e^+e^- Annihilation

Physics Laboratory 3

Visva Loganathan vloganathan@student.ethz.ch

Teaching Assistant

Giorgia Bonomelli gbonomelli@phys.ethz.ch

Abstract In this experiment, the characteristic signatures of electron–positron annihilation from a ^{22}Na source were investigated using NaI(Tl) scintillators coupled to photomultiplier tubes. An exponential fit to the PMT anode signal yielded a fluorescence decay time of $\tau = (240.4 \pm 1.7)\,\text{ns}$, compatible with literature values for NaI(Tl). The MCA spectrum showed the expected ^{22}Na structures, including the 511 keV and 1275 keV photopeaks, Compton continua and edges, and a low-energy Pb X-ray contribution. Gaussian fits gave photopeak positions of $(511.78 \pm 0.08)\,\text{keV}$ and $(1270.83 \pm 0.32)\,\text{keV}$, with energy resolutions of $(8.48 \pm 0.04)\%$ and $(4.89 \pm 0.06)\%$. The fitted Compton edges, $E_c = (340.8 \pm 0.8)\,\text{keV}$ and $E_c = (1063.5 \pm 1.5)\,\text{keV}$, agree with theoretical expectations. The available coincidence spectrum showed a pronounced peak at $\mu = (510.0 \pm 9.7)\,\text{keV}$, but remaining low-energy and Compton-like contributions indicate that the coincidence selection was not fully effective. Overall, the experiment reproduced the main expected spectral signatures while illustrating limitations from detector resolution, calibration uncertainties, background, and imperfect coincidence selection.

ETH Zürich, May 1, 2026

Visva Loganathan

1 Introduction

The study of electron-positron (e^+e^-) annihilation represents a fundamental exploration of the symmetry between matter and antimatter and serves as a direct experimental verification of mass-energy equivalence. In this process, an electron and its antiparticle, the positron, undergo a total conversion of their rest masses into electromagnetic energy, typically resulting in the emission of two gamma-ray photons. This phenomenon is not only a cornerstone of quantum electrodynamics but also the operational basis for critical modern technologies, most notably Positron Emission Tomography (PET) in medical imaging and Positron Annihilation Spectroscopy (PAS) in materials science [1].

In the laboratory setting, positrons are generated through the β^+ decay of a ^{22}Na radioactive source. Once emitted, these positrons interact with the surrounding medium, losing kinetic energy through various scattering processes until they reach thermal equilibrium. The subsequent annihilation with an electron primarily occurs from a state of low momentum, which, due to the conservation of linear momentum and energy, requires the emission of two photons in exactly opposite directions. Each of these photons carries an energy of approximately 511 keV, corresponding to the rest mass of the electron. A significant aspect of this interaction is the potential formation of Positronium, a short-lived bound state whose decay characteristics depend on the relative spin alignment of the particles.

The primary objective of this experiment is to detect these annihilation photons and analyze their energy distribution using NaI(Tl) inorganic scintillators coupled to photomultiplier tubes. The initial phase of the work focuses on the technical characterization of the detection system, including the determination of the optimal operating voltage for the photomultipliers and the estimation of the fluorescence decay time of the NaI(Tl) crystal from the anode signal waveforms. By calibrating the resulting energy spectra, the various mechanisms of photon interaction with matter, specifically the photoelectric effect and Compton scattering, can be identified and analyzed in detail.

Furthermore, the experiment investigates the use of coincidence logic between two separate detectors. In the intended coincidence configuration, discriminators and a coincidence unit are used to select events in which both detectors register signals within a short time window. If correctly implemented, this suppresses random background and preferentially selects events associated with the two back-to-back 511 keV annihilation photons. In the present measurement, the available coincidence spectrum is therefore used to study both the expected enhancement of the annihilation peak and the practical limitations of an imperfect coincidence selection.

2 Experimental Setup

The experimental configuration is designed to detect and process the collinear 511 keV gamma rays resulting from positron annihilation. The physical arrangement of the apparatus, shown in Figure 1, integrates a ^{22}Na radioactive source, two scintillation detectors, and a dedicated chain of nuclear electronics for signal processing and coincidence logic.

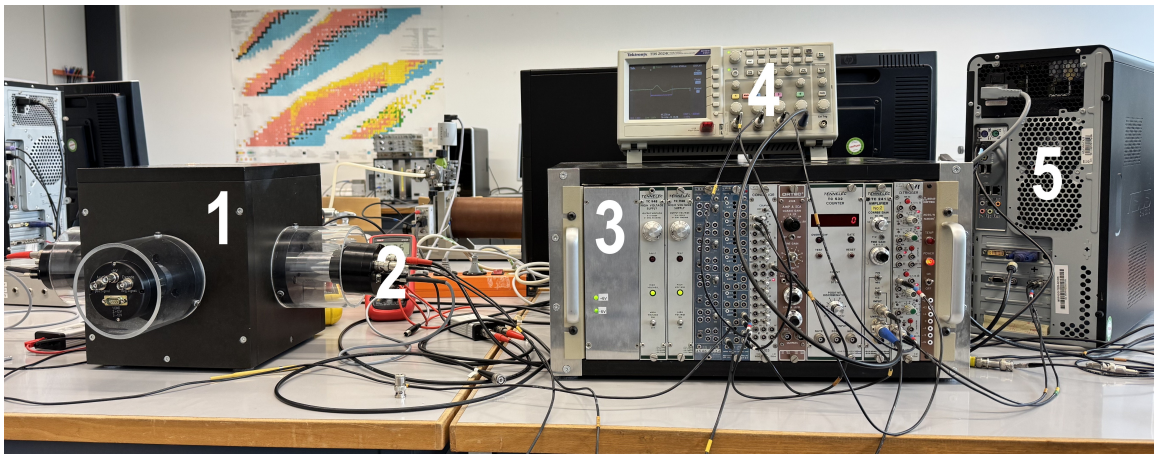

Figure 1: Labeled photograph of the laboratory setup. (1) NaI(Tl) scintillator crystal and Photomultiplier Tube (PMT) housing. (2) Voltage Meter (3) NIM (Nuclear Instrument Module) Crate containing the electronic processing chain. (4) Digital oscilloscope for signal verification. (5) Data acquisition PC acting as a Multi-Channel Analyzer (MCA).

Detectors and Physical Arrangement

The primary radiation source is a ^{22}Na isotope, which undergoes β^+ decay, emitting positrons that subsequently annihilate within the source holder. To detect the resulting photons, two thallium-doped sodium iodide (NaI(Tl)) inorganic scintillators are employed (Figure 1, Label 1). These crystals are chosen for their high light yield and stopping power in the sub-MeV range. Each scintillator is optically coupled to a Photomultiplier Tube (PMT), which converts the scintillation light into electrical pulses.

The PMTs require a stable high-voltage supply (Figure 1, Label 2), which is adjusted for each tube to achieve an optimal gain-to-noise ratio, typically determined through the measurement of counting curves.

Electronic Signal Processing

The electronic chain, housed within the NIM crate (Figure 1, Label 3), is split into two main branches to handle energy spectroscopy and timing logic simultaneously. The signals are extracted from both the anode and the dynode of the PMTs. The anode signal, characterized by its fast rise time, is used for timing and coincidence measurements. It is processed through a Timing Filter Amplifier (TFA) and a Leading Edge Discriminator

to generate standardized logic pulses. These are fed into a coincidence module to identify simultaneous events within a narrow time window ($\Delta\tau$). The resulting logic and raw pulse shapes are monitored using the digital oscilloscope (Figure 1, Label 4).

In parallel, the dynode signals are utilized for energy reconstruction. These pulses are sent to a pre-amplifier and a linear shaping amplifier to produce Gaussian-like pulses with amplitudes proportional to the deposited energy. These shaped signals are processed by an Analog-to-Digital Converter (ADC) and recorded by the data acquisition PC (Figure 1, Label 5). In the intended coincidence configuration, the output of the coincidence unit is used as a gate signal for the ADC, so that only events accompanied by simultaneous signals in both scintillators are recorded. If correctly implemented, this configuration suppresses random background and preferentially selects the 511 keV annihilation photons.

3 Results & Discussion

PMT Signal Shape and Decay Time

As a first step, the pulse shapes of the photomultiplier tube (PMT) were investigated with the oscilloscope. Both the dynode and the anode output were recorded. A typical PMT pulse exhibits a very fast rising edge followed by a slower decay. Figure 2 shows a qualitative oscilloscope trace of the dynode signal in persistence mode. The pulse shape is consistent with the expected response of a NaI(Tl) scintillation detector coupled to a PMT.

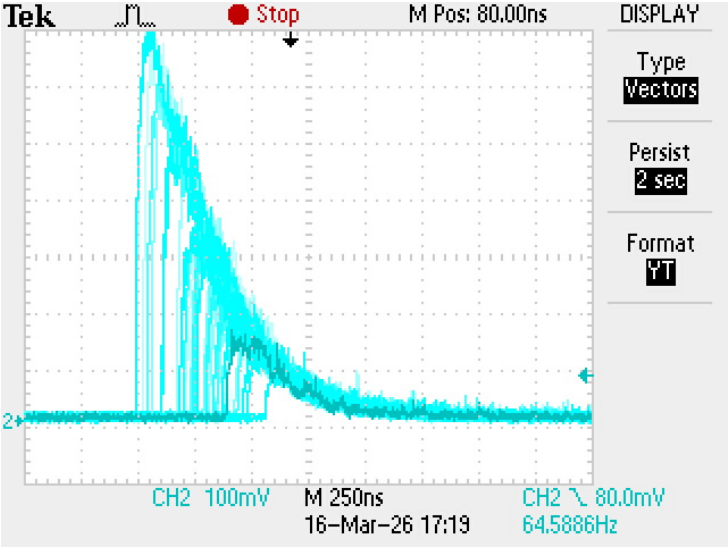

Figure 2: Qualitative oscilloscope trace of the PMT dynode signal in persistence mode. The signal shows a fast rise followed by a slower decay, characteristic of the scintillation process in NaI(Tl).

For the quantitative determination of the scintillation decay time, only the anode pulse was used, since its falling edge is expected to reflect the fluorescence decay of the NaI(Tl) scintillator more directly. The corresponding waveform together with the exponential fit is shown in Fig. 3.

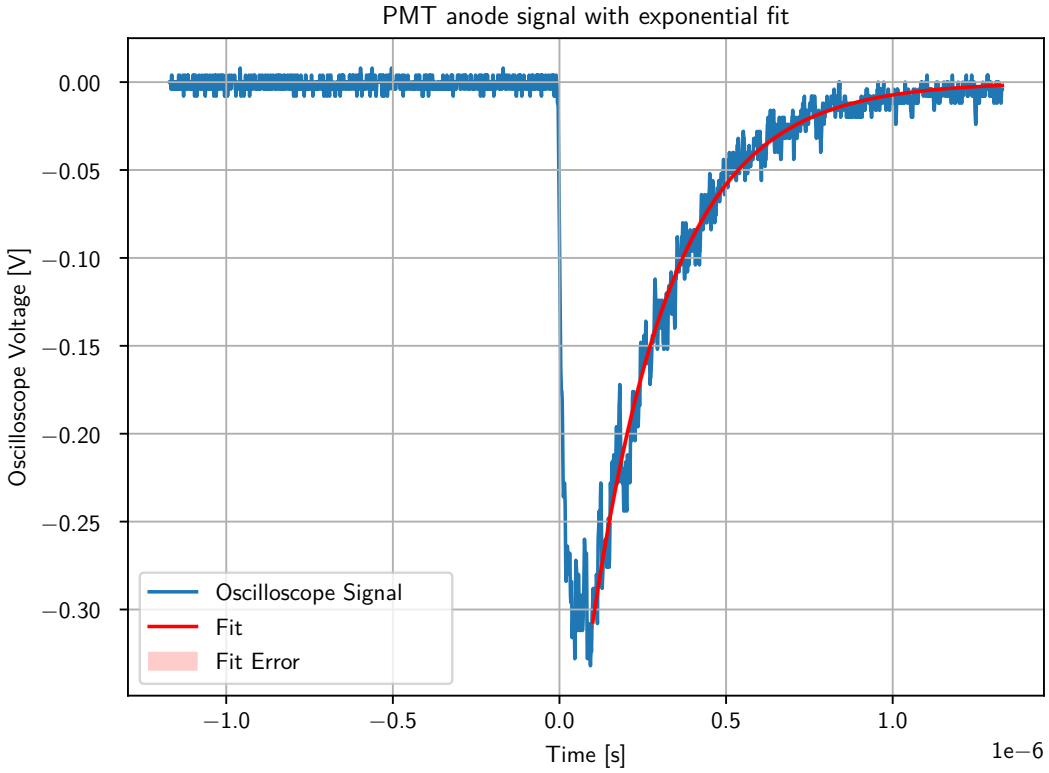

Figure 3: Oscilloscope trace of the PMT anode signal together with the exponential fit to the falling edge. The pulse exhibits a fast onset followed by an approximately exponential decay, as expected for the scintillation light emission in NaI(Tl).

The falling edge of the anode pulse was fitted with an exponential function of the form

$$V(t) = A\exp(-\lambda t), \tag{1}$$

where A is the pulse amplitude and λ is the decay constant. The scintillation decay time is then given by

$$\tau = \frac{1}{\lambda}. \tag{2}$$

From the fit to the anode pulse, the decay constant was obtained as

$$\lambda = (4.16 \pm 0.03) \times 10^6\,\mathrm{s}^{-1}. \tag{3}$$

Using Eq. (2), this corresponds to a fluorescence decay time of

$$\tau = (2.404 \pm 0.017) \times 10^{-7}\,\mathrm{s} = (240.4 \pm 1.7)\,\mathrm{ns}. \tag{4}$$

The measured decay time is consistent with the literature value of approximately 250 ns for NaI(Tl) at room temperature reported in [4]. A more precise literature value including an uncertainty is given by [5]:

$$\tau_{\mathrm{lit}} = (239 \pm 3)\,\mathrm{ns}. \tag{5}$$

The reduced chi-square of the fit was found to be

$$\chi^2_{\text{red}} = 0.935, \tag{6}$$

which indicates that the falling edge is well described by a simple exponential model.
To quantify the agreement with the literature, a compatibility test was performed using

$$z = \frac{|\tau_{\text{meas}} - \tau_{\text{lit}}|}{\sqrt{\sigma^2_{\text{meas}} + \sigma^2_{\text{lit}}}}, \tag{7}$$

with

$$\tau_{\text{meas}} = (240.4 \pm 1.7)\,\text{ns}.$$

This yields

$$z = \frac{|240.4 - 239|}{\sqrt{1.7^2 + 3^2}} \approx 0.41. \tag{8}$$

Since $z < 2$, the measured decay time is compatible with the literature value within uncertainties. This confirms that the recorded anode pulse is consistent with the scintillation process in the NaI(Tl) crystal.

