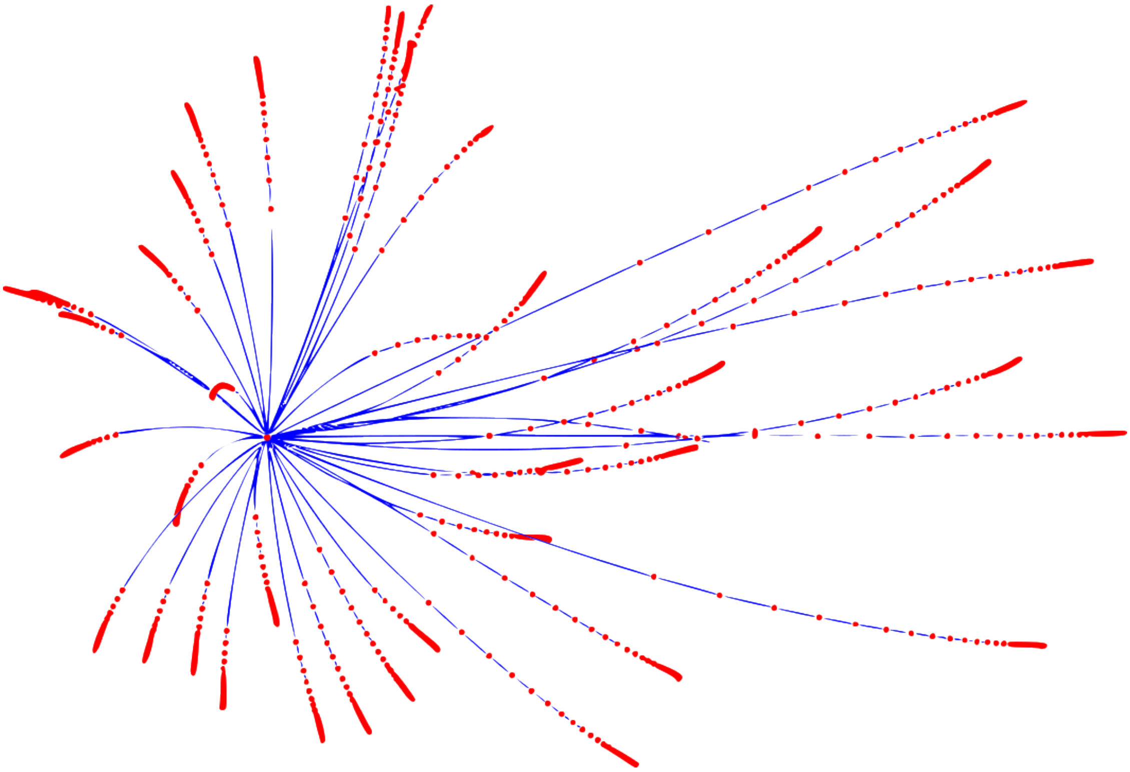
CHALMERS
UNIVERSITY OF TECHNOLOGY



Sorting Simulations

Preparing fission experiments through simulations

Master's thesis in Physics

BJÖRN JOHANSSON

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Sorting Simulations

Preparing fission experiments through simulations

Björn Johansson

Department of Physics
Chalmers University of Technology
Gothenburg, Sweden 2025

Sorting Simulations
Preparing fission experiments through simulations
Björn Johansson

Supervisor: Andreas Heinz, Department of Physics
Examiner: Andreas Heinz, Department of Physics

Master's Thesis 2025
Department of Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Simulation of α particles from a point source inside a magnetic field. Made with GEANT4.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Sorting Simulations
Preparing fission experiments through simulations
Björn Johansson
Department of Physics
Chalmers University of Technology

Abstract

This master thesis focuses on development and tests of the analysis software to include newly added detectors before the start of a novel fission experiment to study ^{230}Ac scheduled for July 2025 at the ISOLDE facility at CERN. By converting simulated data to mimic the actual experimental data format it is possible to test the analysis software in advance. Comparing data output from the software with the same data before the conversion allows to spot bugs in the software. Thus the software can be prepared and tested before the experiment, as it removes the need to use measured data for development.

The experiment will study fission of ^{230}Ac after (d,p) reactions of ^{229}Ac at 8 MeV/u. While the Si array for detecting emitted protons has been used in many experiments, handling of three new detectors must be prepared. Two Si CD detectors for fission fragments, four position sensitive Si strip detectors for luminosity monitoring as well as 36 $CeBr_3$ crystals for γ rays.

