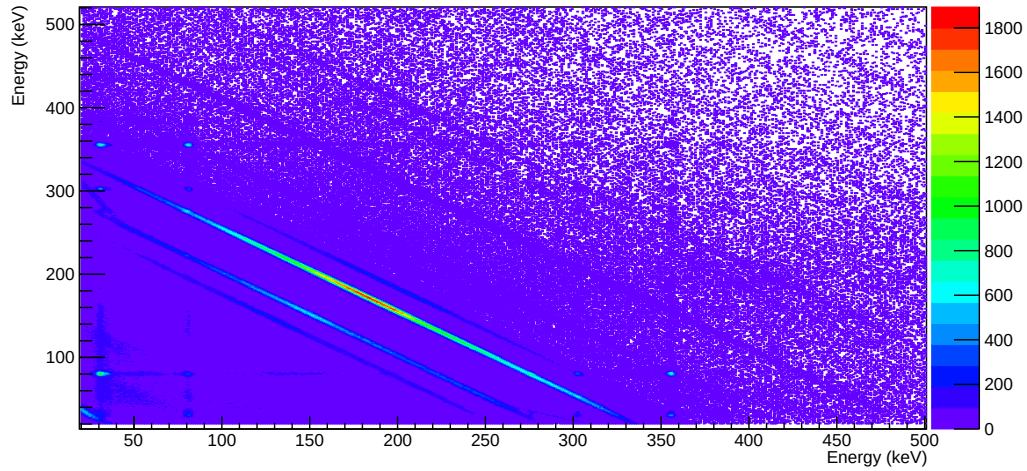


Figure 3.16: The gamma-gamma angular correlation matrix for segment 2 (x-axis) and segment 1 (y-axis) for the ^{133}Ba decay.

together to increase the statistics. As an example the summed matrix for angle $\theta = 16^\circ$ is shown in Fig. 3.17. This matrix corresponded to the sum of 8 angular correlation matrices. A 3D-plot for this matrix is also shown in 3.18, where most of the coincidence peaks are visible.

As there are 122 different angles, 122 summed angular correlation matrices can be analyzed.

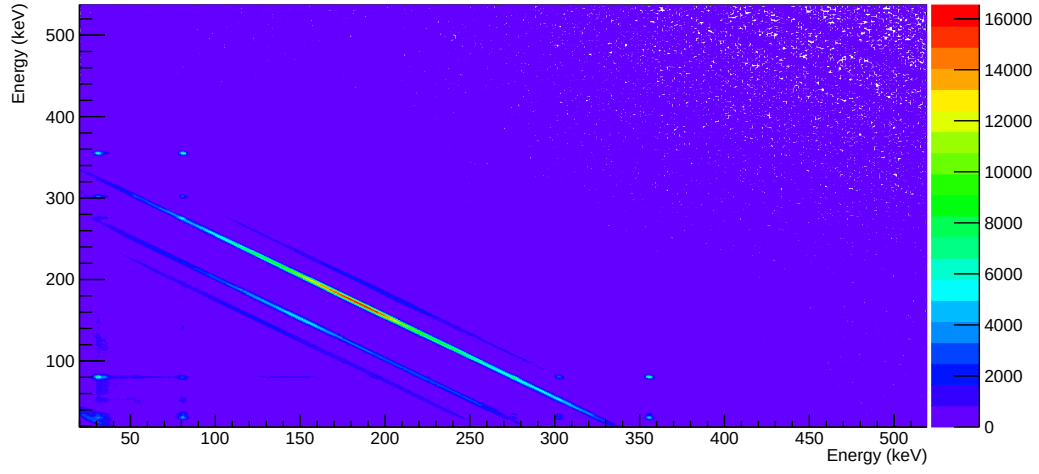


Figure 3.17: A summed gamma-gamma angular correlation matrix corresponding to the first angle $\theta = 16^\circ$, containing 8 angular correlation matrices.

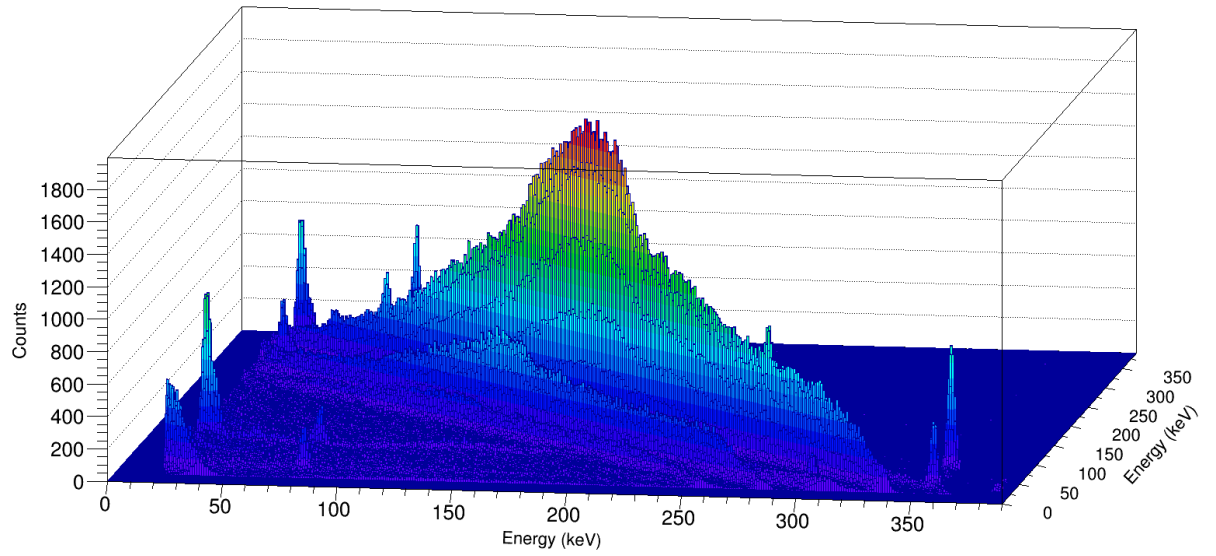


Figure 3.18: A summed gamma-gamma angular correlation matrix corresponding to $\theta = 16^\circ$.

3.5.8 Coincidence gates

In order to determine the coincidence intensity of the gamma rays of interest, a gated spectrum was generated. Gating means selecting a range of channels on one of the axes of the 2-dimensional gamma-gamma matrix and projecting a 1-dimensional spectrum of the other axis. Examples are shown in Fig. 3.19 and Fig. 3.20. The projected spectrum called “gated spectrum” contains all the gamma rays that are in coincidence with the gating transition.

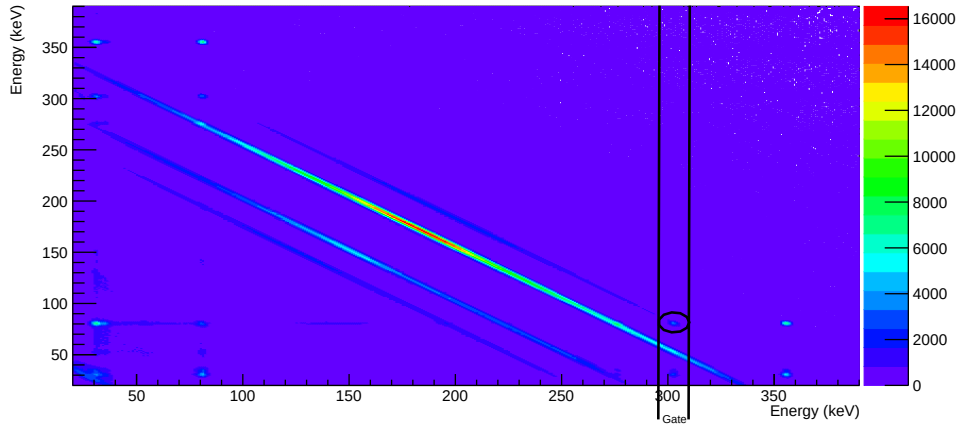


Figure 3.19: Setting a gate on the gamma-gamma angular correlation matrix. In this case the gate is set on the 303 keV gamma-ray, registered in segment 2 (x -axis), creating a gated spectrum in segment 1 (y -axis), shown in Fig. 3.20. The coincidence peak is indicated by the black circle.

In Fig. 3.19 an example where a gate on the 303 keV gamma ray is set on segment 2 (x -axis) is shown by the black lines. The matrix is sliced along these lines and projected onto the y -axis. This gated spectrum is shown in Fig. 3.20. The area of interest is indicated by the black circle in Fig. 3.19. This area corresponds to the coincidence events when the 303 keV gamma ray is detected in coincidence with the 81 keV gamma ray.

As shown in Fig. 3.20, the generated gated spectrum also contains background peaks. The two coincidence peaks at 53 keV and 81 keV correspond to true coincidences as these

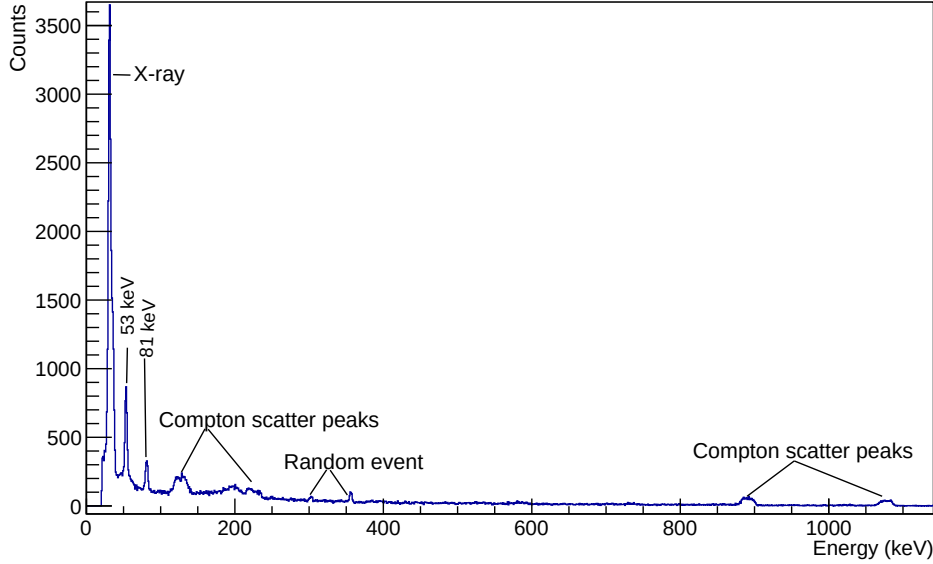


Figure 3.20: A spectrum gated on the 303 keV gamma-ray. The observed coincidence peaks are 81 and 53 keV transitions in coincidence with the 303 keV gamma-ray. The spectrum illustrates the different peaks produced by different backgrounds. The peaks are labeled in the spectrum according to the type of background that produces them. The spectrum was extracted from the summed angular correlation matrix for angle $\theta = 16^\circ$.

gamma-rays are in coincidence with the 303 keV transition. There are also other peaks in the spectrum such as the 303 keV and the 356 keV peaks. They are random coincidence peaks. They appear in the rare cases where two gamma rays are detected in coincidence, but these gamma rays are emitted in two different nuclear decays. Lastly, there are broad bumps labeled Compton scattering peaks. These arise as a result of the Compton scattering of the 356 keV, 383 keV, 1273 keV and 1460 keV gamma rays, where 303 keV was deposited in segment 2 (on the x -axis, where the gate is set), while the remaining energy

is detected in segment 1 (y -axis, where the projection is made).

3.6 Background subtractions

There are a number of sources that contribute to the background in the spectra. These are: room background, coincidence summing, random coincidences, continuous Compton background and Compton scatter peaks. The first step to correct for background is to identify the sources of the background that affect the analysis. Therefore the ^{133}Ba and ^{60}Co spectra with well known energy peaks were investigated and all additional peaks in the spectra were identified.

Coincidence summing is the detection of two gamma rays at the same time in the same detector, such that they are recorded as one event. Some coincidence summing was observed in the data mostly due to coincidence of two X-rays. It did not considerably affect the analysis because in most cases the coincidence summing peaks did not overlap with the peaks of interest.

Gamma-ray peaks observed due to room background were also identified. These are: 511 keV, 1173 keV, 1274 keV, 1332 keV and 1461 keV. They had different energies from the gamma rays of interest and also did not affect the analysis.

The background contributions that significantly affected the measurements and in some cases made the analysis quite complicated were the random coincidences, the Compton scatter peaks and the background under the gate. This section aims to explain in detail how these the background contributions were analyzed and corrected for.

3.6.1 Background under the gate

A gate is set on a range of channels at the peak of interest. Therefore there is always a contribution from events that are not in coincidence with the peak of interest but with the background underneath, see Fig. 3.21. To describe such events a background spectrum is generated by setting a gate on the left (lower energy) and on the right (higher energy) of the peak of interest, see Fig. 3.21. The sum of the two background gated spectra produces a spectrum of the events that are in coincidence with the background under the peak Fig. 3.22. This is called (in this work) the background gate and should be taken on regions that do not include any gamma-ray peaks. In most cases the background gate is chosen such that it has the same width as the gate on the peak of interest, if not, it is normalized to the number of channels of the gate on the peak. Then the background gated spectrum is subtracted from the spectrum gated on the peak of interest to remove the background under the peak. The equation used is as follows;

$$A_C = C_A - k * C_{Bg}, \quad (3.6)$$

where A_C is the background-corrected spectrum, k is a normalizing coefficient, C_A is the gated spectrum and C_{Bg} is the background-gated spectrum.

Figure 3.22 illustrates the gated spectrum in navy and the background spectrum in red. The two coincidence summing peaks at 62 keV and 70 keV are due to coincidence summing of the k_α and k_β X-rays at 31 keV and 35 keV.

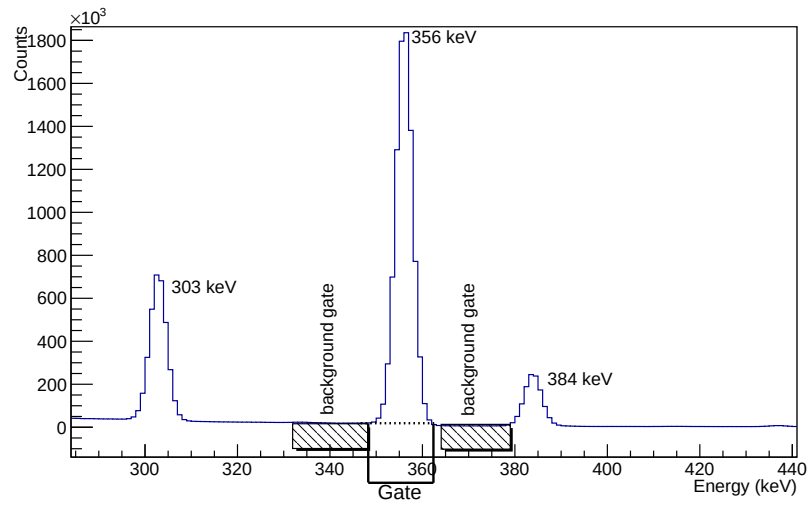


Figure 3.21: A gate set on the 356-keV photopeak and background gates set on the left and on the right of the peak of interest. The gated spectra obtained are shown in Fig. 3.22.

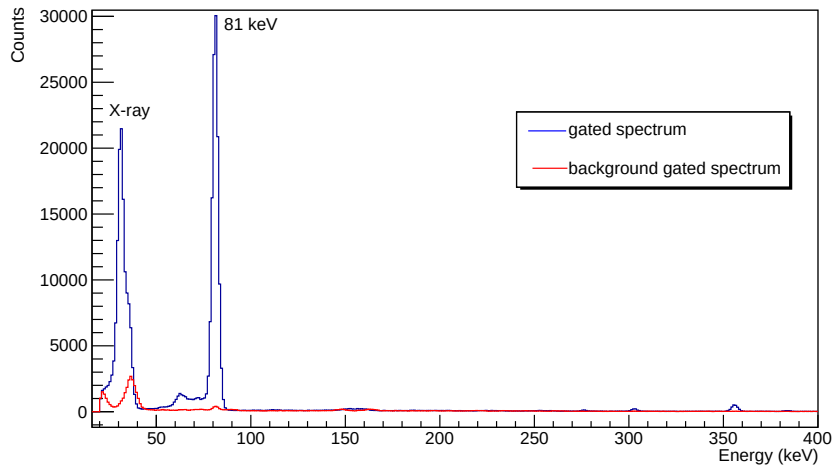


Figure 3.22: A spectrum gated on the 356-keV photopeak, shown in blue and a sum of the two spectra gated on the background on the left and on the right of the photopeak, shown in red.

3.6.2 Compton scatter peaks

In some cases, broad peaks were observed in the gated spectrum. These peaks were not part of the background under the gate nor of the random coincidences. For instance the spectrum gated on the 384-keV gamma ray in Fig. 3.23 shows such peaks. They are

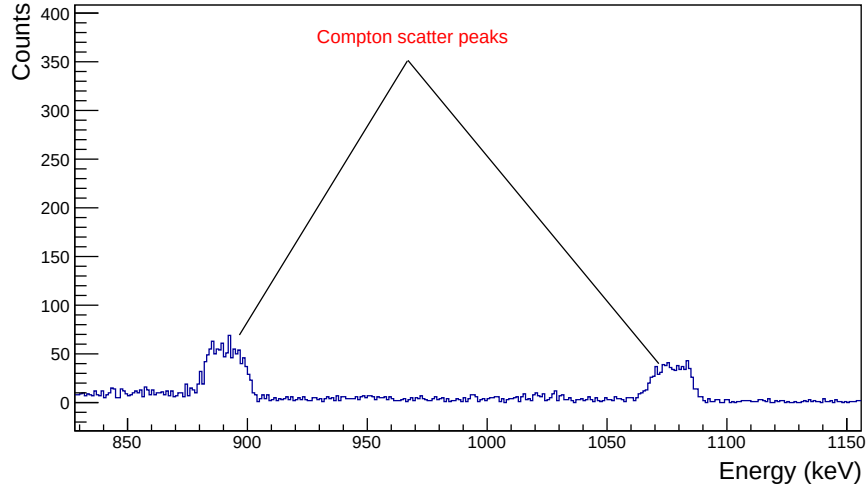


Figure 3.23: An example of two Compton scatter peaks.

called Compton scatter peaks. They are caused by the Compton scattering of gamma rays with energy higher than the energy of the gating gamma ray. For example Fig. 3.23 shows a portion of the spectrum gated on 384 keV shown in Fig. 3.20. The two peaks are caused by the scattering of the 1274 keV and 1461 keV gamma rays. In this case the 1274 keV gamma ray deposited 384 keV in segment 1 and the rest of the energy was absorbed in segment 2. So when the gate on 384 keV is set on segment 1 (x-axis of the matrix) and projected on segment 2 (y-axis of the matrix) the energy absorbed in segment two is registered. This energy is the difference of the gamma-ray energy 1274 keV and energy gate, 303 keV, producing a broad bump at $E = 971$ keV. The same applies to the 1461 keV scatter peak, an energy of 384 keV is deposited in segment 1 while the rest of the 1461 keV gamma ray is absorbed in segment 2. As a result the Compton scatter peaks

appear at $E = 1157$ keV.

One characteristic feature of a Compton scatter peak is that it does not have a Gaussian shape. It has a rectangular like shape with the flat top and width that matches the width of the gate. The second characteristic feature of a Compton scatter peak is that it shifts if the gated is shifted. For instance the spectrum showed in navy in Fig. 3.24 is gated on 303 keV.

It shows a Compton scatter peak at $E = 971$ keV and $E = 1157$ keV. However if

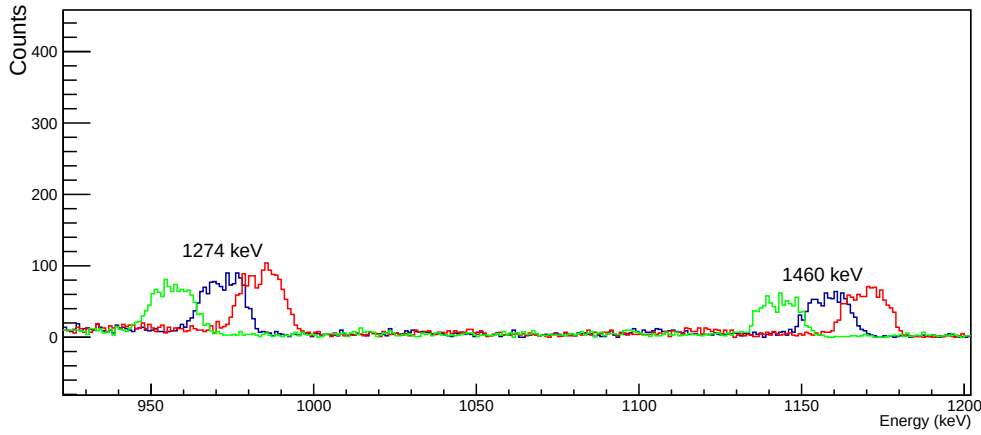


Figure 3.24: The 1274 keV and the 1460 keV Compton scatter peaks produced in the spectra gated on the 303 keV (in navy), 283 keV (red) and 323 keV (green). The green and red spectra are a result of the gates on the left and right of the peak.

the gate is set on 283 keV the Compton scatter peaks shifts with 20 keV to higher energy (red spectrum). In most cases the Compton scatter peaks do not affect the analysis because they do not overlap with the peaks of interest. However, there are cases where they do overlap with a peak of interest.

Consider for instance the 303 - 81 keV cascade of ^{133}Ba , shown in Fig. 3.25. Consider an angular correlation measurement between detector 1 and detector 2 of crystal A, noting that these segments are next to each other (and the angle θ between them is small). Assuming that the cascade of interest is $3/2^+ \rightarrow 5/2^+ \rightarrow 7/2^+$ and the gamma rays of 303 keV are detected in segment 1 and 81 keV detected in segment 2. There is a

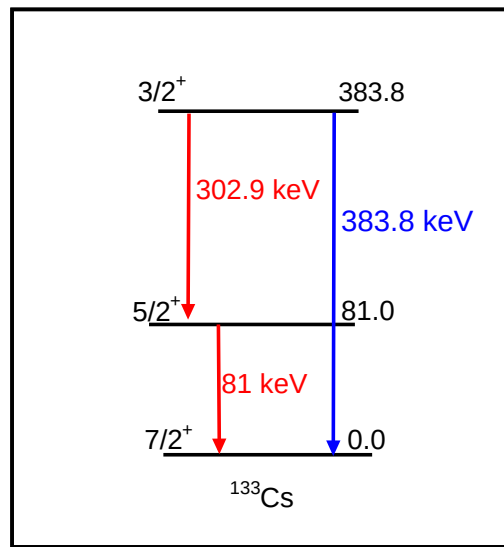


Figure 3.25: An example of transitions that result in a Compton scatter peak under the peak of interest.

probability that the 384 keV gamma ray from ^{133}Ba will deposit 303 keV in segment 1 and scatter into segment 2 where it will deposit the rest of its energy of 81 keV. Since scattering is a rapid process this event falls within the time gate set for coincidence events and is recorded as such. Such events result in a Compton scatter peak at 81 keV. Therefore the true coincidence peak at 81 keV will lie on top of a Compton scatter peak. This makes

it hard to deduce the shape of background under the 81 keV peak and to measure its area.

In order to illustrate how Compton scatter peaks are identified and how they affect the area of the coincidence peak, and how to correct for them, the analysis of the 303 - 81 keV cascade is described. Figure 3.26 shows three gated spectra where the 303-keV gates were set on the same set on the same angular correlation matrix corresponding to $\theta = 16^\circ$. The first gate was set on the 303 keV gamma ray with a width of 12 channels, see Fig. 3.26a. In this case the coincidence peak at 81 keV seems to have tails on both sides and it is very unclear where the background is. The second gate was set with a width of 20 channels, see Fig. 3.26b, and in this spectrum the Compton scatter peak starts to show because the gate is wider. The last gate on the 303-keV gamma ray was set with a width of 30 channels, see Fig. 3.26c. This allowed a clear identification of the height of the Compton scatter peak background. Therefore one could measure the area of the 81 keV above the background.

As an alternative we attempted to subtract the contribution of the Compton scatter peak. The 276 keV gate produce a Compton scatter background under the 160 keV true coincidence peaks, as shown in Fig. 3.27b. Background gates were set on the left and on the right of the 276 keV peak (as described in Section 3.6.1 and Fig. 3.22), each with half of the width of the 276 keV gate. They produce shifted Compton scatter peaks shown in red and green in Fig. 3.27a. These background-gated Compton scatter peaks were aligned with the coincidence peak as shown in Fig. 3.27b. The pink spectrum is the sum of the two shifted background bumps. It was subtracted from the gated spectrum (navy). The Compton scatter corrected spectrum is shown in Fig. 3.28. This method of subtraction proved to be successful. In some cases however the normalization was a bit difficult. For instance, when the Compton-scatter background was not flat under the peak. Angular

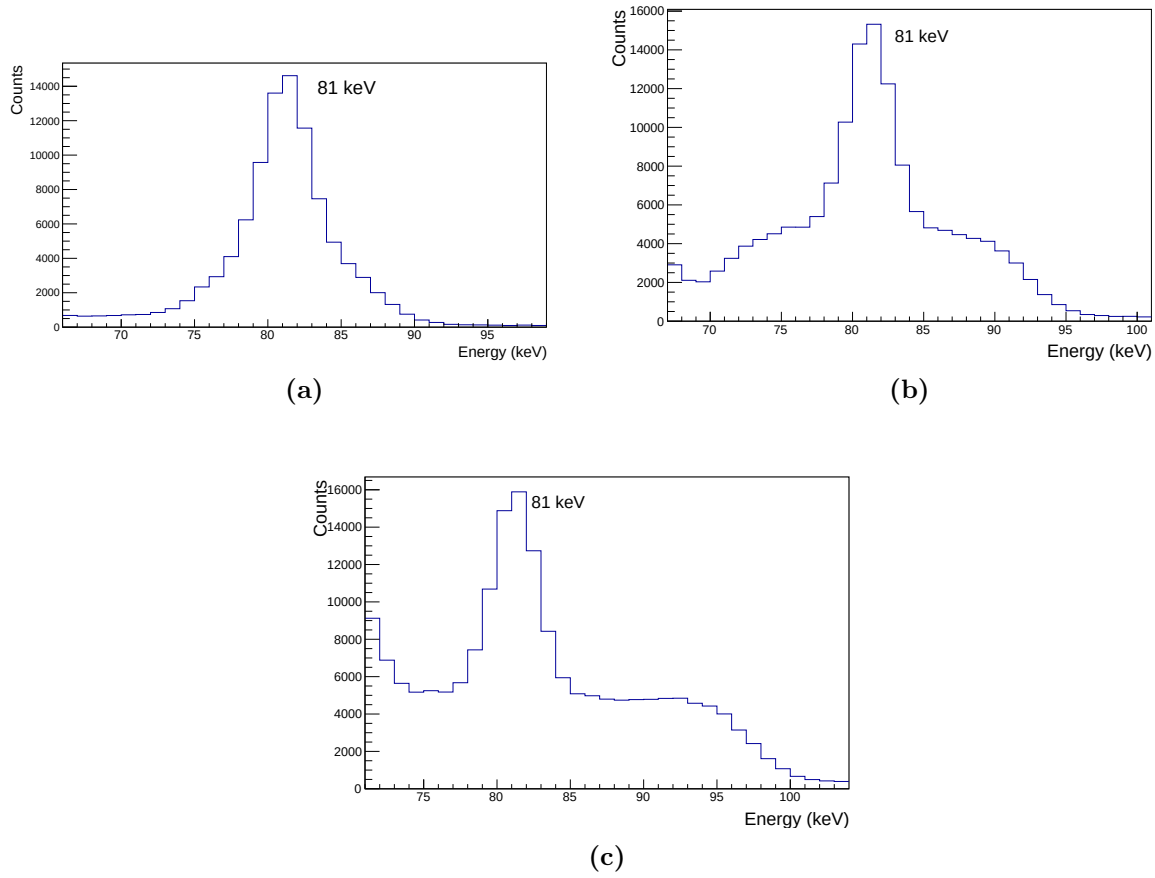
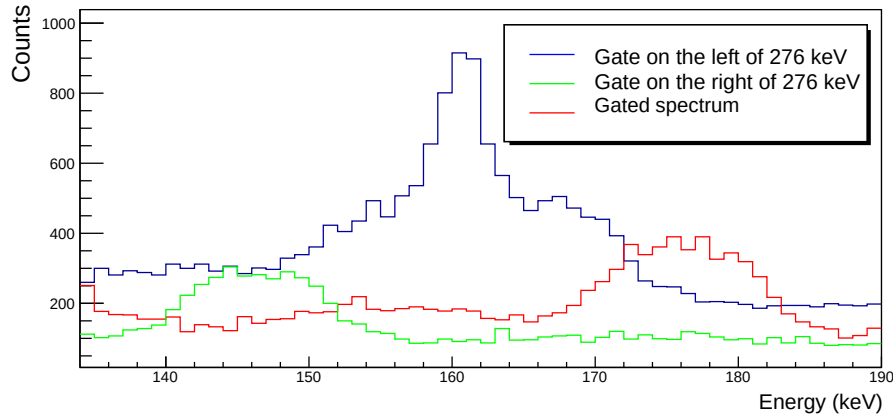
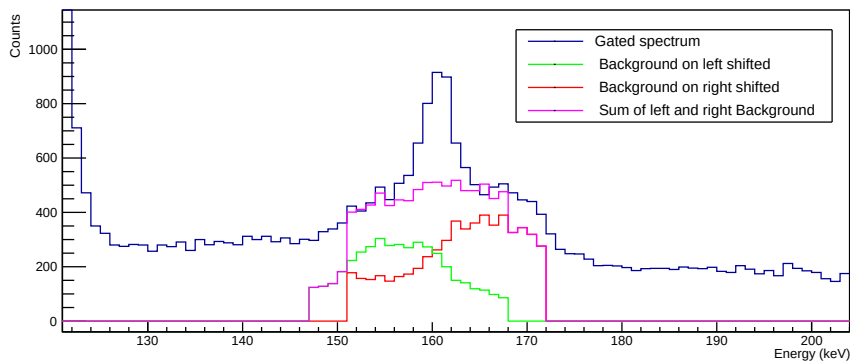


Figure 3.26: (a) A gate set on the 303-keV photopeak in ^{133}Ba , with a gate width of 12 channels. (b) Gate set on the 303-keV photopeak in ^{133}Ba , with a gate width of 20 channels. (c) Gate set on the 303-keV photopeak in ^{133}Ba , with a gate width of 30 channels.

correlation measurements taken in detector pairs that are next to each other (small θ) were significantly affected by Compton scatter peaks, particularly to the 276 - 160 keV and 303 - 81 keV cascades.



(a)



(b)

Figure 3.27: (a) The 160-keV coincidence peak lying on top of a Compton-scatter background. The blue histogram was generated by gating on the 276-keV gamma ray. The red and green histograms were generated by gating on the background under the 276-keV peak (gating on left and right of the 276-keV peak). (b) Same as (a) but the two Compton-scatter background histograms were shifted to align with the coincidence peak. The pink histogram is the sum of the background spectra.

3.6.3 Random coincidences

An event created when two gamma rays are detected in coincidence but are, in fact, gamma rays emitted from two different nuclei that happened by chance to decay at the same time is known as random coincidence. Random coincidences are affected by two

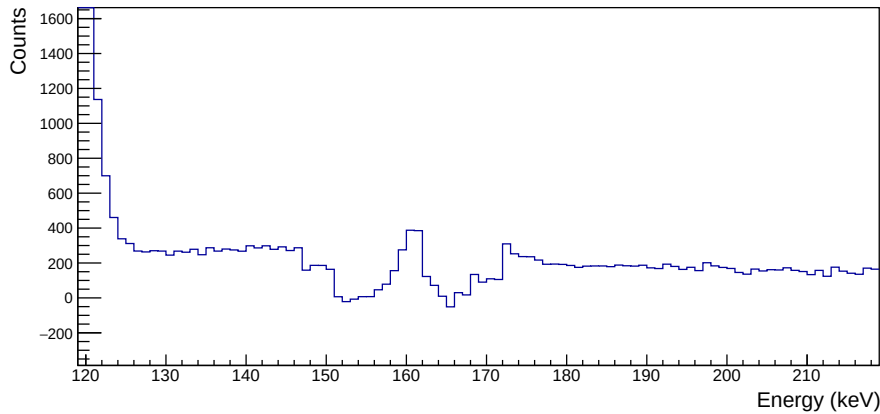


Figure 3.28: The Compton-scatter corrected spectrum. The gated spectrum and the background-gated spectrum are shown in Fig. 3.27a.

factors. The first is the width of the coincidence window. If the time gate is large it is more probable that some fraction of the data will be caused by accidental coincidences. The larger the coincidence gate, the larger the probability for random coincidences. The random coincidence rate can be decreased by decreasing the width of the coincidence window, but it can not be completely eliminated. Conversely a too narrow coincidence window could cut out some true coincidences. The second factor is the activity of the source, if the source is very active, there is a large probability that more than one nucleus will decay within the coincidence window thus increasing the chance of random coincidences.

The random coincidence events can affect to some extent the coincidence intensity of interest, therefore they should be corrected. After a gate had been set on the gamma-gamma angular correlation matrix, the generated coincidence spectrum contained true coincidence peaks but also random coincidence peaks.

An example of a coincidence spectrum produced when a gate is set on 356 keV is shown

in Figs. 3.29 and 3.30. The 356-keV gamma ray is in coincidence with the 81-keV transition, therefore the 81-keV gamma ray peak in Fig. 3.29 is a true coincidence peak. Only the 81-keV transition is in coincidence with the 356-keV gamma ray, therefore no other gamma-ray peaks are expected in the coincidence spectrum. However, this is not the case because we can see other peaks in the gated spectrum.

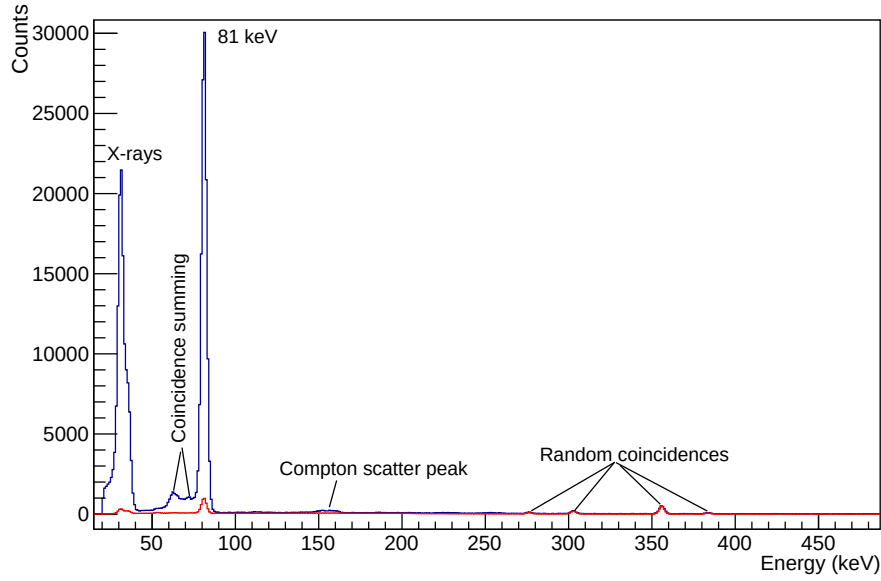


Figure 3.29: A coincidence spectrum generated by gating on 356 keV in the angular correlation matrix for $\theta = 16^\circ$, shown in navy. The spectrum in red is the normalized direct histogram, which is used to correct for random coincidence events.