In contrast, the dynode signal is not suitable for extracting the intrinsic fluorescence decay time of the scintillator. Although it also shows a decaying pulse shape, its time structure is affected by the internal electron multiplication stages of the PMT. Therefore, the dynode pulse does not directly represent the pure scintillation light emission of the NaI(Tl) crystal.

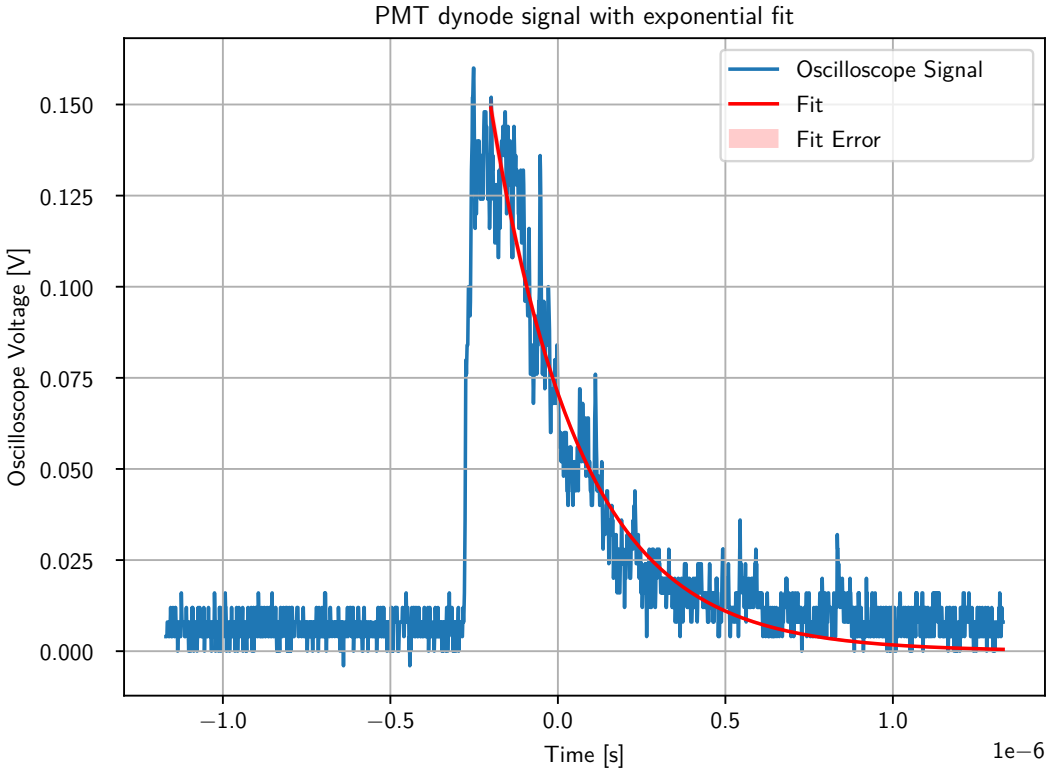

Figure 4: Oscilloscope trace of the PMT dynode signal with an exponential fit shown for comparison. While the signal also exhibits a decaying behaviour, it is influenced by the internal PMT amplification chain and is therefore not used for the determination of the intrinsic NaI(Tl) fluorescence decay time.

For completeness, the fit to the dynode pulse yielded

$$\lambda_{\text{dynode}} = (3.723 \pm 0.045) \times 10^6\,\text{s}^{-1}, \tag{9}$$

corresponding to

$$\tau_{\text{dynode}} = (2.686 \pm 0.032) \times 10^{-7}\,\text{s} = (268.6 \pm 3.2)\,\text{ns}, \tag{10}$$

with

$$\chi^2_{\text{red, dynode}} = 0.828. \tag{11}$$

This value is reported only as a comparison of pulse shapes and not as a measurement of the scintillator fluorescence decay time.

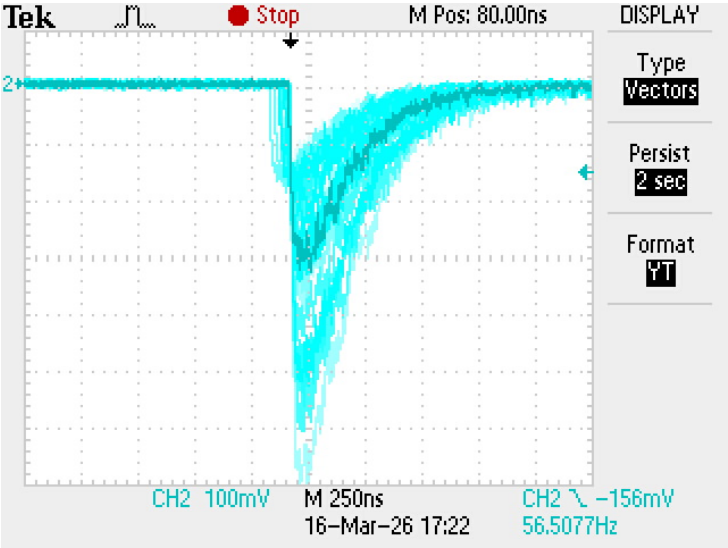

Figure 5: Oscilloscope trace of the PMT anode signal recorded in persistence mode. The superposition of many events produces a broad distribution of pulse amplitudes. In addition, a more pronounced band is visible at a characteristic amplitude, indicating that pulses of this height occur more frequently than others.

In persistence mode, the oscilloscope superimposes many subsequently recorded pulses, such that frequently occurring pulse heights appear more prominently than rare ones. Besides the broad continuous distribution of amplitudes, Fig. 5 shows a more pronounced band at a characteristic pulse height. This feature is interpreted as the pulse amplitude corresponding to the 511 keV annihilation photons. Since these photons are produced very frequently in the decay of the ^{22}Na source, signals with this amplitude occur more often than others and therefore appear as a clearly enhanced line in persistence mode.

PMT Working Point Characterization

In order to determine a suitable operating voltage for the photomultiplier tube (PMT), the counting rate was measured as a function of the applied high voltage. The resulting counting curve is shown in Fig. 6. At low voltages, the pulse amplitudes are too small to pass the discriminator threshold and only a small number of counts is registered. With increasing high voltage, the PMT gain increases and therefore the count rate rises significantly.

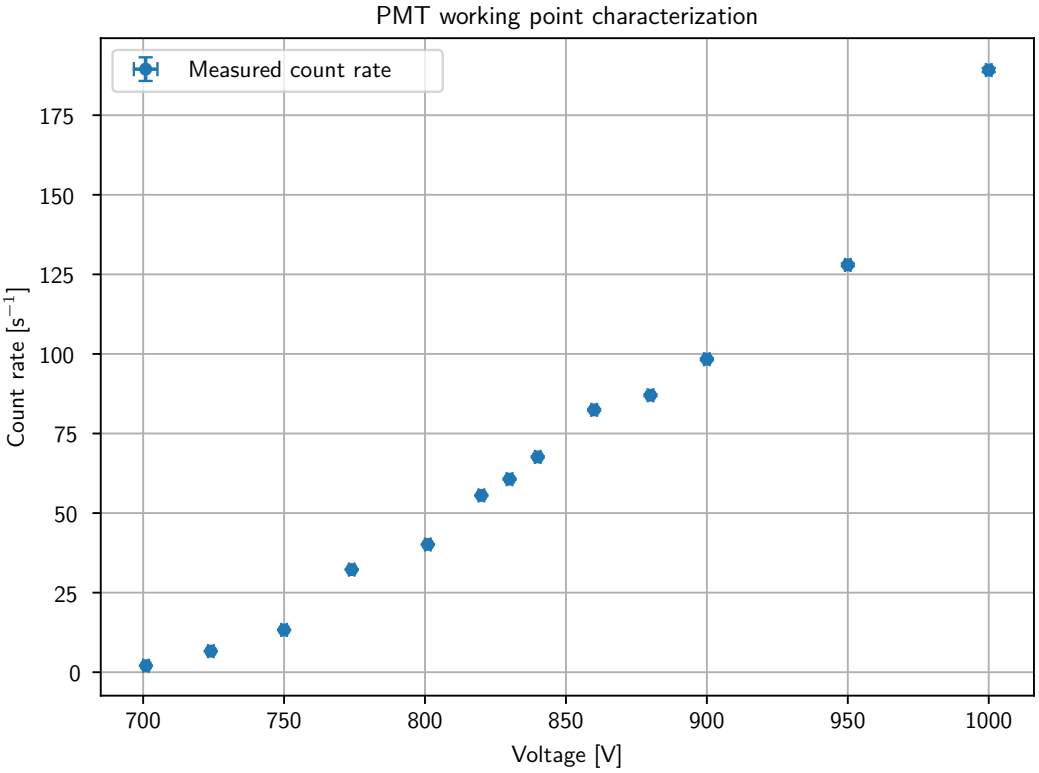

Figure 6: Counting curve of the PMT, showing the measured count rate as a function of the applied high voltage. The count rate increases strongly with voltage, reflecting the increasing PMT gain.

Ideally, the PMT counting curve exhibits a plateau region in which the count rate remains approximately constant over a range of high voltages. In this regime, small fluctuations of the applied voltage or gain have only little influence on the measured rate, making it the preferred working region of the PMT. In the present measurement, however, no clearly flat plateau was observed in the scanned voltage range. Instead, the count rate increases over the full interval. In the intermediate voltage region, the dependence appears approximately linear, whereas above about 900 V the data suggest a steeper, possibly exponential increase. Due to the limited number of measurement points, these regions cannot be determined quantitatively with high precision.

The PMT working point was therefore chosen as a practical compromise. The operating voltage was set in the intermediate region, where the pulse amplitudes are large enough to pass the discriminator threshold reliably, but still below the high-voltage region where the count rate starts to increase more rapidly. For most subsequent measurements, the applied voltage was approximately $V_{\mathrm{WP}} = 826\,\mathrm{V}$, close to the initially chosen value of about 830 V.

This choice provides stable detection conditions while avoiding operation in the region of rapidly increasing count rate, where electronic noise, gain fluctuations, or signal distortions could become more relevant.

The expected counting rate per scintillator can be estimated from the geometrical accep-

tance of the detector and the present activity of the ^{22}Na source. Using the detector radius $r = 1.27\,\text{cm}$ (half the diameter of 2.54 cm), the source-to-detector distance $d = 5\,\text{cm}$, and assuming full detection efficiency, the geometrical fraction of an isotropic photon flux intercepted by one scintillator is approximately

$$\frac{\Omega}{4\pi} \approx \frac{r^2}{4d^2} \approx 0.016.$$

The present activity of the ^{22}Na source can be estimated from radioactive decay using

$$A(t) = A_0 \left(\frac{1}{2}\right)^{t/T_{1/2}}, \tag{12}$$

where $A_0 = 370\,\text{kBq}$ is the initial activity at the time of purchase and $T_{1/2} = 2.6\,\text{y}$ is the half-life of ^{22}Na. Assuming a time interval of approximately $t = 17\,\text{y}$, this gives

$$A \approx 370\,\text{kBq} \cdot \left(\frac{1}{2}\right)^{17/2.6} \approx 4\,\text{kBq}. \tag{13}$$

Thus, the present activity of the source is approximately 4 kBq.
So the expected event rate for the two 511 keV annihilation photons is therefore

$$R \approx 2A\frac{\Omega}{4\pi} \approx 130\,\text{s}^{-1}.$$

The measured count rate at the chosen working point (about $60\,\text{s}^{-1}$) is somewhat lower but remains of the same order of magnitude. This deviation is reasonable since the estimate assumes idealized geometrical acceptance and full detection efficiency, while in reality, threshold effects, incomplete energy deposition, and electronic losses reduce the observed rate.

3.1 Hand-made Integrated Spectrum

As a first energy-spectrum measurement, a hand-made integrated spectrum was recorded using the single channel analyzer (SCA) and the scalar unit. For this measurement, the lower threshold of the SCA window was varied step by step, while the corresponding count rate was recorded for each setting. The resulting spectrum is shown in Fig. 7.

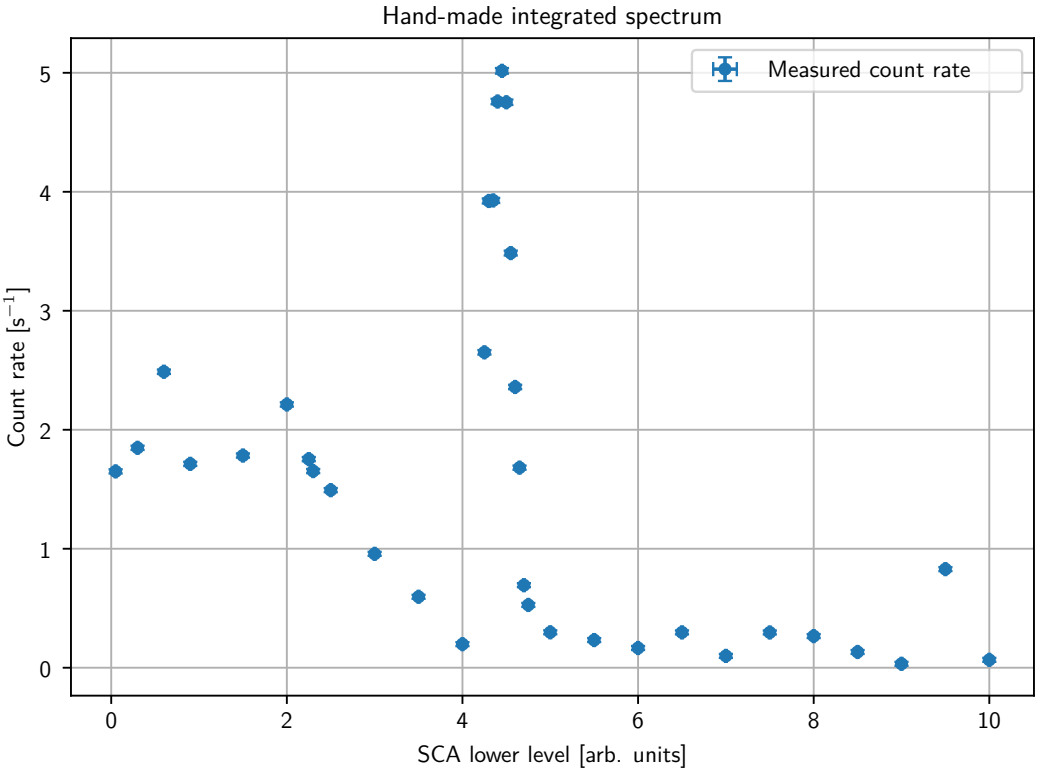

Figure 7: Hand-made integrated spectrum obtained with the SCA and scalar. The measured count rate is plotted as a function of the SCA lower level, given in arbitrary units.