The work in this thesis allow the analysis software to be used as a diagnostic tool for these new detectors already during the setup phase prior to the experiment.

Keywords: ISOLDE, simulations, Monte Carlo, Geant4, Si detectors, calibration, DAQ

Acknowledgements

I would like to thank my supervisor and examiner, Andreas Heinz, for assisting me during this thesis project and helping me to understand the physics that I have been pondering about. I also wish to express my gratitude to Håkan Johansson for his detailed feedback on this thesis and for answering my programming questions as well as teaching me about data aquisition in particular. I want to thank Hans Törnqvist as well for helping me with solving programming issues when they arose. I am grateful for all the interesting physics discussions and anecdotes Björn Jonson shared with me. I also appreciate all the help I recieved from the two PhD students Anna Kawęcka and Maria Vittoria Managlia. I appreciate many interesting discussions with both Erik Jensen and the amanuens Alice Svärdström. It has been fun and exciting to work with the entire group!

I also want to thank the ISS collaboration for all the help, escpecially to Liam Gaffney for helping me with implementing my contributions to the analysis program and for answering my questions related to ISS.

Björn Johansson, Göteborg, Juni 2025

Contents

1

Modern Experiments

Modern day subatomic physics experiments often take place at large accelerator complexes like CERN or GSI. These facilities can produce exotic particles and ions at very high energies and intensities, making it possible for researchers to test theoretical predictions against experimental data and deepen our understanding of the universe's fundamental structure. Such experiments often require very complex setups with many, often highly segmented, detectors that need to be placed where they maximise the probability of measuring the physics observables of interest. The collected signals are transmitted to data aquisition (DAQ) systems in order to record the measured data for subsequent analysis. Advanced analysis software is then used to reconstruct and process the data, extracting meaningful physics of interest. Given the complexity and scale of these experiments, the raw data recorded can amount to several terabytes necessitating robust data handling and storage solutions.

Given the large scope of these experiments, a lot of preparations need to be made as researchers are only given a limited amount of beam time at the facilities to conduct their experiment. Tests beforehand need to be made in order to ensure the best placement of detectors and that all parts, including software, are in working order during the limited time that an experiment takes place. Everything that can be prepared beforehand is time saved that can be spent on solving issues that may arise during the experiment.

One way to determine good detector positions is to simulate the experiment in advance. This is a cheap way to test several builds without the need to physically construct each of them and it does not need a DAQ system or fully developed analysis software. We can, however, also use simulations to test the analysis software by mimicing the raw data output from the DAQ [1]. Then, the software cannot differentiate if the given data comes from a real experiment or from a simulation. This means that we can test the software with simulated events which would otherwise require a particle accelerator. This means that the analysis tools can be tested before we collect any data using the limited exotic beams.

An upcoming experiment this summer is IS739 which will take place at ISOLDE[1]

---

[1] Isotope mass Separator On-Line facility. A rumor is that danish scientists suggested that

facility at CERN. The experiment aims to study fission of ^{230}Ac, a neutron rich isotope with a short lifetime, and will combine new equipment with an already present experimental setup, the ISOLDE Solenoidal Spectrometer (ISS). Extensive preparations are underway before the experiment to test the new equipment and integrate it with the existing setup. A method to test analysis software with simulated data is described in Chapter 3. In the next subsection, the motivation for the experiment IS739 is summarised.