The 276 keV, 303 keV, 356 keV, and 384 keV peaks in Fig. 3.29 are random coincidence peaks. It is clear that the random coincidence peaks are small in comparison with the true coincidence peak. However, since they may contribute to the area of the true coincidence peak, see Fig. 3.30, they were corrected for. The ratio, k , of the area of one random coincidence peak from the coincidence spectrum, A_r , and the area of the same peak in the direct spectrum A_d was calculated:

$$k = \frac{A_r}{A_d}. \quad (3.7)$$

This factor was used to normalize the direct spectrum and then subtract it from the gated spectrum. This was done for corresponding matrices and direct spectra. For example, if the matrix that has to be corrected for random coincidences was a sum of coincidence matrices for which detectors 1, 10, 19 and 28 were placed on the y-axis, the direct spectrum used to correct for this was also a sum of the direct spectra for the same detectors 1, 10, 19 and 28. The normalized direct spectrum used to correct for randoms is shown in Figs 3.29 and 3.30.

The gated spectrum produced after randoms subtraction is shown in Fig. 3.31. It is

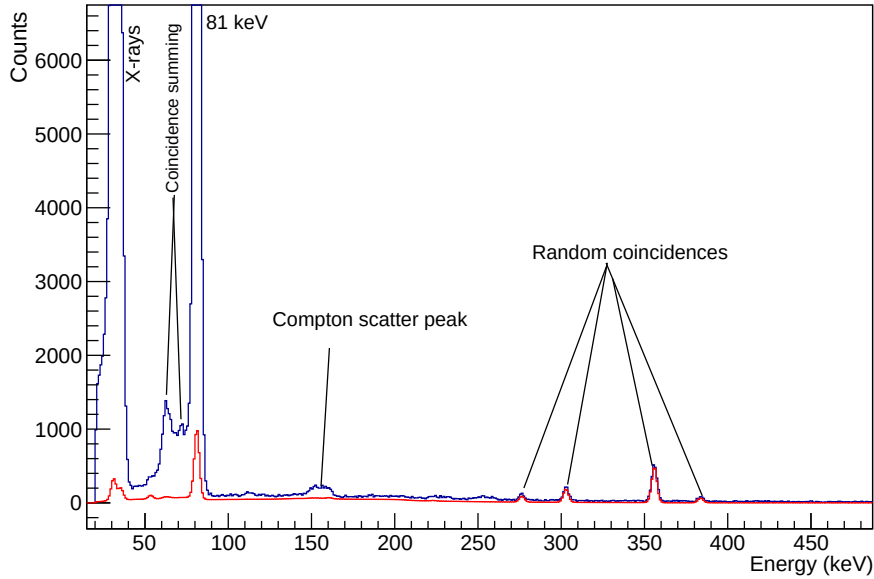


Figure 3.30: The same as in Fig. 3.29, but expanded on the y-axis to better illustrate the random coincidence peaks. The bump at ~ 155 keV is a result of the Compton scattering of the 511 keV annihilation gamma ray. The bumps at 62 and 70 keV are caused by the coincidence summing of the X-rays ($K_\alpha = 31$ keV and $K_\beta = 35$ keV).

clear that the randoms were successfully removed. The background correction technique

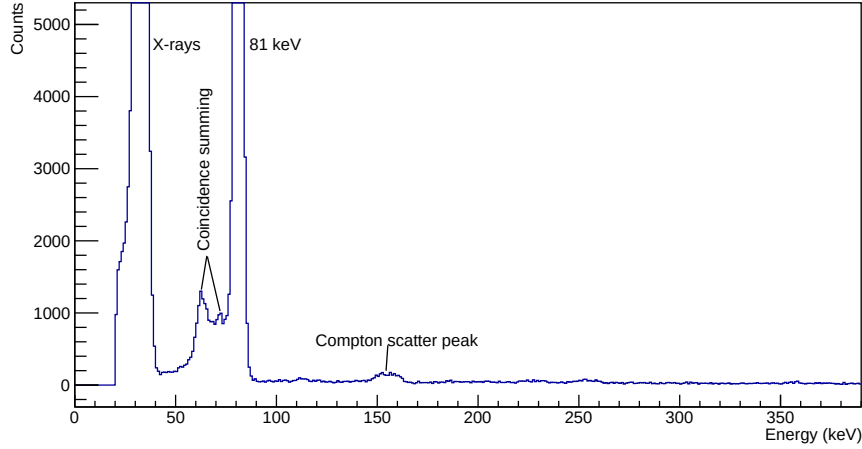


Figure 3.31: A spectrum gated on 356 keV after random coincidences correction.

used in this work involved: *(i)* correcting for background under the gate; *(ii)* removing Compton scatter bump if needed; and *(iii)* removing random coincidences. Such gated background-corrected spectra were used to measure the area of the coincidence peaks and deduce the coincidence intensity as a function of the angle θ .

3.6.4 Peak fitting

To extract the coincidence intensity the peak of interest, e.g. the 81 keV peak in the background corrected gated spectrum, the peak of interest was fitted (see Fig. 3.32). Peak fitting is a basic tool in gamma spectroscopy. In the present work the peak was described as a combination of a Gaussian and a polynomial curve. The polynomial term was used to describe the background under the peak, while the Gaussian described the peak. The fit parameters were varied to describe the peak of interest best.

Most peaks of interest in this work were fitted with a Gaussian and the resulting Gaussian parameters were used to calculate the area of the peak. The built-in ROOT Fit

function was applied. It uses a χ^2 minimization technique to find the best fit. It defines a parameter χ^2 which is equal to the sum of the squares of the differences between the experimental values and the fitted function values. The function considers the fit with the smallest χ^2 as the best fit.

An example where the 356-keV peak is fitted is shown in Fig. 3.32. The area of the

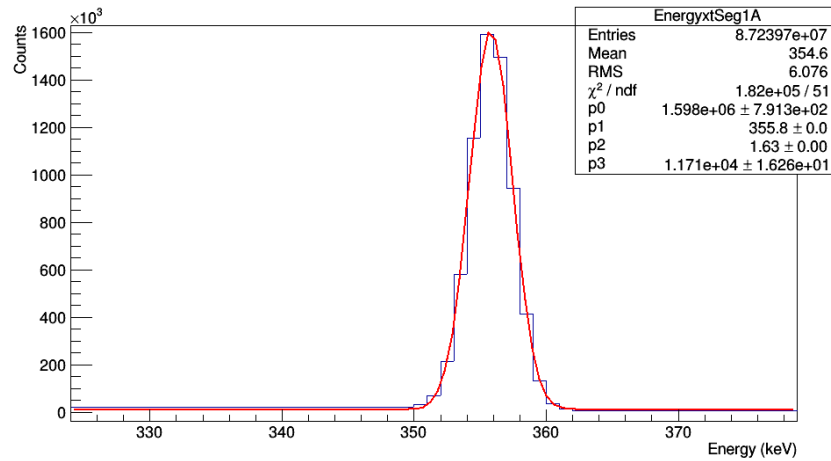


Figure 3.32: A fit for the 356-keV gamma ray using a Gaussian and a first order polynomial. p0 is the height of the peak, p1 is the centroid of the peak, p2 is the width of the peak, p3 describes the height of the background.

peak was calculated as,

$$A = ac \cdot \sqrt{2\pi}, \quad (3.8)$$

where a is the height of the peak and c is the Gaussian width σ . The uncertainty of the measurement is,

$$\Delta A = A \times \sqrt{\left(\frac{\Delta a}{a}\right)^2 + \left(\frac{\Delta c}{c}\right)^2}. \quad (3.9)$$

In some cases, it was difficult to fit the gamma-ray peaks in ^{133}Ba , particularly the low energy 53 keV and 81 keV peaks. In such cases an alternate way of calculating the area of the peak was used. The peak area was calculated as a sum of the counts in each

channel that belongs to the peak. To evaluate a background region under the peak, the regions on the left and on the right of the peak were carefully selected. Each background region was defined over several channels on the left and on the right of the peak and the averaged height for each of them was determined, shown as red dots in Fig. 3.33. Using the averaged background height on the left and on the right the height of the background under the peak was calculated, shown in green in Fig. 3.33 and subtracted from the peak area. In this case the statistical uncertainty on the peak area was taken as $\sqrt{A}$. In most cases the uncertainty caused by the background was neglected because it was too small compared to the total area of the peak.

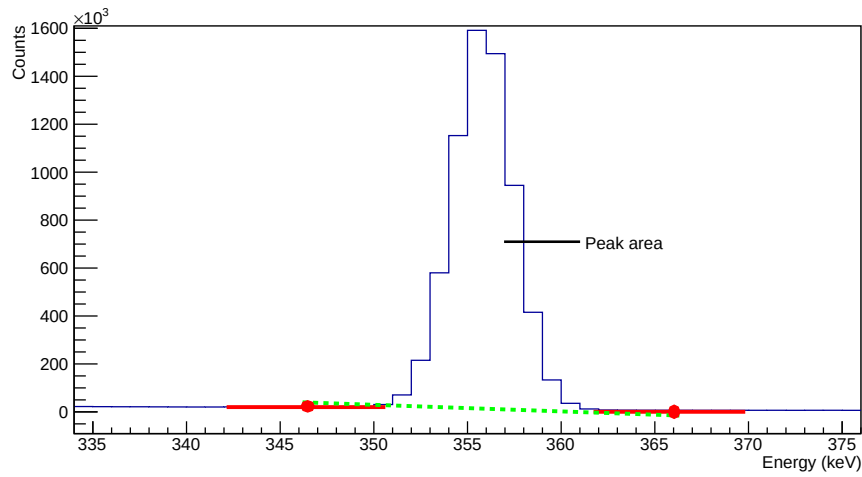


Figure 3.33: The 356-keV photopeak. Background regions were selected on the left and on the right of the peak, shown in red.

3.6.5 Efficiencies

The efficiency of a detector is given by the ratio between the number of the detected gamma rays and the number of the emitted gamma rays. It depends on the energy of the gamma ray and the solid angle of the detector; in our case the solid angle of each segment since the segments of the detector are used as individual detectors. The efficiency of the segments is needed in order to determine the relative angular correlation intensity. The efficiency for the iThemba LABS segmented clover detector was extracted from the singles spectra by fitting the gamma-ray peaks and calculating their area. The singles spectra contain all the recorded events during the data collection.

For a pair of detectors used where γ_1 is detected in segment i and γ_2 is detected in segment j , the relative efficiencies ϵ_i and ϵ_j correspond to the areas of the peaks for γ_1 in the direct spectrum of s_i and for γ_2 in the direct spectrum of s_j , respectively. Since several angular correlation matrices which correspond to the same angle θ were summed together (refer to Section 3.5.5), the efficiency was calculated as a sum of different terms. For instance take the case of a sum of four angular correlation matrices, $Matrice_1_3 + Matrice_10_12 + Matrice_19_21 + Matrice_28_30$. The efficiency is given by;

$$\epsilon_{tot} = \epsilon_1\epsilon_3 + \epsilon_{10}\epsilon_{12} + \epsilon_{19}\epsilon_{21} + \epsilon_{28}\epsilon_{30}. \quad (3.10)$$

In a general form, the efficiency that corresponds to a summed angular correlation matrix is given by,

$$\epsilon_{tot} = \sum_{ij} \epsilon_i \epsilon_j, \quad (3.11)$$

where the indices i and j correspond to the segment pairs for each angular correlation matrix that is part of the summed matrix. The code that extracts the single segment efficiencies is given in Appendix E and the code that sums the angular correlation matrices according to similar angles is given in Appendix F.

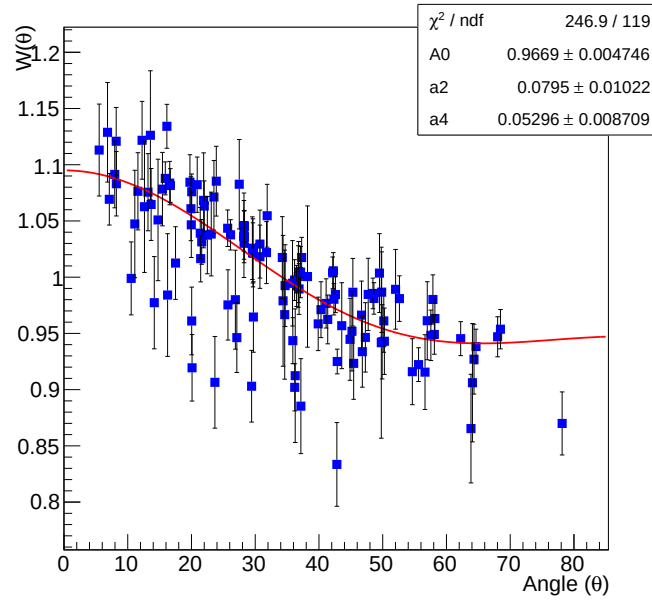
3.6.6 Angular correlation intensities

The crucial part in the angular correlation analysis is to determine the relative coincidence intensity at different angles formed by the different detector combinations. The relative coincidence intensity in a segment pair, e.g. $(seg_i E_1, seg_j E_2)$ is given by the number of coincidences between the two detectors for E_1 and E_2 divided by the product of detector efficiencies.

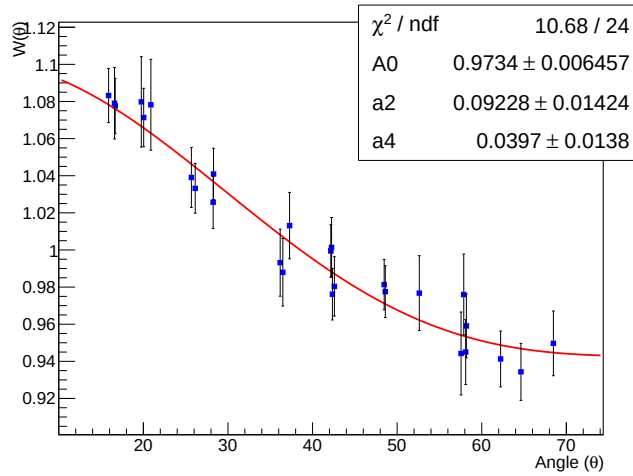
$$W(\theta) = \frac{A_{ij}}{\epsilon_i \epsilon_j}, \quad (3.12)$$

where A_{ij} is the area of the coincidence peak measured in the background corrected spectrum, and ϵ_i and ϵ_j are the efficiencies of the segments i and j respectively for energies E_1 and E_2 , respectively. The coefficients i and j span detectors 0 – 35, but the crystal cores were excluded from the analysis. The relative coincidence intensity for each angle was a sum of many detector pairs, and the efficiencies were also summed as discussed in Section 3.6.5. It was considered that the uncertainty on $W(\theta)$ is approximately equal to the uncertainty in the area of the coincidence peak, because the errors on the efficiencies were too small, thus they were neglected.

During the analysis it was observed that the $W(\theta)$ data points measured for angles that involve front segments only, formed a pattern, while the data points involving back segments are more scattered (see Fig. 3.34a), where data points for all 122 angles are shown for the 1173 - 1332 keV cascade. The back segment data have lower statistics and much larger uncertainty. But it seems that apart from that something else plays a role, as they show larger scattering, also that the ^{133}Ba source has low energy gamma rays, and several of them have negligible probability of being detected by the back segments. Therefore, as a first step for both ^{60}Co and ^{133}Ba , sources the front segments data were only considered. The angular correlation results for the 1173 - 1332 keV cascade in ^{60}Co for the front segment only shown in Fig. 3.34b. Thus the number of points used in this study decreased to 32. Note that the data points still cover an opening angular range of $\approx [10^\circ - 70^\circ]$.



(a)



(b)

Figure 3.34: Angular correlation fits for the 1173 - 1332 keV cascade, using data for: **(a)** all 122 angles, **(b)** the angles that correspond to combinations of front segments only.

The code that performs the background corrections and extracts the angular correlation intensities is given in Appendix G.

3.7 Extracting the a_k coefficients and calculating δ

After the angular correlation data points $W(\theta)$ were plotted, they were fitted using Legendre polynomials:

$$W(\theta) = A_0[1 + a_2P_2(\cos\theta) + a_4P_4(\cos\theta)], \quad (3.13)$$

where A_0 , a_2 and a_4 are angular correlation coefficients and P_2 and P_4 are the Legendre polynomials, given in Table 2.3. From the fit, the a_k coefficients and their uncertainties were extracted. Such fits are shown in Fig. 3.34a and Fig. 3.34b and reflect the multiplicities and mixing ratios of the transition.

Assume that a cascade of two gamma rays is investigated as shown in Fig. 3.35, and that the first gamma ray decays from a state with unknown spin I_i to a state with spin

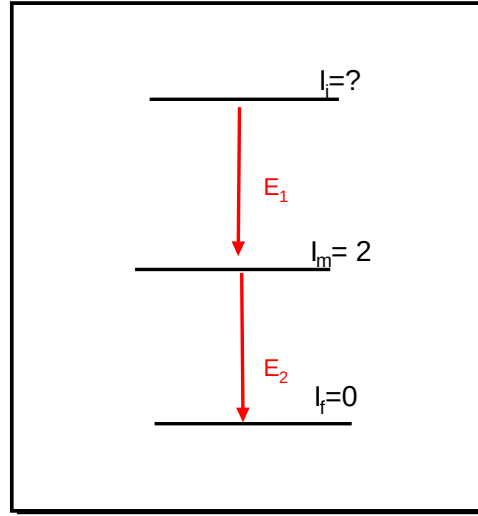


Figure 3.35: A cascade of two gamma rays.

$I_m = 2$, while the second gamma ray decays from this $I_m = 2$ state to $I_f = 0$ state. The most likely spins for I_i are 0, 1, 2, 3 and 4. Theoretical a_2 and a_4 coefficients for these cascades are shown in Fig. 3.36. Note that the coefficients change as a function of the mixing ratio if a mixed $M1 + E2$ transition is considered. The fit of the experi-

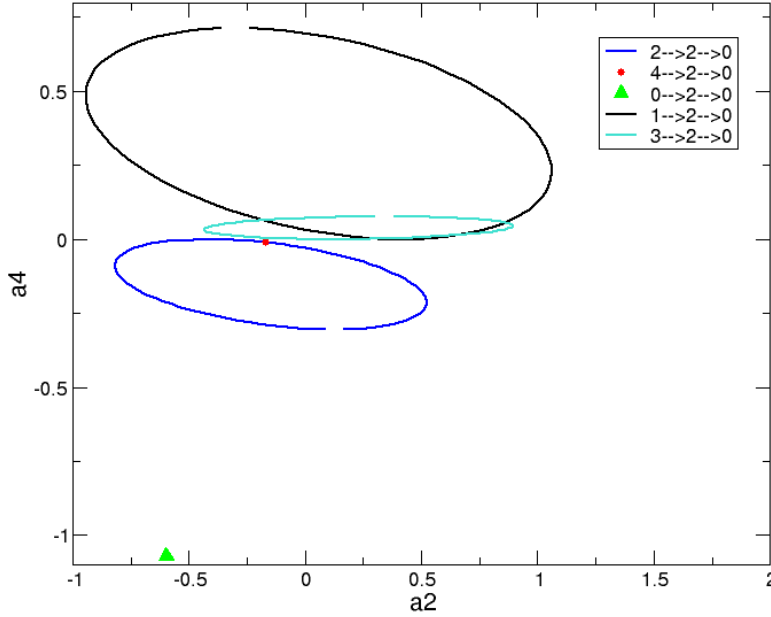


Figure 3.36: An a_2 versus a_4 plot for the possible spin assignments of $I_i = 0, 1, 2, 3, 4$ with varying mixing ratios.

mental angular correlation data delivers the experimental a_2 and a_4 coefficients. Taking into account the uncertainty in these coefficients, in this work they are compared with the theoretical a_2 and a_4 coefficients to determine the unknown spin I_i , and possibly the mixing ratio. In all cases considered here the spins of the three states I_i , I_m , and I_f were known, as well as the mixing ratio of one of the two transitions. The aim was to deduce the multipole mixing ratio δ of the second transition. The mixing ratio was calculated using the DELTA code [51]. It calculates the χ^2 value as a function of the (unknown) mixing ratio δ , taking into account the experimental a_2 and a_4 coefficients. It yields δ

as the value at which the χ^2 function has a minimum. For instance the χ^2 function for the 1332 - 1173 keV cascade assuming the 1173-keV gamma ray has a mixed $E2(+M3)$ multipolarity while the 1332 keV is a pure $E2$ transition, is shown in Fig. 3.37. The error

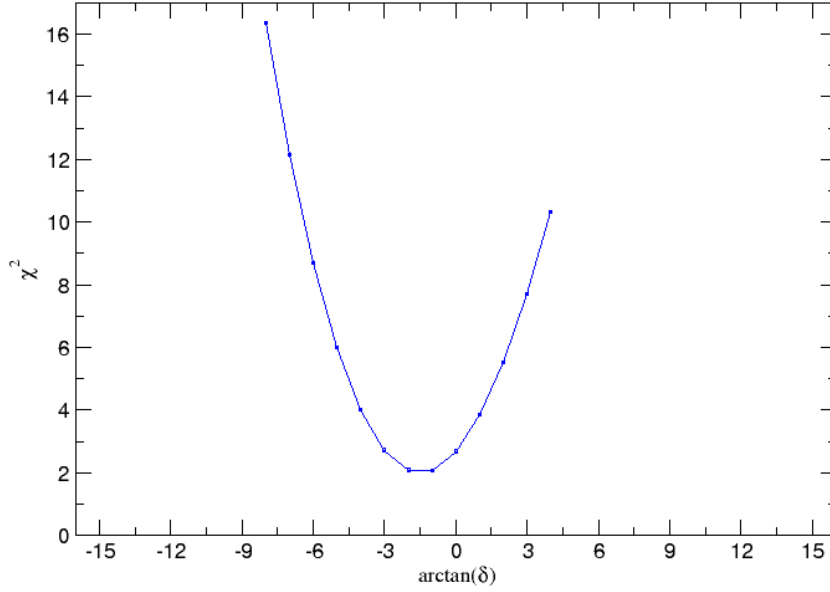


Figure 3.37: A χ^2 versus $\text{atan}(\delta)$. The χ^2 minimum determines the value of δ . The data is for the 1173 - 1332 keV cascade in ^{60}Co . The 1332-keV transition is assumed to be a pure $E2$, while the 1173 keV transition is assumed to be of $E2(+M3)$ multipolarity. A mixing ratio of $\delta = -0.026^{+0.043}_{-0.041}$ was deduced.

bars on the δ are extracted by determining δ_- and δ_+ which are the lower and higher limits of δ . These values represent the value of δ for $\chi^2 = 1.1 \times \chi^2_{\min}$, as shown in figure 3.38.

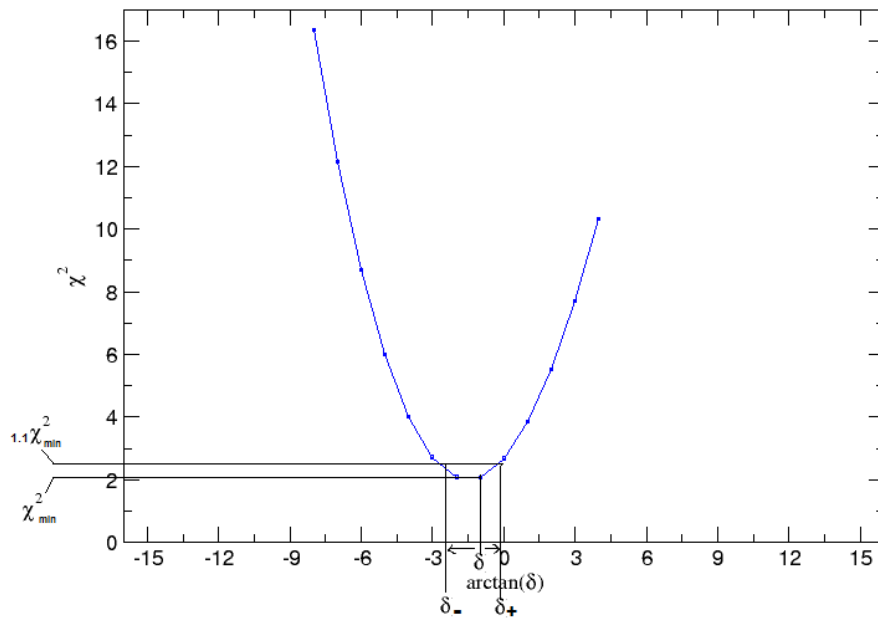


Figure 3.38: An example of how the error bars on δ are determined from a χ^2 plot.

Chapter 4

Results and discussion

The angular correlation functions for several cascades in ^{60}Co and ^{133}Ba were studied and the results are presented below.

4.1 The cascade of 1173 - 1332 keV in ^{60}Co

^{60}Co data were collected for ≈ 12 hours and analyzed in order to test the angular correlation technique and to establish the precision of the iThemba LABS segmented clover detector for such studies. The level scheme of ^{60}Co is shown in Fig. 4.1 [52]. The strong $4^+ \rightarrow 2^+ \rightarrow 0^+$ 1173 – 1332 keV cascade was studied.

Following the approach discussed in Chapter 3, gamma-gamma coincidence matrices for each segment-segment combination were constructed and those that correspond to the same angle were summed together.

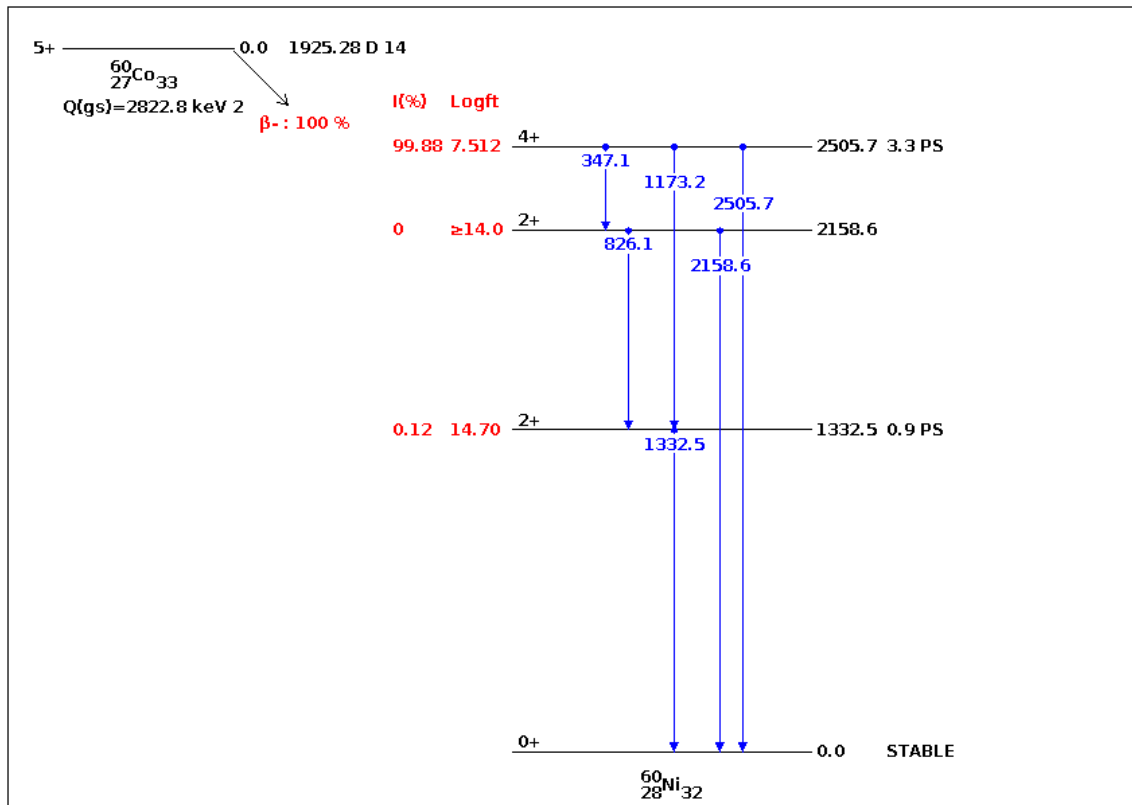


Figure 4.1: The level scheme of ^{60}Co populated by the decay of ^{60}Ni .

As an example the angular correlation matrix for the angle $\theta = 16^\circ$ is shown in Fig. 4.2. The circle indicates the peak that corresponds to the coincidence events of 1173 and 1332 keV gamma rays. In addition to the coincidence events, the Compton scattering

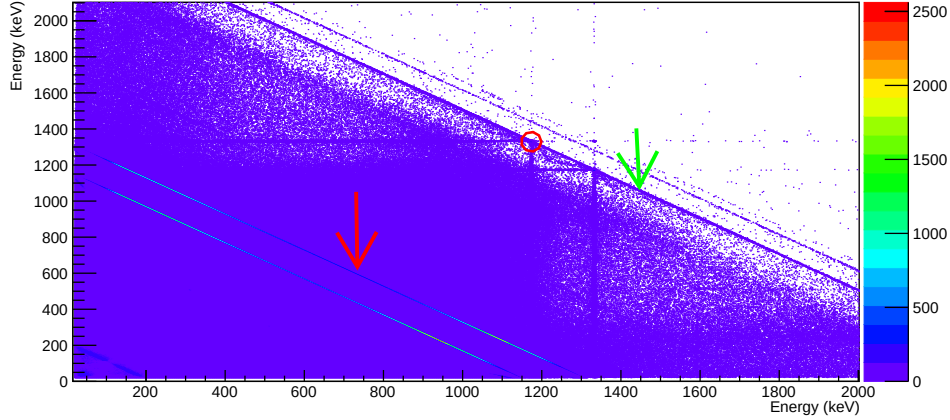


Figure 4.2: The summed angular correlation matrix of ^{60}Co for the angle $\theta = 16^\circ$. The circle indicates the coincidence of the 1173 keV and 1332 keV gamma rays, where 1173 keV is fully absorbed in the detectors on the x -axis of the matrix, while 1332 keV is fully absorbed in the detectors on the y -axis of the matrix. The arrows indicate Compton scattering ridges.

events are visible in the matrix as diagonal lines (ridges). As the angle between the two segments is small there is a large number of Compton scattering events. The Compton ridge crosses the x and y -axis at the energy of the gammaray that Compton scatters. For example, the red arrow indicates the Compton scattering ridge of the 1332 keV gamma ray. The green arrow indicates a different type of Compton scattering. It corresponds to $E = 2505$ keV which is a sum of the 1173 and 1332 keV gamma rays. It corresponds to events when one gamma ray e.g. the 1173 keV gamma ray is detected in detector 1 while the second gamma ray, e.g. the 1332 keV gamma ray scatters from detector 2 to detector 1 giving a total energy deposition of $1173 + 1332 = 2505$ keV, but a smooth Compton scattering type of spectrum in both detectors.

A gate was set on the 1173 keV peak on the x -axis producing a coincidence spectrum where the 1332 peak was of interest. Then a background gate was also set on the same axis on the left and right of the 1173 keV peak to create the background-gated spectrum. The latter was normalized and subtracted from the photopeak-gated spectrum to remove the background under the gate. There was no need for Compton-scatter peak correction. The random coincidences were subsequently corrected by using the direct spectra for the detectors that are on the y -axis of the matrix as explained in Section 3.6.3. The resulting spectra mentioned above are shown in Fig. 4.3.

After all background subtractions the number of the 1173 – 1332 keV coincidences

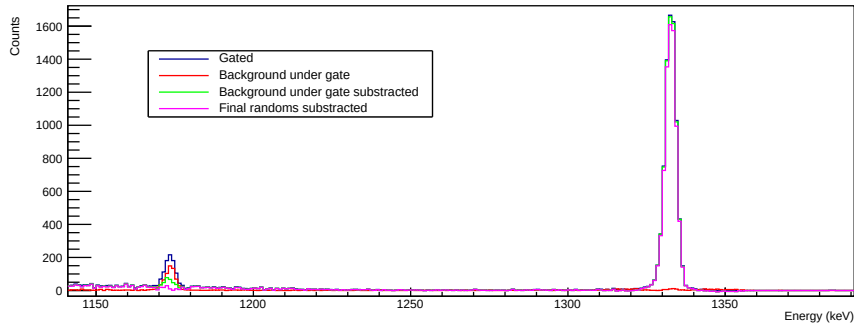


Figure 4.3: The strong gamma-ray peak corresponds to the 1332 keV coincidence peak. The pink histogram is the final output after all background subtractions have been done. The small gamma-ray peak corresponds to 1173 keV and it is a result of mostly random coincidences and background under the gate.

which corresponds to the peak area of the 1332 keV peak in the background-corrected spectrum gated on the 1173 keV transition was determined for angle $\theta = 16^\circ$ as described in Section 3.6.5. The intensity, $W(\theta)$, of the angular correlation function was calculated for $\theta = 16^\circ$ using the coincidence area and the efficiency. This procedure was followed for all 32 angles and the $W(\theta)$ data points were plotted in Fig. 4.4.

The experimental results were compared with three different theoretical angular correlation functions assuming a different spin assignment for the highest-energy level. Assuming that the last transition in the cascade is $2^+ \rightarrow 0^+$, three different options for the first transition were considered that is: $4^+ \rightarrow 2^+ \rightarrow 0^+$, $0^+ \rightarrow 2^+ \rightarrow 0^+$ and $2^+ \rightarrow 2^+ \rightarrow 0^+$. In the latter case the first transition is assumed to be pure dipole. The results in Fig. 4.4 suggest that the theoretical curve for the $4^+ \rightarrow 2^+ \rightarrow 0^+$ cascade fits best the experimental data. Note that the data allow a clear distinction between a stretched quadrupole ($4^+ \rightarrow 2^+ \rightarrow 0^+$ cascade) and a pure unstretched dipole ($2^+ \rightarrow 2^+ \rightarrow 0^+$ cascade). Such a distinction is often difficult with multidetector arrays, for instance the AFRODITE array [53].

In order to test how precisely the mixing ratio can be determined it was assumed that the first transition has a mixed E2+M3 multipolarity, while the second is a pure E2 transition. The data points were fit using Eq. 3.13, see Fig. 4.5.

The A_0, a_2 and a_4 coefficients were extracted from the fit and are shown in Fig. 4.5. The mixing ratio for the first transition was calculated using the code DELTA and these angular correlation coefficients. The final result was $\delta = -0.026^{+0.043}_{-0.041}$. δ was determined in the code by the χ^2 minimization. Figure 4.6 shows the calculated χ^2 versus $\arctan(\delta)$ for this transition. The value of the mixing ratio δ is in agreement with the internationally accepted value of $\delta = -0.0025 \pm 0.0022$ [52, 54], obtained on the basis of averaging the results from several experiments [55].