The horizontal axis is given in arbitrary units, since at this stage no direct calibration to physical energy units has yet been performed. Therefore, the plot should be interpreted qualitatively as the count rate obtained for different discriminator settings rather than as a fully calibrated energy spectrum.

A pronounced maximum (the photo-peak) is visible around an SCA lower level of approximately 4.4 to 4.6 arb. units, indicating that pulses in this amplitude range occur particularly frequently. At larger lower-level settings, the count rate drops strongly, as only very large pulses remain above threshold. At small lower-level settings, the count rate is also relatively high, since a large fraction of the pulse-height distribution contributes.

Thus, the hand-made spectrum already provides a first qualitative impression of the pulse-height distribution of the detected γ-rays. However, due to the limited number of measurement points, the finite SCA window width, and the lack of an energy calibration, the physical features of the spectrum, such as the photopeak, Compton edge, and the high-energy tail, cannot yet be identified with high precision. A more detailed and quantitatively interpretable spectrum is obtained in the following section using the multichannel analyzer (MCA).

3.2 Automated Spectrum Acquisition with MCA and Energy Calibration

In contrast to the hand-made spectrum obtained with the SCA and scalar, the multichannel analyzer (MCA) records the full pulse-height spectrum automatically.

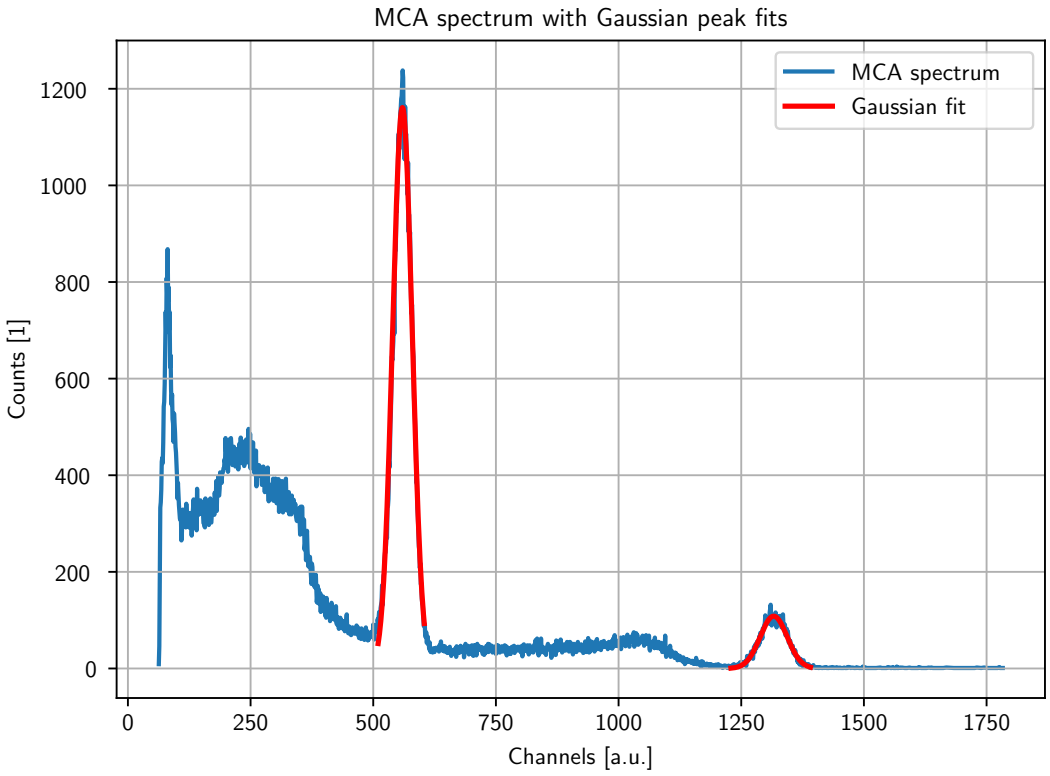

Figure 8: Raw MCA spectrum plotted as registered counts versus channel number. The two dominant photopeaks are clearly visible and were fitted locally to determine their channel positions.

The raw MCA spectrum in Fig. 8 shows two clearly visible photopeaks, which were fitted locally to determine their channel positions for the subsequent energy calibration. To convert the horizontal axis from channel number to energy, the channel positions of the two fitted photopeaks were used together with the origin as calibration points, and a linear calibration function was determined.

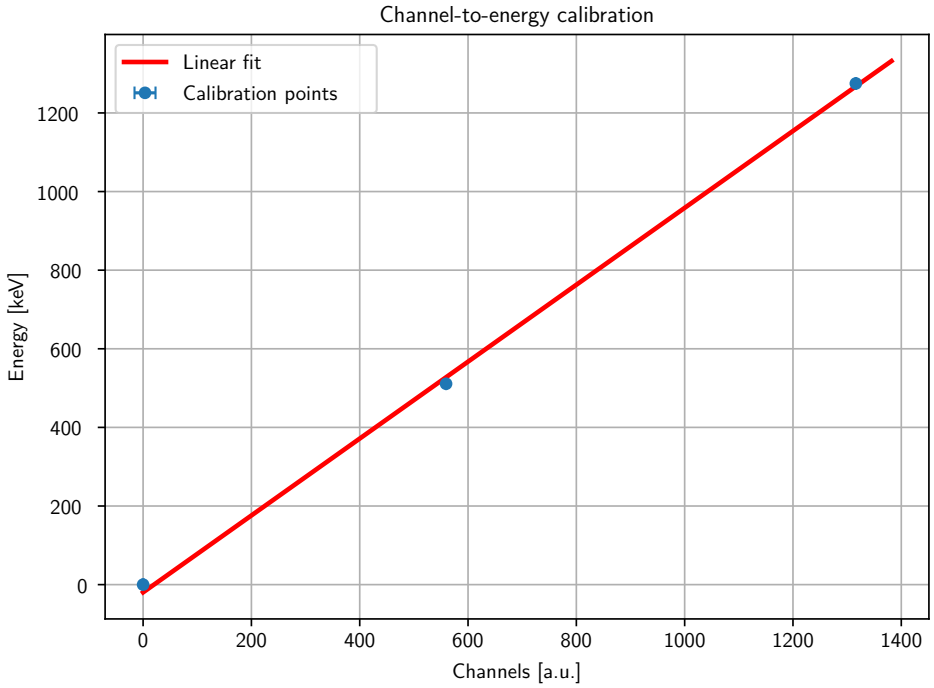

Figure 9: Linear calibration of the MCA channel number to deposited energy using the two fitted photopeaks and the origin.

The MCA channels represent intervals of pulse amplitude; after calibration, they correspond to deposited energies in keV. The channel-to-energy conversion was modeled by a linear function

$$E(c) = a + b\,c,$$

where c denotes the MCA channel number. Using the two fitted photopeak positions together with the origin, the calibration fit yielded

$$a = (-50.0 \pm 14.7)\,\text{keV}, \qquad b = (1.0036 \pm 0.0229)\,\text{keV/channel}.$$

Thus, the conversion function used for the energy calibration is

$$E(c) = (-50.0 \pm 14.7)\,\text{keV} + (1.0036 \pm 0.0229)\,\text{keV/channel} \cdot c.$$

The fitted intercept is not compatible with zero within uncertainties, since

$$\frac{|a|}{\sigma_a} = \frac{50.0}{14.7} \approx 3.4.$$

This indicates a residual offset in the calibration. Although the origin was included as a calibration point, the intercept was left as a free fit parameter. The non-zero intercept therefore indicates that the three calibration points are not perfectly described by a single linear relation with vanishing offset. Possible reasons are systematic effects such

as uncertainties in the fitted peak positions, a non-perfectly linear ADC response, ADC pedestal effects, or offsets in the electronic baseline. Physically, however, a vanishing pulse height should correspond to zero deposited energy. Therefore, a calibration in which the intercept is fixed to zero, i.e. $E(c) = bc$, would be better motivated.
The resulting calibrated spectrum is shown in Fig. 10. Compared to the hand-made SCA spectrum, the MCA spectrum reproduces the same main physical structures with significantly improved resolution and much finer sampling.

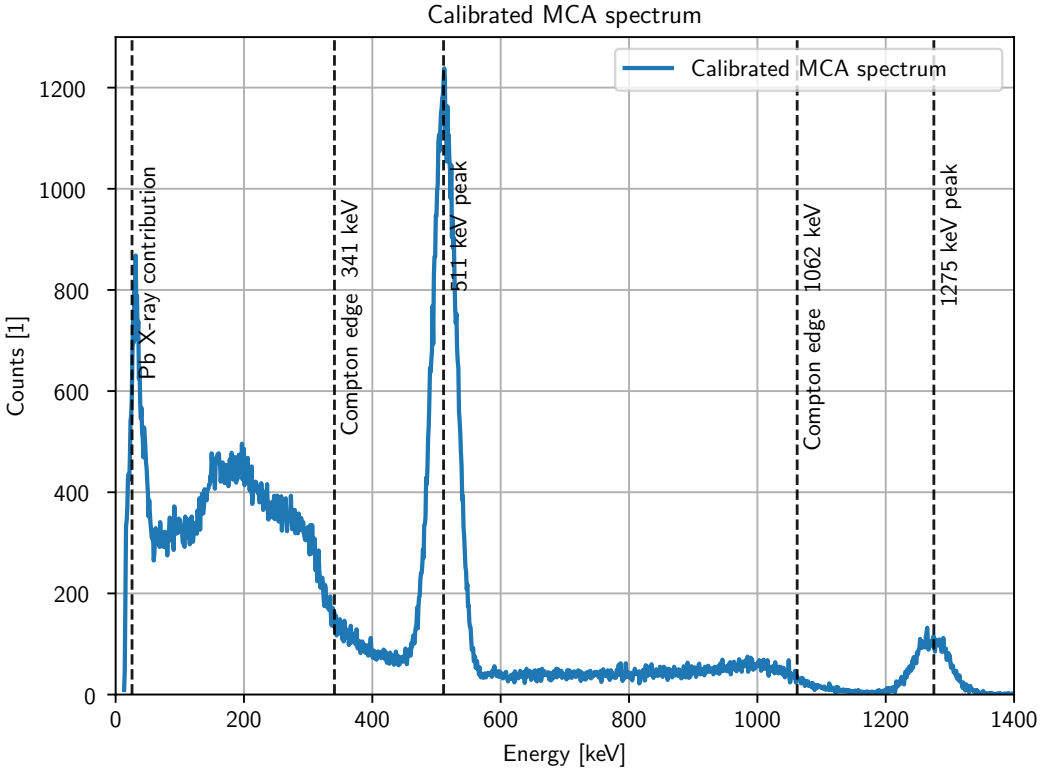

Figure 10: Calibrated MCA spectrum shown as counts versus deposited energy. The two main photopeaks at 511 keV and 1275 keV are clearly visible.

The two dominant photopeaks at 511 keV and 1275 keV originate from the ^{22}Na decay scheme: the first corresponds to the annihilation photons, while the second is due to the de-excitation photon from ^{22}Ne. Each of these photons also produces a Compton continuum, ending at its characteristic Compton edge. The edges at approximately 341 keV and 1062 keV correspond to the maximum energy transfer in Compton scattering of the 511 keV and 1275 keV photons, respectively. At low energies, an additional broad structure is visible in the calibrated spectrum. It is attributed to characteristic Pb X-rays produced in the lead shielding and detected by the scintillator. Since this feature is broad and may consist of several unresolved Pb X-ray lines together with background contributions, it is referred to as a Pb X-ray contribution rather than as a well-defined single peak.

Gaussian Fits of the Two Photopeaks

Photopeaks are commonly modeled by Gaussian functions, since the response of a spectrometer to monoenergetic γ rays is often approximately Gaussian. Physically, this can be understood from the fact that the recorded pulse height is influenced by several stochastic processes, including scintillation-light production, light collection, photoelectron generation, and electron multiplication in the photomultiplier. The combined statistical fluctuations of these processes lead, to a good approximation, to a Gaussian peak shape. Therefore, a Gaussian model provides a reasonable description of the photopeaks in the measured spectrum [3, 6].

To determine the peak positions more precisely, the two visible γ-photopeaks were fitted separately in restricted energy intervals. A Gaussian function with a constant background,

$$f(E) = A \exp\left(-\frac{(E-\mu)^2}{2\sigma^2}\right) + C, \tag{14}$$

was used. The fit ranges were chosen locally around each peak in order to reduce distortions from neighboring spectral structures, in particular from the Compton continuum on the low-energy side of the 511 keV peak.