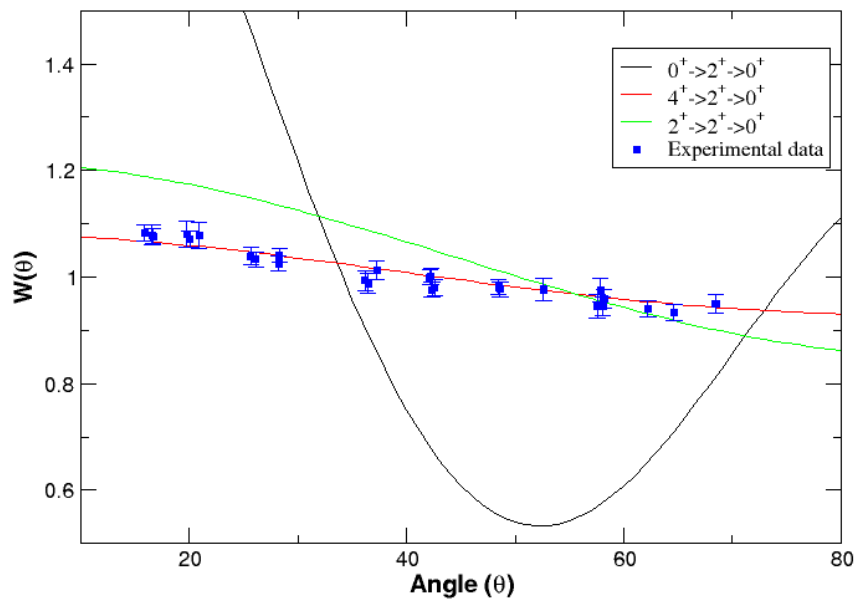


Figure 4.4: The experimental data compared with the theoretical angular correlations functions for different cascades.

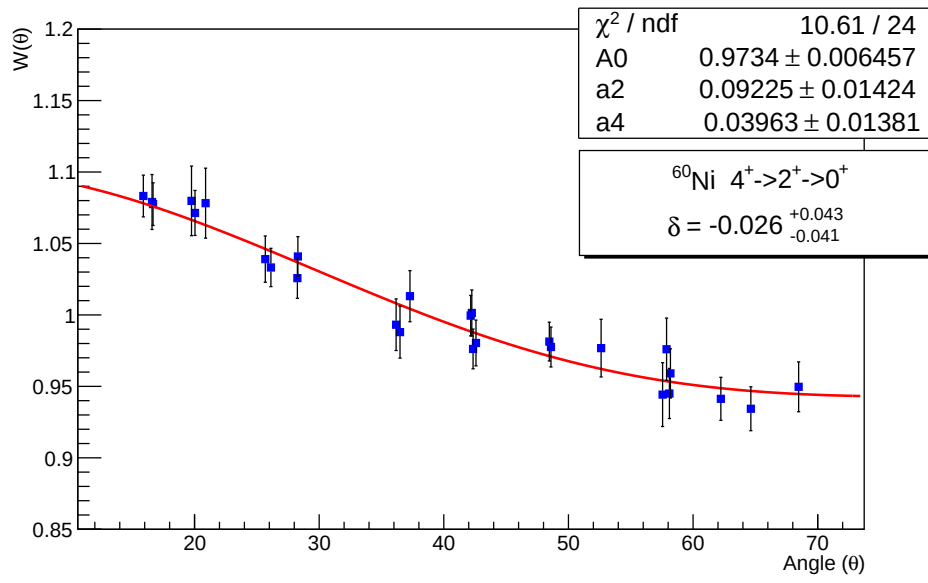


Figure 4.5: Experimentally measured $W(\theta)$ for the 1173 - 1332 keV cascade in ^{60}Co . Only the front segments were considered in the analysis. The data were fitted with Legendre polynomials (red) to extract the A_0 , a_2 and a_4 angular correlation coefficients.

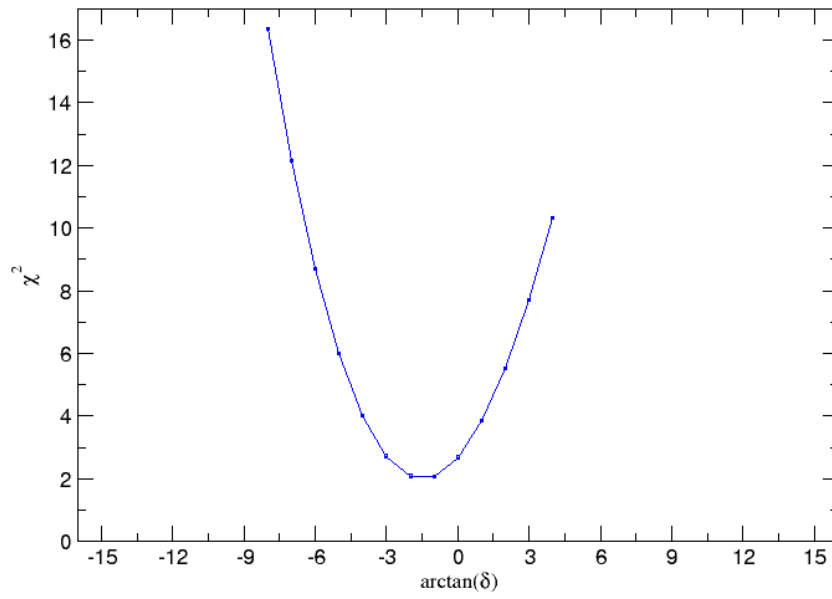


Figure 4.6: χ^2 versus $\arctan(\delta)$ for the 1173 keV transition in the ^{60}Co decay. A mixing ratio of $\delta = -0.026^{+0.043}_{-0.041}$ was deduced.

4.2 Gamma-ray cascades in ^{133}Ba

The ~ 12 hours data collected with the ^{133}Ba source were analyzed, in particular the 356 keV - 81 keV, 53 keV - 384 keV, 276 keV - 161 keV and the 303 keV - 81 keV cascades. The last two cascades are weaker and are also affected by Compton scatter peak background and made the analysis more complicated. The next sections explain in detail how the analysis for these four cascades was carried out. The level scheme of ^{133}Ba shown in Fig. 4.7, lists nine gamma-ray transitions with the following multipolarities;

- 53.2 keV $M1+E2$
- 223.2 keV $M1+E2$
- 302.9 keV $M1+E2$
- 79.6 keV $M1+E2$
- 160.6 keV $M1+E2$
- 81.0 keV $M1+E2$
- 276.4 keV $E2$
- 356.0 keV $E2$
- 383.8 keV $E2$

The aim of this work is to measure δ for the $M1 + E2$ mixed transitions. This could not be done for the weak 79.6 keV and 223.2 keV transitions. The 79.6 keV transition could not be resolved by the detector since it is very close to the 81 keV gamma ray.. The 223.2 keV is a very weak transition, which in addition is strongly affected by Compton scatter peak background, so it was not possible to extract reliable results on its mixing ratio.

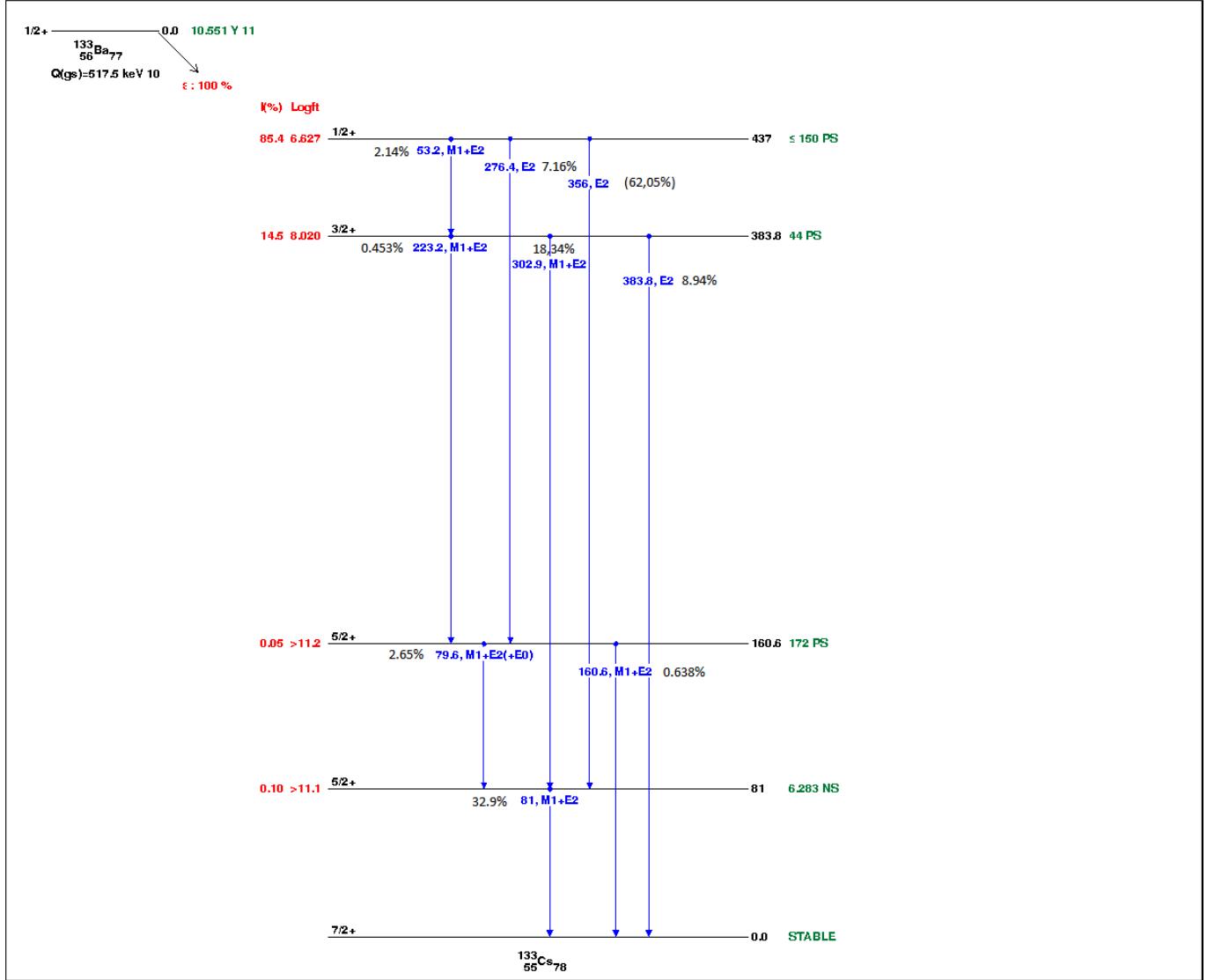


Figure 4.7: The adopted level scheme of ^{133}Cs , populated by the decay of ^{133}Ba . It shows the gamma-rays of interest and their intensities and multiplicities, from [52].

4.2.1 The 356 - 81 keV cascade

A gamma-ray transition can be in coincidence with more than one transition, therefore there could be more than one cascade that can be used for the determination of the multipole mixing ratio of one transition. It proved easier to start with the transitions which were strong since the background subtraction process was easier. The first cascade analyzed was the 356 - 81 keV cascade. The gate was set on the 356-keV transition, for example see Fig. 4.8, and a projection spectrum with the coincidence information was generated.

The background under the gate and the random coincidences background were gen-

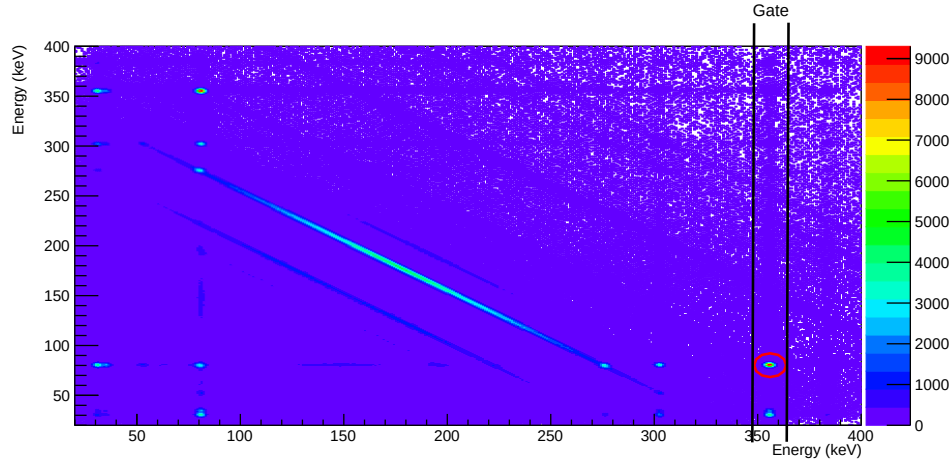


Figure 4.8: The summed angular correlation matrix for $\theta = 15^\circ$. The red circle indicates the area of interest where the coincidence between the 356 keV and the 81 keV transitions appears. The black lines indicate where the gate was set.

erated, see Fig. 4.9, and removed from the gated spectrum, as discussed in Chapter 3.

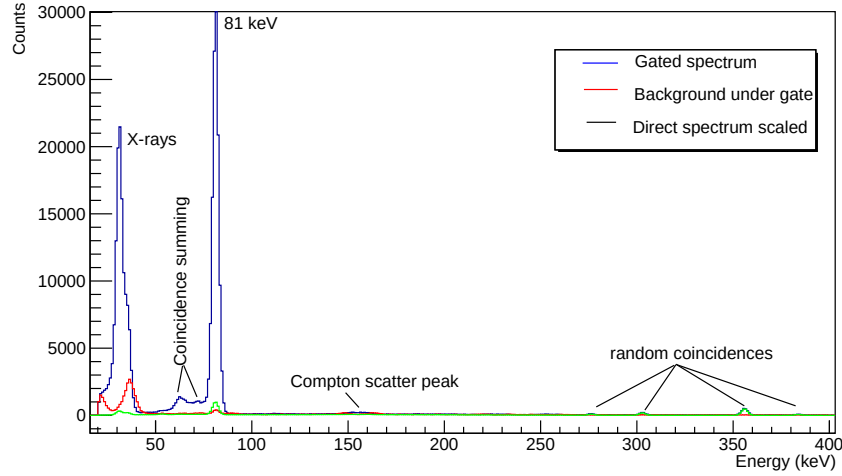


Figure 4.9: A spectrum (in navy) generated by gating on the 356 keV transition in the summed angular correlation matrix for $\theta = 15^\circ$. The red spectrum was generated by setting gates on the left and on the right regions near the gate and represents the background under the gate. The green spectrum is the normalized direct spectrum used to remove the random coincidence contribution.

The final background-subtracted gated spectrum on 356 keV is shown in Fig. 4.10 and was used to measure the area of the 81-keV coincidence peak.

One had to be careful when measuring the area of the 81-keV peak, because on the left of the peak there was a bump from coincidence summing, see Fig. 4.10. It appears due to the coincidence summing of the X-rays with energy $E_{K_\alpha} = 31$ keV and $E_{K_\beta} = 35$ keV. It was not very clear where the background under the 81 keV peak lies. Thus a number of measurements were taken to determine by how much a different background subtraction affected the area of the 81 keV peak. In these measurements the shape of the background was varied as shown in Fig. 4.11 and the effect on the measured peak area was determined.

The colored dotted lines in Fig. 4.11 illustrate how the background was varied. The

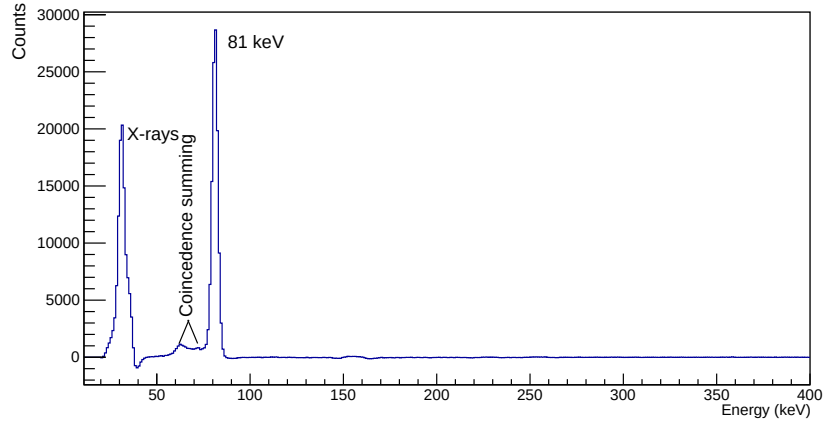


Figure 4.10: The background-subtracted gated on the 356 keV spectrum. It was produced using the summed angular correlation matrix for $\theta = 15^\circ$.

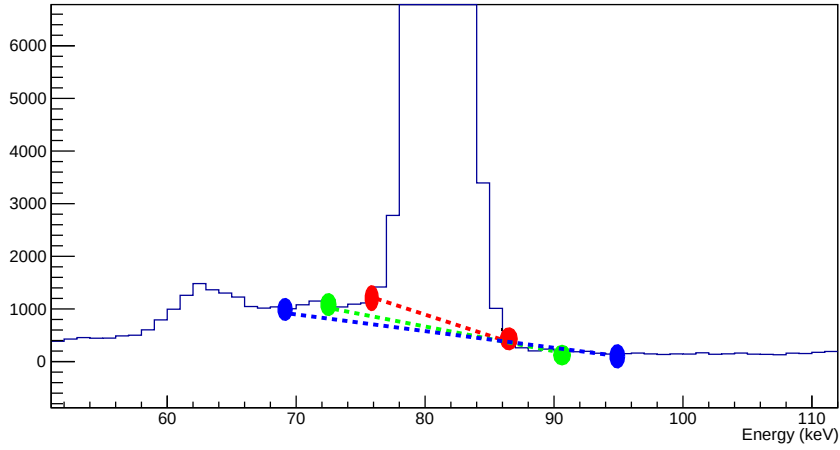


Figure 4.11: A portion of the background-subtracted spectrum gated on 356 keV (shown in Fig. 4.10). The dotted lines were used to test different backgrounds under the 81 keV peak.

area above the dotted line was considered as part of the peak. The red dotted line takes the peak area with no background from the area where the peak is situated, while for example the blue dotted line assumes a lower background. The angular correlation data were calculated with the background taken as the red line shown in Fig. 4.11. However

the uncertainties were increased to cover the slightly larger values obtained by the other measurements. The angular correlation data points are shown in Fig. 4.12.

Fig. 4.12 shows the experimental results compared with the angular correlations func-

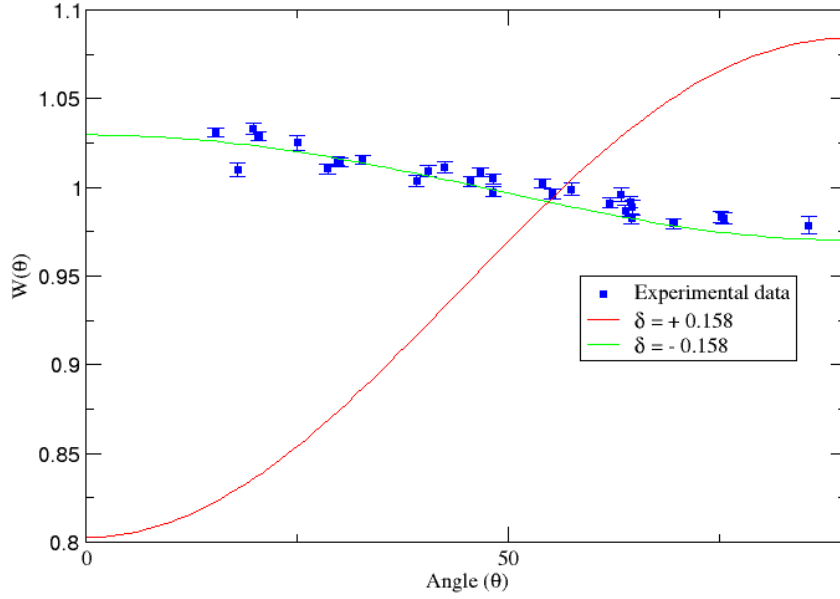


Figure 4.12: Experimental results fitted with an angular correlation function that used (i) the result of this work (green) and (ii) the accepted value in NNDC (red).

tions assuming different mixing ratios. The internationally accepted value for δ for the 81 keV transition is 0.158(5). It is based on a number of angular correlation and internal conversion measurements. However the accepted value has no sign. Thus, in Fig. 4.12 our angular correlation data is compared with the theoretical angular correlation function assuming $\delta = \pm 0.158$. The plot highlights the importance of the sign of the mixing ratio. A wrong sign of δ produces a completely different angular correlation function.

Our data were then fitted as shown in Fig. 4.13 to extract the A_0 , a_2 and a_4 angular correlation coefficients. These coefficients were later used with the code DELTA to determine δ . Figure 4.13 shows the experimental $W(\theta)$ data fitted with Eq. 3.13.

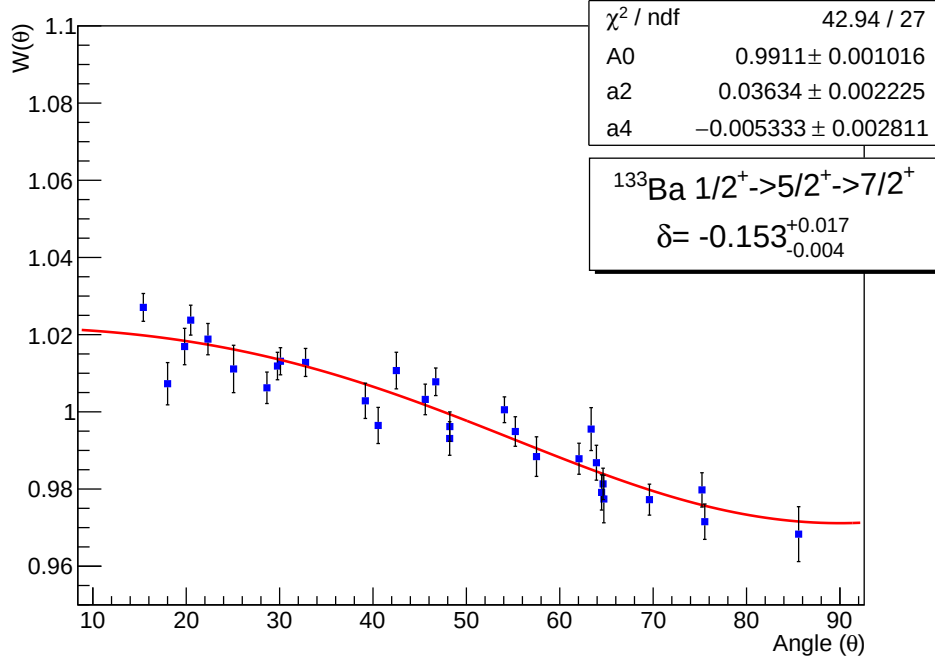


Figure 4.13: An angular correlation fit for the 356 – 81 keV cascade. Angular correlation coefficients of $a_2 = 0.036$ and $a_4 = -0.005$ were deduced.

From the fit of the experimental data, shown in Fig. 4.12, the a_2 and a_4 coefficients were extracted and used to calculate δ for the $M1 + E2$ 81-keV transition. Figure 4.14 shows the χ^2 versus $\arctan(\theta)$ curve with the $\chi^2 = 2.84$ and a value of $\delta = -0.153^{+0.017}_{-0.004}$. The calculated value of δ in this work agrees with the accepted value of $\delta = 0.158(5)$ from [56]. Moreover the phase (sign) of the multipole mixing ratio was determined to be negative.

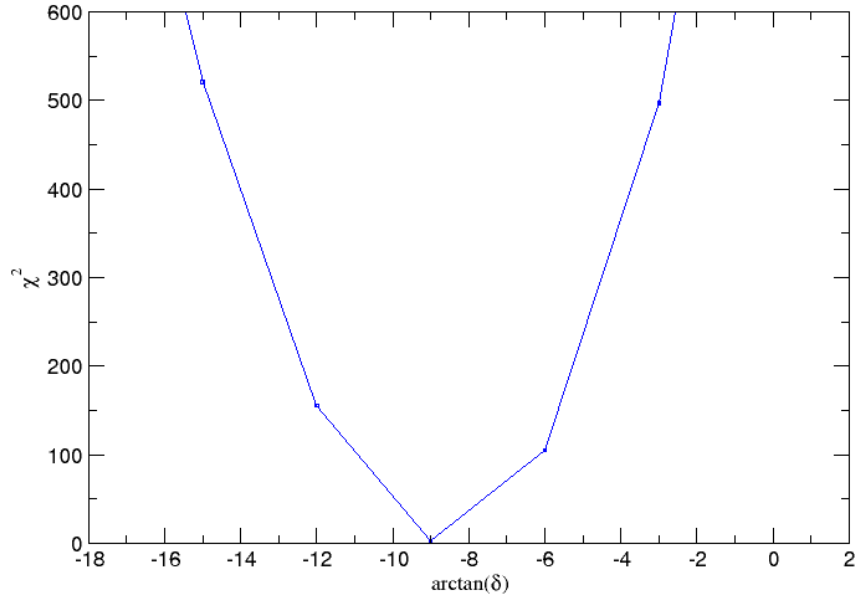


Figure 4.14: The χ^2 versus $\arctan(\theta)$ plot for the experimental angular correlation coefficients of the $\frac{1}{2}^+ \rightarrow \frac{5}{2}^+ \rightarrow \frac{7}{2}^+$ cascade with a range of mixing ratios. The deduced value of δ is $\delta = -0.153^{+0.017}_{-0.004}$.

4.2.2 The 53 - 384 keV cascade

Generally, the gate would be set on the first transition of the cascade and the intensity of the second transition in the cascade would be measured. For the 53 - 384 keV cascade this analysis proved to be difficult particularly at small angles θ . For instance, Figs. 4.15 and 4.16 show the projections spectrum (on the x -axis) of the angular correlation matrices for $\theta = 75^\circ$ and $\theta = 16^\circ$, respectively.

The 53 keV lies on a large background and has low counts. Thus gating on it introduced much background.

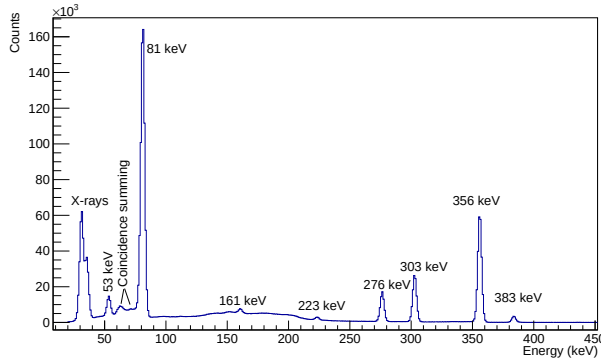


Figure 4.15: A projection on the x -axis of the summed angular correlation matrix for angle $\theta = 75^\circ$.

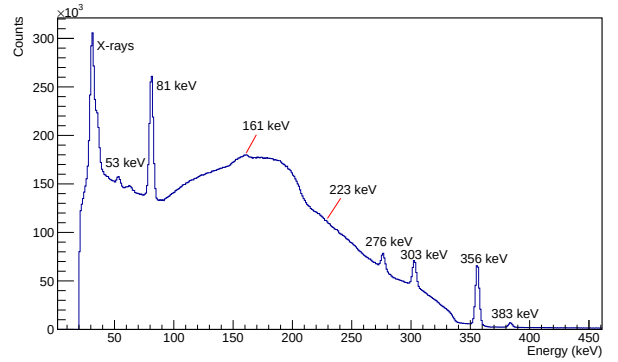


Figure 4.16: A projection on the x -axis of the summed angular correlation matrix for angle $\theta = 16^\circ$.

However, the same region of interest (see the red circle in Fig. 4.17), can be analyzed in a different way, by setting a gate on the 384 keV transition along the y -axis which is then projected on the x -axis. The analysis code was adjusted to change the axis of projection. Note that the efficiency remains unchanged, as the same combination of detectors is used. For background subtractions the axes were also changed accordingly. Note that the 384 keV peak is a pronounced peak in a region with low background.

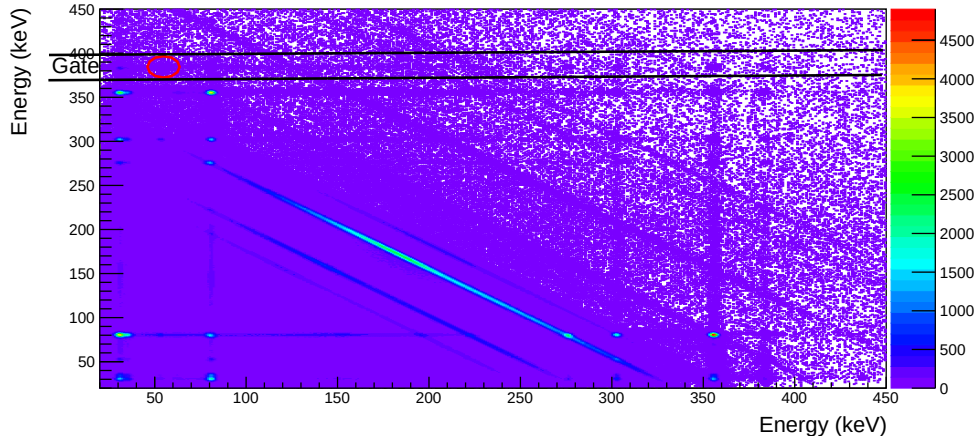


Figure 4.17: The summed correlation matrix for angle $\theta = 16^\circ$. The red circle indicates the 53 - 384 keV coincidence peak. The black lines indicate the 384 keV gate on the y -axis.

The spectrum gated on 384 keV, the background under the gate and the random coincidences spectra are shown in Fig. 4.18. The random coincidence events did not affect the area of the 53 keV peak.

The area of the 53 keV coincidence peak was then measured in the final background-subtracted gated spectrum, shown in Fig. 4.19. The peak was fitted with a Gaussian and a first-order polynomial. The area was used to calculate the angular correlation intensity. The analysis was carried out for all 32 angles and the results for the angular correlation intensity, $W(\theta)$, are shown in Fig. 4.18.

The accepted value of δ for the 53 keV transition is $\delta = 0.08^{+0.02}_{-0.03}$ [56] without a sign. Figure 4.20 shows the experimental angular correlation data compared to the theoretical angular correlation functions for $\delta = \pm 0.08$.

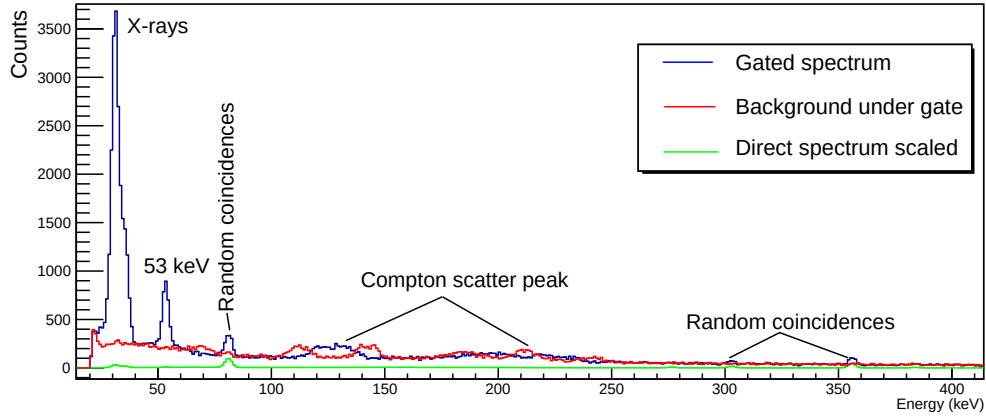


Figure 4.18: A spectrum (in navy) generated by gating on the 384-keV transition in the summed angular correlation matrix for angle $\theta = 18^\circ$. The red spectrum was generated by setting gates on the left and on the right regions near the gate and represents the background under the gate. The green spectrum is the normalized direct spectrum used to remove the random coincidence contribution.

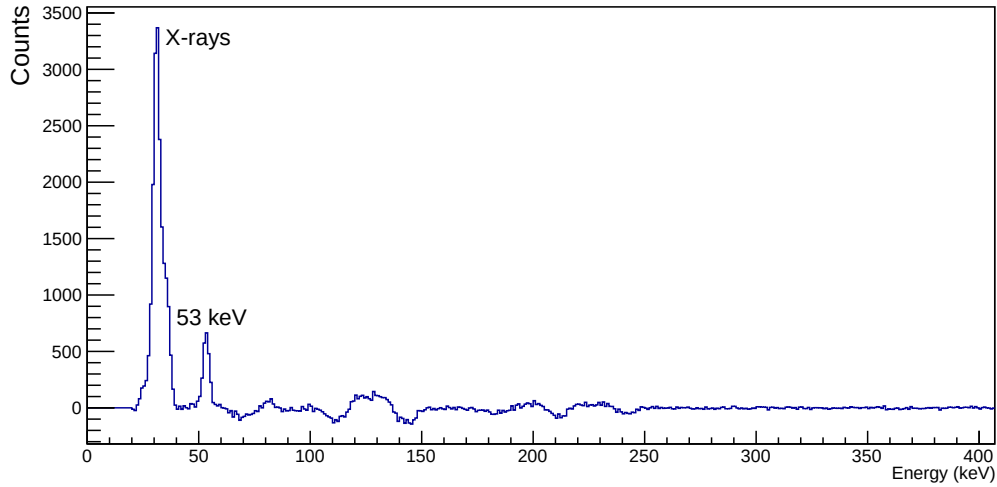


Figure 4.19: The background-subtracted spectrum gated on 384 keV, from the summed angular correlation matrix for angle $\theta = 16^\circ$.

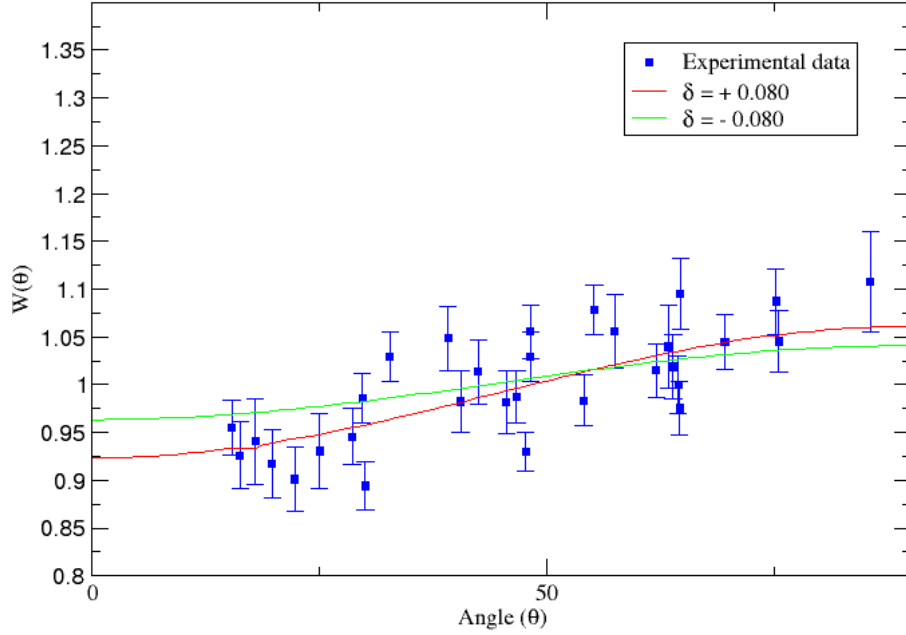


Figure 4.20: Experimental results were compared with the angular correlation functions that use $\delta = \pm 0.08$ for the 53-keV transition.

The experimental data for the 53 - 384 keV cascade were fitted with Legendre polynomials, as shown in Fig. 4.21, and the a_2 and a_4 angular correlation coefficients were extracted.

The experimental angular correlation coefficients a_2 and a_4 were used to calculate the value of the multipole mixing ratio. Figure 4.22 shows the χ^2 versus $\arctan(\theta)$ curve with $\chi^2 = 0.025$ suggesting a value of $\delta = +0.079$.

The internationally accepted multipole mixing ratio for the mixed $M1 + E2$ 53 keV transition is $\delta = 0.08^{+0.02}_{-0.03}$ [56]. In the present work $\delta = +0.079^{+0.074}_{-0.065}$ was obtained for the

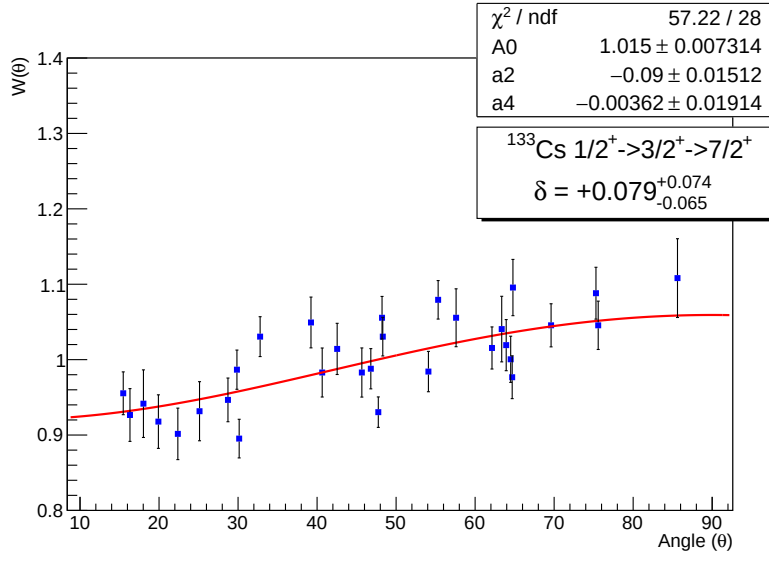


Figure 4.21: The angular correlation fit for the 53 - 384 keV cascade. Angular correlation coefficients of $a_2 = -0.090$ and $a_4 = -0.004$ were deduced.

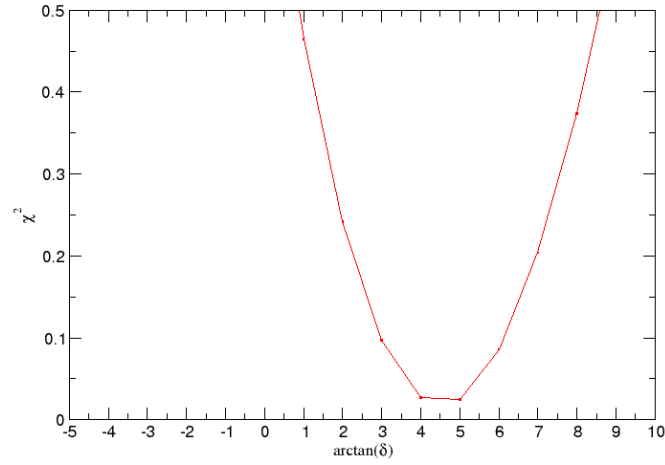


Figure 4.22: The χ^2 versus $\arctan(\theta)$ plot for the experimental angular correlation coefficients of the $\frac{1}{2}^+ \rightarrow \frac{3}{2}^+ \rightarrow \frac{7}{2}^+$ cascade. A value of $\delta = +0.079^{+0.074}_{-0.065}$ is extracted for the 53-keV transition.

mixing ratio of 53-keV transition. The phase of the mixing ratio for this transition was determined to be positive, which is a new result.

4.2.3 The 303 - 81 keV cascade

For the 303 - 81 keV cascade the gate was set on the 303 keV peak, as shown in Fig. 4.23. The 81 keV peak of interest lies on top of a Compton-scatter peak, produced by the

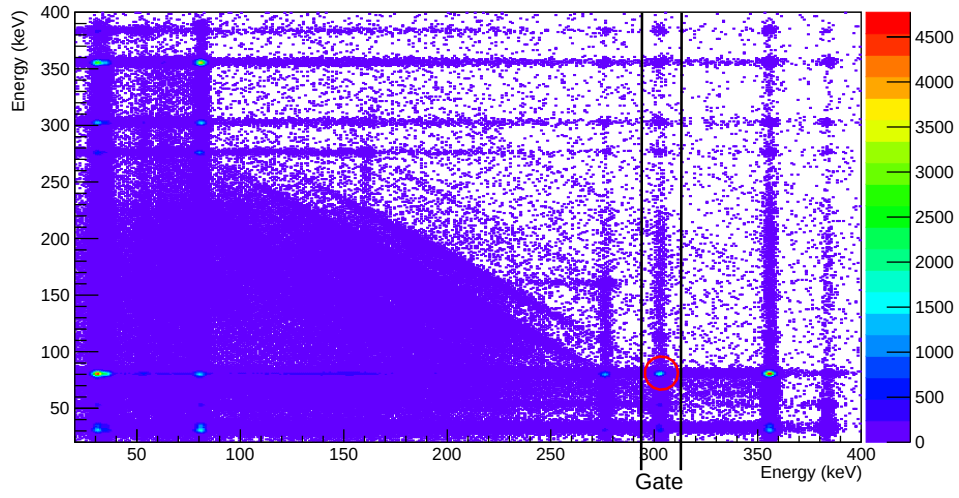


Figure 4.23: The summed angular correlation matrix for angle $\theta = 18^\circ$. The 303 keV gate is indicated by black lines and the region of interest for the 303 - 81 keV cascade is shown by a red circle.

scattering of the 384 keV transition, see Fig. 4.24. A large Compton scatter peak appears also under the 53 keV peak, produced by the Compton scattering of the 356 keV transition.

To visualize the Compton scatter peaks for small angles the gate on 303 keV was made wide, as the width of the Compton scatter bump matches that of the gate. The wide gate allows to visualize the Compton scatter peak background, see Fig. 4.25. In this case no

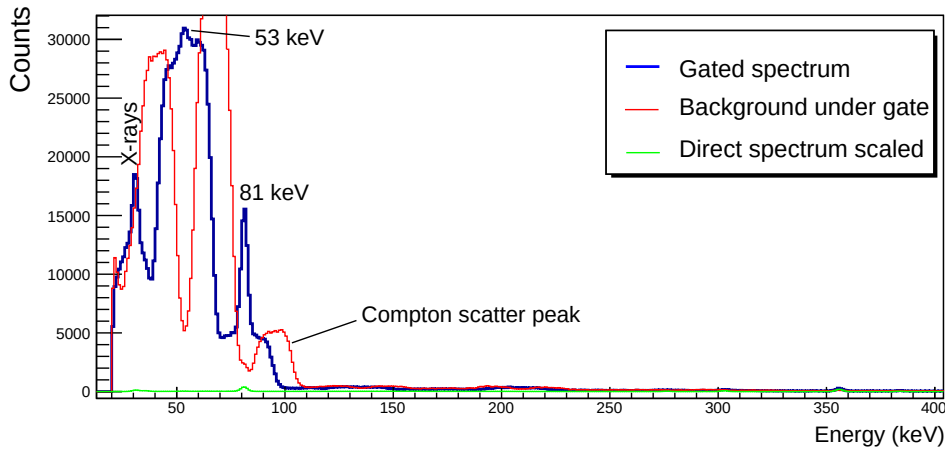


Figure 4.24: As Fig. 4.18, but the gate is set on 303 keV. The coincidence peaks appear on top of a Compton-scatter background.

correction for background under the peak was made as described in Section 3.6.2. The problem is that gating on the background region on the left and on the right of the peak produces a background spectrum where the Compton-scatter bump is shifted in energy. Subtracting such a spectrum from the gated 303 keV spectrum creates negative regions.

An attempt to shift the bumps in the background spectra was made but it was not as successful. Thus, the gated spectrum was not corrected for background under the gate. This type of background is not expected to affect considerably the area of the 81 keV peak, as there is no 81-keV peak in the background spectrum shown in Fig. 4.24 in red. The background due to random coincidences was corrected as described in Section 3.6.3 and finally, the background-corrected gated 303-keV spectrum is shown in Fig. 4.25.

For small angle θ the 81-keV peak was fitted and the area was calculated from the fit parameters. The uncertainty on the area, calculated from the fit was doubled to account for the fact that the spectrum was not corrected for background under the gate. The angular correlation values were then calculated using the peak areas and the corresponding

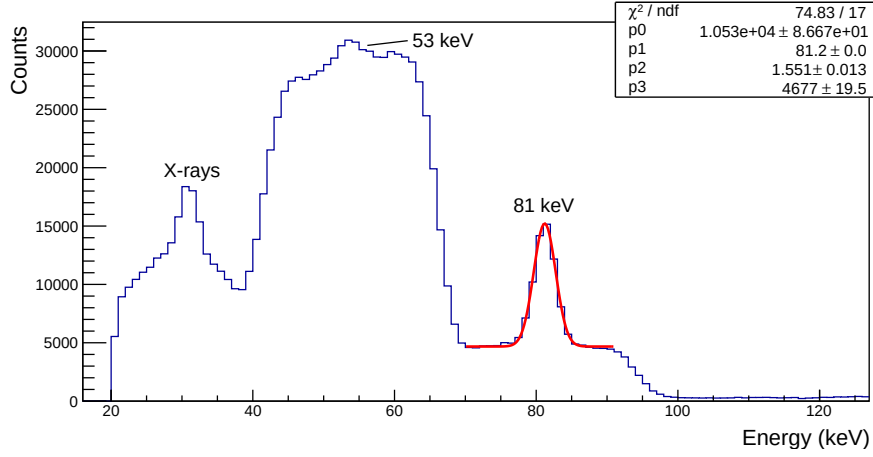


Figure 4.25: A portion of the 303 keV gated spectrum (corrected for randoms), with the 81 keV coincidence peak fitted with a Gaussian and a polynomial. The fit parameters are shown in the right corner of the figure, where $p0$ is the height of the peak, $p1$ is the centroid of the peak, $p2$ is the Gaussian parameter σ , and $p3$ is the height of the background above zero.

efficiencies.

The internationally accepted value for the multipole ratio of the 303 keV transition is $\delta = 0.022 \pm 0.020$ [56], but it has no sign. Figure 4.26 compares our experimental results for angular correlation with the calculated $W(\theta)$ assuming ± 0.022 for the 303 keV transition. In the calculations a value of $\delta = -0.158$ was used for the 81 keV transition. As $|\delta|$ is small, the sign of δ affects the angular correlation function to a lesser extent.

The results for the 303 – 81 keV cascade are shown on Fig. 4.27. The data were fitted with Legendre polynomials and the a_2 and a_4 angular correlation coefficients were extracted. The experimental angular correlation coefficients a_2 and a_4 were used to calculate the value of the multipole mixing ratio for the 303 keV gamma ray. Figure 4.28

shows a χ^2 versus $\arctan(\theta)$ curve which yields a value of $\delta = +0.068 \pm 0.036$. The internationally accepted value of δ for the 303 keV transition is 0.022(20). It compares well within uncertainties with the value measured in this work. Note that the phase of the mixing ratio is a new result.

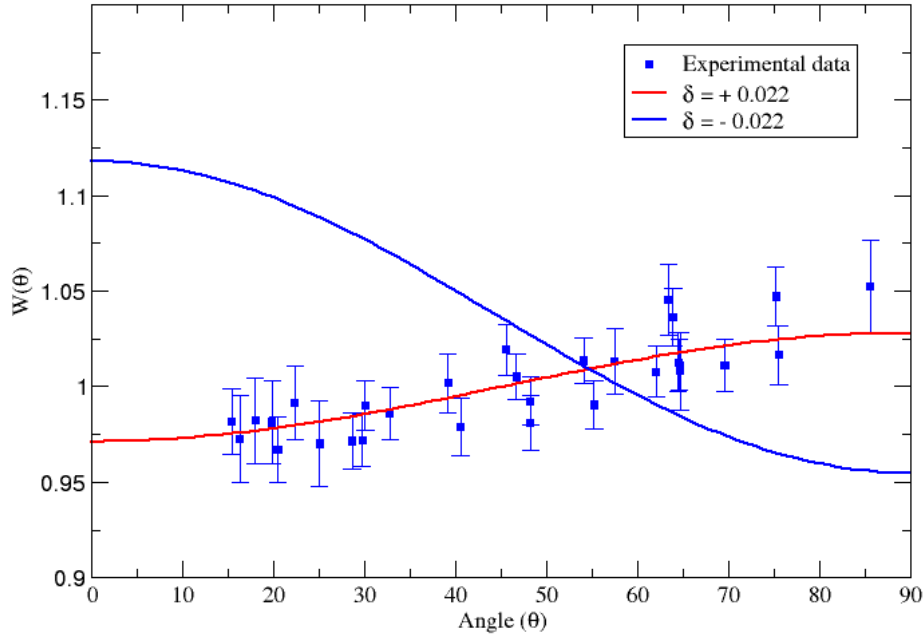


Figure 4.26: Experimental angular correlation data for the 303 - 81 keV cascade compared with the theoretical angular correlation functions assuming $\delta = \pm 0.022$.

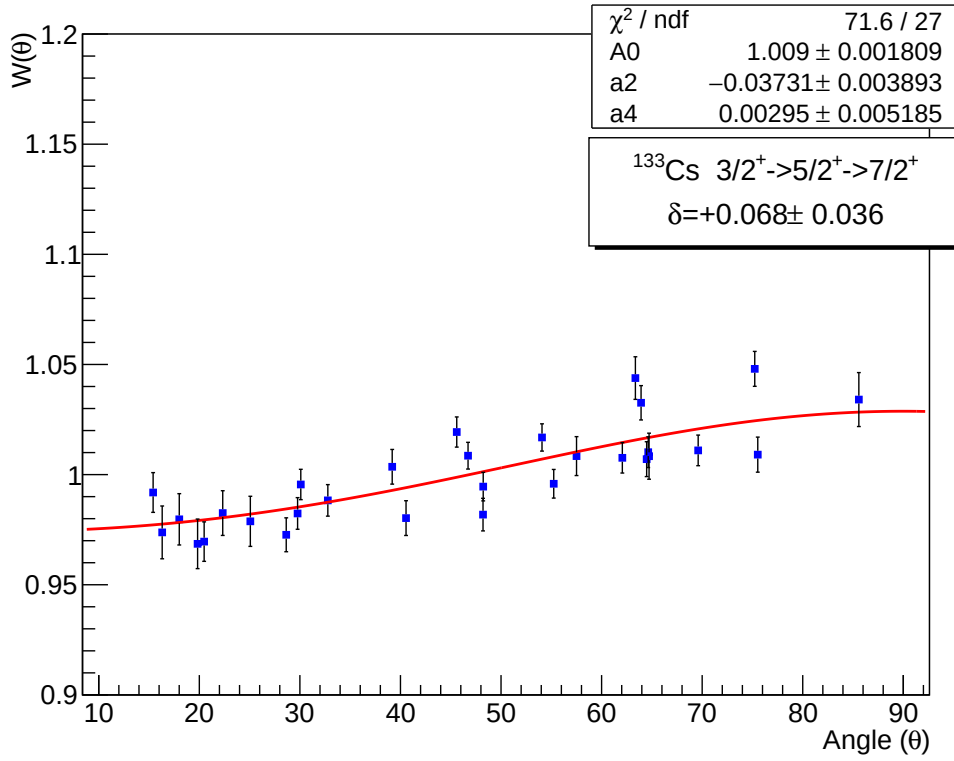


Figure 4.27: The experimental angular correlation data for the 303 - 81 keV cascade is fitted to extract the angular correlation coefficients. Angular correlation coefficients of $a_2 = -0.037$ and $a_4 = 0.003$ were deduced.

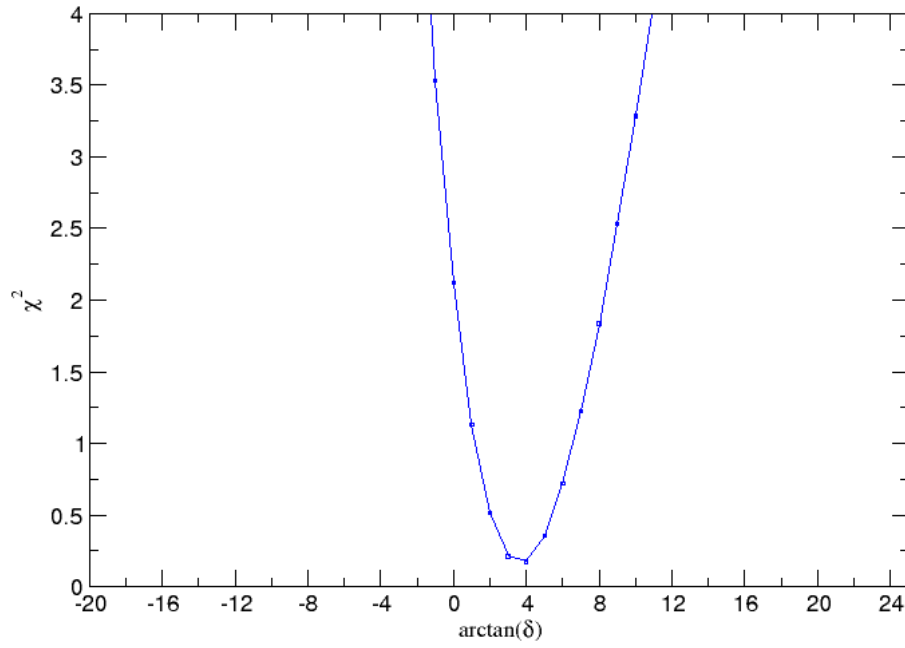


Figure 4.28: The χ^2 versus $\arctan(\theta)$ plot for the experimental angular correlation coefficients of the $\frac{3}{2}^+ \rightarrow \frac{5}{2}^+ \rightarrow \frac{7}{2}^+$ cascade with a range of mixing ratios. The deduced value of δ is $+0.068 \pm 0.036$.

4.2.4 The 276 - 161 keV cascade

A wide gate was set on the 276-keV gamma ray, (see Fig. 4.29), to visualize better the Compton scatter background.

For large angles θ the 161 keV peak was not affected by the Compton-scatter peak

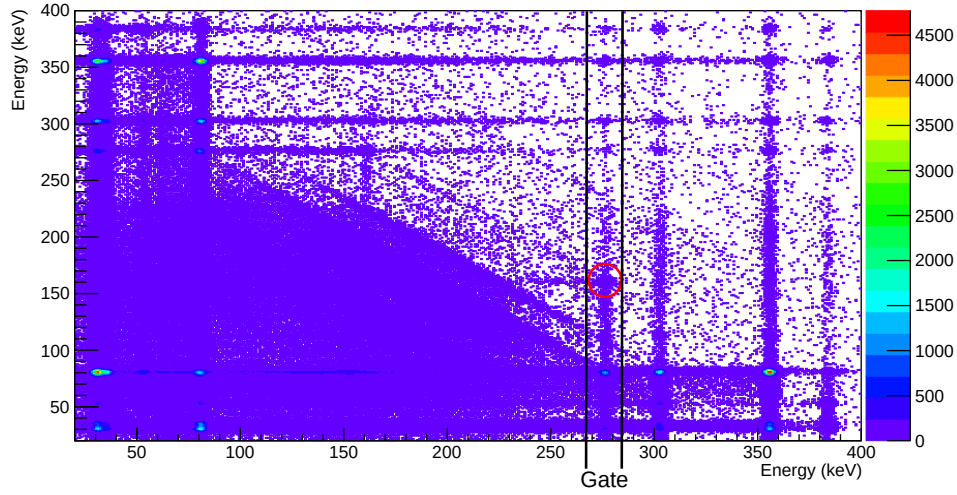


Figure 4.29: The summed angular correlation matrix for angle $\theta = 18^\circ$, with the 276 keV gate indicated by black lines and the region of interest for the 276 – 161 keV cascade is shown by a red circle.

background. Thus the gated spectra were corrected for both background under the peak and background due to the random coincidences. Then the area of the 161-keV peak was measured. However, for small angles θ the 161-keV and 81-keV peaks lie on top of considerable Compton-scatter peaks, see Fig. 4.30 and 4.31.

In such cases the correction for background under the gate could not be done because the Compton scatter peaks were shifting in on the background gated spectra, introducing

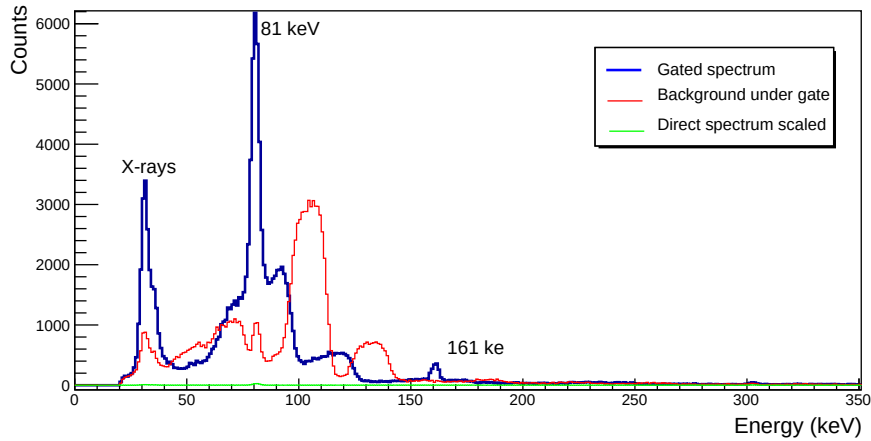


Figure 4.30: Spectra generated by gating on 276 keV in the summed angular correlation matrix for $\theta = 18^\circ$. The 53 keV, 81 keV and 161 keV peaks are lying on a Compton-scatter background. The scaled direct spectrum is shown in green and the background-gated spectrum is shown in red. The Compton-scatter peaks shift in the red spectrum.

negative regions in the background-gated spectrum. For small angles the gated spectrum was corrected for random coincidences only, as described in Section 3.6.3. For all angles the 161-keV peak was fitted using a Gaussian and a first order polynomial, as shown in Fig. 4.31. The area and the uncertainties on the area of these peaks were calculated from the fit parameters and for small angles the uncertainties were doubled in order to account for the fact that the background under the gate was not corrected for.

The internationally accepted value for the multipole ratio of the 161 keV transition is $\delta = 0.96(5)$ [56], but it has no sign. Figure 4.32 shows the experimental angular correlation data compared with the theoretical angular correlation functions $W(\theta)$ assuming $\delta = \pm 0.96$. This figure also emphasizes the importance of the sign of the mixing ratio. An opposite sign of δ produces a very different angular correlation function.

The experimental results for the 276 - 161 keV cascade are shown on Fig. 4.33. The data

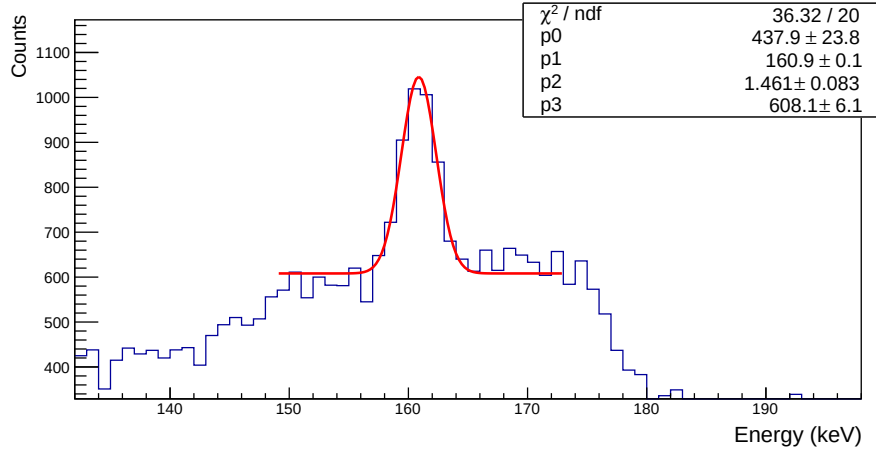


Figure 4.31: A portion of the 276 keV gated spectrum after background subtraction. The 161 keV peak is fitted with a Gaussian and a polynomial. The parameters are shown in the right corner of the figure, where $p0$ is the height of the peak, $p1$ is the centroid of the peak, $p2$ is the Gaussian parameter σ , and $p3$ is the height of the background.

were fitted with Legendre polynomials and the a_2 and a_4 angular correlation coefficients were extracted. The experimental angular correlation coefficients a_2 and a_4 were used to calculate the value of the multipole mixing ratio for the 161 keV gamma ray.

The experimental angular correlation coefficients a_2 and a_4 were used to calculate the value δ . Figure 4.34 shows a χ^2 versus $\arctan(\theta)$ fit which yields a value of $\delta = +1.165^{+0.983}_{-0.504}$, in agreement with the accepted value in the NNDC. However, note, that the phase of the mixing ratio is a new result.

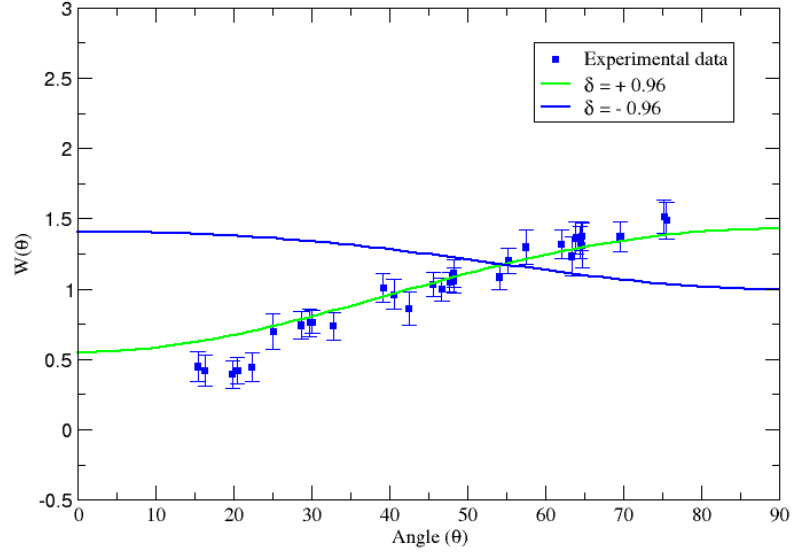


Figure 4.32: Experimental angular correlation data for 276 - 161 keV cascade compared with the theoretical angular correlation functions assuming $\delta = \pm 0.96$.

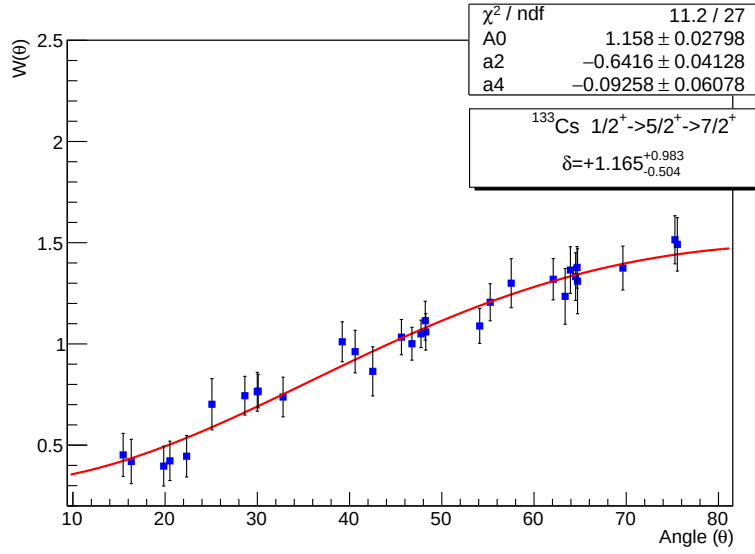


Figure 4.33: Angular correlation fit for the 276 - 161 keV cascade. Angular correlation coefficients of $a_2 = -0.641$ and $a_4 = -0.092$ were deduced.

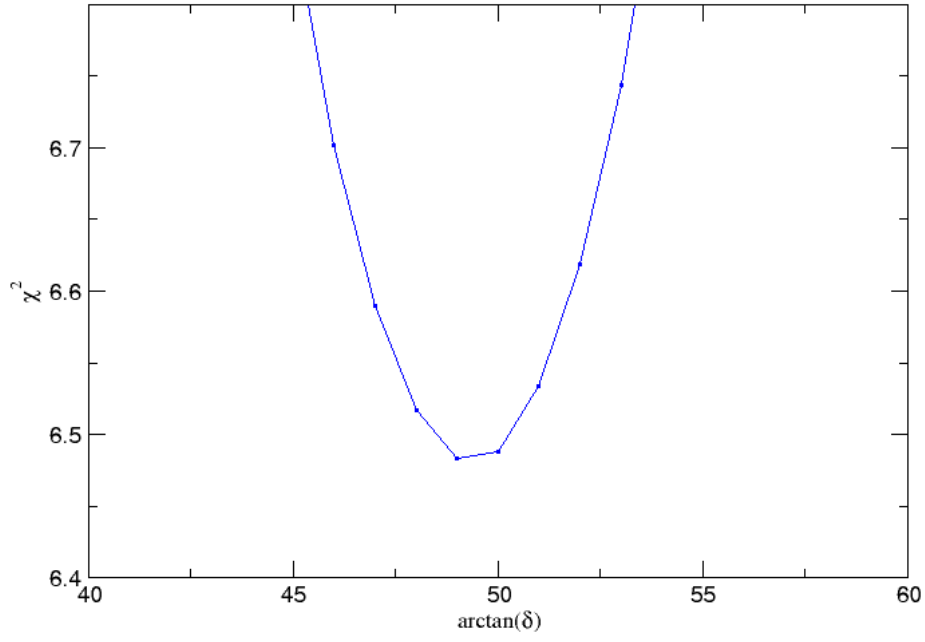


Figure 4.34: The χ^2 versus $\arctan(\theta)$ plot for the experimental angular correlation coefficients of the $\frac{1}{2}^+ \rightarrow \frac{5}{2}^+ \rightarrow \frac{7}{2}^+$ cascade with a range of mixing ratios. The deduced value of the mixing ratio δ for the 161 keV transition is $+1.165^{+0.983}_{-0.504}$.

4.2.5 Summary of the angular correlation results for ^{133}Ba

A summary of the results compared to the previously measured multipole mixing ratios is given in Table 4.1. The measured multipole mixing ratios δ are in good agreement with the previous measurements. Note that the signs of the mixing ratio were also measured and these are new results.

Table 4.1: Summarized results and accepted multipole mixing ratios from the NNDC [56].

γ	Spins	δ (Nuclear Data Sheets)	δ (Experiment) from this work
53.2 keV	$1/2^+ \rightarrow 3/2^+$	$0.08^{+0.02}_{-0.03}$	$+0.079^{+0.074}_{-0.065}$
79.6 keV	$5/2^+ \rightarrow 5/2^+$	0.124(15)	—
81 keV	$5/2^+ \rightarrow 7/2^+$	0.158(5)	$-0.153^{+0.017}_{-0.004}$
161 keV	$5/2^+ \rightarrow 7/2^+$	0.96(5)	$+1.165^{+0.983}_{-0.504}$
223.2 keV	$3/2^+ \rightarrow 5/2^+$	$-0.114(14)$	—
302.9 keV	$3/2^+ \rightarrow 5/2^+$	0.022(20)	$+0.068 \pm 0.036$

With respect to the precision our results are very good. In some cases the uncertainty of δ is similar to the uncertainty of the internationally accepted values, e.g. the lower limit for δ for the 81 keV transition. In most cases the uncertainty is a factor of 2 or 3 larger, e.g. the uncertainties of δ for the 53 and 303 keV transitions. This is very encouraging, taking into account that: (i) our results were recorded over a short period of ~ 12 hours, (ii) not all experimental data were analyzed, but the data for the front segments only, (iii) the international accepted values are deduced after averaging over the results of several experiments, including both angular correlation and internal conversion experiments. We can improve further the precision of our analysis, by taking into account the back segments, by evaluating the attenuation coefficients due to the limiting opening angle of the segments, by averaging data for similar angles to increase the statistics etc. Such an analysis is planned for the near future. In general our work suggests that the segmented

clover has excellent capacity for angular correlation measurements.

During the analysis a Compton scattering from what seems to be a 437-keV gamma ray was observed in the gamma-gamma angular correlation matrices. ^{133}Ba is not known to have such a transition. Therefore, an analysis to determine what caused this Compton scattering was done, and the details of this are discussed in the next chapter.

Chapter 5

Searching for a possible new 437 keV transition in ^{133}Ba

During the angular correlation analysis, a Compton scattering ridge corresponding to $E = 437$ keV was observed in the gamma-gamma angular correlation matrices. Generally, Compton scattering is identified by a diagonal line (continuous ridge) in the matrix which corresponds to a constant sum of two energies detected on the two detectors represented on the x and y -axis, e.g. $E = E_x + E_y = \text{constant}$. The two energies correspond to the deposited energy of the scattered gamma ray in the two detectors. Figure 5.1 shows a summed gamma-gamma angular correlation matrix which corresponds to angle $\theta = 18^\circ$. As a 437-keV ridge, appears in this matrix it is important to understand how is it formed.

If there was a 437 keV gamma ray emitted by the ^{133}Ba source, observing a Compton ridge formed by 437 keV would be normal because all gamma rays with high enough energy to scatter form Compton ridges. However such a gamma ray is not known. Therefore we had to investigate how the 437-keV ridge was formed. First we checked for a 437 keV peak in the singles data. It was indeed observed in the direct spectra of the front segments, see for instance Fig. 5.2.

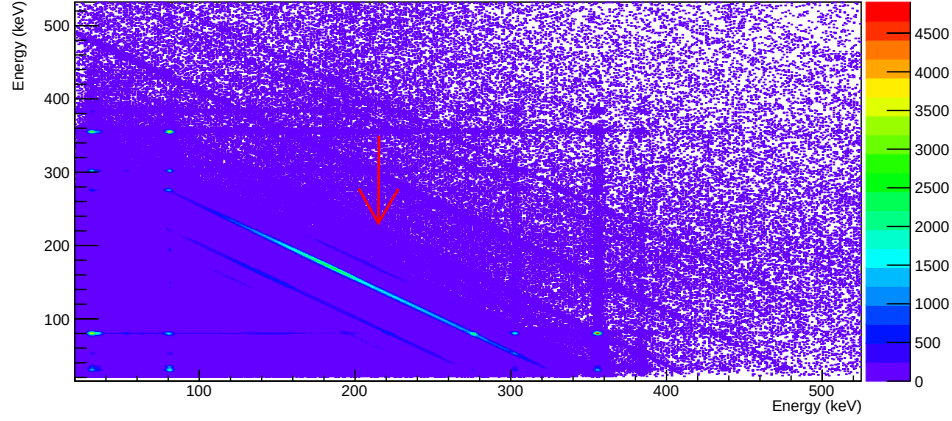


Figure 5.1: The gamma-gamma angular correlation matrix for angle $\theta = 18^\circ$. The red arrow indicates the 437 keV Compton scattering ridge.