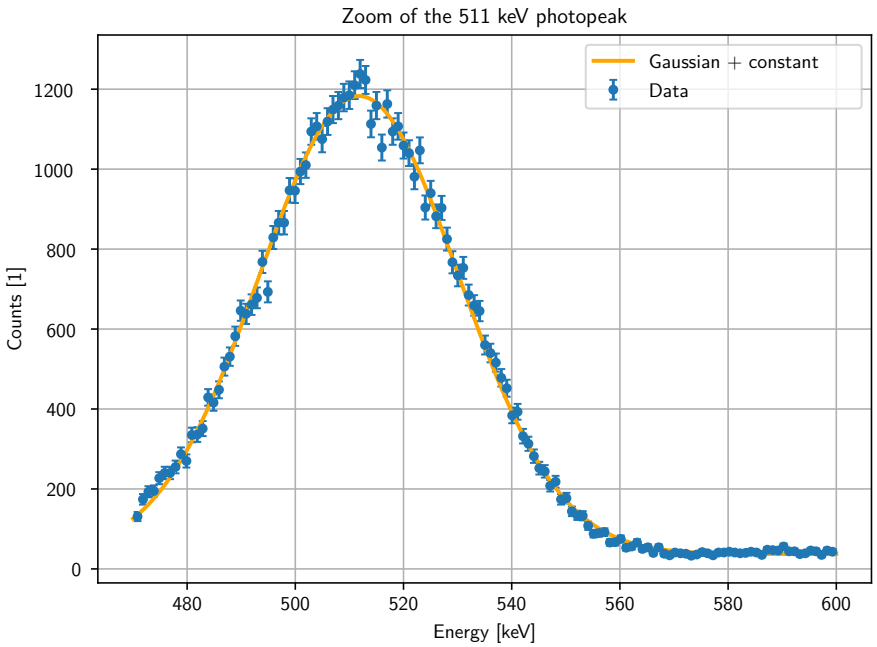

Figure 11: Zoom of the 511 keV photopeak together with the Gaussian fit with constant background.

The fitted peak position is compatible with the expected annihilation-photon energy of 511 keV. The finite peak width reflects the detector energy resolution and statistical fluctuations in the scintillation and readout process.

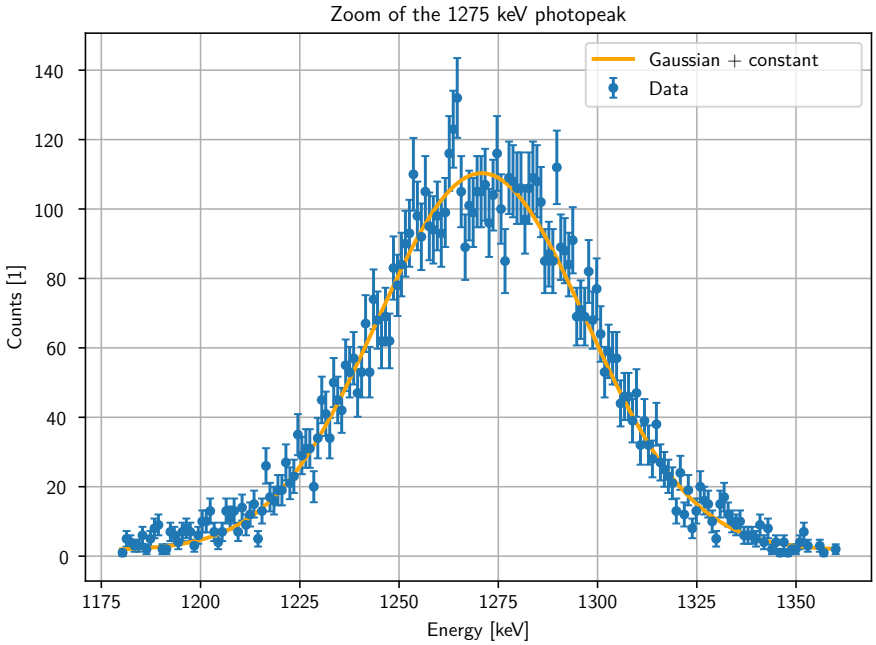

Figure 12: Zoom of the 1275 keV photopeak together with the Gaussian fit with constant background.

The second fitted peak corresponds to the 1275 keV de-excitation photon from ^{22}Ne. For both peaks, the fit quality was evaluated through the chi-square value and is reported together with the fitted parameters in table 1.

Table 1: Fit results for the two visible γ-photopeaks, obtained with a Gaussian function and constant background. The 511 keV peak was fitted in the range 470–600 keV, while the 1275 keV peak was fitted in the range 1180–1360 keV. For the uncertainty of the centroid μ, the error of the mean was used rather than the uncertainty returned directly by the fit.

511 keV photopeak

Parameter	Value	Std. error
A	1145.8	6.5
μ [keV]	511.78	0.08
σ [keV]	18.47	0.09
C	37.1	1.1
χ^2	172.47	
DoF	125	
χ^2_{red}	1.380	
p	0.00317	

1275 keV photopeak

Parameter	Value	Std. error
A	108.6	1.6
μ [keV]	1270.83	0.32
σ [keV]	26.43	0.34
C	1.69	0.34
χ^2	211.40	
DoF	172	
χ^2_{red}	1.229	
p	0.02190	

The fitted peak positions are consistent with the expected γ-ray energies of ^{22}Na. For the uncertainty of the peak positions, the error of the mean was used,

$$\sigma_\mu = \frac{\sigma}{\sqrt{N_{\text{peak}}}},$$

where N_{peak} was estimated from the area of the fitted Gaussian. The first peak was found at $(511.78 \pm 0.08)\,\text{keV}$, while the second peak was obtained at $(1270.83 \pm 0.32)\,\text{keV}$. The small deviation of the second peak from 1275 keV is likely due to residual calibration uncertainties and other systematic effects not included in the purely statistical fit errors. The goodness-of-fit results are $\chi^2_{\text{red}} = 1.380$ with $p = 0.00317$ for the 511 keV peak and $\chi^2_{\text{red}} = 1.229$ with $p = 0.02190$ for the 1275 keV peak. While the reduced chi-square values are close to 1, the rather small p-values indicate that the Gaussian plus constant background model does not fully describe the data. This is likely due to an under-modeling of the peak regions: the photopeaks are superimposed on a non-flat Compton background, and the detector response may contain small asymmetries or tail contributions that are not included in a simple Gaussian model with constant background. In addition, the statistical uncertainties are small in bins with high count rates, so even small systematic deviations contribute significantly to the chi-square. A more realistic fit model would include a sloped or polynomial background, or possible asymmetric tail contributions.

The detector energy resolution was determined from the fitted peak widths using

$$R_{\text{FWHM}} = \frac{\Delta E}{E} = 2.35\frac{\sigma}{E}. \tag{15}$$

For the 511 keV photopeak, this gives

$$R_{\text{FWHM}}(511\,\text{keV}) = (8.48 \pm 0.04)\%. \tag{16}$$

For the 1275 keV photopeak, the corresponding value is

$$R_{\text{FWHM}}(1275\,\text{keV}) = (4.89 \pm 0.06)\%. \tag{17}$$

As expected, the detector shows a better relative energy resolution at higher energy.

Fits of the Compton Edges

In addition to the two photopeaks, the Compton edges associated with the 511 keV annihilation photons and the 1275 keV de-excitation photons were fitted. A Compton edge is not peak-like, but corresponds to the maximum energy transferred to the recoil electron in Compton scattering. Therefore, the edges were modeled phenomenologically by a smeared step function with a linear background,

$$f(E) = a + bE + \frac{A}{2}\left[1 - \text{erf}\left(\frac{E - E_c}{\sqrt{2}\,\sigma}\right)\right], \tag{18}$$

where the error-function term accounts for the smoothing of the ideal edge due to the finite detector energy resolution. The parameter E_c gives the fitted Compton-edge position.

The theoretical Compton-edge energy is given by the maximum kinetic energy transferred to the recoil electron,

$$E_{c,\mathrm{th}} = E_\gamma \left(1 - \frac{1}{1 + 2E_\gamma/(m_e c^2)}\right), \tag{19}$$

where $m_e c^2 = 511\,\mathrm{keV}$. This gives $E_{c,\mathrm{th}} \approx 341\,\mathrm{keV}$ for $E_\gamma = 511\,\mathrm{keV}$ and $E_{c,\mathrm{th}} \approx 1062\,\mathrm{keV}$ for $E_\gamma = 1275\,\mathrm{keV}$.

The 511 keV Compton edge was fitted in the range 250–420 keV, while the 1275 keV Compton edge was fitted in the range 900–1150 keV.

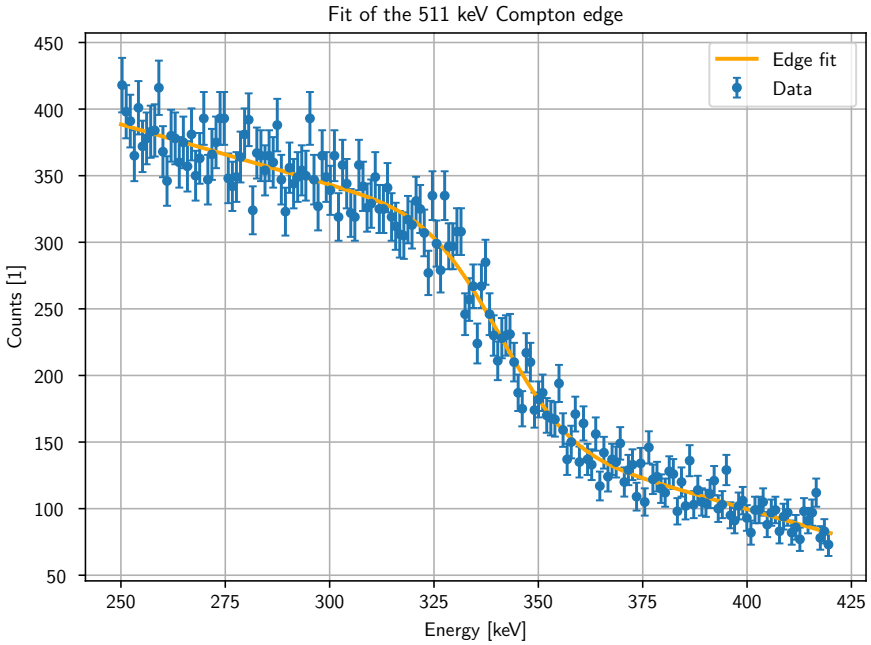

Figure 13: Fit of the Compton edge associated with the 511 keV photons. The theoretical edge position is approximately 341 keV.

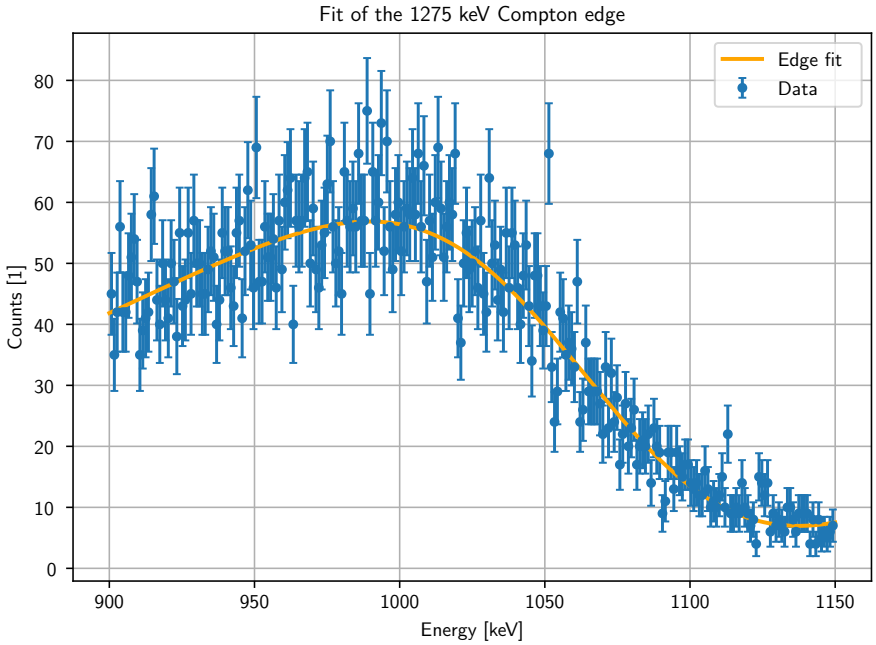

Figure 14: Fit of the Compton edge associated with the 1275 keV photons. The theoretical edge position is approximately 1062 keV.

Table 2: Fit results for the two Compton edges, obtained with the smeared step model of Eq. (18) and a linear background.

511 keV **Compton edge**

Parameter	Value	Std. error
A	153.77	10.01
E_c [keV]	340.83	0.78
σ [keV]	13.24	1.44
a	460.01	32.81
b	-0.9010	0.0825
χ^2	182.43	
DoF	169	
χ^2_{red}	1.079	
p	0.22718	

1275 keV **Compton edge**

Parameter	Value	Std. error
A	93.33	11.34
E_c [keV]	1063.51	1.48
σ [keV]	46.21	4.14
a	-253.48	53.48
b	0.2245	0.0454
χ^2	240.42	
DoF	250	
χ^2_{red}	0.962	
p	0.65694	

The fit to the 511 keV Compton edge yields

$$E_c(511\,\text{keV}) = (340.8 \pm 0.8)\,\text{keV}, \tag{20}$$

which is compatible with the theoretical expectation of approximately 341 keV. The goodness of fit, with $\chi^2_{\text{red}} = 1.079$ and $p = 0.227$, indicates that the smeared step model provides an adequate description of the data in the chosen fit range.

For the 1275 keV photon, the fitted Compton-edge position is

$$E_c(1275\,\text{keV}) = (1063.5 \pm 1.5)\,\text{keV}. \tag{21}$$

This is also compatible with the theoretical value of approximately 1062 keV. The fit quality is good, with $\chi^2_{\text{red}} = 0.962$ and $p = 0.657$, showing that the same phenomenological model also describes the higher-energy Compton edge well. The larger fitted width for this edge reflects the broader transition region in the measured spectrum and the lower statistics compared to the 511 keV Compton edge.