However, the observation of a 437-keV peak in this spectrum is inconclusive because, it might represent a new transition, or it could be a result of coincidence summing. If the 437 keV peak is produced by coincidence summing, the gamma rays that can sum up to 437 keV from ^{133}Ba are:

$$437 \text{ keV} = 160.6 \text{ keV} + 276.4 \text{ keV}$$

$$437 \text{ keV} = 53.2 \text{ keV} + 383.8 \text{ keV}$$

$$437 \text{ keV} = 81 \text{ keV} + 356 \text{ keV}$$

This requires that both gamma rays that add up to 437 keV are detected in the same segment. Therefore examining the direct spectra in the back segments can provide useful information, because observing both gamma rays in a back segment is highly unlikely because one of the transitions in each case has energy $E < 161 \text{ keV}$ and can not be detected by the back segments as it does not penetrate that deep. Therefore, a 437-keV peak can not be produced in back segments by coincidence summing.

The back segment spectra showed a very small peak, see for instance Fig. 5.3. Note

that the direct spectra of back segments 5, 6, and 8 were used only, as segment 7 is placed on the inner side, next to the hole between the crystals, thus gamma rays may reach segment 7 through the whole without passing through a thick layer of germanium material.

This suggests that there could be a new 437 keV, $\frac{1}{2}^+ \rightarrow \frac{7}{2}^+$ transition in the ^{133}Ba

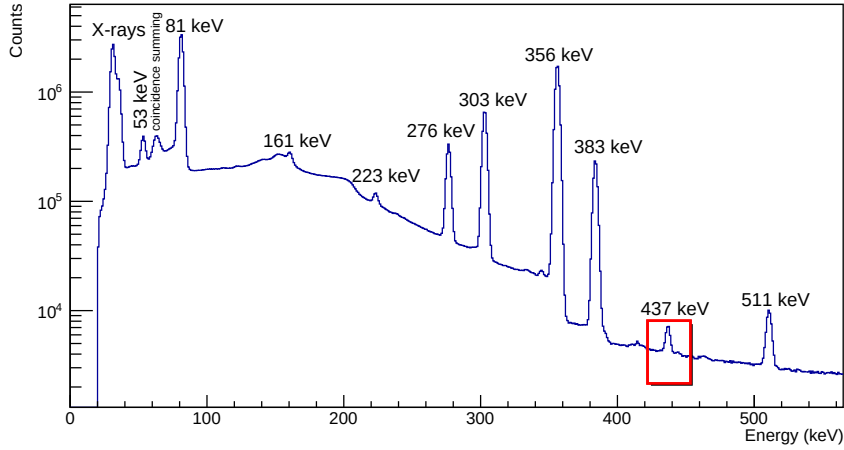


Figure 5.2: The gamma spectrum for segment 1 of crystal B. The box in red indicates the 437-keV peak. This spectrum can not distinguish between coincidence summing of multiple gamma-rays and a single gamma-ray transition.

gamma decay. This motivated a further analysis to investigate if this is indeed a new gamma transition. We investigated what is creating the ridge in the angular correlation matrices, for the front segments. Could for instance some kind of Compton scattering produce a 437-keV ridge? In order to investigate which gamma rays scatter, a cut at $x = y$ on the angular correlation matrices was performed, see for instance Fig. 5.4.

Basically a cut is taking a portion of the angular correlation matrix and projecting it onto one of the axes, similar to gating. In this case the $(x = y)$ gate cuts through the Compton ridges and generates a projected spectrum with peaks that reflect the Compton scatter

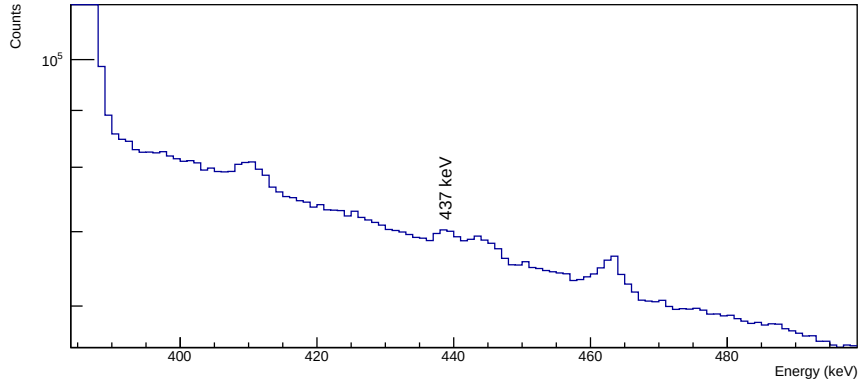


Figure 5.3: Summed spectrum from the direct spectra of the back segments 5, 6, and 8 of crystals A, B, C and D.

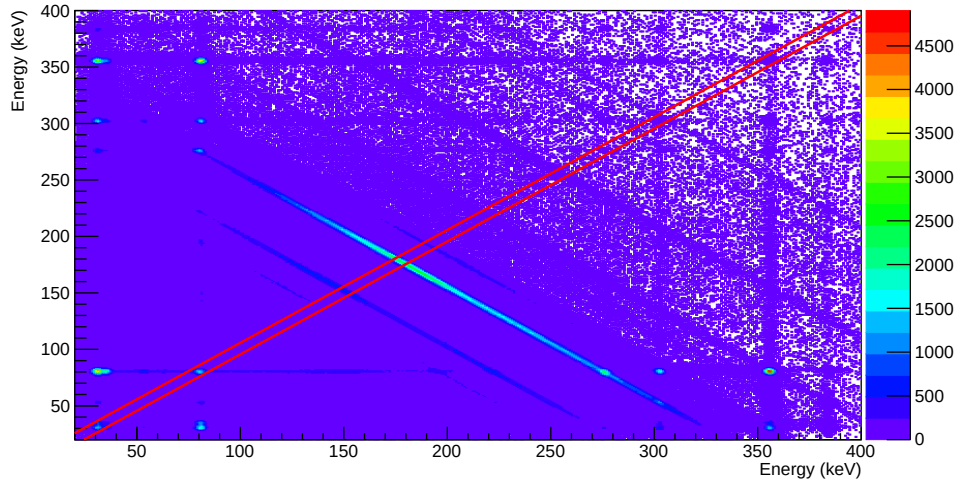


Figure 5.4: A cut along $x = y$ angular correlation matrix for $\theta = 18^\circ$. The projection of this cut on segment 1 is shown in Fig. 5.5 as the navy histogram.

ridges, see Fig. 5.5. It is important to note that since the cut is at $x = y$ it means that the gamma ray deposits half of its energy on each segment. So doubling the energy of the peaks in Fig. 5.5 gives the correct energy of the gamma ray that scattered and produced the ridge.

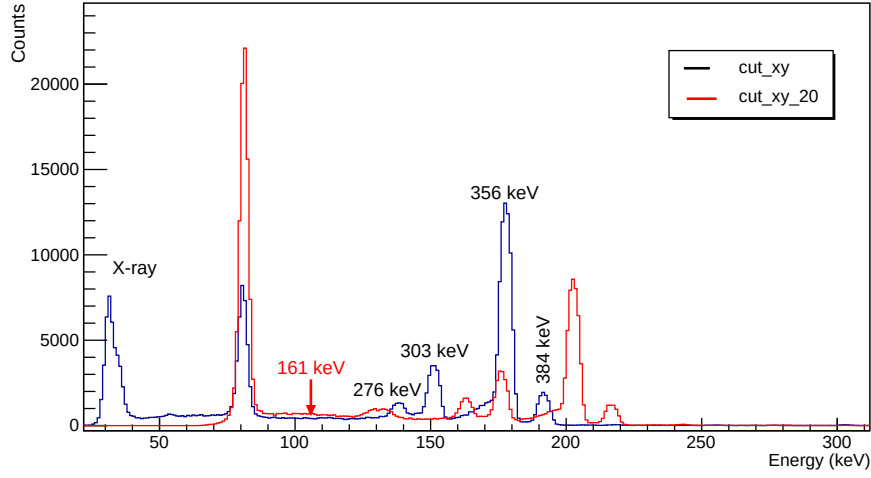


Figure 5.5: A projection spectra produced by a cut through $x = y$ (navy) of the angular correlation matrix for $\theta = 18^\circ$ and by a cut that is shifted by 20 keV, see Fig. 5.6 (red). The peaks represent Compton-scatter ridges, except the X-rays and the 81 keV peak. The gamma rays that correspond to each ridge are labeled with their energies.

The navy histogram in Fig. 5.5 corresponds to the cut at $x = y$. To determine the energy of the gamma rays that scattered the peak energies are doubled and written on top of the peaks in Fig. 5.5. There is a peak at 81 keV which could be the scattering of the 161 keV gamma ray. On the other hand there is strong coincidence between the 81 keV and 80 keV gamma rays. The coincidence peak of these two transitions lies exactly on the continuous Compton scatter ridge, as seen in Fig. 5.4.

To test this further another cut was made with a shift of 20 keV in order to avoid the 81 - 80 keV coincidence peak, see Fig. 5.6. All the peaks that correspond to Compton ridges shift (see red spectrum in Fig. 5.5). If there was a 161-keV ridge its peak would have also shifted and would have appeared where the red arrow points in Fig. 5.5. Instead the 81 keV peak remains unshifted. Therefore, the spectrum shows no scatter from the

161-keV gamma ray, which confirms that only gamma rays with energy $E > 161$ keV in ^{133}Ba scatter. The ridges generated by the Compton scatter of the 276 keV, 303 keV and 384 keV gamma rays are identified as peaks in the navy spectrum.

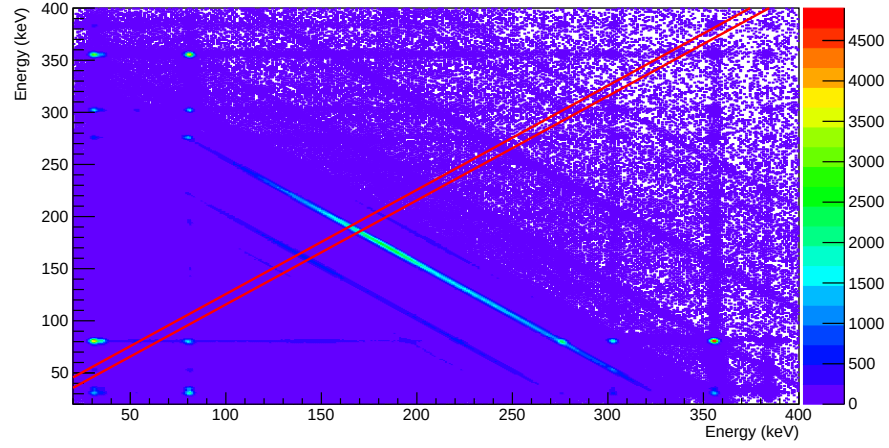


Figure 5.6: A cut along $x = y$ shifted by 20 keV. The projection of this cut on segment 1 is shown in Fig. 5.5 in red.

Therefore the 437 keV Compton ridge cannot be produced by Compton scattering of both gamma rays in the cascades of ^{133}Ba . There is one more option, that is, the observed 437 ridge could be produced by Compton scattering of one of the gamma rays while the other one is fully absorbed. For instance, in the case of $437 \text{ keV} = 53 \text{ keV} + 384 \text{ keV}$, it is possible that the 53-keV gamma ray is fully absorbed while the 384-keV gamma ray scatters. If the 53 keV transition is absorbed in segment 1 and the 384-keV gamma ray deposits part of its energy in segment 1 and then scatters into segment 2 it will contribute to the Compton ridge. In addition, such events will look somewhat similar to a Compton scatter of a 437-keV gamma ray. An alternative way to produce the 437-keV ridge is if the 53-keV gamma ray is absorbed in segment 2 and the 384-keV gamma ray scatters from segment 1 to segment 2. Both these possibilities are illustrated in Fig. 5.7.

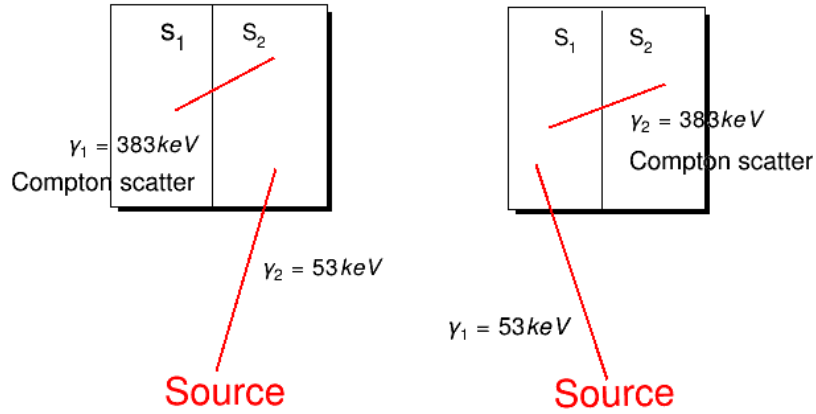


Figure 5.7: Illustration of the Compton scattering of the 384-keV gamma ray in coincidence with the 53-keV transition, contributing to a 437-keV Compton ridge.

The interaction shown on the left in Fig. 5.7, can be described if a cut along the 384-keV Compton ridge is made (see such cuts illustrated in Fig. 5.9) and projected on segment 1, see the red spectrum in Fig. 5.8. It represents the Compton scattering ridge of the 384-keV transition. The peaks in the spectrum correspond to the coincidence of the 81-keV and 303-keV gamma rays with total energy of 384 keV. The continuum, however, corresponds to the 384-keV Compton. The interaction shown on the right in Fig. 5.7 can be described by the same cut if the spectrum is shifted by 53 keV to higher energies. This shift reflects the fact that 53 keV is always absorbed in segment 1. The sum yields spectra shown in Fig. 5.8 describes the Compton ridge at 437 keV produced by the 53 - 384 keV cascade.

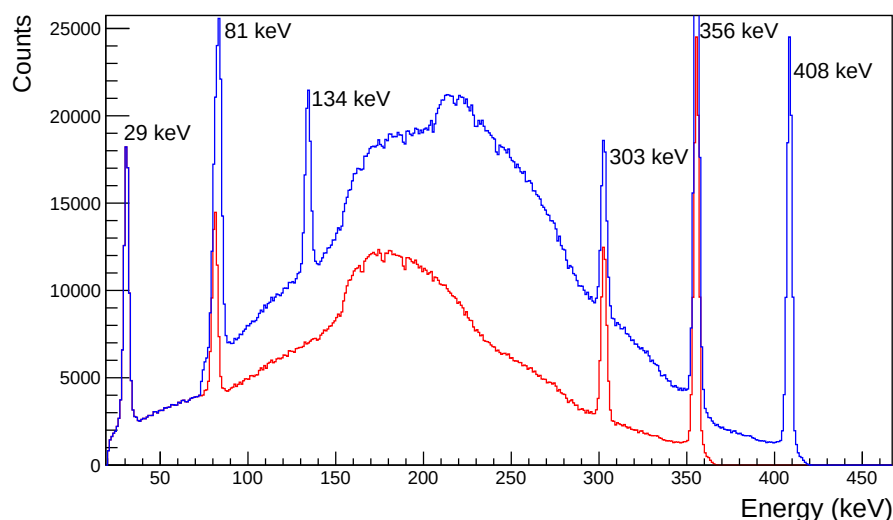


Figure 5.8: Projection of the Compton scattering ridge of the 384 keV gamma-ray (red) in the summed angular correlation matrix for angle $\theta = 18^\circ$. The peak at 28 keV is a Compton-scatter peak, $384 \text{ keV} = 356 \text{ keV} + 28 \text{ keV}$. This spectrum represents events in the 437-keV ridge, produced as illustrated in Fig. 5.7 left. The navy spectrum represents all Compton scatter events in the 437-keV ridge, produced by the 53 - 384 cascade as illustrated in Fig. 5.7 left and right. For more details see text.

If a cut is taken along the 437-keV ridge the red spectrum in Fig. 5.10 is produced. The

peaks correspond to true coincidences between two different gamma rays. For example, the 53-keV transition is in coincidence with the 384-keV transition and the 81-keV gamma ray is in coincidence with the 356-keV gamma ray and the 161-keV transition is in coincidence with the 276-keV transition. Each coincidence pair sums up to 437 keV. Thus the peaks can be explained by such coincidences. It is interesting to find out what exactly is producing the continuous background in the spectrum. The continuum could be created by a new 437-keV gamma-ray that scatters in two segments. One alternative is that the continuum is created by a Compton scattering of the 276, 356 and 384-keV gamma-rays in coincidence with the 161, 81 and 53-keV transitions, respectively, of the type described in Figs. 5.7 and 5.8. Cuts along the ridges that correspond to 276, 356 and 384-keV were created (see Fig. 5.9) and projected. Spectra as shown in Fig. 5.8 were produced for each of the three cascades. They were summed and the total spectrum is shown (in navy in Fig 5.10).

The question is can such events account for the shape of the 437 keV Compton scatter spectrum shown in Fig. 5.10. The red spectrum created as a sum of Compton scatter events matches reasonably well the 437-keV spectrum. Therefore there is no evidence for a new 437-keV transition although, the navy spectrum does not exactly match the shape of the 437 keV Compton scatter (red). As the agreement is reasonable, it seems that if there is a new 437-keV transition it must be very weak. One can further investigate this, using GEANT4 simulations for the geometry of the iThemba LABS segmented clover detector, and test the shape of the simulated 437-keV ridge in comparison with the observations, as well as work with larger statistics and investigate data from the back segments too.

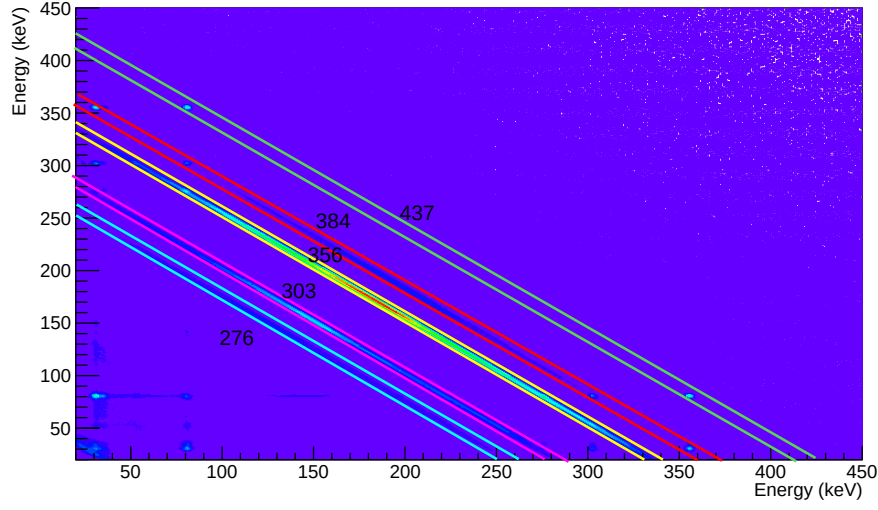


Figure 5.9: Cuts along the 276 keV, 356 keV, 384 keV and 437 keV Compton ridges. The cuts are represented in different colors and labeled with the corresponding gamma-ray energies.

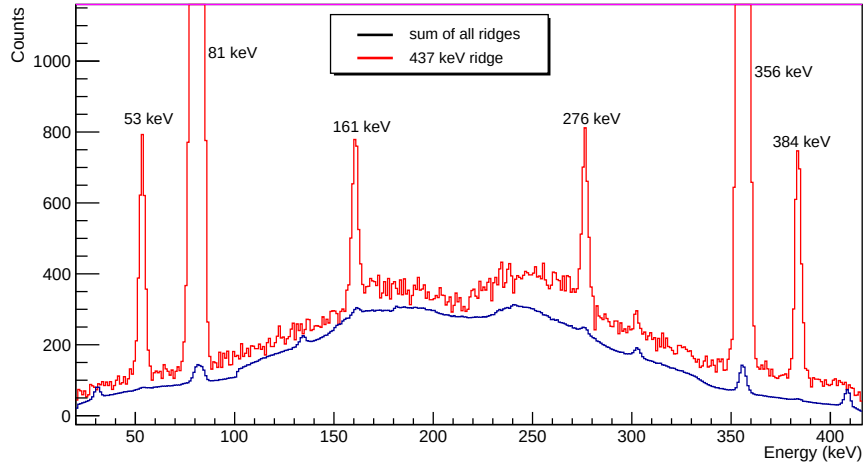


Figure 5.10: The projection of the cut along the 437-keV ridge in red. The spectrum in navy represents the Compton-scatter events from the 53 - 384 keV, 356 - 81 keV, and 276 - 161 keV cascades that contribute to the 437-keV ridge. The contributions from each cascade was deduced as described in the text, and as shown in Fig. 5.8 (navy) for the 53 - 384 keV cascade.

Chapter 6

Conclusions

Gamma-gamma angular correlation measurements were performed in order to develop a methodology for angular correlation measurements and to establish the precision for mixing ratio studies with the iThemba LABS segmented clover detector. The first measurement performed was with a ^{60}Co source that β decays to ^{60}Ni . Data was collected for ~ 12 hours. About a quarter of the data, that is the data for the front-front segment combination was used. We were able to easily distinguish between a stretched dipole and unstretched quadrupole transition in the decay of ^{60}Co . In order to find out with what precision the multipole mixing ratios can be determined, we assumed that the 1173 keV transition in the $4^+ \rightarrow 2^+ \rightarrow 0^+$ (1173–1332) cascade is a mixed transition and measured $\delta = -0.026^{+0.043}_{-0.041}$. This result is in agreement with the accepted value of δ in the NNDC [54, 55], and the error bars of the measurement are a factor of ~ 20 larger than the error bars of the accepted value from previously performed experiments.

The second measurement was with a ^{133}Ba source that β decays to ^{133}Cs . The data was also collected for ~ 12 hours using only the front-front segment combination. Mixing ratios for four transitions in the decay of the ^{133}Ba source were measured successfully. The measured δ values were compared with accepted the values from NNDC [52] and are in good agreement. Our error bars are larger than the error bars on the accepted values

for δ by a factor ranging from 0.8 to 3.7 for three of the transitions while for the fourth transition a larger error bar by a factor of 10 - 20 larger was obtained. From the work done it can be concluded that spin assignments for excited nuclear states are possible and using angular correlation measurements the angular correlation coefficients and multipole mixing ratios can be measured with good precision, for instance for the 81 keV transition in ^{133}Ba the mixing ratio was determined with a precision of 0.004.

It is very encouraging that with just 12 hours of data we were able to measure multipole mixing ratios with such good precision. It is expected that with an improvement in the analysis method, with larger data collection and including back segments too, even better precision can be achieved. It is important to note that despite having a small data set new results on the signs of the multipole mixing ratios were determined. Note that measurements have also been done using the Internal Conversion Coefficients (ICC) method [57, 58]. However the method developed in this work seems to be much better, as it allows both good precision, and information on the sign of δ . In conclusion, the developed technique for angular correlation measurements with the iThemba LABS segmented clover detector will allow the assignment of unknown spins and parities, the determination of angular correlation coefficients and multipole mixing ratios in future experiments with very good precision.

Further improvements in this work include: **(i)** using GEANT4 simulations to determine the coordinates of the gamma-ray interactions in the detector volume in order to determine the angular correlation angles with high precision, **(ii)** an improvement in the analysis method to also include data from the back segments, **(iii)** establishing a method to better determine the area and its error bar in the measurements for transitions affected by Compton scatter peaks, **(iv)** determining the attenuation factor, taking into account the solid angle of the segments and the geometry of the detector. Due to the complexity

of the geometry of the iThemba LABS segmented clover detector and to time constraints the geometric factor was not calculated, however such future work is being planned.

Appendix A

Code written by T.D. Bucher

```
#include "Riostream.h"
#include "Byteswap.h"
#include "TTree.h"
#include "TH2.h"
#include "TH1.h"
#include "TF1.h"
#include "TChain.h"
#include <iostream>
#include <fstream>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include <stddef.h>
#include "Rtypes.h"
#include "TROOT.h"
#include "TSystem.h"
#include <iomanip>
#include "TFile.h"
#include "TBrowser.h"
#include "TCanvas.h"
#include <sstream>

using namespace std;
using std::cout;
using std::endl;

class SimSort {
public:
    SimSort();
    ~SimSort();
    Double_t timeBuf[256][2048], timeBuf2[2048], timeins, pow16_8, pow16_4;
    Int_t empful[256], worful[2][256], wordins[256], wins, words3;
    UShort_t ggmatt[4096][4096];
    Int_t ggsize, g, gg, tableADC[4][16384];
    Double_t etot[32], cz[32];
    Double_t deddet[32], dedet[32], det[32];
    Int_t matrix;
    Double_t ETOT[32];
    ofstream datagg;
    ofstream info, info1, info2;
    ifstream datafs[128]; // attention: pb if there are more than 128 runs in an experiment
    ifstream dataws[128]; // attention: pb if there are more than 128 runs in an experiment
    UShort_t wordBuf[256][8192], wordPtr3[16384], wordPtr4[32768];
    UShort_t *wordPtr, *ep, wordPtr2[16384];
    UInt_t myPtr[4][2048];
    ULong_t myPtr2[4][2048];
    ULong_t myPtr3[4][16384];
    UInt_t myPtr4[4][16384];
    Double_t E1, E2, Energy[10], Offsetarray[50][50], Gainarray[50][50], OffsetarrayLC[50][50], GainarrayLC[50][50];
```

```

UShort_t Crystal[4],crystalbuf[10], DetID[10];
Int_t words,ADC_events,eof,blocks,check,check2,stepblock,stepblock2,ult;
Int_t endian, row, Offset, Colm;
Short_t eventenergy;
ULong_t EVTIME0,EVTIME1, EVTIME2,EVTIME3,EVTIMEA,timest[32],timeref,TimeStmp[10],jj;
UShort_t CFTime,tracelength;
ULong_t TrailEsum,ESleading,GapEsum,Baseline;
UShort_t headerlgth,crateid,slotid,channelno,header;
Int_t ReadData(Char_t *dataf, Int_t blkSize);
Int_t GainMatch(Int_t=0, Double_t=0, Int_t=0);
//Int_t ConvertDataPixie16();
Int_t Tree(Char_t * = NULL);
Int_t fConvert;
Int_t SetOffset(const char * = NULL);
Int_t SetGain(const char * = NULL);
Int_t GainDrift(const char * = NULL);
Int_t OffsetDrift(const char * = NULL);
//Int_t SetGainc(const char * = NULL);
//Int_t SetOffsetc(const char * = NULL);
Int_t GainDriftCorr(Int_t=0, Double_t=0, Int_t=0,Int_t=0);
Int_t fGain;
Int_t fOffset;
//Int_t fGainDrift;
//Int_t fOffsetc;
//Double_t fAdcGainc[100];
Double_t fAdcOffset[100];
//Double_t fAdcOffsetc[100];
Double_t fAdcGain[100],addback[100],fGainDriftCorr;

```

```
private:
```

```

    Int_t fWrite;
    Int_t fRead;
    Int_t j, t;
};
//-----//
//***** open and reads the gain file *****/
//*****

Int_t SimSort::SetGain(const char *gainfName)
{
    Int_t rowA=0;
    Int_t colA=0;
    string lineA;
    Double_t x;
    // char buf[36][36];
    GainarrayLC[50][50]=0;
    Double_t fileIN;
    //string filename;
    ifstream fileINf;
    // cin >> filename;
    fileINf.open(gainfName);
    cout<<"read number of columns"<<endl;
    if (fileINf.fail()){cerr<< "file you try to open is closed";
    exit(1);
    }
    cout<<"\n"<<endl;
    if (fileINf.is_open()){
        while(getline(fileINf,lineA)){
            istream streamA(lineA); // use buf for input
            colA=0;
            while (streamA >> x){
                GainarrayLC[rowA][colA]=x;
                colA++;
            }
            rowA++;
        }
    }
}

```

```

}

cout << " #number of rows= " << rowA << endl;
cout << " #number of columns= " << colA << endl;
cout << " "<<endl;
/*for (int i=0; i<rowA; i++){
for (int j=0; j<colA; j++){
cout<< setw(6)<<GainarrayLC[i][j]<< " ";
}
cout<<endl;
}*/

//cout<<"value= "<<Gainarray[3][0]<<endl;
return 0;

return fGain;
}

//-----//
//***** Reads the offset *****//

Int_t SimSort::SetOffset(const char *offsetfName)
{
Int_t rowA=0;
Int_t colA=0;
string lineA;
Double_t x;
// char buf[36][36];
OffsetarrayLC[50][50]=0;
Double_t fileIN;
//string filename;
ifstream fileINf;
// cin >> filename;
fileINf.open(offsetfName);
cout<<"read number of columns"<<endl;
if (fileINf.fail()){cerr<< "file you try to open is closed";
exit(1);
}
cout<<"\n"<<endl;
if (fileINf.is_open()){
while(getline(fileINf,lineA)){
istringstream streamA(lineA); // use buf for input
colA=0;
while (streamA >> x){
OffsetarrayLC[rowA][colA]=x;
colA++;
}
rowA++;
}
}

cout << " #number of rows= " << rowA << endl;
cout << " #number of columns= " << colA << endl;
cout << " "<<endl;
/*for (int i=0; i<rowA; i++){
for (int j=0; j<colA; j++){
cout<< setw(6)<<OffsetarrayLC[i][j]<< " ";
}
cout<<endl;
}*/

//cout<<"value= "<<Offsetarray[3][0]<<endl;
return 0;

return fOffset;
}

//-----//

```

```

////////////////////////////////////
/***** Reads the GainCoef *****/
////////////////////////////////////
Int_t SimSort::GainDrift(const char *xtalkfName)
{
    Int_t rowA=0;
    Int_t colA=0;
    string lineA;
    Double_t x;
    // char buf[36][36];
    Gainarray[50][50]=0;
    Double_t fileIN;
    //string filename;
    ifstream fileINf;
    // cin >> filename;
    fileINf.open(xtalkfName);
    cout<<"read number of columns"<<endl;
    if (fileINf.fail()){cerr<< "file you try to open is closed";
    exit(1);
    }
    cout<<"\n"<<endl;
    if (fileINf.is_open()){
    while(getline(fileINf,lineA)){
    istringstream streamA(lineA); // use buf for input
    colA=0;
    while (streamA >> x){
    Gainarray[rowA][colA]=x;
    colA++;
    }
    rowA++;
    }
    }

    cout <<" #number of rows= " << rowA << endl;
    cout <<" #number of columns= " << colA << endl;
    cout << " "<<endl;
    /*for (int i=0; i<rowA; i++){
    for (int j=0; j<colA; j++){
    cout<< setw(6)<<Gainarray[i][j]<< " ";
    }
    cout<<endl;
    }*/
    cout<<"value= "<<Gainarray[0][0]<<endl;
    cout<<"value= "<<Gainarray[1][0]<<endl;
    cout<<"value= "<<Gainarray[2][0]<<endl;
    return 0;
    }

//-----//
////////////////////////////////////
/***** Reads the GainCoef *****/
////////////////////////////////////
Int_t SimSort::OffsetDrift(const char *xtalkfName)
{
    Int_t rowA=0;
    Int_t colA=0;
    string lineA;
    Double_t x;
    // char buf[36][36];
    Offsetarray[50][50]=0;
    Double_t fileIN;
    //string filename;
    ifstream fileINf;
    // cin >> filename;
    fileINf.open(xtalkfName);
    cout<<"read number of columns"<<endl;
    if (fileINf.fail()){cerr<< "file you try to open is closed";
    exit(1);
    }
    cout<<"\n"<<endl;

```

```

if (fileINf.is_open()){
while(getline(fileINf,lineA)){
istringstream streamA(lineA); // use buf for input
colA=0;
while (streamA >> x){
Offsetarray[rowA][colA]=x;
colA++;
}
rowA++;
}
}

cout << " #number of rows= " << rowA << endl;
cout << " #number of columns= " << colA << endl;
cout << " "<<endl;
/*for (int i=0; i<rowA; i++){
for (int j=0; j<colA; j++){
cout<< setw(6)<<Offsetarray[i][j]<< " ";
}
cout<<endl;
}*/
cout<<"value= "<<Offsetarray[0][0]<<endl;
cout<<"value= "<<Offsetarray[1][0]<<endl;
cout<<"value= "<<Offsetarray[2][0]<<endl;
return 0;
}

//-----//
//-----//
//***** Initialise all the tables (calibration...) *****//
//-----//

SimSort::SimSort()
{}

SimSort::~SimSort()
{}

//-----//
//-----//
//***** Read data *****//
//-----//

Int_t SimSort::ReadData(Char_t *dataf, Int_t blkSize)
{
    struct addressWord {
        UShort_t data;
        UShort_t group;
        UShort_t item;
        UShort_t type;
    };
    typedef struct addressWord Addr_t;
    struct contentWord {
        UShort_t data;
    };
    typedef struct contentWord Cont_t;
    // const Int_t blkSize = 7488; // in bytes
    const Int_t nWdChar = blkSize / sizeof(Char_t);
    const Int_t nWdInt = blkSize / sizeof(UShort_t);
    Int_t pbyte;
    Char_t bytes[blkSize];
    Char_t *bytesPtr = bytes;
    info1.open("Readdata.txt",ios::app);
    fRead=1;
    {

        if (! datafs[0].is_open()) {
            datafs[0].open(dataf,ifstream::binary);
            if (datafs[0].is_open()) info1 << "ReadData: Reading raw data file \"" << dataf << "\"..."<< endl;
            else info1 << "ReadData: Couldn't find raw data file \"" << dataf << "\"..."<< endl;
        }
        if(datafs[0].eof() == 0) {

```


125

```

    if (Seg==7){if ((ii < (runsize*(row+1))) && (ii>1792251252)){ Energy[Seg] = ((GainarrayLC[row][Colm]*(EEnergy))+OffsetarrayLC[row][Colm]);}}
if(row==35){if (Seg==8){if ((ii > (runsize*row)) && (ii<1792506099 )){ Energy[Seg] = ((Gainarray[row][Colm]*(EEnergy))+Offsetarray[row][Colm]);}}
    if (Seg==8){if ((ii < (runsize*(row+1))) && (ii>1792506099)){ Energy[Seg] = ((GainarrayLC[row][Colm]*(EEnergy))+OffsetarrayLC[row][Colm]);}}
return 0;
}
//-----//
//***** Convert XIA raw data format *****//
//*****

Int_t SimSort::Tree(Char_t *fName)
{
typedef struct contentWord Cont_t;

Int_t ult1,i,ii,iii,x,k, s, kk,diff,diff1, l,flag;
info.open("Mod3Hist.txt",ios::app);
//info2.open("WfileMOP2.txt",ios::app);
flag=1;
diff1=8;
diff=0;
eof=1;
x=0;
l=0;
jj=0;
TH1D *EH[3000][10];

Float_t valuep1[10][4], valuep2[10][4],errorp1[10][4],errorp2[10][4];
for(kk=0;kk<9;kk++){Energy[kk]=0; TimeStamp[kk]=0; DetID[kk]=0;}
datafs[0].open(fName,ifstream::binary);
//for(ii=0;ii<100000000;ii++){//dd++;
while (!datafs[0].eof()){
//info<<"In l= " <<l<<endl;
EH[l][0] = new TH1D(Form("hist0_%d",l),"EnergyCore;Energy (keV);Counts",20000,0,10000);
EH[l][1] = new TH1D(Form("hist1_%d",l),"EnergySeg1;Energy (keV);Counts",20000,0,10000);
EH[l][2] = new TH1D(Form("hist2_%d",l),"EnergySeg2;Energy (keV);Counts",20000,0,10000);
EH[l][3] = new TH1D(Form("hist3_%d",l),"EnergySeg3;Energy (keV);Counts",20000,0,10000);
EH[l][4] = new TH1D(Form("hist4_%d",l),"EnergySeg4;Energy (keV);Counts",20000,0,10000);
EH[l][5] = new TH1D(Form("hist5_%d",l),"EnergySeg5;Energy (keV);Counts",20000,0,10000);
EH[l][6] = new TH1D(Form("hist6_%d",l),"EnergySeg6;Energy (keV);Counts",20000,0,10000);
EH[l][7] = new TH1D(Form("hist7_%d",l),"EnergySeg7;Energy (keV);Counts",20000,0,10000);
EH[l][8] = new TH1D(Form("hist8_%d",l),"EnergySeg8;Energy (keV);Counts",20000,0,10000);
//info<<"jj = " <<jj<<endl;
for(ii=jj;ii<(50000000+jj);ii++){ row = Int_t(ii/50000000);
    ReadData(fName,32);
myPtr2[0][0] = (ULong_t) wordPtr2[2];
myPtr2[0][1] = (ULong_t) wordPtr2[3]; myPtr2[0][1] = myPtr2[0][1] << 16;
myPtr2[0][2] = (ULong_t) wordPtr2[4]; myPtr2[0][2] = myPtr2[0][2] << 32;
EVTTIME0 = myPtr2[0][0] | myPtr2[0][1] | myPtr2[0][2];
//cout<<" "<<EVTTIME0<<" "<< ii<<endl;
    if (wordPtr2[x]==33360){if ( wordPtr2[6]>32000 ){ Energy[0]=-32767+wordPtr2[6];}
else {Energy[0]=wordPtr2[6];}
        DetID[0]=wordPtr2[0];
        EH[l][0]->Fill(Energy[0]);
        Energy[0]=0;}
    else if (wordPtr2[x]==33364){if ( wordPtr2[6]>32000 ){ Energy[1]=-32767+wordPtr2[6];}    ///S1
    else {Energy[1]=wordPtr2[6];}
        DetID[1]=wordPtr2[x];
        GainDriftCorr(1,Energy[1],row,ii);
        EH[l][1]->Fill(Energy[1]);
        Energy[1]=0;}
    else if (wordPtr2[x]==33365){if ( wordPtr2[6]>32000 ){ Energy[2]=-32767+wordPtr2[6];}    ///S2
    else {Energy[2]=wordPtr2[6];}
        DetID[2]=wordPtr2[x];
        GainDriftCorr(2, Energy[2], row,ii);
        EH[l][2]->Fill(Energy[2]);
        Energy[2]=0;}
    else if (wordPtr2[x]==33366){if ( wordPtr2[6]>32000 ){ Energy[3]=-32767+wordPtr2[6];}    ///S3
    else {Energy[3]=wordPtr2[6];}
        DetID[3]=wordPtr2[x];
        GainDriftCorr(3, Energy[3], row,ii);

```

```

EH[1][3]->Fill(Energy[3]);
Energy[3]=0;}
else if (wordPtr2[x]==33367){if ( wordPtr2[6]>32000 ){ Energy[4]=-32767+wordPtr2[6];}   ///S4
else {Energy[4]=wordPtr2[6];}
DetID[4]=wordPtr2[x];
GainDriftCorr(4, Energy[4], row,ii);
EH[1][4]->Fill(Energy[4]);
Energy[4]=0;}
else if (wordPtr2[x]==33368){if ( wordPtr2[6]>32000 ){ Energy[5]=-32767+wordPtr2[6];}   ///S5
else {Energy[5]=wordPtr2[6];}
DetID[5]=wordPtr2[x];
GainDriftCorr(5, Energy[5], row,ii);
EH[1][5]->Fill(Energy[5]);
Energy[5]=0;}
else if (wordPtr2[x]==33369){if ( wordPtr2[6]>32000 ){ Energy[6]=-32767+wordPtr2[6];}   ///S6
else {Energy[6]=wordPtr2[6];}
DetID[6]=wordPtr2[x];
GainDriftCorr(6, Energy[6], row, ii);
EH[1][6]->Fill(Energy[6]);
Energy[6]=0;} // cout<<DetID[6]<<" "<<Energy[6]<<" "<< ii<<endl;}
else if (wordPtr2[x]==33370){if ( wordPtr2[6]>32000 ){ Energy[7]=-32767+wordPtr2[6];}   ///S7
else {Energy[7]=wordPtr2[6];}
DetID[7]=wordPtr2[x];
GainDriftCorr(7, Energy[7], row,ii);
EH[1][7]->Fill(Energy[7]);
Energy[7]=0;}
else if (wordPtr2[x]==33371){if ( wordPtr2[6]>32000 ){ Energy[8]=-32767+wordPtr2[6];}   ///S8
else {Energy[8] = wordPtr2[6];}
DetID[8]=wordPtr2[x];
GainDriftCorr(8, Energy[8], row,ii);
EH[1][8]->Fill(Energy[8]);
Energy[8]=0;}

}
//info<<"value jj= "<<jj<<" ii= "<<ii<<endl;
jj=(jj+500000000);
//info<<"new value jj= "<<jj<<" ii= "<<ii<<endl;
TFile *f = TFile::Open(Form("HistACrM3_%.d.root",1), "RECREATE");

TF1 *P1 = new TF1("P1","gaus(0)+pol1(3)",0,10000);
TF1 *P2 = new TF1("P2","gaus(0)+pol1(3)",0,10000);
/////////////////////////
//info<<"Core "<<endl;
//info2<<"Core "<<endl;
P1->SetParameters(268800,500,3,0,0); //core
P2->SetParameters(1600,7865,3,0,0);
EH[1][0]->Fit("P1","", "",470,530);
EH[1][0]->Fit("P2","", "",7850,8100);

for (i=0;i<3;i++) {
    valuep1[0][i] = P1->GetParameter(i);
    errorp1[0][i] = P1->GetParError(i);}
//info<<valuep1[0][i]<<" += "<<errorp1[0][i]<<" "<< 1 <<endl;
//info<<" "<<endl;
for (i=0;i<3;i++) {
    valuep2[0][i] = P2->GetParameter(i);
    errorp2[0][i] = P2->GetParError(i);}

/////////////////////////
//info<<"Seg 1 "<<endl;
//info2<<"Seg 1 "<<endl;
P1->SetParameters(20000,480,8.13,0,0); //Seg1
P2->SetParameters(20,8580,10.06,0,0);
P1->FixParameter(2,8.13);
P2->FixParameter(2,10.06);
EH[1][1]->Fit("P1","", "",400,600);
EH[1][1]->Fit("P2","", "",8600,9000);
for (i=0;i<3;i++) {
    valuep1[1][i] = P1->GetParameter(i);

```

```

        errorp1[1][i] = P1->GetParError(i);
        // info<<valuep1[1][i]<<" += "<<errorp1[1][i]<<" "<< 1 <<endl;
    }//info<<" "<<endl;
    for (i=0;i<3;i++) {
        valuep2[1][i] = P2->GetParameter(i);
        errorp2[1][i] = P2->GetParError(i);
        // info<<valuep2[1][i]<<" += "<<errorp2[1][i]<<" "<< 1 <<endl;

        }//info<<" "<<endl;
    if (valuep2[1][0] < 1){
P2->SetParameters(20,8620,10.06,0,0);
P2->FixParameter(2,10.06);
EH[1][1]->Fit("P2","", "",8400,9000);}
    for (i=0;i<3;i++) {
        valuep2[1][i] = P2->GetParameter(i);
        errorp2[1][i] = P2->GetParError(i);
        //info<<valuep2[3][i]<<" += "<<errorp2[3][i]<<" "<< 1 <<endl;

    }
    if (valuep2[1][0] < 1){
P2->SetParameters(20,8675,10.06,0,0);
P2->FixParameter(2,10.06);
EH[1][1]->Fit("P2","", "",8400,9000);}
    for (i=0;i<3;i++) {
        valuep2[1][i] = P2->GetParameter(i);
        errorp2[1][i] = P2->GetParError(i);
        //info<<valuep2[3][i]<<" += "<<errorp2[3][i]<<" "<< 1 <<endl;

    }
}
//////////*****//////////
//info<< "Seg 2 "<<endl;
//info2<< "Seg 2 "<<endl;
P1->SetParameters(20000,490,8.00,0,0); //Seg2
P2->SetParameters(10,8860,8.98,0,0);
P1->FixParameter(2,8.00);
P2->FixParameter(2,8.98);
EH[1][2]->Fit("P1","", "",400,600);
EH[1][2]->Fit("P2","", "",8800,9000);
    for (i=0;i<3;i++) {
        valuep1[2][i] = P1->GetParameter(i);
        errorp1[2][i] = P1->GetParError(i);
        //info<<valuep1[2][i]<<" += "<<errorp1[2][i]<<" "<< 1 <<endl;
    }//info<<" "<<endl;
    for (i=0;i<3;i++) {
        valuep2[2][i] = P2->GetParameter(i);
        errorp2[2][i] = P2->GetParError(i);
        //info<<valuep2[2][i]<<" += "<<errorp2[2][i]<<" "<< 1 <<endl;
    }//info<<" "<<endl;

    if (valuep2[2][0] < 1){
P2->SetParameters(15,8880,8.98,0,0);
P2->FixParameter(2,8.98);
EH[1][2]->Fit("P2","", "",8800,9000);}
    for (i=0;i<3;i++) {
        valuep2[2][i] = P2->GetParameter(i);
        errorp2[2][i] = P2->GetParError(i);}
    if (valuep2[2][0] < 1){
P2->SetParameters(15,8920,8.98,0,0);
P2->FixParameter(2,8.98);
EH[1][2]->Fit("P2","", "",8800,9000);}
    for (i=0;i<3;i++) {
        valuep2[2][i] = P2->GetParameter(i);
        errorp2[2][i] = P2->GetParError(i);
        //info<<valuep2[3][i]<<" += "<<errorp2[3][i]<<" "<< 1 <<endl;

    }
}
//////////*****//////////
//info<< "Seg 3 "<<endl;
//info2<< "Seg 3 "<<endl;

```

```

P1->SetParameters(26000,510,9.73,0,0); //Seg3
P2->SetParameters(10,9140,8.29,0,0); //Seg3
P1->FixParameter(2,9.73);
P2->FixParameter(2,8.29);
EH[1][3]->Fit("P1","", "", 400,650);
EH[1][3]->Fit("P2","", "", 9000,9300);
for (i=0;i<3;i++) {
    valuep1[3][i] = P1->GetParameter(i);
    errorp1[3][i] = P1->GetParError(i);
    //info<<valuep1[3][i]<<" += "<<errorp1[3][i]<<" "<< 1 <<endl;
} //info<<" "<<endl;
for (i=0;i<3;i++) {
    valuep2[3][i] = P2->GetParameter(i);
    errorp2[3][i] = P2->GetParError(i);
    //info<<valuep2[3][i]<<" += "<<errorp2[3][i]<<" "<< 1 <<endl;

}
if (valuep2[3][0] < 1){
P2->SetParameters(10,9160,8.29,0,0);
P2->FixParameter(2,8.29);
EH[1][3]->Fit("P2","", "", 9000,9300);}
for (i=0;i<3;i++) {
    valuep2[3][i] = P2->GetParameter(i);
    errorp2[3][i] = P2->GetParError(i);
    //info<<valuep2[3][i]<<" += "<<errorp2[3][i]<<" "<< 1 <<endl;

}
if (valuep2[3][0] < 1){
P2->SetParameters(10,9200,8.29,0,0);
P2->FixParameter(2,8.29);
EH[1][3]->Fit("P2","", "", 9000,9300);}
for (i=0;i<3;i++) {
    valuep2[3][i] = P2->GetParameter(i);
    errorp2[3][i] = P2->GetParError(i);
    //info<<valuep2[3][i]<<" += "<<errorp2[3][i]<<" "<< 1 <<endl;

}

}

//info<<" "<<endl;
//////////*****////////////////
//info<< "Seg 4 "<<endl;
//info2<< "Seg 4 "<<endl;
P1->SetParameters(25000,455,8.43,0,0); //Seg4
P2->SetParameters(15,8130,9.05,0,0); //Seg4
P1->FixParameter(2,8.43);
P2->FixParameter(2,9.05);
EH[1][4]->Fit("P1","", "", 390,600);
EH[1][4]->Fit("P2","", "", 8200,8800);
for (i=0;i<3;i++) {
    valuep1[4][i] = P1->GetParameter(i);
    errorp1[4][i] = P1->GetParError(i);
    // info<<valuep1[4][i]<<" += "<<errorp1[4][i]<<" "<< 1 <<endl;
} //info<<" "<<endl;
for (i=0;i<3;i++) {
    valuep2[4][i] = P2->GetParameter(i);
    errorp2[4][i] = P2->GetParError(i);
    //info<<valuep2[4][i]<<" += "<<errorp2[4][i]<<" "<< 1 <<endl;

} //info<<" "<<endl;

if (valuep2[4][0] < 1){
P2->SetParameters(15,8140,9.05,0,0);
P2->FixParameter(2,9.05);
EH[1][4]->Fit("P2","", "", 8000,8300);}
for (i=0;i<3;i++) {
    valuep2[4][i] = P2->GetParameter(i);
    errorp2[4][i] = P2->GetParError(i);
}

```

```

if (valuep2[4][0] < 1){
P2->SetParameters(15,8185,9.05,0,0);
P2->FixParameter(2,9.05);
EH[1][4]->Fit("P2","", "",8000,8300);}
for (i=0;i<3;i++) {
    valuep2[4][i] = P2->GetParameter(i);
    errorp2[4][i] = P2->GetParError(i);
}
//////////*****//////////
//info<< "Seg 5 "<<endl;
//info2<< "Seg 5 "<<endl;
P1->SetParameters(1500,2180,7.64,0,0); //Seg5
P2->SetParameters(50,8960,8.71,0,0); //Seg5
P1->FixParameter(2,7.64);
P2->FixParameter(2,8.71);
EH[1][5]->Fit("P1","", "",2000,2300);
EH[1][5]->Fit("P2","", "",8800,9200);

for (i=0;i<3;i++) {
    valuep1[5][i] = P1->GetParameter(i);
    errorp1[5][i] = P1->GetParError(i);
    // info<<valuep1[5][i]<<" += "<<errorp1[5][i]<<" "<< 1 <<endl;

} //info<<" "<<endl;
for (i=0;i<3;i++) {
    valuep2[5][i] = P2->GetParameter(i);
    errorp2[5][i] = P2->GetParError(i);}
if (valuep2[5][0] < 1){
P2->SetParameters(35,8980,8.71,0,0);
P2->FixParameter(2,8.71);
EH[1][5]->Fit("P2","", "",8800,9200);}
for (i=0;i<3;i++) {
    valuep2[5][i] = P2->GetParameter(i);
    errorp2[5][i] = P2->GetParError(i);
    //info<<valuep2[3][i]<<" += "<<errorp2[3][i]<<" "<< 1 <<endl;

}

if (valuep2[5][0] < 1){
P2->SetParameters(35,9000,8.71,0,0);
P2->FixParameter(2,8.71);
EH[1][5]->Fit("P2","", "",8800,9200);}
for (i=0;i<3;i++) {
    valuep2[5][i] = P2->GetParameter(i);
    errorp2[5][i] = P2->GetParError(i);
}

//////////*****//////////
//info<< "Seg 6 "<<endl;
//info2<< "Seg 6 "<<endl;
P1->SetParameters(1500,2210,7.83,0,0); //Seg6
P2->SetParameters(30,7900,8.96,0,0); //Seg6
P1->FixParameter(2,7.83);
P2->FixParameter(2,8.96);
EH[1][6]->Fit("P1","", "",2100,2400);
EH[1][6]->Fit("P2","", "",7800,8310);

for (i=0;i<3;i++) {
    valuep1[6][i] = P1->GetParameter(i);
    errorp1[6][i] = P1->GetParError(i);
    //info<<valuep1[6][i]<<" += "<<errorp1[6][i]<<" "<< 1 <<endl;
} //info<<" "<<endl;
for (i=0;i<3;i++) {
    valuep2[6][i] = P2->GetParameter(i);
    errorp2[6][i] = P2->GetParError(i);
    //info<<valuep2[6][i]<<" += "<<errorp2[6][i]<<" "<< 1 <<endl;

} //info<<" "<<endl;
if (valuep2[6][0] < 1){
P2->SetParameters(30,7915,8.96,0,0);

```

```

P2->FixParameter(2,8.96);
EH[1][6]->Fit("P2","", "", 7800,8300);}
for (i=0;i<3;i++) {
    valuep2[6][i] = P2->GetParameter(i);
    errorp2[6][i] = P2->GetParError(i);

}

if (valuep2[6][0] < 1){
P2->SetParameters(30,7930,8.96,0,0);
P2->FixParameter(2,8.96);
EH[1][6]->Fit("P2","", "", 7800,8300);}
for (i=0;i<3;i++) {
    valuep2[6][i] = P2->GetParameter(i);
    errorp2[6][i] = P2->GetParError(i);
    //info<<valuep2[3][i]<<" += "<<errorp2[3][i]<<" "<< 1 <<endl;

}

//////////*****//////////
//info<< "Seg 7 "<<endl;
//info2<< "Seg 7 "<<endl;
P1->SetParameters(3000,2260,8.36,0,0); //Seg7
P2->SetParameters(35,8100,9.15,0,0); //Seg7
P1->FixParameter(2,8.36);
P2->FixParameter(2,9.15);
EH[1][7]->Fit("P1","", "", 2100,2400);
EH[1][7]->Fit("P2","", "", 8000,8300);
for (i=0;i<3;i++) {
    valuep1[7][i] = P1->GetParameter(i);
    errorp1[7][i] = P1->GetParError(i); }//info<<" "<<endl;
for (i=0;i<3;i++) {
    valuep2[7][i] = P2->GetParameter(i);
    errorp2[7][i] = P2->GetParError(i);}

if (valuep2[7][0] < 1){
P2->SetParameters(35,8120,9.15,0,0);
P2->FixParameter(2,9.15);
EH[1][7]->Fit("P2","", "", 8000,8300);}
for (i=0;i<3;i++) {
    valuep2[7][i] = P2->GetParameter(i);
    errorp2[7][i] = P2->GetParError(i);

}

if (valuep2[7][0] < 1){
P2->SetParameters(35,8130,9.15,0,0);
P2->FixParameter(2,9.15);
EH[1][7]->Fit("P2","", "", 8000,8300);}
for (i=0;i<3;i++) {
    valuep2[7][i] = P2->GetParameter(i);
    errorp2[7][i] = P2->GetParError(i);

}

//info<<" "<<endl;
//////////*****//////////
//info<< "Seg 8 "<<endl;
//info2<< "Seg 8 "<<endl;
P1->SetParameters(1200,2120,10.5,0,0); //Seg8
P2->SetParameters(30,8720,10.7,0,0); //Seg8
P1->FixParameter(2,10.5);
P2->FixParameter(2,10.7);
EH[1][8]->Fit("P1","", "", 2000,2300);
EH[1][8]->Fit("P2","", "", 8600,8900);

for (i=0;i<3;i++) {
    valuep1[8][i] = P1->GetParameter(i);
    errorp1[8][i] = P1->GetParError(i);

}

//info<<" "<<endl;
for (i=0;i<3;i++) {
    valuep2[8][i] = P2->GetParameter(i);

```

```

        errorp2[8][i] = P2->GetParError(i);
        // info<<valuep2[8][i]<<" += "<<errorp2[8][i]<<" "<< 1 <<endl;

    }
    if (valuep2[8][0] < 1){
    P2->SetParameters(30,8740,10.7,0,0);
    P2->FixParameter(2,10.7);
    EH[1][8]->Fit("P2","", "",8600,8900);}
    for (i=0;i<3;i++) {
        valuep2[8][i] = P2->GetParameter(i);
        errorp2[8][i] = P2->GetParError(i);
    }

    info<<valuep1[0][1]<<" "<<valuep1[1][1]<<" "<< valuep1[2][1]<<" "<<valuep1[3][1]
    <<" "<<valuep1[4][1]<<" "<<valuep1[5][1]<<" "<<valuep1[6][1]<<" "<<valuep1[7][1]
    <<" "<<valuep1[7][1]<<" "<<valuep2[0][1]<<" "<<valuep2[1][1]<<" "<<valuep2[2][1]
    <<" "<<valuep2[3][1]<<" "<<valuep2[4][1]<<" "<<valuep2[5][1]<<" "<<valuep2[6][1]
    <<" "<<valuep2[7][1]<<" "<<valuep2[8][1]<<" "<<1 <<endl;

    for (k=0; k<9;k++){

    EH[1][k]->Write();}
    l++; }
    TCanvas *Ff = new TCanvas("EnergyFront","FrontEnergySpectra",1);
    Ff->Divide(3,3);
    for(s=0; s<9; s++){
    Ff->cd(s+1);
    EH[1][s]->Draw();}
    return 0;
}
////////////////////////////////////////
//Int_t mm,xr,xc,kk; //include the gain matching here just after the gaindrift

```

Appendix B

Code written by T.D. Bucher

```
#include "Riostream.h"
// #include "SimSort.h"
// #include "lib/SimSortCom.h"
// #include "lib/Bytes.h"
#include "Byteswap.h"
#include "TTree.h"
#include "TH2.h"
#include "TChain.h"
#include <iostream>
#include <fstream>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include <stddef.h>
#include "Rtypes.h"
#include "TR00T.h"
#include "TSystem.h"
#include <iomanip>
#include "TFile.h"
#include "TBrowser.h"
using namespace std;
using std::cout;
using std::endl;
class SimSort {
public:
    SimSort();
    ~SimSort();
    Double_t timeBuf[256][2048], timeBuf2[2048], timeins, pow16_8, pow16_4;
    Int_t empful[256], worful[2][256], wordins[256], wins, words3;
    Int_t ggsize, g, gg, tableADC[4][16384];
    Double_t etot[32], cz[32];
    ofstream datagg;
    ofstream info, info1;
    ifstream datafs[10]; // attention: pb if there are more than 128 runs in an experiment
    ofstream dataws; // attention: pb if there are more than 128 runs in an experiment
    UShort_t wordBuf[256][8192], wordPtr3[4][16384], wordPtr4[32768];
    UShort_t adcdet[400], adcData[4][2048], adcList[4][2048];
    UShort_t adcRepair[2048], *wordPtr, *ep, wordPtr2[4][16384], crystalbuf[32];
    UInt_t myPtr[4][2048], myPtr4[4][16384];
    ULong_t myPtr2[4][2048], myPtr3[4][16384];
    UShort_t myEnerg[4][2048];
    Int_t words[4], ADC_events, eof, blocks, check, check2, stepblock, stepblock2, ult, endian;
    Short_t eventenergy;
    ULong_t EVTTIME, EVTtime[1], mergedArray[16384], time[4], time0, time1, time2;
    UShort_t CFDtime, tracelength, trace[400];
    ULong_t TrailEsum, ESleading, GapEsum, Baseline;
    UShort_t headerlgth, crateid, slotid, channelno, header;
    Int_t ReadData(Char_t * = NULL, Int_t=0, Int_t = 0);
    Int_t Checkdata(Char_t * = NULL, Int_t=0);
```

```

Int_t WriteData(Char_t * = NULL, Int_t=0);
//Int_t ConvertDataPixie16();
Int_t MinTime(Char_t * = NULL, Char_t * = NULL;
Char_t * = NULL, Char_t * = NULL, Char_t * = NULL);
Int_t postpone, seq, wordxia1[4], wordxia2, ultxia;
Int_t fConvert;

private:
Int_t fWrite;
Int_t fRead;
Int_t j, t;
};
//-----//
//***** Initialise all the tables (calibration...) *****/
//*****
SimSort::SimSort()
{}
SimSort::~SimSort()
{}

//-----//
//***** Read data *****/
//*****

Int_t SimSort::ReadData(Char_t *dataf, Int_t hh, Int_t skbyte)
{
    struct addressWord {
        UShort_t data;
        UShort_t group;
        UShort_t item;
        UShort_t type;
    };
    typedef struct addressWord Addr_t;
    struct contentWord {
        UShort_t data;
    };
    typedef struct contentWord Cont_t;
    const Int_t blkSize = 32; // in "no traces" bytes
    const Int_t nWdChar = blkSize / sizeof(Char_t);
    const Int_t nWdInt = blkSize / sizeof(UShort_t);
    Int_t pbyte;
    Char_t bytes[blkSize];
    Char_t *bytesPtr = bytes;
    //Char_t *wbytes= new Data[]
    fRead=1;
    ult=0;

    {
    if (! datafs[hh].is_open()) {
    datafs[hh].open(dataf, ifstream::binary);
    if (datafs[hh].is_open()) cout << "ReadData: Reading raw data file \"" << dataf << "\"..."<< endl;
    else cout << "ReadData: Couldn't find raw data file \"" << dataf << "\"..."<< endl;
    }
    if(datafs[hh].eof() == 0) {datafs[hh].seekg(-skbyte,
    ios_base::cur);pbyte = datafs[hh].tellg();
    if (datafs[hh].read(bytesPtr, nWdChar)) {
    blocks++;
    wordPtr = (UShort_t *) bytesPtr;
    for (j=0; j < nWdInt; j++){
    if(endian) wordPtr2[hh][j]=Rbswap_16(wordPtr[j]);
    else wordPtr2[hh][j]=wordPtr[j];
    }
    words[hh]=nWdInt;
    }
    else {
    fRead=0;
    eof=0;
    datafs[hh].close();
    }
    }
}

```

```

    }
    if((blocks-check)>=0) {
        //cout << "Block read: " << blocks<< "[1A" << endl;
        check=check+stepblock;
    }
    }
    else {
        datafs[hh].close();
        cout << " Ouch ! " << endl;
    }
    }
    return fRead;
}

//-----//
//-----//
//***** Write one block of data *****//
//-----//
Int_t SimSort::WriteData(Char_t *dataw, Int_t hh)
{
    Int_t i;
    //info1.open("infomergechono.txt",ios::app);
    const Int_t wblkSize = 32; // in bytes, 16 words x 2 bytes each
    const Int_t wnWdChar = wblkSize / sizeof(Char_t);
    const Int_t wnWdInt = wblkSize / sizeof(USHort_t);
    //cout <<"wnWdChar "<< wnWdChar << endl;
    Char_t bytes[wblkSize];
    Char_t *wbytesPtr;
    //info1 << "modno "<< hh << " time " << time[hh] << endl;
    if (! dataws.is_open()) {
        dataws.open(dataw,ofstream::binary);
        if (dataws.is_open()) fWrite==1;
        else cout << "WriteData: Couldn't open data file \"" << dataw<< "\"..."<< endl;
    }
    for (i=0; i < wnWdInt; i++){
        wordPtr3[hh][i] = wordPtr2[hh][i];
        //dataws[hh].write(wbytesPtr,wblkSize);
    }
    wbytesPtr= (Char_t*) wordPtr3[hh];
    //means storing the starting address of 1st row to the pointer variable wbytesPtr.
    dataws.write(wbytesPtr,32); //32 is the block size
    //cout << *wordPtr3 << endl;
    info1.close();
    return fWrite;
}

//-----//
//-----//
//***** Checking data *****//
//-----//
Int_t SimSort::Checkdata(Char_t *ffName, Int_t hh)
{
    Int_t jj,coreID,segID,segmiss,segmis1,sbyte,recheck;
    Int_t x=7;
    segmiss = 0;
    segmis1 = 0;
    recheck = 1;
    sbyte = 0;
    Int_t Ethhit = 20;
    Int_t count=0;
    //while(recheck !=0){
    //if ( wordPtr2[hh][6]<32000 && wordPtr2[hh][6] > Ethhit){
    recheck = 0; segmiss = 0; segmis1 = 0;
    myPtr2[hh][0]=(ULong_t) wordPtr2[hh][x-5];
    myPtr2[hh][1]=(ULong_t) wordPtr2[hh][x-4]; myPtr2[hh][1] = myPtr2[hh][1] << 16;
    myPtr2[hh][2]=(ULong_t) wordPtr2[hh][x-3]; myPtr2[hh][2] = myPtr2[hh][2] << 32;
    time[hh]=myPtr2[hh][0] | myPtr2[hh][1] | myPtr2[hh][2];

}

//-----//
//***** Merging data *****//

```

```

////////////////////////////////////
Int_t SimSort::MinTime(Char_t *fName0, Char_t *fName1,
Char_t *fName2, Char_t *fName3, Char_t *wName1)
{
UShort_t tt;
info.open("infomerge.txt",ios::app);
datafs[0].open(fName0,ifstream::binary);
datafs[1].open(fName1,ifstream::binary);
datafs[2].open(fName2,ifstream::binary);
datafs[3].open(fName3,ifstream::binary);
Int_t ii,lk,sbyte1;
sbyte1 = 0;
ReadData(fName0,0,sbyte1); Checkdata(fName0,0);
ReadData(fName1,1,sbyte1); Checkdata(fName1,1);
ReadData(fName2,2,sbyte1); Checkdata(fName2,2);
ReadData(fName3,3,sbyte1); Checkdata(fName3,3);
Double_t ll;
Int_t x=7;
//cout << time[0] << endl;
//cout << time[1] << endl;
//cout << time[2] << endl;
//cout << time[3] << endl;
//for(ii=0;ii<100;ii++){ cout << "I am inside the loop"<<endl;
while (!datafs[0].eof() && !datafs[1].eof()
&& !datafs[2].eof() && !datafs[3].eof()){
if (time[0]<= time[1]){
if (time[0]<= time[2]){
if (time[0]<= time[3]) { WriteData(wName1,0);
ReadData(fName0,0,sbyte1);
Checkdata(fName0,0);
//info << "old time[0] " << time[0] << endl;
}
else {WriteData(wName1,3);
ReadData(fName3,3,sbyte1);
Checkdata(fName3,3);
//info << "old time[3] " << time[3] << endl;
}
}
else {if (time[2]<= time[3]){ WriteData(wName1,2);
ReadData(fName2,2,sbyte1);
Checkdata(fName2,2);
//info << "old time[2] " << time[2] << endl;
}
else{WriteData(wName1,3);
ReadData(fName3,3,sbyte1);
Checkdata(fName3,3);
}
}
else if (time[1]<= time[2]){
if (time[1]<= time[3]) { WriteData(wName1,1);
ReadData(fName1,1,sbyte1);
Checkdata(fName1,1);
}
else { WriteData(wName1,3);
ReadData(fName3,3,sbyte1);
Checkdata(fName3,3);
}
}
else if (time[2]<= time[3]) { WriteData(wName1,2);
ReadData(fName2,2,sbyte1);
Checkdata(fName2,2);
}
else { WriteData(wName1,3);
ReadData(fName3,3,sbyte1);
Checkdata(fName3,3);}}
while(!datafs[0].eof() && !datafs[1].eof()
&& !datafs[2].eof()){
if (time[0]<= time[1]){
if (time[0]<= time[2]){ WriteData(wName1,0);

```

```

        ReadData(fName0,0,sbyte1);
        Checkdata(fName0,0);
    }
    else {WriteData(wName1,2);
    ReadData(fName2,2,sbyte1);
    Checkdata(fName2,2);
    }
}
else if (time[1]<= time[0]){
if (time[1]<= time[2]){WriteData(wName1,1);
    ReadData(fName1,1,sbyte1);
    Checkdata(fName1,1);
}
else { WriteData(wName1,2);
    ReadData(fName2,2,sbyte1);
    Checkdata(fName2,2);
    }
}
}
while(!datafs[1].eof() && !datafs[2]
.eof() && !datafs[3].eof()){
if (time[1]<= time[2]){
if (time[0]<= time[3]){ WriteData(wName1,1);
    ReadData(fName1,1,sbyte1);
    Checkdata(fName1,1);
}
else {WriteData(wName1,3);
    ReadData(fName3,3,sbyte1);
    Checkdata(fName3,3);
}
}
else if (time[2]<= time[1]){
if (time[2]<= time[3]){ WriteData(wName1,2);
    ReadData(fName2,2,sbyte1);
    Checkdata(fName2,2);
}
else { WriteData(wName1,3);
    ReadData(fName3,3,sbyte1);
    Checkdata(fName3,3);
    }
}
}
while(!datafs[0].eof()&& !datafs[2].
eof() && !datafs[3].eof()){
if (time[0]<= time[2]){
if (time[0]<= time[3]){ WriteData(wName1,0);
    ReadData(fName0,0,sbyte1);
    Checkdata(fName0,0);
}
else {WriteData(wName1,3);
    ReadData(fName3,3,sbyte1);
    Checkdata(fName3,3);
}
}
else if (time[2]<= time[0]){
if (time[2]<= time[3]){WriteData(wName1,2);
    ReadData(fName2,2,sbyte1);
    Checkdata(fName2,2);
}
else { WriteData(wName1,3);
    ReadData(fName3,3,sbyte1);
    Checkdata(fName3,3);
    }
}
}
}
while(!datafs[0].eof() && !datafs[1].eof() && !datafs[3].eof()){
if (time[0]<= time[1]){
if (time[0]<= time[3]){ WriteData(wName1,0);
    ReadData(fName0,0,sbyte1);
    Checkdata(fName0,0);

```

```

}
else {WriteData(wName1,3);
ReadData(fName3,3,sbyte1);
Checkdata(fName3,3);
}
}
else if (time[1]<= time[0]){
if (time[1]<= time[3]){ WriteData(wName1,1);
ReadData(fName1,1,sbyte1);
Checkdata(fName1,1);
}
else { WriteData(wName1,3);
ReadData(fName3,3,sbyte1);
Checkdata(fName3,3);      }
}
}
while(!datafs[0].eof() && !datafs[1].eof()){
if (time[0]<= time[1]){
WriteData(wName1,0);
ReadData(fName0,0,sbyte1);
Checkdata(fName0,0); //info << "time[0] " << time[0] << endl;
}
else {
WriteData(wName1,1);
ReadData(fName1,1,sbyte1);
Checkdata(fName1,1); //info << "time[2] " << time[2] << endl;
}
}
while(!datafs[1].eof() && !datafs[2].eof()){
if (time[1]<= time[2]){
WriteData(wName1,1);
ReadData(fName1,1,sbyte1);
Checkdata(fName1,1);// info << "time[0] " << time[0] << endl;
}
else {
WriteData(wName1,2);
ReadData(fName2,2,sbyte1);
Checkdata(fName2,2);
}
}
while(!datafs[2].eof() && !datafs[3].eof()){
if (time[2]<= time[3]){
WriteData(wName1,2);
ReadData(fName2,2,sbyte1);
//info << "old time[0] " << time[0] << endl;
Checkdata(fName2,2); // info << "time[0] " << time[0] << endl;
}
else {
WriteData(wName1,3);
ReadData(fName3,3,sbyte1);
Checkdata(fName3,3);// info << "time[2] " <<time[2] << endl;
}
}
while(!datafs[0].eof() && !datafs[3].eof()){
if (time[0]<= time[3]){
WriteData(wName1,0);
ReadData(fName0,0,sbyte1);
Checkdata(fName0,0); // info << "time[0] " << time[0] << endl;
}
else {
WriteData(wName1,3);
ReadData(fName3,3,sbyte1);
Checkdata(fName3,3);// info << "time[2] " << time[2] << endl;
}
}
while(!datafs[1].eof() && !datafs[3].eof()){
if (time[1]<= time[3]){
WriteData(wName1,1);
ReadData(fName1,1,sbyte1);
Checkdata(fName1,1); //info << "time[0] " << time[0] << endl;
}
}