Fit of the Pb X-ray Contribution

At low energies, an additional structure is visible in the calibrated MCA spectrum. This feature is attributed to characteristic X-rays emitted by the lead shielding. Photons from the ^{22}Na source can interact with the lead, for example via photoelectric absorption, creating inner-shell vacancies. The subsequent atomic de-excitation produces characteristic Pb X-rays, which can then be detected by the NaI(Tl) scintillator.
To estimate the effective position of this low-energy contribution, the structure was fitted after the channel-to-energy calibration. Since the feature is broad and sits on a non-trivial background, it was modeled phenomenologically by a Gaussian function with a linear background,

$$f(E) = A \exp\left[-\frac{(E-\mu)^2}{2\sigma^2}\right] + mE + c. \tag{22}$$

The fit was performed in the range 34.5–94.5 keV.

Table 3: Fit results for the low-energy Pb X-ray contribution, obtained with the model of Eq. (22) in the fit range 34.5–94.5 keV.

Parameter	Value	Std. error
A	2042.66	396.65
μ [keV]	51.39	1.63
σ [keV]	20.42	1.83
m	37.2145	6.9770
c	-3389.20	752.38
χ^2	256.19	
DoF	49	
χ^2_{red}	5.228	
p	$< 10^{-4}$	

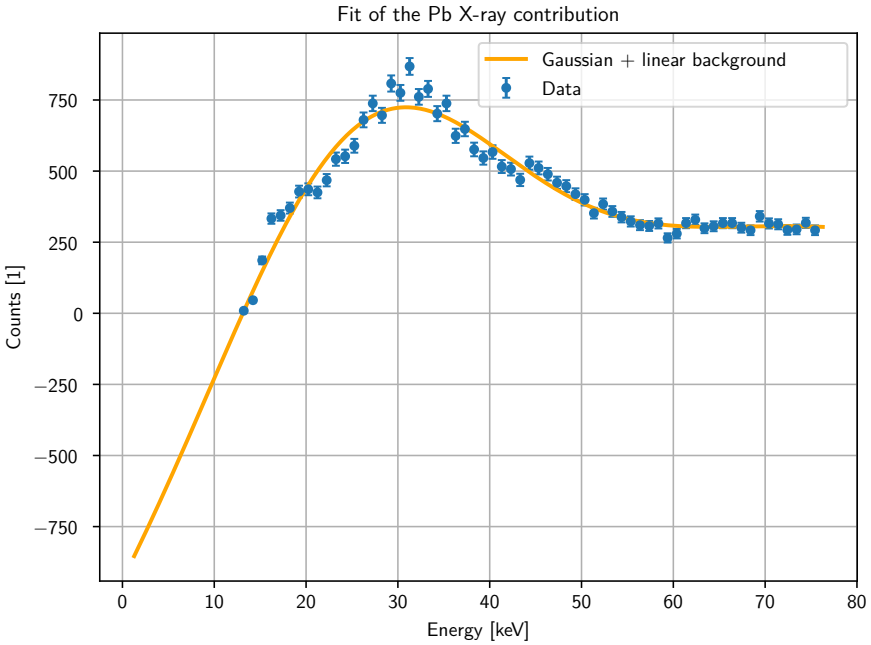

Figure 15: Fit of the low-energy Pb X-ray contribution using a Gaussian function with a linear background.

The fit gives an effective centroid of

$$\mu_{\mathrm{Pb}} = (51.4 \pm 1.6)\,\mathrm{keV}. \tag{23}$$

However, the goodness of fit is poor, with $\chi^2_{\mathrm{red}} = 5.228$ and a p-value close to zero. This shows that the simple Gaussian plus linear background model does not fully describe the low-energy structure. The fitted value should therefore not be interpreted as a precise characteristic X-ray energy, but rather as an effective position of the observed low-energy contribution.

The characteristic Pb K-X-ray lines are expected mainly in the range of approximately 73–85 keV [2]. The lower fitted centroid is likely influenced by the limited energy resolution of the NaI(Tl) detector, the non-trivial background shape in the low-energy region, and the residual offset of the energy calibration, which affects low energies most strongly. Nevertheless, the presence of a pronounced low-energy structure is qualitatively consistent with secondary X-ray production in the lead shielding.

3.3 Spectrum Acquisition with Coincidence Requirement

As a final step, the spectrum was intended to be recorded under a coincidence condition between the two detectors. In an ideal coincidence measurement, the MCA should only accept events for which both scintillators produce signals within the coincidence time window. This requires the output of the coincidence unit to be used as the gate signal for the ADC during data acquisition. In this configuration, the recorded spectrum should be

dominated almost exclusively by the 511 keV annihilation peak, since the two annihilation photons are emitted approximately back-to-back and can therefore trigger both detectors simultaneously.
For the available coincidence data set, the spectrum was converted to energy using the calibration corresponding to the ADC settings of the coincidence measurement,

$$E(c) = (-0.1 \pm 1.4)\,\mathrm{keV} + (0.954 \pm 0.018)\,\mathrm{keV/channel} \cdot c, \tag{24}$$

where c denotes the MCA channel number. This calibration was used because the coincidence data were recorded with ADC settings corresponding to this channel-to-energy relation. Applying the single-detector MCA calibration from subsection 3.2 would shift the observed annihilation peak away from 511 keV.

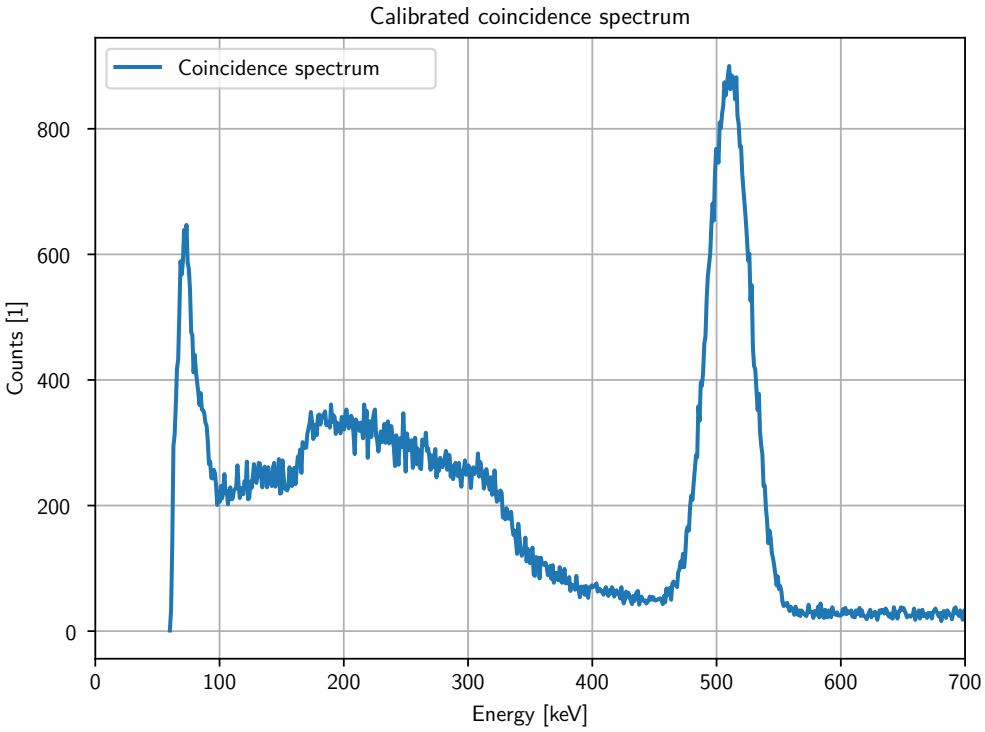

Figure 16: Calibrated coincidence spectrum shown on a linear scale. A pronounced peak is visible near the expected 511 keV annihilation-photon energy.

The calibrated spectrum shows a pronounced peak close to the expected 511 keV annihilation photon energy. However, sizeable low-energy and Compton-like contributions are still visible. In an ideal coincidence-gated spectrum, these structures should be strongly suppressed because uncorrelated single-detector events should not satisfy the coincidence condition. Their presence indicates that the coincidence condition was not properly ensured during the acquisition of the available data, most likely because the coincidence output was not correctly used as the ADC gate. Since the data set contains only the final MCA histogram and no event-by-event timing information, this selection cannot be

applied retroactively in the analysis.

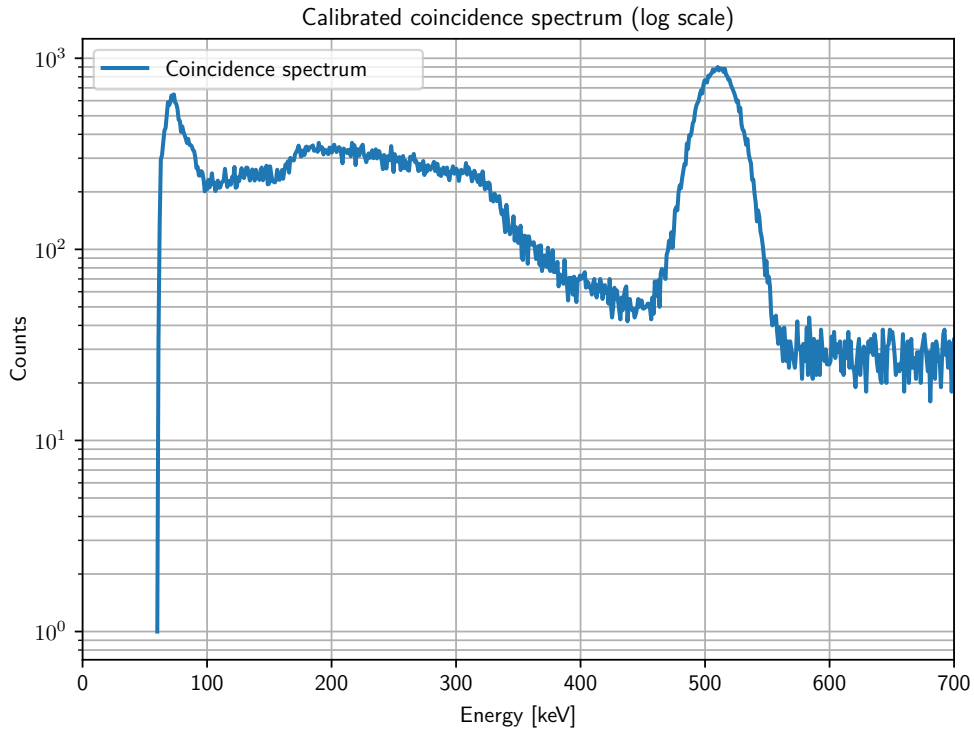

Figure 17: Calibrated coincidence spectrum shown on a logarithmic scale. The logarithmic representation makes the remaining non-peak contributions outside the main annihilation peak more visible.

The logarithmic plot confirms that the non-peak contributions are not negligible. In particular, the low-energy structure and the Compton-like continuum remain clearly visible. This is not expected for a properly coincidence-gated spectrum, where almost all contributions unrelated to simultaneous detection of the two annihilation photons should be suppressed. Ideally, besides the dominant 511 keV annihilation peak, only much smaller residual contributions should remain. Therefore, the measured distribution should be interpreted with caution: although the 511 keV peak is clearly present, the spectrum does not represent an ideal coincidence-gated measurement.

To determine the position of the dominant annihilation peak more precisely, the peak was fitted locally in the range 470–550 keV. A Gaussian function with a constant background,

$$f(E) = A \exp\left(-\frac{(E-\mu)^2}{2\sigma^2}\right) + C, \tag{25}$$

was used.

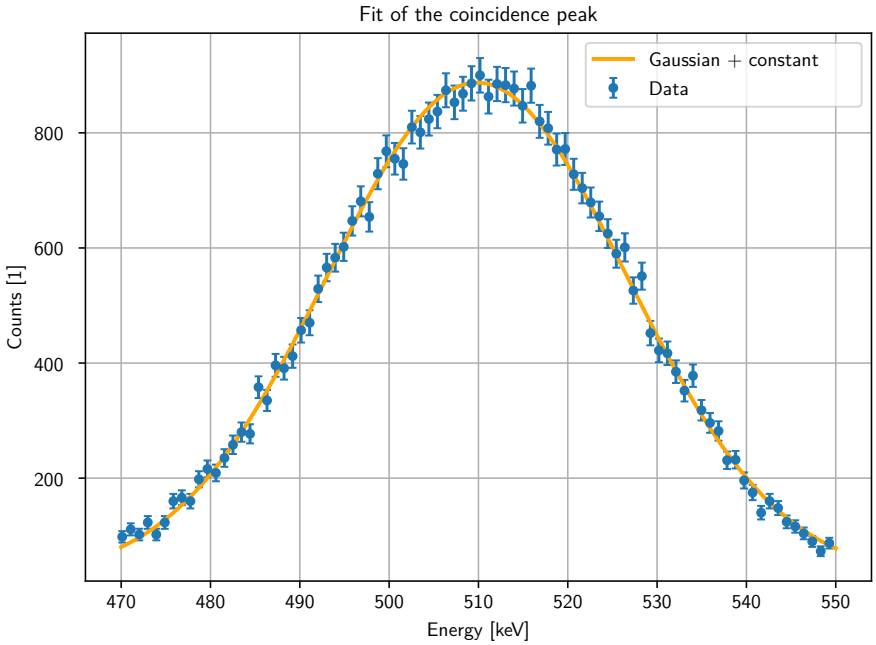

Figure 18: Fit of the dominant coincidence peak with a Gaussian function and constant background.

The fit was performed in MCA channels and the result was then converted to energy using the calibration above, including the propagated calibration uncertainty. This yields a peak position of

$$\mu = (510.0 \pm 9.7)\,\text{keV}, \tag{26}$$

which is compatible with the expected annihilation-photon energy of 511 keV within uncertainties. The fitted width is

$$\sigma = (16.85 \pm 0.38)\,\text{keV}. \tag{27}$$

The goodness of fit is satisfactory within the chosen local fit range, with

$$\chi^2 = 76.04, \qquad \text{DoF} = 80, \qquad \chi^2_{\text{red}} = 0.950, \tag{28}$$

corresponding to a p-value of 0.605. This indicates that the Gaussian plus constant background model provides a good description of the dominant peak itself. However, this local fit does not imply that the full spectrum is an ideal coincidence spectrum, since the broader distribution still contains significant non-coincident background contributions.

The photon energy spectrum measured without a coincidence gate corresponds to the ordinary single-detector MCA spectrum, in which all recorded events are accepted. It therefore contains the full set of structures discussed above, including the 511 keV and 1275 keV photopeaks, the associated Compton continua, the Compton edges, and the low-energy Pb X-ray contribution.

In contrast, a correctly gated coincidence spectrum should only accept events accompanied by a simultaneous signal in the opposite detector. As a result, it should be much more selective. The 511 keV annihilation peak should remain as the dominant structure, since the two annihilation photons are emitted approximately back-to-back and are therefore especially likely to fulfill the coincidence condition. At the same time, uncorrelated background events, the 1275 keV single-photon contribution, and most Compton-like structures should be strongly reduced.

The available measured spectrum does show a clear enhancement of the 511 keV peak, but the remaining low-energy and Compton-like structures are still too pronounced for an ideal coincidence-gated spectrum. This suggests that the coincidence gate was not fully effective during the measurement. Thus, the data demonstrate the presence of the annihilation peak, but they should not be interpreted as a clean coincidence spectrum in which only back-to-back annihilation-photon events were selected.

4 Conclusion

In this experiment, the characteristic signatures of electron–positron annihilation from a ^{22}Na source were successfully observed with NaI(Tl) scintillators coupled to photomultiplier tubes. The oscilloscope study of the PMT pulses showed the expected fast rise and slower exponential decay, and the fluorescence decay time extracted from the anode signal, $\tau = (240.4 \pm 1.7)\,\text{ns}$, was found to be in good agreement with the literature value for NaI(Tl).

The energy spectra obtained with the MCA clearly exhibited the main expected structures of the γ-ray spectrum, in particular the photopeaks at 511 keV and 1275 keV, the corresponding Compton continua, the Compton edges, and the low-energy Pb X-ray contribution. A linear channel-to-energy calibration was established from the fitted photopeak positions, although a residual calibration offset remained visible in the fitted intercept. The detector energy resolution was determined at both photopeak energies. In addition, the fitted Compton-edge positions for both the 511 keV and 1275 keV photons were found to be compatible with the theoretical expectations.

Finally, the available coincidence spectrum showed a pronounced peak close to the expected 511 keV annihilation-photon energy. However, sizeable low-energy and Compton-like contributions remained visible, indicating that the coincidence condition was not fully selective during the data acquisition. Therefore, the measurement should not be interpreted as an ideal coincidence-gated spectrum. Nevertheless, the clear enhancement of the annihilation peak demonstrates the basic principle of selecting events associated with the two back-to-back annihilation photons. Overall, the experiment reproduced the expected qualitative behavior of the detector system and photon spectra, while also illustrating practical limitations arising from finite detector resolution, background, calibration uncertainties, and imperfect coincidence selection.

5 Appendix

Acknowledgments

I would like to thank my lab partner, Emile Sauthier, for the good collaboration throughout this experiment. We carried out the measurements together and jointly collected the data used in this report. I am also especially grateful to him for sharing his Python code with me, which was very helpful for understanding and performing the data analysis.

I would also like to thank our teaching assistant, Giorgia Bonomelli, for her supervision and support during the experiment. Her guidance and explanations were very helpful throughout the laboratory work and the interpretation of the results.

AI Usage Declaration

This report was prepared with assistance from ChatGPT (OpenAI) for language polishing, formatting suggestions, and for helping to develop, debug, and correct the syntax of the Python code. The AI was not used to generate raw data, to select results, or to determine the final conclusions. All scientific content (methods, calculations, results, and interpretation) was critically reviewed and validated by the authors. The authors take full responsibility for the accuracy of the analysis and for any remaining errors. No confidential or personal data were provided to the AI system during the preparation of this report.

References

[1] Giorgia Bonomelli. *e^+e^- Annihilation Experiment*. Physics Practical Laboratory Manual. Latest version: 21/02/2024; V2 by Giorgia Bonomelli (2024); based on V1 by Aldo Antognini and Chiara Casella (2010); laboratory manual for the Physics Practical, ETH Zürich. ETH Zürich, Department of Physics, 2024.

[2] HORIBA Jobin Yvon. *Table of X-ray Emission Lines*. https://www.horiba.com/fileadmin/uploads/Scientific/Documents/XRay/emission_lines.pdf. Energies of principal K, L and M X-ray emission lines; accessed 2026-05-01. 2005.

[3] International Atomic Energy Agency. *Practical Aspects of Operating a Neutron Activation Analysis Laboratory*. Tech. rep. Vienna: IAEA, 2016. URL: https://www-pub.iaea.org/MTCD/Publications/PDF/AQ-48_web.pdf.

[4] Luxium Solutions, LLC. *NaI(Tl) and Polyscin® NaI(Tl): Sodium Iodide Scintillation Material*. Manufacturer data sheet, accessed 2026-04-06. Aug. 2023. URL: https://luxiumsolutions.com/sites/default/files/2023-08/142266_Luxium_Sodium-Iodide-Material-Data-Sheet_FIN.pdf.

[5] P. Sibczynski et al. "Study of NaI(Tl) scintillator cooled down to liquid nitrogen temperature". In: *Journal of Instrumentation* 7.11 (2012), P11006. DOI: 10.1088/1748-0221/7/11/P11006.

[6] Shuheng Wen et al. "Temperature Peak-Drift Correction Method for NaI(Tl) Detectors Using the Background Peak". In: *Sensors* 24.8 (2024), p. 2621. DOI: 10.3390/s24082621. URL: https://www.mdpi.com/1424-8220/24/8/2621.

Python Code

dataAnalyser.py

```
import numpy as np
import matplotlib.pyplot as plt

def poly(a, x) :
    x = np.asarray(x)
    a = np.asarray(a)

    # Shape: (len(x), len(a))
    powers = x[:, None] ** np.arange(len(a))[None, :]

    # Multiply each column by its coefficient and sum
    return np.sum(a * powers, axis=1)

def gauss(beta, x) :
    mu = beta[0]
    sigma=beta[1]
    return beta[2] * np.exp(-(x - mu)**2 / sigma**2 / 2)


def plotData(ax, x, y, sx, sy, label='Data Points') :
    ax.errorbar(x, y, sy, sx, linestyle='None', capsize=2, label=label)
    ax.grid()
    ax.legend()

def plotFit(ax, x, fun, beta, sigma_beta=0, label='Fit') :
    """
    fun(beta, x)
    """
    label_error = label + ' Error'
    x = np.linspace(np.min(x), np.max(x), 500)
    ax.plot(x, fun(beta, x), color=(1,0,0,1), label=label)
    ax.fill_between(x, fun(beta + sigma_beta, x), fun(beta -sigma_beta, x),
    color=(1,0,0,0.2), label=label_error)
    ax.legend()


def funFit(fun, x, y, sigma_x, sigma_y, beta0, ax='', plotDataPoints=True) :
    """
```

```
    fun(beta, x)
    """
    from scipy.odr import Model, Data, RealData, ODR
    model = Model(fun)
    data = RealData(x, y, sigma_x, sigma_y)
    beta0 = np.array(beta0)
    odr = ODR(data, model, beta0 = beta0)
    out = odr.run()
    beta = out.beta
    cov_beta = out.cov_beta
    sigma_beta = np.sqrt(np.diag(cov_beta))

    # plot
    if ax != '' :
        if plotDataPoints :
            plotData(ax, x, y, sigma_x, sigma_y)
            plotFit(ax, x, fun, beta, sigma_beta)
        else :
            plotFit(ax, x, fun, beta, sigma_beta)

    for i in range(len(beta)) :
        if i != 0 :
            print(r" & ", end='')
        print(r"${:} \pm {:}$".format(beta[i], sigma_beta[i]), end='')
    print(r" \\")

    return beta, cov_beta

def linearFit(x,y,sigma_x, sigma_y, ax='', plotDataPoints=True) :
    beta, cov_beta = funFit(poly, x, y, sigma_x, sigma_y, [0,1], ax,
    plotDataPoints)
    return beta, cov_beta

def sortData(data, ind, tol) :
    shape = data.shape
    masks = []
    val = []
    included = False
    for i in range(shape[1]) :
        mask = (data[ind] <= (data[ind][i] + tol)) * (data[ind] >=
    data[ind][i] - tol)
```

```
        for m in masks :
            if np.any((mask * m) != 0) :
                a = mask + m
                m = a > 0
                included = True
            else :
                pass
        if not included :
            masks.append(mask)
            val.append(data[ind][i])
        included = False

        result = []
        for i in range(len(masks)) :
            b = np.ones(shape) * masks[i]
            b[b == 0] = np.nan
            res = b * data
            #res = res[~np.isnan(res)]
            result.append(res)

    return result, val


def chiSquared(model, beta, x, y, sigma_y) :
    m = model(beta, x)
    nu = len(y) - len(beta)
    return np.sum(np.power((y - m), 2) / np.power(sigma_y, 2)) / nu


def intervalOfInterest(x, rest, a=-np.inf, b=np.inf) :
    """
    gets the interval of interest but in a system of two values
    """
    bigger = x >= a
    smaller = x <= b
    bigger = np.float32(bigger)
    smaller = np.float32(smaller)
    bigger[bigger == 0] = np.nan
    smaller[smaller == 0] = np.nan
    x_new = bigger * smaller * x
    x_new = x_new[~np.isnan(x_new)]
```

```
118
119     n = len(rest)
120     rest_new = [[] for i in range(n) ]
121     for i in range(n) :
122         rest_new[i] = bigger * smaller * rest[i]
123         rest_new[i] = rest_new[i][~np.isnan(rest_new[i])]
124     return x_new, rest_new
125
126 def sortArrays(arr):
127     """
128     Sorts the first numpy array from min to max and applies
129     the same permutation to all other arrays in the list.
130
131     Parameters:
132         arr (list of np.ndarray): list of arrays of the same length
133
134     Returns:
135         list of np.ndarray: list of permuted arrays
136     """
137     if not arr:
138         return []
139
140     # Get the permutation that sorts the first array
141     order = np.argsort(arr[0])
142
143     # Apply the same permutation to all arrays
144     return [a[order] for a in arr]
145
146 def restrict_arrays_to_other_array(arrs, values, other_values) :
147     mask = np.zeros(len(arrs[0]))
148     for i in range(len(other_values)) :
149         mask += values == other_values[i]
150     mask[mask == 0] = np.nan
151     res = [[] for i in range(len(arrs))]
152     for i in range(len(arrs)) :
153         a = mask * arrs[i]
154         res[i] = a[~np.isnan(a)]
155
156     return res
157
158
```

```
159 def plotFromOscillo(dirs, path, measureNumber = [], save = False) :
160     if measureNumber == [] :
161         measureNumber = [i for i in range(len(dirs))]
162     for i in range(len(dirs)) :
163 
164         fig = plt.figure()
165         ax1 = fig.add_subplot()
166         ax2 = ax1.twinx()
167         color2="tab:orange"
168         color1="tab:blue"
169         ax2.set_ylabel('Voltage [V]', color=color2)
170         ax1.set_ylabel('Voltage [V]', color=color1)
171         ax2.tick_params(axis='y', labelcolor=color2)
172         ax1.tick_params(axis='y', labelcolor=color1)
173         data1 = np.loadtxt(path + dirs[i] + '/F000' + str(measureNumber[i]) +
    "CH1.CSV", delimiter=',', usecols=(3,4)).T
174         data2 = np.loadtxt(path + dirs[i] + '/F000' + str(measureNumber[i]) +
    "CH2.CSV", delimiter=',', usecols=(3,4)).T
175 
176         #plotData(ax, data1[0], data1[1], 0, 0)
177         #plotData(ax, data2[0], data2[1], 0, 0)
178         ax1.plot(data1[0], data1[1], 'x-', label='Input signal')
179         ax2.plot(data2[0], data2[1], 'x-', label='Analog Output', color=color2)
180 
181         ax1.set_xlabel('Time [s]')
182 
183         ax1.legend()
184         ax2.legend()
185 
186         plt.grid()
187         if save :
188             plt.savefig('../report/fig/' + dirs[i] + ".pdf")
189         plt.show()
190 
191 
192 class Value() :
193     def __init__(self,name, x, sx, unit='') :
194         self.name = name
195         self.x = x
196         self.sx = sx
197         self.unit = unit
```

```
198
199 class Data() :
200     def __init__(self, filename) :
201         data = np.loadtxt(filename, delimiter=',', skiprows=1).T
202         self.N = len(data)
```

scopeFluorescenceTime.py

```python
from dataAnalyser import *

path = '../results/16-03/'
dirs = ["ALL0000", "ALL0001"]
channel = [2, 2]
n = [0, 1]

names = ["dynode_000_fit", "anode_000_fit"]
titles = ["PMT dynode signal with exponential fit", "PMT anode signal with
    exponential fit"]
ext = ".pdf"

beginn = [-2e-7, 1e-7]

def expon(beta, x):
    return beta[1] * np.exp(-beta[0] * x)

for i in range(len(dirs)):
    fig = plt.figure()
    ax = fig.add_subplot()

    data = np.loadtxt(
        path + dirs[i] + "/F000" + str(n[i]) + "CH" + str(channel[i]) + ".CSV",
        delimiter=',',
        usecols=(3, 4)
    ).T

    t = data[0]  # s
    V = data[1]  # V

    # plot raw oscilloscope signal first, so fit stays visible in foreground
    ax.plot(t, V, label='Oscilloscope Signal', zorder=1)

    # select fit region
    t1, [V1] = intervalOfInterest(t, [V], beginn[i], np.inf)

    beta0 = np.array([1, 1])
    beta, cov_beta = funFit(expon, t1, V1, 1e-8, .01, beta0, ax, False)

    print(cov_beta)
```

```
40
41      chi2 = chiSquared(expon, beta, t1, V1, .01)
42      print(chi2)
43
44      ax.set_title(titles[i], fontsize=11)
45      ax.set_xlabel('Time [s]')
46      ax.set_ylabel('Oscilloscope Voltage [V]')
47      ax.legend()
48      plt.grid()
49      plt.tight_layout()
50      plt.savefig("../report/fig/" + names[i] + ext)
51      plt.show()
```