```

```

}
else {
WriteData(wName1,3);
ReadData(fName3,3,sbyte1);
Checkdata(fName3,3); // info << "time[2] " <<time[2] << endl;
}
}
while(!datafs[0].eof() && !datafs[2].eof()){
if (time[0]<= time[3]){
WriteData(wName1,0);
ReadData(fName0,0,sbyte1);
Checkdata(fName0,0); //info << "time[0] " << time[0] << endl;
}
else {
WriteData(wName1,2);
ReadData(fName2,2,sbyte1);
Checkdata(fName2,2); //info << "time[2] " <<time[2] << endl;
}
}
while(!datafs[0].eof()){
WriteData(wName1,0);
ReadData(fName0,0,sbyte1);
Checkdata(fName0,0);
}
while(!datafs[1].eof()){
WriteData(wName1,1);
ReadData(fName1,1,sbyte1);
Checkdata(fName1,1);
}
while(!datafs[2].eof()){
WriteData(wName1,2);
ReadData(fName2,2,sbyte1);
Checkdata(fName2,2);
}
while(!datafs[3].eof()){
WriteData(wName1,3);
ReadData(fName3,3,sbyte1);
Checkdata(fName3,3);
}
if (! dataws.is_open()) {
cout << "closing file" << endl;
dataws.close();
cout << "file closed" << endl;
}
else cout << "file was closed" << endl;
info.close();
return 0;
}

```

Appendix C

Code written by T.D. Bucher

```
#include "Riostream.h"
// #include "SimSort.h"
// #include "lib/SimSortCom.h"
// #include "lib/Bytes.h"
#include "Byteswap.h"
#include "TTree.h"
#include "TH2.h"
#include "TH1.h"
#include "TChain.h"
#include <iostream>
#include <fstream>
#include <sstream>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include <stddef.h>
#include "Rtypes.h"
#include "TROOT.h"
#include "TSystem.h"
#include <iomanip>
#include "TFile.h"
#include "TBrowser.h"
#include "TObjArray.h"
#include "TObject.h"
#include "TKey.h"
#include "TMapFile.h"
using namespace std;
using std::cout;
using std::endl;

class SimSort {
public:
    SimSort();
    ~SimSort();
    Double_t timeBuf[256][2048], timeBuf2[2048], timeins, pow16_8, pow16_4;
    Int_t empful[256], worful[2][256], wordins[256], wins, words3;
    ofstream datagg;
    ofstream info;
    ifstream datafs[128];
    ofstream dataws[128];
    UShort_t wordBuf[256][8192], wordPtr3[16384], wordPtr4[32768];
    UShort_t UShort_t adcdet[400], adcData[4][2048], adcList[4][2048]
    adcRepair[2048], *wordPtr, *ep, wordPtr2[16384];
    UInt_t myPtr[4][2048];
    ULong_t myPtr2[4][2048];
    UShort_t myEnerg[4][2048];
    Int_t words, ADC_events, eof, blocks, check, check2, stepblock, stepblock2, ult;
    Double_t timedef3, timedef0, timedef1, timedef2, timetestdef;
    Int_t endian, hit, hittre_c[4], hittre_s[4], hitt;
```

```

Int_t hittt, Eth,seghit[4][8], nuhit, mod;
UShort_t eventenergy,eventenergyAcor,Crystal[4];
UShort_t crystalbuf[20][10], DetID[4][10];
Double_t Energy[4][10],Energydrift[4][10],Energyxtc[4][10];
Double_t REnergy[4],REnergync[4];
Double_t Edb157[4][10],Edb175[4][10], Energybuf[20][10];
Double_t TSEnergy[4],TSEnergync[4],TEnergy;
Double_t xtkarray[40][40],Etot[9],Etott,REnergySeg[4][8],EnergySeg[4][8];
Double_t CoreEnergy[4],GDGain[40][40],GDOffset[40][40];
Double_t ULong_t EVTIME0,EVTIME1, EVTIME2,EVTIME3,EVTIMEA;
Double_t timest[32],timeref,TimeStmp[4][10];
Double_t timestmpbuf[32],timestmpsegbuf[20][32],time0;
Double_t time1,time2,time3,timetest, Time,deltaTimeStmp;
UShort_t CFDtime[4][10],CFDtimebuf[20][10],tracelength;
ULong_t TrailEsum,ESleading,GapEsum,Baseline;
UShort_t trace[4][10][400],tracebuf[20][10][400];
UShort_t headerlgth,crateid,slotid,channelno,crystalID;
Int_t ReadData(Char_t * = NULL,Int_t=0);
//Int_t ConvertDataPixie16();
Int_t GainMatch(Int_t=0, Double_t=0, Int_t=0, Int_t=0, Int_t=0);
Int_t CrossTalk(Int_t=0, Int_t=0);
Int_t Tree(Char_t * = NULL);
Int_t CalibTree(Char_t * = NULL);
//Int_t TraceShift(Int_t=0);
Int_t postpone,seq,wordxia1,wordxia2,ultxia;
Int_t fConvert;
Double_t energy1,energy2,energy3,energy4;
Int_t SetOffset(const char * = NULL);
Int_t SetGain(const char * = NULL);
Int_t SetXtalk(const char * = NULL);
Int_t GainDrift(const char * = NULL);
Int_t OffsetDrift(const char * = NULL);
Int_t fGain;
Int_t fOffset;
Double_t fAdcOffset[100];
Double_t fAdcGain[100],addback[100];

private:

    Int_t fWrite;
    Int_t fRead;
    Int_t j, t;
};

//-----//
//***** open and reads the gain file *****//

Int_t SimSort::SetGain(const char *gainfName)
{
    Int_t i;
    char buf[320];
    Float_t gain;
    ifstream gainf;
    gainf.open(gainfName);

    ofstream gaindef;
    i=0;

    while (! gainf.eof()) {
    gainf.getline(buf,320);
    sscanf(buf,"%f",&gain);
    fAdcGain[i] = gain;
    i++;
        fGain++;
    }

    gainf.close();
    return fGain;
}

```

```

}

//-----//
//***** Reads the offset *****//
//-----//

Int_t SimSort::SetOffset(const char *offsetfName)
{
    Int_t i;
    char buf[320];
    Float_t offset;
    ifstream offsetf;
    offsetf.open(offsetfName);

    i=0;

    while (! offsetf.eof()) {
        offsetf.getline(buf,320);
        sscanf(buf,"%f",&offset);
fAdcOffset[i] = offset;
        i++;

        fOffset++;
    }

    offsetf.close();

    return fOffset;
}

//-----//
//***** Reads the xtalk *****//
//-----//

Int_t SimSort::SetXtalk(const char *xtalkfName)
{
    Int_t rowA=0;
    Int_t colA=0;
    string lineA;
    Double_t x;
    // char buf[36][36];
    xtkarray[40][40]=0;
    Double_t fileIN;
    //string filename;
    ifstream fileINf;
    // cin >> filename;
    fileINf.open(xtalkfName);
    cout<<"read number of columns"<<endl;
    if (fileINf.fail()){cerr<< "file you try to open is closed";
    exit(1);
    }
    cout<<"\n"<<endl;
    if (fileINf.is_open()){
        while(getline(fileINf,lineA)){
            istringstream streamA(lineA); // use buf for input
            colA=0;
            while (streamA >> x){
                xtkarray[rowA][colA]=x;
                colA++;
            }
            rowA++;
        }
    }

    cout <<" #number of rows= " << rowA << endl;
    cout <<" # number of columns= " << colA << endl;
    cout << " "<<endl;
    /*for (int i=0; i<rowA; i++){
        for (int j=0; j<colA; j++){
            cout<< setw(6)<<xtkarray[i][j]<< " ";
        }
    }

```

```

cout<<endl;
}*/

//cout<<"value= "<<xtkarray[3][0]<<endl;
return 0;
}

//-----//
////////////////////////////////////////////////////////////////////
//***** Reads the Gain for drift correction *****
////////////////////////////////////////////////////////////////////
Int_t SimSort::GainDrift(const char *gainfName)
{
    Int_t rowG=0;
    Int_t colG=0;
    string lineA;
    Double_t x;
    // char buf[36][36];
    GDGain[40][40]=0;
    Double_t fileIN;
    //string filename;
    ifstream fileINf;
    // cin >> filename;
    fileINf.open(gainfName);
    cout<<"columns and rows Gaindrift coef"<<endl;
    if (fileINf.fail()){cerr<< "file you try to open is closed";
    exit(1);
    }
    cout<<"\n"<<endl;
    if (fileINf.is_open()){
        while(getline(fileINf,lineA)){
            istringstream streamA(lineA); // use buf for input
            colG=0;
            while (streamA >> x){
                GDGain[rowG][colG]=x;
                colG++;
            }
            rowG++;
        }
    }

    cout << " #number of rows= " << rowG << endl;
    cout << " # number of columns= " << colG << endl;
    cout << " "<<endl;

    cout<<"value= "<<GDGain[0][0]<<endl;
    return 0;
}

//-----//
////////////////////////////////////////////////////////////////////
//***** Reads the offset for drift correction *****
////////////////////////////////////////////////////////////////////
Int_t SimSort::OffsetDrift(const char *offsetfName)
{
    Int_t rowA=0;
    Int_t colA=0;
    string lineA;
    Double_t x;
    // char buf[36][36];
    GDOffset[40][40]=0;
    Double_t fileIN;
    //string filename;
    ifstream fileINf;
    // cin >> filename;
    fileINf.open(offsetfName);
    cout<<"columns and rows offset coef"<<endl;
    if (fileINf.fail()){cerr<< "file you try to open is closed";
    exit(1);
    }
    cout<<"\n"<<endl;

```

```

if (fileINf.is_open()){
while(getline(fileINf,lineA)){
istringstream streamA(lineA); // use buf for input
colA=0;
while (streamA >> x){
GDOffset[rowA][colA]=x;
colA++;
}
rowA++;
}
}

cout << " #number of rows= " << rowA << endl;
cout << " # number of columns= " << colA << endl;
cout << " "<<endl;
/*for (int i=0; i<rowA; i++){
for (int j=0; j<colA; j++){
cout<< setw(6)<<GDOffset[i][j]<< " ";
}
cout<<endl;
}*/

cout<<"value= "<<GDOffset[0][0]<<endl;
return 0;
}

//-----//
//-----//
//***** Initialise all the tables (calibration...) *****//
//-----//

SimSort::SimSort()
{}

SimSort::~SimSort()
{}

//-----//
//-----//
//***** Read data *****//
//-----//

Int_t SimSort::ReadData(Char_t *dataf, Int_t blkSize)
{
    struct addressWord {
        UShort_t data;
        UShort_t group;
        UShort_t item;
        UShort_t type;
    };
    typedef struct addressWord Addr_t;
    struct contentWord {
        UShort_t data;
    };
    typedef struct contentWord Cont_t;
    // const Int_t blkSize = 7488; // in bytes
    const Int_t nWdChar = blkSize / sizeof(Char_t);
    const Int_t nWdInt = blkSize / sizeof(UShort_t);
    Int_t pbyte;
    Char_t bytes[blkSize];
    Char_t *bytesPtr = bytes;

    fRead=1;
    ult=0;

    {

        if (! dataf[0].is_open()) {
            dataf[0].open(dataf,ifstream::binary);

```

```

    if (datafs[0].is_open()) cout << "ReadData: Reading raw data file \""
<< dataf << "\"..."<< endl;
    else cout << "ReadData: Couldn't find raw data file \""
<< dataf << "\"..."<< endl;
    }
    if(datafs[0].eof() == 0) {
        if (datafs[0].read(bytesPtr,nWdChar)) {
blocks++;
wordPtr = (UShort_t *) bytesPtr;
//cout << "nuword="<< nWdInt << endl;
pbyte = datafs[0].tellg();
//cout << "number of bytes " << pbyte << endl;

for (j=0; j < nWdInt; j++){ //cout<< j << endl;
    if(endian) wordPtr2[j]=Rbswap_16(wordPtr[j]);
    //this should replace the if above, have to check that endian business
    else wordPtr2[j]=wordPtr[j];
}

words=nWdInt;

    }
    else {
        fRead=0;
        eof=0;
        datafs[0].close();
    }
if((blocks-check)>=0) {
cout << "Block read: " << blocks<< "[1A" << endl;
check=check+stepblock;
}
}
else {
    datafs[0].close();
    cout << " DUCH ! " << endl;
}
}

return fRead;
}

////////////////////////////////////
/***** GAIN MATCHING *****/
////////////////////////////////////
Int_t SimSort::GainMatch(Int_t crysID, Double_t EEnergy, Int_t ID1, Int_t Seg, Int_t row)
{ Int_t nc=1;
  Int_t x = 7;
  Int_t modn;

      Int_t linen;          //line for the gm coefficients
crysID = crysID - 33312;
// modn = Int_t(crysID/16);
    modn = ID1;
Int_t Colm = (modn*9 )+ Seg;
    if (Seg == 0) {linen = ID1 * 16 ;}
    else {linen = ID1*16 + Seg + 3;}
Energydrift[modn][Seg] = GDGain[row][Colm]*(EEnergy+(rand()
/((double)RAND_MAX+1)))+ GDOffset[row][Colm];
//cout<< ID1 << " " << Colm << " " << GDGain[row][Colm] << " "
<< GDOffset[row][Colm]<< endl;
Energy[modn][Seg] = fAdcGain[linen]*(Energydrift[modn][Seg]
+(rand()/((double)RAND_MAX+1)))+ fAdcOffset[linen];
return 0;
}

////////////////////////////////////
/*****Xtalk correction*****/
////////////////////////////////////
Int_t SimSort::CrossTalk(Int_t m, Int_t i)
{ Int_t mm,xx,xc,kk;
ofstream info;
info.open("infoaddxtalk.txt",ios::app);

```

```

Int_t Ethx=0; //cout<< "module= "<< m << " hitt seg= "
<< i << endl; for (j=2; j<10; j++){seghit[j]=0;} //cout << " m= "<< m
<< " i= "<< i << " Intial energy= " << Energy[m][i]<< endl; hittt++;
//Int_t Ej=-50;

Energyxyc[m][i]=0;
// if (Energy[m][i]>Ethx){
xr = (m*8)+i-1;
for (mm=0; mm<4; mm++){
    for (Int_t j=1; j<9; j++){ xc=(mm*8)+j-1;
    if (mm != m && Energy[mm][j]>Ethx){
        Energyxyc[m][i] = (Energyxyc[m][i]+(Energy[mm][j]*xtkarray[xr][xc]));
        //inverted cross-talk matrix;
    }
    else if (mm == m) {Energyxyc[m][i] =
        (Energyxyc[m][i]+(Energy[mm][j]*xtkarray[xr][xc]));}
        //inverted cross-talk matrix;
    } //if (nuhit==2 && Energy[0][6]>Eth && Energy[0][8]>Eth){ cout
    << "I am here "<< hit<< endl; Edbl75[0][3]= Energyxyc[0][6]+Energyxyc[0][8];}
    }
    // info << "m= " << m << " i = " << i << " Energy= "
    << Energy[m][i] << " Energyxyc= "
    << Energyxyc[m][i] << " " << xtkarray[xr][1] << endl;

    // }

return 0;
}
////////////////////////////////////
/***** Filling histogram/spectrum with raw data *****/
////////////////////////////////////

Int_t SimSort::CalibTree(Char_t *fName)
{
    Int_t ii,jj,x,k,gain;
    UShort_t ID;

    TFile f1("Calib.root","recreate");

    ofstream info;
    info.open("infoadd.txt",ios::app);

    datafs[0].open(fName,ifstream::binary);

    Char_t specname[128];
    TH1D *spec[4][10];
    gain = 1;

    cout << " do you want to read earlier calibration "<< endl ;
    cout << " if yes, type 1 " << endl ;
    cin >> gain ;

    for(k=0; k<4; k++){
        for(ii=0; ii<10; ii++){
            sprintf(specname,"Crystal%dsegment%d",k+1,ii);
            spec[k][ii] = new TH1D(specname,specname,10000,-100,9900);
        }
    }

    for(ii=0; ii<100; ii++){
        //while (!datafs[0].eof()){
        ReadData(fName,32);

        x=7;
        ID = wordPtr2[0];
        crystalID = wordPtr2[0];
        if (wordPtr2[6]>32000){Energy[0][0]=-32767+wordPtr2[6];}
        else {Energy[0][0]=wordPtr2[6];}
        //Energy[0][0]= wordPtr2[6];
    }
}

```

```

if(ID == 33312){jj=0;k=0;} if(ID > 33312 && ID < 33314){jj=ID-33313+1;k=0;}
if(ID > 33313 && ID < 33328){jj=ID-33316+2;k=0;}
if(ID == 33328){jj=0;k=1;} if(ID > 33328 && ID < 33330){jj=ID-33329+1;k=1;}
if(ID > 33329 && ID < 33344){jj=ID-33332+2;k=1;}
if(ID == 33344){jj=0;k=2;} if(ID > 33344 && ID < 33360){jj=ID-33345+1;k=2;}
if(ID > 33345 && ID < 33360){jj=ID-33348+2;k=2;}
if(ID == 33360){jj=0;k=3;} if(ID > 33360 && ID < 33362){jj=ID-33361+1;k=3;}
if(ID > 33361){jj=ID-33364+2;k=3;}

if(gain){GainMatch(crystalID,Energy[0][0],0);}
spec[k][jj]->Fill(Energy[0][0]);}

f1.Write();
f1.Close();
info.close();
return 0;
}

///////////////////////////////////////////////////
//***** Convert XIA raw data format *****/
///////////////////////////////////////////////////

Int_t SimSort::Tree(Char_t *fName)
{
typedef struct contentWord Cont_t;

Int_t i,ii,iii,k,kk,dd,m,jj,tt, Ethhit=20;
Int_t row;
ULong64_t l=0;
// Multi Root files ///////////////////////////////////
Int_t fileID = 0;
Long64_t bytesMax = 504857600; /*500MB*/
TFile *f = 0;
TTree *t1 = 0;
typedef struct contentWord Cont_t;
//const char *outputPrefix = "test";
const char *outputPrefix = "Co60_23Aug17_4mod_new";
Long64_t eventID = 0;
f = TFile::Open(TString::Format("%s_%d.root", outputPrefix, fileID++).Data(),"RECREATE");
t1=new TTree("t1","a simple Tree");
////////////////////////////////////

/////define the tree
//TFile f1("Bi207_11oct16_4mod.root","recreate");
//TTree *t1=new TTree("t1","a simple Tree");
//t1->Branch("eventenergy",&eventenergy,"eventenergy/s");
//t1->Branch("tracelength",&tracelength,"tracelength/s");
//t1->Branch("timest",&timest,"timest[32]/1");
t1->Branch("TimeStamp",&TimeStamp,"TimeStamp[4][10]/1");
//t1->Branch("hit",&hit,"hit/i");
t1->Branch("Energy",Energy,"Energy[4][10]/D");
t1->Branch("Energyxtc",Energyxtc,"Energyxtc[4][10]/D");
//t1->Branch("REnergy",REnergy,"REnergy[4]/D");
//t1->Branch("REnergync",REnergync,"REnergync[4]/D");
//t1->Branch("Edb157",Edb157,"Edb157[4][10]/D");
//t1->Branch("Edb175",Edb175,"Edb175[4][10]/D");
//t1->Branch("TEnergy",&TEnergy,"TEnergy/D");
//t1->Branch("Etot",&Etot,"Etot[9]/D");
//t1->Branch("Etott",&Etott,"Etott/D");
//t1->Branch("DetID",DetID,"DetID[4][10]/s");
//t1->Branch("trace",trace,"trace[4][10][400]/s");
//t1->Branch("CFDtime",CFDtime,"CFDtime[4][10]/s");
////////////////////////////////////
//t1->Branch("EnergySeg",&EnergySeg,"EnergySeg[4][10]/D");
//t1->Branch("REnergySeg",&REnergySeg,"REnergySeg[4][10]/D");
//t1->Branch("TSEnergy",&TSEnergy,"TSEnergy[4]/D");
//t1->Branch("TSEnergync",&TSEnergync,"TSEnergync[4]/D");
t1->Branch("hittre_s",&hittre_s,"hittre_s[4]/i");
t1->Branch("hittre_c",&hittre_c,"hittre_c[4]/i");
//t1->Branch("seghit",&seghit,"seghit[4][8]/s");

```

```

//t1->Branch("CoreEnergy",&CoreEnergy,"CoreEnergy[4]/D");

//////////

timeref = 100;

datafs[0].open(fName,ifstream::binary);
for(k=0;k<4;k++){
    timest[k]=0; TEnergy=0; hittre_c[k]=0; hittre_s[k] = 0;
    for(kk=0;kk<10;kk++){ Energybuf[k][kk]=0;Energy[k][kk]=0;
    crystalbuf[k][kk]=0; timestmpsegbuf[k][kk]=0;
        TimeStmp[k][kk]=0; DetID[k][kk]=0;CFDtimebuf[k][kk]=0;CFDtime[k][kk]=0;}
}
for (m=0; m<4; m++) {for (jj=1;jj<9;jj++){seghit[m][jj]=0;}}
//for(ii=0;ii<43400;ii++){ReadData(fName,7488);}

//for(tt=0;tt<2000000000;tt++){//dd++;
//for(tt=0;tt<100;tt++){
while (!datafs[0].eof()){
    l++;
    ReadData(fName,32);
    row = Int_t(1/500000000);

        if ((wordPtr2[6]<32000) && (wordPtr2[6] > Ethhit))
        {
myPtr2[0][0] = (ULong_t) wordPtr2[2];
myPtr2[0][1] = (ULong_t) wordPtr2[3]; myPtr2[0][1] = myPtr2[0][1] << 16;
myPtr2[0][2] = (ULong_t) wordPtr2[4]; myPtr2[0][2] = myPtr2[0][2] << 32;
Time = myPtr2[0][0] | myPtr2[0][1] | myPtr2[0][2];
// info << "Time " << Time << endl;
if (EVTTIME1 == 0) {EVTTIME1 = Time; deltaTimeStmp = 0;}
        else {deltaTimeStmp = (Time - EVTTIME1);}
        if (deltaTimeStmp*deltaTimeStmp < timeref*timeref)
        {
            channelno=wordPtr2[0] &0x000F;
            slotid =(wordPtr2[0] &0x00F0)>>4;
            mod = slotid - 2;
            if (channelno != 0)
            {
                channelno = channelno - 3;
            }
            TimeStmp[mod][channelno] = Time;
            Energy[mod][channelno] = wordPtr2[6];
            CFDtime[mod][channelno] = wordPtr2[5];
            DetID[mod][channelno] = wordPtr2[0];
            // info << "Raw Data " << DetID[mod][channelno] << " "
            << TimeStmp[mod][channelno] << " " << Energy[mod][channelno] << endl;
            GainMatch(DetID[mod][channelno],Energy[mod][channelno],mod,channelno,row);
            //GainMatch(DetID[mod][channelno],Energy[mod][channelno],mod,channelno);
            // info << " GM Data " << DetID[mod][channelno] << " "
            << TimeStmp[mod][channelno] << " " << Energy[mod][channelno] << endl;
        }
        else {
            for (m=0; m<4; m++) {for (jj=1;jj<9;jj++){
                if (Energy[m][jj] > Ethhit)
                {CrossTalk(m,jj);
                // calculate the xtalk-corrected energy on segmemnt jj in module m
                // info << " Crosstalk energies " << Energyxtc[m][jj] << endl;
                if (Energyxtc[m][jj] > Ethhit)
                { hittre_s[m]++; }
                }
            }
            if (Energy[m][0] > Ethhit)
            {
                Energyxtc[m][0] = Energy[m][0];
                hittre_c[m]++;
            } // no crosstalk correction on the core yet

```

```

    } // finish Xtalk correction for the whole event & calculate # hits

    t1->Fill();    // fill the tree

    ////////////////////////////////// New Multi Root files //////////////////////////////////

    if(f)
    if (f->GetBytesWritten() > bytesMax) {
    f->Write("", TObject::kOverwrite);
    f->Close();
    Printf("Write %lld bytes", f->GetBytesWritten());

    f = TFile::Open(TString::Format("%s_%d.root",
    outputPrefix, fileID++).Data(), "RECREATE");
    t1=new TTree("t1", "a simple Tree");
    //t1->Branch("eventenergy", &eventenergy, "eventenergy/s");
    //t1->Branch("tracelength", &tracelength, "tracelength/s");
    //t1->Branch("timest", &timest, "timest[32]/1");
    t1->Branch("TimeStamp", &TimeStamp, "TimeStamp[4][10]/1");
    //t1->Branch("hit", &hit, "hit/i");
    t1->Branch("Energy", Energy, "Energy[4][10]/D");
    t1->Branch("Energyxtc", Energyxtc, "Energyxtc[4][10]/D");
    //t1->Branch("REnergy", REnergy, "REnergy[4]/D");
    //t1->Branch("REnergync", REnergync, "REnergync[4]/D");
    //t1->Branch("Edb157", Edb157, "Edb157[4][10]/D");
    //t1->Branch("Edb175", Edb175, "Edb175[4][10]/D");
    //t1->Branch("TEnergy", &TEnergy, "TEnergy/D");
    //t1->Branch("Etot", &Etot, "Etot[9]/D");
    //t1->Branch("Etott", &Etott, "Etott/D");
    //t1->Branch("DetID", DetID, "DetID[4][10]/s");
    //t1->Branch("trace", trace, "trace[4][10][400]/s");
    //t1->Branch("CFDtime", CFDtime, "CFDtime[4][10]/s");
    //////////////////////////////////
    //t1->Branch("EnergySeg", &EnergySeg, "EnergySeg[4][10]/D");
    //t1->Branch("REnergySeg", &REnergySeg, "REnergySeg[4][10]/D");
    //t1->Branch("TSEnergy", &TSEnergy, "TSEnergy[4]/D");
    //t1->Branch("TSEnergync", &TSEnergync, "TSEnergync[4]/D");
    t1->Branch("hittre_s", &hittre_s, "hittre_s[4]/i");
    t1->Branch("hittre_c", &hittre_c, "hittre_c[4]/i");
    //t1->Branch("seghit", &seghit, "seghit[4][8]/s");
    //t1->Branch("CoreEnergy", &CoreEnergy, "CoreEnergy[4]/D");

    }

    //////////////////////////////////
    // cout << " Tree Data " << endl;
    for (i=0; i<4; i++){for(j=0; j<9; j++){
    // cout << DetID[i][j] << " " << TimeStamp[i][j] << " "
    << Energyxtc[i][j] << " " << hittre_s[i] << " " << hittre_c[i] << endl;
    }}
    for (i=0; i<4; i++){ // initialize
    for(j=0; j<9; j++){
    hittre_s[i] = 0;
    hittre_c[i] = 0;
    TimeStamp[i][j] = 0;
    Energy[i][j] = 0;
    Energyxtc[i][j] = 0;
    CFDtime[i][j] = 0;
    DetID[i][j] = 0;
    }}
    channelno=wordPtr2[0] &0x000F;
    // keep the data as the first entry of the next event
    slotid =(wordPtr2[0] &0x00F0)>>4;
    mod = slotid -2;
    if (channelno != 0)
    {
    channelno = channelno - 3;
    }
    TimeStamp[mod][channelno] = Time;
    Energy[mod][channelno] = wordPtr2[6];
    CFDtime[mod][channelno] = wordPtr2[5];

```

```

        DetID[mod][channelno] = wordPtr2[0];
//          info << "Raw Data "<< DetID[mod][channelno] << " "
<< TimeStmp[mod][channelno] << " " << Energy[mod][channelno] << endl;
GainMatch(DetID[mod][channelno],Energy[mod][channelno],mod,channelno,row);
        EVTIME1 = Time;
//          info << " GM Data "<< DetID[mod][channelno] << " "
<< TimeStmp[mod][channelno] << " " << Energy[mod][channelno] << endl;

        } // end of else
    }
} //end for on tt

if (f){f->Write("", TObject::kOverwrite);
f->Close();}
return 0;
}

```

Appendix D

Code written by T.D. Bucher

```
#include "TROOT.h"
#include "TFile.h"
#include "TTree.h"
#include "TGraph.h"
#include "TStyle.h"
#include "TChain.h"
#include "TCanvas.h"
#include "TBrowser.h"
#include "Rtypes.h"
#include "TH2.h"
#include "TF1.h"
#include "TRandom.h"
#include "TLegend.h"
#include "TStopwatch.h"
#include "TLinearFitter.h"
#include <iostream>
#include <fstream>
#include <math.h>

int Hist_Ba133_hit2_AngCorMat()
{
    TStopwatch timer;
    timer.Start();
    TDate_t now;
    now.Print();
    Char_t filename[201];
    Char_t filename_out[201];
    Char_t matrice_name[200];
    Char_t matrice_title[200];
    ofstream info;
    ofstream spectra;

    //info.open("Multihistne81_302seg1.txt",ios::app);
    TH1D *Time; Time = new TH1D("Time", "Time",400,300,700);
    TH1D *TimeGated; TimeGated = new TH1D("TimeGated", "TimeGated",7500,0,7500);
    //TH3D *E3D; E3D = new TH3D("E-E-T","E-E-T",2000,0,2000,2000,0,2000,400,300,700);

    TH2D *E2D[36];
    Int_t k=31; //the hit segment
    for (int j=0;j<36;j++){
        sprintf(matrice_name,"Energyxyc_2hits_%d_%d",k,j);
        sprintf(matrice_title,"Energyxyc_2hits_%d_%d; Energy (keV); Energy (keV)",k,j);
        cout << matrice_name << " " << matrice_title << "[1A" << endl;
        E2D[j] = new TH2D(matrice_name, matrice_title,2000,0,2000,2000,0,2000);
    } //end j
    cout << "" << endl;

    Int_t i,ii,j,jj,m,mm,numseg,d1,d2;
```

```

ULong_t iii;
Int_t hitt[8],Seghit2[9];
Double_t Ethl=20, Ethh = 32000;
//Int_t hitt[8],i,count[10],h2,h3,hh;
UInt_t hittre_c[4],hittre_s[4];
UShort_t DetID[4][10];
Double_t Energy[4][10],Energyxyc[4][10], Etotc, Etots, Ecs;
ULong_t TimeStmp[4][10];
Double_t tm;
Char_t specname[1734677];
//UShort_t CrystalID[1734677];
// derived class definition
class Histogram {
public:
void Fill(float value);
int GetNbins();
virtual float GetBinContent(int bin);
protected:
float* GetBins() const { return bins; }
private:
int nbins;
float* bins;
};
TChain chain("t1");
for (j=0;j<460;j++){ //741
sprintf(filename,"/media/TIGRESS8TB2/Ba133_20Nov_4mod_%d.root",j);
cout << filename << endl;
chain.Add(filename);
}
cout << " " << endl;

chain.SetBranchStatus("*",0); // disable all the branches
chain.SetBranchStatus("TimeStmp",1);
chain.SetBranchAddress("TimeStmp",&TimeStmp);
chain.SetBranchStatus("Energy",1); // enable one branch
chain.SetBranchAddress("Energy",&Energy);
chain.SetBranchStatus("Energyxyc",1); // enable one branch
chain.SetBranchAddress("Energyxyc",&Energyxyc);
chain.SetBranchStatus("hittre_c",1);
chain.SetBranchAddress("hittre_c",&hittre_c);
chain.SetBranchStatus("hittre_s",1);
chain.SetBranchAddress("hittre_s",&hittre_s);

ULong_t nentries = (ULong_t)chain.GetEntries();
cout << "entries=" << nentries << endl;
for (iii=0;iii<nentries;iii++){
chain.GetEntry(iii);
if (iii%10000 == 0) cout << "Block read: " << iii << "[1A" << endl;

numseg = hittre_s[0] + hittre_s[1] + hittre_s[2] + hittre_s[3];

if(numseg == 2){
mm = k/9; jj = k - 9*mm;
if ((Energyxyc[mm][jj] > Ethl) && (Energyxyc[mm][jj] < Ethh)) {
for (d2 = 0; d2 < 36; d2++){
m = d2/9; j = d2 - 9*m;
if ((d2 != k) && (Energyxyc[m][j] > Ethl) && (Energyxyc[m][j] < Ethh)){
if(((TimeStmp[mm][jj]-TimeStmp[m][j]+500) > 440) &&
((TimeStmp[mm][jj]-TimeStmp[m][j]+500) < 540)){
E2D[d2]->Fill(Energyxyc[mm][jj],Energyxyc[m][j]);
//creates 2D Ang corr matrices
}
Time -> Fill(TimeStmp[mm][jj]-TimeStmp[m][j]+500);
if((Energyxyc[mm][jj]>351)&&(Energyxyc[mm][jj]<361)
&&(Energyxyc[m][j]>76)&&(Energyxyc[m][j]< 86))
{//enter the energies
tm = TimeStmp[mm][jj]-TimeStmp[m][j]+200*d2+300;
TimeGated->Fill(tm-200);
} // end of if(Energy...
```

```

} // end of for (d2 =
} //end of if(Energy...
} // end of if numseg
} // end data taking
cout << "" << endl;
cout<<" I am here"<<endl;
//////////Output root file with histograms////////
sprintf(filename_out,"356_81_Ba133Sep2018_AngCorr_d%d_459_tg.root",k);
cout << filename_out << endl;
TFile *f = TFile::Open(filename_out,"recreate");
//////////Plotting//////////
TCanvas *Em = new TCanvas("Ematrix_k_S2","Ematrix_S1_S2",1);
Em->cd();
E2D[2]->Draw();
for (j=0; j<36; j++) {
if (j != k) E2D[j]->Write();
}
TCanvas *Et = new TCanvas("Time_k_s","Time",1);
Et->cd();
Time->Draw();
Time->Write();
TCanvas *Etg = new TCanvas("Time_gated","Time_gated",1);
Etg->cd(1);
TimeGated->Draw();
TimeGated->Write();
//////////Close output file////////
f->Close();
//////////
spectra.close();
info.close();
Printf("Namaste!");
timer.Stop();
timer.Print();
return 0;
}