findHV.py

```
from dataAnalyser import *

data = np.loadtxt('../results/findHV.csv', delimiter=',', skiprows=1).T

fig = plt.figure()
ax = fig.add_subplot()

# counts / time = count rate
c = data[1] / data[2]
sc = np.sqrt((1 / data[2] * data[4])**2 + (-data[1] / data[2]**2 * data[5])**2)

# plot with circular markers instead of the default "cross-like" appearance
ax.errorbar(
    data[0], c,
    yerr=sc,
    xerr=data[3],
    fmt='o',
    markersize=4.5,
    capsize=3,
    linestyle='None',
    label='Measured count rate'
)

ax.set_title('PMT working point characterization', fontsize=11)
ax.set_xlabel('Voltage [V]')
ax.set_ylabel(r'Count rate [s$^{-1}$]')
ax.legend()
ax.grid()
plt.tight_layout()

plt.savefig('../report/fig/findHV.pdf')
plt.show()
```

manualSpectra.py

```
from dataAnalyser import *

data = np.loadtxt('../results/spectraManual.csv', delimiter=',', skiprows=1).T

fig = plt.figure()
ax = fig.add_subplot()

# count rate = counts / measurement time
c = data[1] / data[2]
sc = np.sqrt(
    (1 / data[2] * data[4])**2 +
    (-data[1] / data[2]**2 * data[5])**2
)

ax.errorbar(
    data[0], c,
    yerr=sc,
    xerr=data[3],
    fmt='o',
    markersize=4.5,
    capsize=3,
    linestyle='None',
    label='Measured count rate'
)

ax.set_title('Hand-made integrated spectrum', fontsize=11)
ax.set_xlabel('SCA lower level [arb. units]')
ax.set_ylabel(r'Count rate [s$^{-1}$]')
ax.legend()
ax.grid()
plt.tight_layout()

plt.savefig('../report/fig/manualSpectra.pdf')
plt.show()
```

spectrumPC.py

```python
import os
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.special import erf
from scipy.stats import chi2 as chi2_dist


# ========================
# Settings
# ========================
DATA_FILE = "../results/pc/specComptplusPeak.csv"
FIG_DIR = "../report/fig"
os.makedirs(FIG_DIR, exist_ok=True)

SHOW_PLOTS = True


# ========================
# Helper functions
# ========================
def gauss_const(x, A, mu, sigma, C):
    return A * np.exp(-(x - mu) ** 2 / (2 * sigma ** 2)) + C


def gauss_linear_bg(x, A, mu, sigma, m, c):
    return A * np.exp(-(x - mu) ** 2 / (2 * sigma ** 2)) + m * x + c


def gauss_quadratic_bg(x, A, mu, sigma, q, m, c):
    return A * np.exp(-(x - mu) ** 2 / (2 * sigma ** 2)) + q * x**2 + m * x + c


def linear(x, a, b):
    return a + b * x


def compton_edge_model(E, A, Ec, sigma, a, b):
    return a + b * E + 0.5 * A * (1 - erf((E - Ec) / (np.sqrt(2) * sigma)))

```

```
41
42 def chi2_and_dof(y, yfit, sigma, npar):
43     chi2 = np.sum(((y - yfit) / sigma) ** 2)
44     dof = len(y) - npar
45     return chi2, dof
46
47
48 def finish_plot(filename):
49     plt.tight_layout()
50     plt.savefig(os.path.join(FIG_DIR, filename))
51     if SHOW_PLOTS:
52         plt.show()
53     else:
54         plt.close()
55
56
57 # =========================
58 # Load MCA spectrum
59 # =========================
60 data = np.loadtxt(DATA_FILE, skiprows=13).T
61
62 cutoff = 1500
63 channels_all = data[0][:cutoff]
64 counts_all = data[2][:cutoff]
65
66 sigma_counts_all = np.sqrt(np.maximum(counts_all, 1))
67
68
69 # =========================
70 # 1) Raw MCA spectrum in channels
71 # =========================
72 fit_ranges_channels = [
73     (510, 604),     # 511 keV peak
74     (1229, 1391)    # 1275 keV peak
75 ]
76
77 p0_channels = [
78     [1150, 560, 20, 40],    # [A, mu, sigma, C]
79     [110, 1316, 28, 5]
80 ]
81
```

```
channel_peak_positions = []
channel_peak_errors = []
channel_fit_results = []

fig = plt.figure()
ax = fig.add_subplot()

ax.plot(channels_all, counts_all, label="MCA spectrum", linewidth=1.8,
    zorder=1)

for i, (xmin, xmax) in enumerate(fit_ranges_channels):
    mask = (channels_all >= xmin) & (channels_all <= xmax)
    x = channels_all[mask]
    y = counts_all[mask]
    sy = sigma_counts_all[mask]

    popt, pcov = curve_fit(
        gauss_const,
        x,
        y,
        p0=p0_channels[i],
        sigma=sy,
        absolute_sigma=True,
        maxfev=10000
    )

    perr = np.sqrt(np.diag(pcov))

    yfit = gauss_const(x, *popt)
    chi2, dof = chi2_and_dof(y, yfit, sy, 4)
    chi2_red = chi2 / dof
    p_value = chi2_dist.sf(chi2, dof)

    channel_peak_positions.append(popt[1])
    channel_peak_errors.append(perr[1])
    channel_fit_results.append((popt, perr, chi2, dof, chi2_red, p_value))

    print(f"\n=== CHANNEL FIT PEAK {i+1} ===")
    print(f"A     = {popt[0]:.3f} pm {perr[0]:.3f}")
    print(f"mu    = {popt[1]:.3f} pm {perr[1]:.3f} channels")
    print(f"sigma = {popt[2]:.3f} pm {perr[2]:.3f} channels")
```

```
    print(f"C     = {popt[3]:.3f} pm {perr[3]:.3f}")
    print(f"chi2  = {chi2:.3f}")
    print(f"DoF   = {dof}")
    print(f"chi2_red = {chi2_red:.3f}")
    print(f"p-value = {p_value:.5f}")

    xfit = np.linspace(xmin, xmax, 500)
    label = "Gaussian + constant fit" if i == 0 else None
    ax.plot(
        xfit,
        gauss_const(xfit, *popt),
        color="red",
        linewidth=2.2,
        label=label,
        zorder=3
    )

ax.set_title("MCA spectrum with Gaussian peak fits", fontsize=11)
ax.set_xlabel("Channels [a.u.]")
ax.set_ylabel("Counts [1]")
ax.legend()
ax.grid()

finish_plot("mca_spectrum_channels.pdf")

channel_peak_positions = np.array(channel_peak_positions)
channel_peak_errors = np.array(channel_peak_errors)


# =========================
# 2) Channel -> energy calibration
# =========================
E_cal = np.array([0.0, 511.0, 1275.0])
ch_cal = np.array([0.0, channel_peak_positions[0], channel_peak_positions[1]])
sch_cal = np.array([1.0, channel_peak_errors[0], channel_peak_errors[1]])

# Weighted linear fit: E = a + b * channel
coeffs, cov = np.polyfit(ch_cal, E_cal, deg=1, w=1 / sch_cal, cov=True)
b_cal, a_cal = coeffs
sb_cal, sa_cal = np.sqrt(np.diag(cov))

```

```
print("\n=== CALIBRATION ===")
print(f"E(channel) = ({a_cal:.4f} pm {sa_cal:.4f}) + ({b_cal:.6f} pm {sb_cal:.6f}) * channel")
print(f"Intercept compatibility with 0: |a|/sigma_a = {abs(a_cal) / sa_cal:.2f}")

fig = plt.figure()
ax = fig.add_subplot()

ax.errorbar(
    ch_cal,
    E_cal,
    xerr=sch_cal,
    fmt="o",
    markersize=5,
    capsize=3,
    linestyle="None",
    label="Calibration points"
)

xfit = np.linspace(0, max(ch_cal) * 1.05, 300)
ax.plot(
    xfit,
    linear(xfit, a_cal, b_cal),
    color="red",
    linewidth=2.2,
    label="Linear fit"
)

ax.set_title("Channel-to-energy calibration", fontsize=11)
ax.set_xlabel("Channels [a.u.]")
ax.set_ylabel("Energy [keV]")
ax.legend()
ax.grid()

finish_plot("channel_energy_calibration.pdf")

# Convert full spectrum to energy
energy_all = linear(channels_all, a_cal, b_cal)


```

```
# =========================
# 3) Fit of the low-energy Pb X-ray contribution
# =========================
# The low-energy structure is broad and sits on a curved background.
# Therefore, a Gaussian + quadratic background is used.

search_min = 0
search_max = 120

search_mask = (energy_all >= search_min) & (energy_all <= search_max)
E_search = energy_all[search_mask]
N_search = counts_all[search_mask]

mu0_pb = E_search[np.argmax(N_search)]

# Fit window around observed low-energy structure
fit_half_width_left = 30
fit_half_width_right = 45

emin_pb = max(search_min, mu0_pb - fit_half_width_left)
emax_pb = min(search_max, mu0_pb + fit_half_width_right)

mask_pb = (energy_all >= emin_pb) & (energy_all <= emax_pb)
x_pb = energy_all[mask_pb]
y_pb = counts_all[mask_pb]
sy_pb = sigma_counts_all[mask_pb]

A0_pb = np.max(y_pb) - np.min(y_pb)
sigma0_pb = 10.0
q0_pb = 0.0
m0_pb = 0.0
c0_pb = np.min(y_pb)

popt_pb, pcov_pb = curve_fit(
    gauss_quadratic_bg,
    x_pb,
    y_pb,
    p0=[A0_pb, mu0_pb, sigma0_pb, q0_pb, m0_pb, c0_pb],
    sigma=sy_pb,
    absolute_sigma=True,
    bounds=(
```

```
        [0, emin_pb, 1, -np.inf, -np.inf, -np.inf],
        [np.inf, emax_pb, 50, np.inf, np.inf, np.inf]
    ),
    maxfev=50000
)

perr_pb = np.sqrt(np.diag(pcov_pb))

yfit_pb = gauss_quadratic_bg(x_pb, *popt_pb)
chi2_pb, dof_pb = chi2_and_dof(y_pb, yfit_pb, sy_pb, 6)
chi2_red_pb = chi2_pb / dof_pb
p_value_pb = chi2_dist.sf(chi2_pb, dof_pb)

lead_xray_position = popt_pb[1]

print("\n=== Pb X-RAY CONTRIBUTION FIT ===")
print(f"Fit range: {emin_pb:.1f} to {emax_pb:.1f} keV")
print(f"A     = {popt_pb[0]:.3f} pm {perr_pb[0]:.3f}")
print(f"mu    = {popt_pb[1]:.3f} pm {perr_pb[1]:.3f} keV")
print(f"sigma = {popt_pb[2]:.3f} pm {perr_pb[2]:.3f} keV")
print(f"q     = {popt_pb[3]:.6f} pm {perr_pb[3]:.6f}")
print(f"m     = {popt_pb[4]:.5f} pm {perr_pb[4]:.5f}")
print(f"c     = {popt_pb[5]:.3f} pm {perr_pb[5]:.3f}")
print(f"chi2  = {chi2_pb:.3f}")
print(f"DoF   = {dof_pb}")
print(f"chi2_red = {chi2_red_pb:.3f}")
print(f"p-value = {p_value_pb:.5f}")

fig = plt.figure()
ax = fig.add_subplot()

ax.errorbar(
    x_pb,
    y_pb,
    yerr=sy_pb,
    fmt="o",
    markersize=4,
    capsize=2,
    linestyle="None",
    label="Data"
)
```

```
284
285 xfit_pb = np.linspace(emin_pb, emax_pb, 500)
286 ax.plot(
287     xfit_pb,
288     gauss_quadratic_bg(xfit_pb, *popt_pb),
289     color="orange",
290     linewidth=2.0,
291     label="Gaussian + quadratic background"
292 )
293
294 ax.set_title("Fit of the Pb X-ray contribution", fontsize=11)
295 ax.set_xlabel("Energy [keV]")
296 ax.set_ylabel("Counts [1]")
297 ax.legend()
298 ax.grid()
299
300 finish_plot("pb_xray_contribution_fit.pdf")
301
302
303 # =========================
304 # 4) Calibrated MCA spectrum with identified structures
305 # =========================
306 fig = plt.figure()
307 ax = fig.add_subplot()
308
309 ax.plot(
310     energy_all,
311     counts_all,
312     linewidth=1.8,
313     label="Calibrated MCA spectrum",
314     zorder=2
315 )
316
317 compton_edge_511 = 341
318 photopeak_511 = 511
319 compton_edge_1275 = 1062
320 photopeak_1275 = 1275
321
322 ax.axvline(lead_xray_position, color="black", linestyle="--", linewidth=1.2,
    alpha=0.9)
```

```
ax.axvline(compton_edge_511, color="black", linestyle="--", linewidth=1.2, alpha=0.9)
ax.axvline(photopeak_511, color="black", linestyle="--", linewidth=1.2, alpha=0.9)
ax.axvline(compton_edge_1275, color="black", linestyle="--", linewidth=1.2, alpha=0.9)
ax.axvline(photopeak_1275, color="black", linestyle="--", linewidth=1.2, alpha=0.9)

ymax = np.max(counts_all)

ax.text(lead_xray_position + 8, 0.90 * ymax, "Pb X-ray contribution",
        rotation=90, va="top", ha="left")
ax.text(compton_edge_511 + 8, 0.90 * ymax, "Compton edge  341 keV",
        rotation=90, va="top", ha="left")
ax.text(photopeak_511 + 8, 0.90 * ymax, "511 keV peak",
        rotation=90, va="top", ha="left")
ax.text(compton_edge_1275 + 8, 0.90 * ymax, "Compton edge  1062 keV",
        rotation=90, va="top", ha="left")
ax.text(photopeak_1275 + 8, 0.90 * ymax, "1275 keV peak",
        rotation=90, va="top", ha="left")

ax.set_title("Calibrated MCA spectrum", fontsize=11)
ax.set_xlabel("Energy [keV]")
ax.set_ylabel("Counts [1]")
ax.set_xlim(0, 1400)
ax.set_ylim(0, 1.05 * ymax)
ax.legend()
ax.grid()

finish_plot("mca_spectrum_energy.pdf")


# =========================
# 5) Photopeak fits in energy
# =========================
fit_ranges_energy = [
    (470, 600),
    (1180, 1360)
]

```

```
p0_energy = [
    [1150, 511, 18, 40],
    [110, 1275, 25, 5]
]

titles = [
    "Zoom of the 511 keV photopeak",
    "Zoom of the 1275 keV photopeak"
]

save_names = [
    "peak_511_zoom.pdf",
    "peak_1275_zoom.pdf"
]

for i, (emin, emax) in enumerate(fit_ranges_energy):
    mask = (energy_all >= emin) & (energy_all <= emax)
    x = energy_all[mask]
    y = counts_all[mask]
    sy = sigma_counts_all[mask]

    popt, pcov = curve_fit(
        gauss_const,
        x,
        y,
        p0=p0_energy[i],
        sigma=sy,
        absolute_sigma=True,
        maxfev=10000
    )

    perr = np.sqrt(np.diag(pcov))

    bin_width = np.mean(np.diff(np.sort(x)))
    A_fit = popt[0]
    sigma_fit = abs(popt[2])

    N_peak = A_fit * np.sqrt(2 * np.pi) * sigma_fit / bin_width
    mu_err_mean = sigma_fit / np.sqrt(N_peak)

    yfit = gauss_const(x, *popt)
```

```
    chi2, dof = chi2_and_dof(y, yfit, sy, 4)
    chi2_red = chi2 / dof
    p_value = chi2_dist.sf(chi2, dof)

    fig = plt.figure()
    ax = fig.add_subplot()

    ax.errorbar(
        x,
        y,
        yerr=sy,
        fmt="o",
        markersize=4,
        capsize=2,
        linestyle="None",
        label="Data"
    )

    xfit = np.linspace(emin, emax, 500)
    ax.plot(
        xfit,
        gauss_const(xfit, *popt),
        color="orange",
        linewidth=2.0,
        label="Gaussian + constant"
    )

    ax.set_title(titles[i], fontsize=11)
    ax.set_xlabel("Energy [keV]")
    ax.set_ylabel("Counts [1]")
    ax.legend()
    ax.grid()

    finish_plot(save_names[i])

    print(f"\n=== ENERGY FIT PEAK {i+1} ===")
    print(f"Fit range: {emin} to {emax} keV")
    print(f"A     = {popt[0]:.3f} pm {perr[0]:.3f}")
    print(f"mu    = {popt[1]:.3f} pm {mu_err_mean:.3f} keV  (error of mean)")
    print(f"sigma = {popt[2]:.3f} pm {perr[2]:.3f} keV")
    print(f"C     = {popt[3]:.3f} pm {perr[3]:.3f}")
```

```
    print(f"N_peak = {N_peak:.1f}")
    print(f"chi2   = {chi2:.3f}")
    print(f"DoF    = {dof}")
    print(f"chi2_red = {chi2_red:.3f}")
    print(f"p-value = {p_value:.5f}")


# =========================
# 6) Compton edge fits
# =========================
compton_edge_fits = [
    {
        "name": "511 keV",
        "emin": 250,
        "emax": 420,
        "p0": [250, 341, 15, 80, -0.1],
        "bounds": (
            [0, 300, 1, -np.inf, -np.inf],
            [np.inf, 380, 80, np.inf, np.inf]
        ),
        "title": "Fit of the 511 keV Compton edge",
        "save": "compton_edge_511_fit.pdf"
    },
    {
        "name": "1275 keV",
        "emin": 900,
        "emax": 1150,
        "p0": [60, 1062, 25, 80, -0.03],
        "bounds": (
            [0, 1000, 1, -np.inf, -np.inf],
            [np.inf, 1120, 120, np.inf, np.inf]
        ),
        "title": "Fit of the 1275 keV Compton edge",
        "save": "compton_edge_1275_fit.pdf"
    }
]

for cfg in compton_edge_fits:
    emin = cfg["emin"]
    emax = cfg["emax"]

```

```
    mask = (energy_all >= emin) & (energy_all <= emax)
    x = energy_all[mask]
    y = counts_all[mask]
    sy = sigma_counts_all[mask]

    popt, pcov = curve_fit(
        compton_edge_model,
        x,
        y,
        p0=cfg["p0"],
        sigma=sy,
        absolute_sigma=True,
        bounds=cfg["bounds"],
        maxfev=20000
    )

    perr = np.sqrt(np.diag(pcov))

    yfit = compton_edge_model(x, *popt)
    chi2, dof = chi2_and_dof(y, yfit, sy, 5)
    chi2_red = chi2 / dof
    p_value = chi2_dist.sf(chi2, dof)

    fig = plt.figure()
    ax = fig.add_subplot()

    ax.errorbar(
        x,
        y,
        yerr=sy,
        fmt="o",
        markersize=4,
        capsize=2,
        linestyle="None",
        label="Data"
    )

    xfit = np.linspace(emin, emax, 500)
    ax.plot(
        xfit,
        compton_edge_model(xfit, *popt),
```

```
        color="orange",
        linewidth=2.0,
        label="Edge fit"
    )

    ax.set_title(cfg["title"], fontsize=11)
    ax.set_xlabel("Energy [keV]")
    ax.set_ylabel("Counts [1]")
    ax.legend()
    ax.grid()

    finish_plot(cfg["save"])

    print(f"\n=== COMPTON EDGE FIT ({cfg['name']}) ===")
    print(f"Fit range: {emin} to {emax} keV")
    print(f"A     = {popt[0]:.3f} pm {perr[0]:.3f}")
    print(f"Ec    = {popt[1]:.3f} pm {perr[1]:.3f} keV")
    print(f"sigma = {popt[2]:.3f} pm {perr[2]:.3f} keV")
    print(f"a     = {popt[3]:.3f} pm {perr[3]:.3f}")
    print(f"b     = {popt[4]:.5f} pm {perr[4]:.5f}")
    print(f"chi2  = {chi2:.3f}")
    print(f"DoF   = {dof}")
    print(f"chi2_red = {chi2_red:.3f}")
    print(f"p-value = {p_value:.5f}")
```

spectrumCoincidence.py

```
from dataAnalyser import *
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.stats import chi2 as chi2_dist


# =========================
# helper functions
# =========================
def gauss_const(x, A, mu, sigma, C):
    return A * np.exp(-(x - mu)**2 / (2 * sigma**2)) + C

def chi2_and_dof(y, yfit, sigma, npar):
    chi2 = np.sum(((y - yfit) / sigma) ** 2)
    dof = len(y) - npar
    return chi2, dof

def channel_to_energy(c, a0, a1):
    return a0 + a1 * c

def propagate_peak_center_to_energy(mu_c, smu_c, a0, sa0, a1, sa1,
    cov_a0_a1=0.0):
    """
    E = a0 + a1 * mu_c
    var(E) = var(a0) + mu_c^2 var(a1) + a1^2 var(mu_c) + 2 mu_c cov(a0,a1)
    """
    mu_E = a0 + a1 * mu_c
    var_E = sa0**2 + (mu_c**2) * sa1**2 + (a1**2) * smu_c**2 + 2.0 * mu_c *
    cov_a0_a1
    smu_E = np.sqrt(max(var_E, 0.0))
    return mu_E, smu_E

def propagate_sigma_to_energy(sigma_c, ssigma_c, a1, sa1):
    """
    sigma_E = |a1| * sigma_c
    approximate propagation:
    var(sigma_E) = sigma_c^2 var(a1) + a1^2 var(sigma_c)
    """
    sigma_E = abs(a1) * sigma_c
```

```
    var_sigma_E = (sigma_c**2) * sa1**2 + (a1**2) * ssigma_c**2
    ssigma_E = np.sqrt(max(var_sigma_E, 0.0))
    return sigma_E, ssigma_E


# =========================
# FINAL CALIBRATION
# =========================
# Use exactly ONE final calibration here.
# I set this to the calibration currently quoted in your report.
# If you later decide on a different final calibration, change only these
    numbers.

a0 = -0.1      # keV
sa0 = 1.4      # keV
a1 = 0.954     # keV / channel
sa1 = 0.018    # keV / channel

# If you know Cov(a0, a1), insert it here in keV^2/channel.
# Otherwise leave it at 0.0.
cov_a0_a1 = 0.0


# =========================
# Load coincidence spectrum
# =========================
data = np.loadtxt('../results/pc/specCoinc.csv', skiprows=13).T

cutoff = 1350
channels = data[0][:cutoff]
counts = data[2][:cutoff]

# Poisson uncertainties
scounts = np.sqrt(np.maximum(counts, 1))


# =========================
# 1) Raw coincidence spectrum in channels
# =========================
fig = plt.figure()
ax = fig.add_subplot()
```

```
79
80  ax.plot(channels, counts, linewidth=1.8, label='Coincidence spectrum')
81
82  ax.set_title('Coincidence spectrum in MCA channels', fontsize=11)
83  ax.set_xlabel('ADC channel')
84  ax.set_ylabel('Counts')
85  ax.legend()
86  ax.grid()
87  plt.tight_layout()
88
89  plt.savefig("../report/fig/coincidence_spectrum_channels.pdf")
90  plt.show()
91
92
93  # =========================
94  # 2) Convert full spectrum to energy for display
95  # =========================
96  energy = channel_to_energy(channels, a0, a1)
97
98  fig = plt.figure()
99  ax = fig.add_subplot()
100
101 ax.plot(energy, counts, linewidth=1.8, label='Coincidence spectrum')
102
103 ax.set_title('Calibrated coincidence spectrum', fontsize=11)
104 ax.set_xlabel('Energy [keV]')
105 ax.set_ylabel('Counts')
106 ax.set_xlim(0, 700)
107 ax.legend()
108 ax.grid()
109 plt.tight_layout()
110
111 plt.savefig("../report/fig/coincidence_spectrum_energy_linear.pdf")
112 plt.show()
113
114
115 # =========================
116 # 3) Coincidence spectrum in energy (log scale)
117 # =========================
118 fig = plt.figure()
119 ax = fig.add_subplot()
```

```
120
121 counts_log = np.maximum(counts, 1)
122
123 ax.semilogy(energy, counts_log, linewidth=1.8, label='Coincidence spectrum')
124
125 ax.set_title('Calibrated coincidence spectrum (log scale)', fontsize=11)
126 ax.set_xlabel('Energy [keV]')
127 ax.set_ylabel('Counts')
128 ax.set_xlim(0, 700)
129 ax.legend()
130 ax.grid(which='both')
131 plt.tight_layout()
132
133 plt.savefig("../report/fig/coincidence_spectrum_energy_log.pdf")
134 plt.show()
135
136
137 # =========================
138 # 4) Fit coincidence peak IN CHANNELS
139 # =========================
140 # Choose the peak window in energy for convenience,
141 # but perform the fit in ADC channels.
142 emin, emax = 470.0, 550.0
143 cmin = (emin - a0) / a1
144 cmax = (emax - a0) / a1
145
146 mask = (channels >= cmin) & (channels <= cmax)
147
148 x_c = channels[mask]
149 y = counts[mask]
150 sy = np.sqrt(np.maximum(y, 1))
151
152 # start values in channels
153 mu0_c = (511.0 - a0) / a1
154 sigma0_c = 15.0 / abs(a1)
155 p0 = [850.0, mu0_c, sigma0_c, 30.0]
156
157 popt, pcov = curve_fit(
158     gauss_const,
159     x_c,
160     y,
```

```
    p0=p0,
    sigma=sy,
    absolute_sigma=True,
    maxfev=10000
)

perr = np.sqrt(np.diag(pcov))
yfit = gauss_const(x_c, *popt)

chi2, dof = chi2_and_dof(y, yfit, sy, 4)
chi2_red = chi2 / dof
p_value = chi2_dist.sf(chi2, dof)

A_c, mu_c, sigma_c, C_c = popt
sA_c, smu_c, ssigma_c, sC_c = perr

# propagate to energy
mu_E, smu_E = propagate_peak_center_to_energy(mu_c, smu_c, a0, sa0, a1, sa1,
    cov_a0_a1)
sigma_E, ssigma_E = propagate_sigma_to_energy(sigma_c, ssigma_c, a1, sa1)


# =========================
# 5) Plot fit on energy axis
# =========================
fig = plt.figure()
ax = fig.add_subplot()

x_E = channel_to_energy(x_c, a0, a1)

ax.errorbar(
    x_E, y, yerr=sy,
    fmt='o',
    markersize=4,
    capsize=2,
    linestyle='None',
    label='Data'
)

xfit_c = np.linspace(np.min(x_c), np.max(x_c), 500)
yfit_plot = gauss_const(xfit_c, *popt)
```

```
xfit_E = channel_to_energy(xfit_c, a0, a1)

ax.plot(xfit_E, yfit_plot, color='orange', linewidth=2.0, label='Gaussian + constant')

ax.set_title('Fit of the coincidence peak', fontsize=11)
ax.set_xlabel('Energy [keV]')
ax.set_ylabel('Counts')
ax.legend()
ax.grid()
plt.tight_layout()

plt.savefig("../report/fig/coincidence_peak_fit.pdf")
plt.show()


# =========================
# 6) Output
# =========================
print("\n=== CALIBRATION USED ===")
print(f"E(channel) = ({a0:.4f} +- {sa0:.4f}) + ({a1:.6f} +- {sa1:.6f}) * channel")
print(f"Cov(a0, a1) = {cov_a0_a1:.6e}")

print("\n=== COINCIDENCE PEAK FIT (IN CHANNELS) ===")
print(f"Fit range: {cmin:.2f} to {cmax:.2f} channels")
print(f"A         = {A_c:.3f} +- {sA_c:.3f}")
print(f"mu_c      = {mu_c:.3f} +- {smu_c:.3f} channels")
print(f"sigma_c   = {sigma_c:.3f} +- {ssigma_c:.3f} channels")
print(f"C         = {C_c:.3f} +- {sC_c:.3f}")
print(f"chi2      = {chi2:.3f}")
print(f"DoF       = {dof}")
print(f"chi2_red  = {chi2_red:.3f}")
print(f"p-value   = {p_value:.5f}")

print("\n=== COINCIDENCE PEAK POSITION (PROPAGATED TO ENERGY) ===")
print(f"mu_E      = {mu_E:.3f} +- {smu_E:.3f} keV")
print(f"sigma_E   = {sigma_E:.3f} +- {ssigma_E:.3f} keV")
```