```

Appendix E

```
int EffArea()
{

    Int_t i,j;
    Int_t BinX1,BinX2,CntX1,CntX2;
    Int_t Area = 0, AreaBg = 0, AreaBgCorr = 0;
    Int_t Pl,Ph,Peak;
    Double_t BgA,BgLeftA, BgRightA, BgB;
    Int_t m,s;
    Char_t infoFileName[200];

    Int_t Ph = -4; Int_t Pl =5;//background gate
    Peak = 223;
    sprintf(infoFileName,"Areas_d1_%d.txt",Peak);
    cout << infoFileName << endl;

    ofstream info;
    info.open(infoFileName,ios::app);
    info << "j   AreaBgCorr   " << endl;

    TCanvas *Ee = new TCanvas("CurrentHist","CurrentHist",1);
    Ee->Divide(6,6);
    //hist->Draw();
    //Get the areas from the Spectra

    cout << "I am here" << endl;

    for (j=0; j< 36; j++){
        m = int(j/9);
        s = j - m * 9;
        cout << m << "   " << s << endl;
        Ee->cd(j+1);
        if (m == 0) {
            if (s == 0) {TH1D *hist(*EnergyxtCoreA);}
            if (s == 1) {TH1D *hist(*EnergyxtSeg1A);}
            if (s == 2) {TH1D *hist(*EnergyxtSeg2A);}
            if (s == 3) {TH1D *hist(*EnergyxtSeg3A);}
            if (s == 4) {TH1D *hist(*EnergyxtSeg4A);}
            if (s == 5) {TH1D *hist(*EnergyxtSeg5A);}
            if (s == 6) {TH1D *hist(*EnergyxtSeg6A);}
            if (s == 7) {TH1D *hist(*EnergyxtSeg7A);}
            if (s == 8) {TH1D *hist(*EnergyxtSeg8A);}
        }
        if (m == 1) {
            if (s == 0) {TH1D *hist(*EnergyxtCoreB);}
            if (s == 1) {TH1D *hist(*EnergyxtSeg1B);}
            if (s == 2) {TH1D *hist(*EnergyxtSeg2B);}
            if (s == 3) {TH1D *hist(*EnergyxtSeg3B);}
```

```

if (s == 4) {TH1D *hist(*EnergyxtSeg4B);}
if (s == 5) {TH1D *hist(*EnergyxtSeg5B);}
if (s == 6) {TH1D *hist(*EnergyxtSeg6B);}
if (s == 7) {TH1D *hist(*EnergyxtSeg7B);}
if (s == 8) {TH1D *hist(*EnergyxtSeg8B);}
}
if (m == 2) {
if (s == 0) {TH1D *hist(*EnergyxtCoreC);}
if (s == 1) {TH1D *hist(*EnergyxtSeg1C);}
if (s == 2) {TH1D *hist(*EnergyxtSeg2C);}
if (s == 3) {TH1D *hist(*EnergyxtSeg3C);}
if (s == 4) {TH1D *hist(*EnergyxtSeg4C);}
if (s == 5) {TH1D *hist(*EnergyxtSeg5C);}
if (s == 6) {TH1D *hist(*EnergyxtSeg6C);}
if (s == 7) {TH1D *hist(*EnergyxtSeg7C);}
if (s == 8) {TH1D *hist(*EnergyxtSeg8C);}
}
if (m == 3) {
if (s == 0) {TH1D *hist(*EnergyxtCoreD);}
if (s == 1) {TH1D *hist(*EnergyxtSeg1D);}
if (s == 2) {TH1D *hist(*EnergyxtSeg2D);}
if (s == 3) {TH1D *hist(*EnergyxtSeg3D);}
if (s == 4) {TH1D *hist(*EnergyxtSeg4D);}
if (s == 5) {TH1D *hist(*EnergyxtSeg5D);}
if (s == 6) {TH1D *hist(*EnergyxtSeg6D);}
if (s == 7) {TH1D *hist(*EnergyxtSeg7D);}
if (s == 8) {TH1D *hist(*EnergyxtSeg8D);}
}
//cout << "I'll draw" << endl;
hist->Draw();
cout << "I'll calculate the area" << endl;
double parameter[50][50], error[50][50], Peak_Area, Error_Peak_area;
double height=-99; double current_height=0;
for(int count=Peak-5; count<Peak+5; count++){

if(current_height>height){
height=current_height;
current_height=hist->GetBinContent(count);
}

}
cout<<current_height<<endl;
TF1 *fit1=new TF1("fit1","gaus(0)+pol1(3)",0,1000);
fit1->SetParameters(current_height,Peak,3,0,0);
hist->Fit("fit1","", "",Peak+Ph,Peak+Pl);
for(int u=0; u<3; u++){
parameter[1][u]=fit1->GetParameter(u);
error[1][u]=fit1->GetParError(u);
cout<< parameter[1][u] << " " << error[1][u] <<endl;
}
Peak_Area = parameter[1][0]*parameter[1][2]*sqrt(2*3.141595);
Error_Peak_area=Peak_Area*sqrt((error[1][0]/parameter[1][0])
*(error[1][0]/parameter[1][0]) + (error[1][2]/parameter[1][2])
*(error[1][2]/parameter[1][2]));
cout<<"Peak Area = "<< Peak_Area <<endl;
info << Peak_Area<<endl;
} //end for(j
info.close();
}

```

Appendix F

```
#include "TGraphErrors.h"
#include "TCanvas.h"
#include "TH1D.h"
#include <iostream>
#include <fstream>
#include <cstring>
#include <sstream>
#include "TROOT.h"
#include "TFile.h"
#include "TTree.h"
#include "TGraph.h"
#include "TStyle.h"
#include "TChain.h"
#include "TCanvas.h"
#include "TBrowser.h"
#include "Rtypes.h"
#include "TH2D.h"
#include "TF1.h"
#include "TRandom.h"
#include "TLegend.h"
#include "TStopwatch.h"
#include "TLinearFitter.h"
#include <math.h>
#include <strstream>
#include <ctype.h>

void SumEfficiencies_Vers2(){

TStopwatch timer;
timer.Start();
TDateTime now;
now.Print();

// Variable declarations
const Int_t n = 122; //Number of angles
const Int_t m = 17; //number of matrices per angle
string x[n][m];
string matrices[1][17];
string s;
Int_t eff_file1[36][2];
Int_t eff_file2[122][2];
Char_t filename[201],filename_out[201], Areas1[200];
Char_t infoFileName[200], matrice_name[20];
long int eff1=0;
long int eff2=0;
long int sumEff=0;
long long int sumAreas[122];
Double_t Angle_Area,Peak,Peak2,dig1_1,dig1_2, dig2_1,dig2_2,P1,Ph,tot_eff_Angle;
Int_t dig1,dig2,dig2i,dig1i,b;

P1 = -10; Ph = 10; //enter width of peak
```

```

Peak = 53; Peak2= 223;
//gamma rays in the cascade of interest in order of emittance

stringstream segment1;
stringstream segment2;
stringstream segment1_1;
stringstream segment2_1;
stringstream segment1_2;
stringstream segment2_2;

ofstream info;
sprintf(infoFileName,"outputNovS_%d_%d.txt",Peak,Peak2); //output textfile
info.open(infoFileName,ios::app);
info << "Angle      "<< "      "<<"Total Efficiency "<<endl;

/////////////////////////Loading the summed matrices/////////////////////////
ifstream in("Angles_arranged.txt");
if(!in) {
    cout << "Cannot open input file!!!" << endl;
}
if(in){
    for(int i = 0; i<n; i++){
        //cout <<"i="<< i << endl;
        for (int j=0; j<m; j++){
            //cout <<"j="<< j << endl;
            in >> x[i][j];
            //cout<<x[i][j]<<endl;
        }
    }
    in.close();
}

/////////////////////////LOADING Eff files/////////////////////////
    sprintf(filename,"Areas_d1_%d.txt",Peak); cout<<filename<<endl;
    ifstream in(filename);
    if(!in) {cout << "Cannot open input file!!!" << endl; }
    if(in){
        for(int g = 0; g<36; g++){
            for (int h=0; h<1; h++){
                in >> eff_file1[g][h];
            }
            //cout<<eff_file1[g][h]<<endl; }
        }
        in.close();

    sprintf(filename,"Areas_d1_%d.txt",Peak2);cout<<filename<<endl;
    ifstream in(filename);
    if(!in) {cout << "Cannot open input file!!!" << endl; }
    if(in){
        for(int gg = 0; gg<36; gg++){
            for (int hh=0; hh<1; hh++){
                in >> eff_file2[gg][hh];
            }
            // cout<<eff_file2[gg][hh]<<endl; } }
        in.close();
    }

    //////////////////////////*****Calculating the efficiencies*****////////////////////////
    for(int ii = 0; ii<122; ii++){
        for (int w=1; w<m; w++){

            if (x[ii][w]!="0"){
                s = x[ii][w];
                int g = s.length();
                for (b=0; b<g; b++){
                    //cout<<b<<endl;
                    strcpy(matrice_name, s.c_str());

                    if(g<20){
                        //////////////////////////detector1////////////////////////
                        segment1 << matrice_name[16];
                        segment1 >> dig1;

                        eff1= eff_file1[dig1][0];

```

```

        cout<<"Detector 1 effiecieny = "<<eff1<<endl;

//////////detector2//////////
segment2 << matrice_name[18];
segment2 >> dig2;
eff2 = eff_file2[dig2][0];
cout<<"Detector 2 efficiency = "<<eff2<<endl;

tot_eff_Angle= eff1 * eff2;

cout <<"Energyxyc_2hits_"<<dig1<<"_"<<dig2<<" = "<<tot_eff_Angle<<endl;

segment1.clear(); segment2.clear();
if(g>20){
//////////detector1//////////
segment1_1 << matrice_name[16];
segment1_1 >> dig1_1;

segment1_2 << matrice_name[17];
segment1_2 >> dig1_2;

dig1 = dig1_1*10 + dig1_2;
eff1 = eff_file1[dig1][0];
cout<<"Detector 1 effiecieny = "<<eff1<<endl;
//////////detector2//////////
segment2_1 << matrice_name[19];
segment2_1 >> dig2_1;

segment2_2 << matrice_name[20];
segment2_2 >> dig2_2;

dig2 = dig2_1*10 + dig2_2;
eff2 = eff_file2[dig2][0];
tot_eff_Angle= eff1 * eff2;
cout<<"Detector 2 efficiency = "<<eff2<<endl;
cout <<"Energyxyc_2hits_"<<dig1<<"_"<<dig2<<" = "<<tot_eff_Angle<<endl;

segment1_1.clear(); segment1_2.clear();
segment2_1.clear(); segment2_2.clear();
}
if(g!=19 && g!=21){ if (isdigit(matrice_name[17])){
//////////detector1//////////
segment1_1 << matrice_name[16];
segment1_1 >> dig1_1;

segment1_2 << matrice_name[17];
segment1_2 >> dig1_2;
dig1 = dig1_1*10 + dig1_2;
eff1 = eff_file1[dig1][0];
cout<<"Detector 1 effiecieny = "<<eff1<<endl;

//////////detector2//////////
segment2 << matrice_name[19];
segment2 >> dig2;
eff2 = eff_file2[dig2][0];
tot_eff_Angle= eff1 * eff2;

cout<<"Detector 2 efficiency = "<<eff2<<endl;
cout <<"Energyxyc_2hits_"<<dig1<<"_"<<dig2<<" = "<<tot_eff_Angle<<endl;

segment1_1.clear(); segment1_2.clear();
segment2.clear();
}
if (!isdigit(matrice_name[17])){
//////////detector1//////////
segment1 << matrice_name[16];
segment1 >> dig1;
eff1=eff_file1[dig1][0];
cout<<"Detector 1 effiecieny = "<<eff1<<endl;
//////////detector2//////////

```

```

    segment2_1 << matrice_name[18];
    segment2_1 >> dig2_1;
    segment2_2 << matrice_name[19];
    segment2_2 >> dig2_2;

    dig2 = dig2_1*10 + dig2_2;
    eff2 = eff_file2[dig2][0];
    tot_eff_Angle= eff1 * eff2;
    cout<<"Detector 2 efficiency = "<<eff2<<endl;
    cout <<"Energyxyc_2hits_"<<dig1<<"_"<<dig2<<" = "<<tot_eff_Angle<<endl;
segment1.clear();
segment2_1.clear(); segment2_2.clear();

}
}
sumAreas[ii]+=tot_eff_Angle[w];
cout<<"Efficiency for angle "<< x[ii][0]<<" "<<sumAreas[ii]<<endl;
}
}info << x[ii][0]<<" "<<sumAreas[ii]<<endl;
}

Printf("Done!");
timer.Stop();
timer.Print();
}

```

Appendix G

Code written by T.D. Bucher

```
#include "TROOT.h"
#include "TFile.h"
#include "TTree.h"
#include "TGraph.h"
#include "TStyle.h"
#include "TChain.h"
#include "TCanvas.h"
#include "TBrowser.h"
#include "Rtypes.h"
#include "TH2.h"
#include "TF1.h"
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include "TRandom.h"
#include "TLegend.h"
#include "TStopwatch.h"
#include "TLinearFitter.h"
#include <iostream>
#include <fstream>
#include <math.h>

void correct_bg(){
TStopwatch timer;
timer.Start();
TDateTime now;
now.Print();

Char_t filename[125],filename1[125],filename2[125],filename5[125];
Char_t filename3[125],filename4[125],filename6[125];
Char_t filename8[125],filename9[125],filename10[125],filename11[125];
Char_t filename_out[125],filename_out1[125],filename_out6[125];
Char_t filename_out2[125],filename_out5[125];
Char_t dirfile_name[125],filename_out3[125],filename_out4[125];
Char_t dirfile_name4[125],filename_out8[125];
Char_t filename_out9[125],filename_out10[125],filename_out11[125];
Int_t G1 = -12; Int_t Gh = 12; //enter width of gate
Int_t Gate = 303; //enter energy of gate
Int_t P1 = -12; Int_t Ph = 12; //enter width of peak
Int_t B1 = -12; Int_t Bh = 12; //enter width of background
Double_t Area = 0, AreaBg = 0, AreaBgCorrgs = 0,AreaBgCorrds = 0;
Double_t AreaBgCo = 0,AreaBgNo = 0, BgLeftA, BgRightA, sum, Bufbin,newbin;
Double_t BinX1, BinX2, BgA, BgB, bggatelf, bggatert, bgw, Coefbg, Coefbg_squared;
Double_t coefficient=0;
Char_t file_name[201];
Int_t Peak=356;//random peak
Int_t Peak2=81;//coincident peak
bggatelf = (Gate+G1);
bggatert = (Gate+Gh);
```

```

bgw=12;
Coefbg=(Gh-G1+1)/((bggatelf-(bggatelf-bgw)+1)+(bggatert+bgw)-bggatert)+1);
cout<<"Bg Coeff = "<<Coefbg<<endl;
Coefbg_squared=Coefbg*Coefbg;
float Coef;
TH1D *gated_hist[500];
TH1D *dir_hist_scale[500];
TH1D *BG_gated_hist[500];
TH1D *Bgleft[500];          //background on the left of peak
TH1D *Bgright[500];         //background on the right of peak
TH1D *Bgsubgated[500];      // Total background
TH1D *BG_gated_hist1[500];  // background under the peak removed
TH1D *BgErrorSpec[500];
TH1D *Bgleft_shifted[500];
TH1D *Bgright_shifted[500];
TH1D *BgCompSc[500];
TH1D *BgUnderPeak[500];
Char_t infoFileName[100];
Char_t infoErrorName[100];
////////////////////////////////////
ofstream info;
sprintf(infoFileName,"Areas_%d_%d.txt",Gate,Peak2);
info.open(infoFileName,ios::app);
ofstream infoError;
sprintf(infoErrorName,"Error_areas%d_%d.txt",Gate,Peak2);
infoError.open(infoErrorName,ios::app);
infoError << "Angle# " << " ErrorSpec Area" <<endl;
//info <<"Angle#"<<" Peak Area " << " Error " << " p0 "
<<" p0Error "<<" p2 "<<"p2Error"<<endl;
info <<"Angle#"<<" Peak Area " <<endl;
TFile *output = TFile::Open("303_81.root","recreate"); //Final_output_10Oct_303_81
TFile h ("summed_matrix_10Oct.root");// Angular correlation gated spectra

for(int z=0;z<122;z++){
sprintf(filename,"dir_hist_scale_%d",z);
sprintf(filename_out,"dir_hist_scale_%d; Counts; Energy (keV)",z);
dir_hist_scale[z]= new TH1D(filename,filename_out,2000,0,2000);

sprintf(filename1,"Final_BG_hist_%d",z);
sprintf(filename_out1,"Final_Bg_corr_hist_%d; Counts; Energy (keV)",z);
BG_gated_hist[z]= new TH1D(filename1,filename_out1,2000,0,2000);
sprintf(filename2,"gated_hist_%d",z);
sprintf(filename_out2,"gated_hist_%d; Counts; Energy (keV)",z);
gated_hist[z]= new TH1D(filename2,filename_out2,2000,0,2000);

sprintf(filename3,"Backgroundl_%d",z);
sprintf(filename_out3,"Backgroundl_%d; Counts; Energy (keV)",z);
Bgleft[z]= new TH1D(filename3,filename_out3,2000,0,2000);
sprintf(filename8,"Bgr_shifted_%d",z);
sprintf(filename_out8,"Bgr_shifted_%d; Counts; Energy (keV)",z);
Bgleft_shifted[z] = new TH1D(filename8,filename_out8,2000,0,2000);
sprintf(filename4,"Backgroundr_%d",z);
sprintf(filename_out4,"Backgroundr_%d; Counts; Energy (keV)",z);
Bgright[z]= new TH1D(filename4,filename_out4,2000,0,2000);
sprintf(filename9,"Bgl_shifted_%d",z);
sprintf(filename_out9,"Bgl_shifted_%d; Counts; Energy (keV)",z);
Bgright_shifted[z] = new TH1D(filename9,filename_out9,2000,0,2000);
sprintf(filename10,"Total_Bg_Substr_%d",z);
sprintf(filename_out10,"Total_Bg_Substr_%d; Counts; Energy (keV)",z);
BgCompSc[z]= new TH1D(filename10,filename_out10,2000,0,2000);
sprintf(filename5,"Background_Under_peak%d",z); //
sprintf(filename_out5,"Backgroundh_%d; Counts; Energy (keV)",z);
BG_gated_hist1[z]= new TH1D(filename5,filename_out5,2000,0,2000);
sprintf(filename6,"ErrorSpectra_%d",z);
sprintf(filename_out6,"ErrorSpectra_%d; Counts; Energy (keV)",z);
BgErrorSpec[z]= new TH1D(filename6,filename_out6,2000,0,2000);
sprintf(filename11,"BgUnderPeak_%d",z);
sprintf(filename_out11,"BgUnderPeak_%d; Counts; Energy (keV)",z);
BgUnderPeak[z]= new TH1D(filename11,filename_out11,2000,0,2000);
}

```

```

for (int ii=0;ii<122;ii++){
sprintf(file_name,"Summed_Matrix_%d",ii); cout<<file_name<<endl;
TH2D *hist2D = (TH2D*)h.Get(file_name);
gated_hist[ii] = hist2D->ProjectionY(Form("Gated_spec_%d",ii),Gate+G1,Gate+Gh);
Bgleft[ii]= hist2D->ProjectionY(Form("Bgleft_%d",ii),bggatelf-bgw,bggatelf);
Bgright[ii]=hist2D->ProjectionY(Form("Bgright_%d",ii),bggatert, bggatert+bgw);
Bgsubgated[ii]= (TH1D*)Bgleft[ii]->Clone(Form("TotalBgSpec_%d",ii)); //need to look at
Bgsubgated[ii]->Add(Bgright[ii]); //total background spectrum
//cout <<" Bg Left low= " << bggatelf-bgw << " Bg Left high= " << bggatelf << endl;
//cout <<" Bg Right low= " << bggatert<< " Bg Right high= " << bggatert+bgw << endl;
Bgsubgated[ii]->Scale(Coeffbg);
BG_gated_hist1[ii]=(TH1D*)gated_hist[ii]->Clone(Form("Total_Bg_Substr_%d",ii));
//BG_gated_hist1[ii]->Add(Bgsubgated[ii],-1);
BG_gated_hist[ii]=(TH1D*)BG_gated_hist1[ii]
->Clone(Form("BgUnderPeakSubstra_spec_%d",ii));
////////////////////shifting to remove compton scattering //////////////////
/*
//double kright=Peak2/(Peak2-bgw-4); cout<<kright<<endl;
//double kleft=Peak2/(Peak2+bgw+4); cout<<kleft<<endl;
for (Int_t q=174;q<=204;q++){
Bgleft_shifted[ii]->SetBinContent(q-bgw,(Bgleft[ii]->GetBinContent(q)));
//Bgright_shifted[ii]->Scale(kright);
for (Int_t q=118;q<=148;q++){Bgright_shifted[ii]
->SetBinContent(q+bgw,(Bgright[ii]->GetBinContent(q)*1);} // shift left
//Bgleft_shifted[ii]->Scale(kleft);
//for (Int_t q=78;q<=86;q++){
//BgUnderPeak[ii]->SetBinContent(q,(Bgright[ii]->GetBinContent(q)));
//BgUnderPeak[ii]->Scale(2*kleft);
BgCompSc[ii]= (TH1D*)Bgleft_shifted[ii]->Clone(Form("Total_Comp_ScBg_%d",ii));
BgCompSc[ii]->Add(Bgright_shifted[ii]);
//BgCompSc[ii]->Add(BgUnderPeak[ii]);
//BG_gated_hist1[ii]=(TH1D*)gated_hist[ii]->Clone(Form("Total_Bg_Substr_%d",ii));
BG_gated_hist1[ii]->Add(BgCompSc[ii],-1);
*/
////////////////////
BinX1 = (Peak + P1);
BinX2 = (Peak + Ph);
Area = 0; AreaBg = 0; AreaBgCorrgrs = 0;
BgA = (BG_gated_hist1[ii]->GetBinContent(BinX2)-BG_gated_hist[ii]
->GetBinContent(BinX1))/(BinX2-BinX1);
BgB = BG_gated_hist1[ii]->GetBinContent(BinX1)-BgA * BinX1;

for (int d = BinX1; d <= BinX2; d++) {
Area = Area + BG_gated_hist1[ii]->GetBinContent(d);
AreaBg = AreaBg + BgA * d + BgB;
AreaBgCorrgrs = Area - AreaBg;}
cout << "Area of " <<Peak<< " background subtracted gated spectrum = "
<< AreaBgCorrgrs << endl;
////////////////////
TFile *f=new TFile("Summed_Direct_specs.root");// direct spectra
f->cd();
sprintf(dirfile_name,"Summed_direct_%d",ii);
cout<<dirfile_name<<endl;
TH1D *dir_hist = (TH1D*)f->Get(dirfile_name);
BinX1 = (Peak + P1);
BinX2 =( Peak + Ph);

BgLeftA = 0; BgRightA = 0;
for (int c = BinX1+B1; c <= BinX1; c++) {
BgLeftA = BgLeftA + dir_hist->GetBinContent(c);
// cout << c <<" " << dir_hist->GetBinContent(c) << " " << BgLeftA << endl;
}
BgLeftA = -BgLeftA/B1;
for (int c = BinX2; c <= BinX2+Bh; c++) {
BgRightA = BgRightA + dir_hist->GetBinContent(c);
//cout<<c<<" " <<dir_hist->GetBinContent(c)<<" " <<BgRightA<< endl;
}
BgRightA = BgRightA/Bh;
Area = 0; AreaBg = 0; AreaBgCorrds = 0;
BgA = ( BgRightA - BgLeftA)/(BinX2+Bh/2-BinX1-B1/2);
BgB = BgLeftA - BgA * (BinX1+B1/2);

```

```

for (int a = BinX1; a < BinX2+1; a++) {
Area = Area + dir_hist->GetBinContent(a);
AreaBg = AreaBg + BgA * a + BgB;
AreaBgCorrds = Area - AreaBg;}

cout<<"Area of " <<Peak<< "in direct spectrum = "<<AreaBgCorrds << endl;
//////////
Coef=(AreaBgCorrds/AreaBgCorrds); cout<<"The scaling coefficient is "<<Coef<<endl;
dir_hist_scale[ii]= (TH1D*)dir_hist->Clone(Form("dir_hist_scale_%d",ii));
dir_hist_scale[ii]->Scale(Coef);
BinX1 = (Peak + Pl);
BinX2 = (Peak + Ph);
BgLeftA = 0; BgRightA = 0;
for (int bb = BinX1+B1; bb <= BinX1; bb++) {
BgLeftA = BgLeftA + dir_hist_scale[ii]->GetBinContent(bb);
}
BgLeftA = -BgLeftA/B1;
for (int cc = BinX2; cc <= BinX2+Bh; cc++) {
BgRightA = BgRightA + dir_hist_scale[ii]->GetBinContent(cc);
}
BgRightA = BgRightA/Bh;
Area = 0; AreaBg = 0; AreaBgNo = 0;
BgA = ( BgRightA - BgLeftA)/(BinX2+Bh/2-BinX1-B1/2);
BgB = BgLeftA - BgA * (BinX1+B1/2);

for (int dd = BinX1; dd <= BinX2; dd++) {
Area = Area + dir_hist_scale[ii]->GetBinContent(dd);
AreaBg = AreaBg + BgA * dd + BgB;
AreaBgNo = Area - AreaBg;}
cout << "Area of " << Peak << " direct spectra after normalisation = "
<< AreaBgNo << endl;
BG_gated_hist1[ii]->Add(dir_hist_scale[ii],-1);
//////////fitting peak2 area//////////

double parameter[500][500], error[500][500];
double Peak_Area[130], Error_Peak_area[130];
double height=-99; double current_height=0;
for(int count=Peak2-6; count<Peak2+6; count++){
if(current_height>height){
height=current_height;
current_height=BG_gated_hist1[ii]->GetBinContent(count);
}
}
cout<<current_height<<endl;
TF1 *fit1=new TF1("fit1","gaus(0)+pol1(3)",0,10000);
fit1->SetParameters(current_height,Peak2,3,0,0);
BG_gated_hist1[ii]->Fit("fit1","","",Peak2-10,Peak2+10);
for(int u=0; u<3; u++){
parameter[1][u]=fit1->GetParameter(u);
error[1][u]=fit1->GetParError(u);
cout<< parameter[1][u] << " " << error[1][u] <<endl;
}
Peak_Area[ii] = parameter[1][0]*parameter[1][2]*sqrt(2*3.141595);
Error_Peak_area[ii]=Peak_Area[ii]*sqrt((error[1][0]/parameter[1][0])
*(error[1][0]/parameter[1][0])
+(error[1][2]/parameter[1][2])*(error[1][2]/parameter[1][2]));
cout<<"Peak Area = "<< Peak_Area[ii] <<endl;
info << Peak_Area[ii] <<" "<< Error_Peak_area[ii]<<endl;
output->cd();
gated_hist[ii]->Write();
Bgleft[ii]->Write();
Bgright[ii]->Write();
//BgUnderPeak[ii]->Write();
//Bgleft_shifted[ii]->Write();
//Bgright_shifted[ii]->Write();
//BgCompSc[ii]->Write();
Bgsubgated[ii]->Write();
dir_hist_scale[ii]->Write();
BG_gated_hist[ii]->Write(); //Background under the peak subtracted gated spectra

```

```
BG_gated_hist1[ii]~>Write(); //final background subtracted hist
}
output->Close();
Printf("Done!");
timer.Stop();
timer.Print();
}
```

Appendix H

```
#include "TGraphErrors.h"
#include "TCanvas.h"
#include "TH1D.h"
#include <iostream>
#include <fstream>

void Fit_acf(){
const Int_t n = 27; // number of data points
Double_t x[n] = {0};
Double_t y[n] = {0.0};
Double_t ex[n] = {0.0};
Double_t ey[n] = {0.0};

ifstream in("cobalt.txt"); // name of the input file
if(!in) {
    cout << "Cannot open input file!!!" << endl;
}
if(in){
    for(int i = 0; i<n ; i++){
        in >> x[i] >> y[i] >> ey[i];
    }
    in.close();
}

for(int i = 0; i<n ; i++){
cout << x[i] << ", " << " " << y[i] << ", " << " " << ey[i] << ", " << " " << endl;
}

//Plot Satellite Charge
// TCanvas *c1 = new TCanvas("c1","SatellitechargeB1_1stater",200,10,700,500);
TGraphErrors *gr = new TGraphErrors(n,x,y,NULL,ey);
gr->SetTitle("SatellitechargeB1");
gr->SetMarkerColor(4);
gr->SetMarkerStyle(21);
gStyle->SetOptTitle(0);
gStyle->SetOptStat(1101);
gStyle->SetOptFit(1);

TCanvas *cVOS1 = new TCanvas("1173_1332","Co front",0,0,600,600);
cVOS1->cd();
TH1 *hVOS1 = (TH1*) gr->GetHistogram();
hVOS1->SetXTitle("Angle (#theta)");
hVOS1->SetYTitle("W(#theta)");
//hVOS1->SetMinimum(0.7);
//hVOS1->SetMaximum(1.2);
gr->Draw("AP");
TF1 *fit = new TF1("fit","[0]*(1+ 0.5*[1]*(3*cos(x*(3.14159265359/180))
^2-1)+0.125*[2]*(35*cos(x*(3.14159265359/180))^4-30*cos(x*(3.14159265359/180))^2+3))",0,90);
fit->SetParameters(0.9943,0.034,-0.009); // initial values for (a0,a2,a4)
gr->Fit(fit, "");
gr->GetXaxis()->SetRangeUser(0, 90);
}
```

References

- [1] D. R. Hamilton, *Phys. Rev.* **58**, 122 (1940).
- [2] E. L. Brady, M. Deutsch, *Phys. Rev.* **78**, 558 (1950).
- [3] D. E. Klema, *Phys. Rev.* **109**, 1652 (1958).
- [4] L. I. Yin, L. Wiedenbeck, *Nucl. Phys.* **54**, 86 (1964).
- [5] M. Roteta, E. Garcia-torano, *Appl. Radia. and Isot.* **49**, 1349 (1998).
- [6] J. Eberth, J. Simpson, *et al.*, *Prog. Part. Nucl. Phys.* **60**, 283 (2008).
- [7] I. Y. Lee, R. M. Clark, *et al.*, *Nucl. Phys. A* **746**, 255 (2004).
- [8] D. Habs, *et al.*, *Prog. Part. Nucl. Phys.* **38**, 111 (1997).
- [9] J. Simpson, F. Azaiez, G. deFrance, *et al.*, *Acta Phys. Hung. A.* **12**, 159 (2000).
- [10] S. Akkoyun, A. Algora, B. Alikhani, *et al.*, *Nucl. Instrum. Methods Phys. Res. A* **668**, 26 (2012).
- [11] H. C. Scraggs, C. Pearson, *et al.*, *Nucl. Instrum. Methods Phys. Res. A* **543**, 431 (2005).
- [12] J. Smith, *et al.*, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **922**, 47 (2019).

-
- [13] M. Austin, *APS Division of Nuclear Physics Meeting Abstracts* (2015), p. EA.001.
- [14] A. D. Maclean, Gamma-Gamma Angular Correlation Measurements With GRIFFIN, Master's thesis (2016).
- [15] P. D. Schmelzenbach, The study of ^{150}Sm through the Beta decay of ^{150}Pm , ^{150m}Eu and ^{150g}Eu , Ph.D. thesis (2003).
- [16] K. S. Krane, D. Halliday, *Introductory Nuclear Physics* (John Wiley & Sons, 1987).
- [17] D. Halliday, R. Resnick, J. Walker, *Fundamentals of Physics, Extended ed.* (Wiley, 2015).
- [18] D. J. Griffiths, *Introduction to electrodynamics 3rd ed.* (Printice-Hall, Inc, 1999).
- [19] J. Blatt, V. Weisskopf, *Theoretical Nuclear Physics 4th ed.*, Dover Books on Physics (John Wiley & Sons, 1991).
- [20] J. D. Jackson, *Classical Electrodynamics, 3rd ed.* (John Wiley & Sons, 1962).
- [21] K. S. Krane, R. M. Steffen, R. M. Wheeler, *At. Data Nucl. Data Tables* **11**, 351 (1973).
- [22] A. J. Ferguson, *Angular correlation methods in gamma-ray spectroscopy* (North-Holland Publishing Company-Amsterdam, 1965).
- [23] R. R. Roy, B. P. Nigam, *Nuclear Physics theory and experiment, 1st ed.* (Wiley Eastern Pvt.Ltd, 1996).
- [24] G. Goertzel, *Phys. Rev.* **70**, 897 (1946).
- [25] A. Messiah, *Quantum Mechanics 2nd ed.* (Dover publications, Inc., 1995).
- [26] K. S. Krane, *Tables of coefficients for analysis of angular distribution of gamma radiation from oriented nuclei.*

- [27] S. M. Wong, *Introductory Nuclear Physics 2nd ed.* (Englewood Cliff, 2008).
- [28] T. Yamazaki, *Nucl. Data Sheets Section A* **3**, 1 (1967).
- [29] K. S. Krane, *Phys. Rev. C* **10**, 1197 (1974).
- [30] A. Backlin, B. Fogelberg, *Nucl. Phys A* **138**, 429 (1969).
- [31] K. S. Krane, *At. Data Nucl. Data Tables* **16**, 383 (1975).
- [32] K. S. Krane, *At. Data Nucl. Data Tables* **25**, 29 (1980).
- [33] https://energyeducation.ca/encyclopedia/Band_gap. Accessed: 2017-09-30.
- [34] G. F. Knoll, *Radiation Detection and Measurement, 3rd ed.* (John Wiley & Sons, 2000).
- [35] R. D. Evans, *The Atomic Nucleus* (Tata McGraw-Hill Publishing, 1955), tmh edn.
- [36] N. Zettili, *Quantum Mechanics: Concepts and Applications* (Wiley, 2009).
- [37] <https://sites.google.com/site/puenggphysics/home/unit-iv/Compton-effect>. Accessed: 2017-04-08.
- [38] D. Ashok, F. Thomas, *Introduction To Nuclear And Particle Physics 2nd ed.* (World Scientific Publishing Company, 2003).
- [39] <http://electrons.wikidot.com/pair-production-and-annihilation>. Accessed: 2017-04-08.
- [40] <https://www.xia.com/>. Accessed: 2017-04-08.
- [41] Multi instance data acquisition system, <http://nnsa.dl.ac.uk/MIDAS/>.
- [42] R. Kumar, *ROOT: A Data Analysis and Data Mining Tool from CERN* (2008).

- [43] <https://root.cern.ch/>. Accessed: 2017-03-13.
- [44] S. P. Noncolela, T. D. Bucher, *et al.*, *Acta Phys. Pol. B* **48**, 347 (2017).
- [45] T. D. Bucher, S. P. Noncolela, *et al.*, *Phys. Scr.* **92**, 114004 (2017).
- [46] S. Agostinelli, J. Allison, *et al.*, *Nucl Instrum Methods Phys Res A* **506**, 250 (2003).
- [47] D. Duprez, The ithemba labs clover detector and geant4 simulations (2017). Stellenbosch University.
- [48] J. Allison, K. Amako, *et al.*, *IEEE Transactions on Nuclear Science* **53**, 270 (2006).
- [49] M. A. Schumaker, Measurements and simulation-based optimization of tigress hpge detector array performance, Ph.D. thesis, The University of Guelph, The address of the publisher (1993).
- [50] Cosine theorem. encyclopedia of mathematics., http://www.encyclopediaofmath.org/index.php?title=Cosine_theorem&oldid=13758.
- [51] L. P. Ekstrom, *Comput. Phys. Commun* **32**, 399 (1984).
- [52] National nuclear data center, brookhaven national laboratory, <https://www.nndc.bnl.gov/>. Accessed: 2017-02-20.
- [53] R. T. Newman, J. J. Lawrie, *et al.*, *Balkan Physics Letters* p. 182 (1998).
- [54] E. Browne, J. Tuli, *Nuclear Data Sheets* **114**, 1849 (2013).
- [55] K. S. Krane, J. R. Sites, H. A. Smith, W. A. Steyert, *Nucl. Instr. and Meth.* **169**, 573 (1980).
- [56] Y. Khazov, A. Rodionov, F. Kondev, *Nuclear Data Sheets* **112**, 855 (2011).
- [57] W. H. Trzaska, *Nucl. Instrum. Methods Phys. Res A* **297**, 223 (1990).

- [58] K. Yadav, G. V. Manohar, *Proceedings of the DAE Symp. on Nucl. Phys.* **59**, 288 (2014).