1.1 The place where heavy elements are born

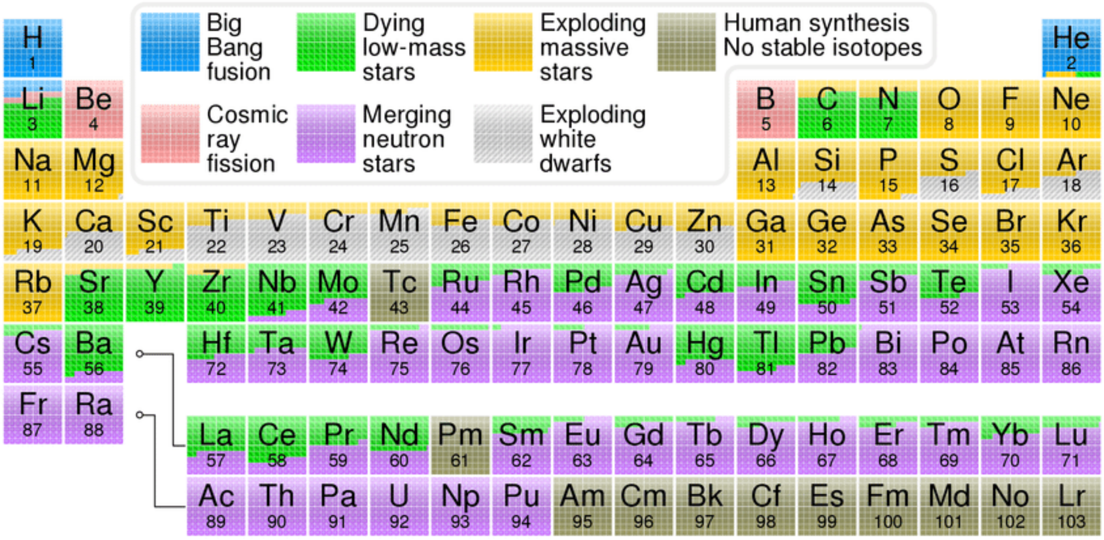


Figure 1.1: Periodic table where the colours of each element depicts the astrophysical site of its origin. Figure from Wikimedia Commons [3].

In 2017, observations of a neutron star merger were made [4], a phenomenon were two neutron stars collide and create a rare environment with an enormous flux of free neutrons. Light particles in this neutron flux can capture neutrons in quick succession, increasing their mass. As the particles capture more neutrons, they become heavier and more unstable and at some point undergo β-decay which increases the nuclear charge by one unit (and also emits an electron as well as an anti-electron neutrino). For each neutron captured, the nuclei will become heavier and for each beta decay the nuclei will gain one nuclear charge unit. This process is referred to as the rapid neutron capture process, or *r-process*, and is one out of

---

the DE, which completes the abbreviation, should stand for Danish Engineering [2].

several processes in which elements heavier than iron are made [5]. Many of the heaviest elements can be created in neutron star mergers, indicated by the purple colour in Fig 1.1. At some point, these neutron rich nuclei become heavy enough to undergo fission which imposes a limit on the mass of heavy elements which can be formed in our universe.

The two fragments formed during fission can subsequently pick up more neutrons, continuing the r-process. Exploring the role fission has in the r-process could improve our understanding of the creation of the heaviest of elements. However, the r-process involves nuclei far from the line of stability in the nuclear chart and these nuclei have very short lifetimes. This makes it challenging or even impossible to create them in a laboratory. It is important to note that such isotopes of interest need to be produced during the experiment and as close to the detectors as possible. Otherwise they would decay, or possibly fission, before they are surrounded by detectors that can measure them.

IS739 is an early step towards exploring the role fission has in the r-process. By measuring fission events of neutron rich ^{230}Ac nuclei we can test how well theoretical models describe fission of nuclei with higher neutron excess. It is a good candidate because fission of ^{230}Ac has not been studied before and the isotope is neutron rich, but still relatively close to the stability line. The isotope ^{230}Ac has a half life of 122 ± 3 s [6] before it beta decays into ^{230}Th. This is too short to produce a target made of ^{230}Ac and to perform an experiment lasting several days with it. Instead, a beam of ^{230}Ac with sufficient intensity to perform an experiment can be produced at the ISOLDE facility. In addition, most of the setup at ISS is already in place, apart from a set of new detectors added specifically to detect fission events.

1.2 From detector output to observables

For the IS739 experiment, four detector types will be used. The first one is the Liverpool on-axis silicon array (ISS array) which has been used for many experiments prior [7–9]. The other three detector types are: CD shaped silicon detectors to detect the fission fragments, resistive silicon strip detectors for luminosity measurments and $CeBr_3$ crystals to measure the energy of γ photons and these detectors are new to the setup. The analysis software, `ISSSort` [10], needs to be updated in order to process the data from the new detectors.

The data from the detectors are delivered by the DAQ in a compact data format to a storage device. A software then has to unpack and separate the data from each detector. That data can be analysed per detector and then the combination of data from multiple detectors allows to extract the wanted physics observables.

The work of this thesis focuses on integrating the new detectors in `ISSSort` and test the software before the experiment by using simulated data as a substitute for experimental data. The data from the simulations can either be directly analysed or converted and processed by `ISSSort`. The advantage of this scheme is that we can compare the data before and after `ISSSort`, making it easier to see if the data quality is preserved. The status of the software after finishing this project is that the data from all detectors can be unpacked in `ISSSort` and that the software is also able to analyse the data from the luminosity detectors, while the fragment and γ detectors have some analysis routines implemented. This is illustrated in Figure 1.2.

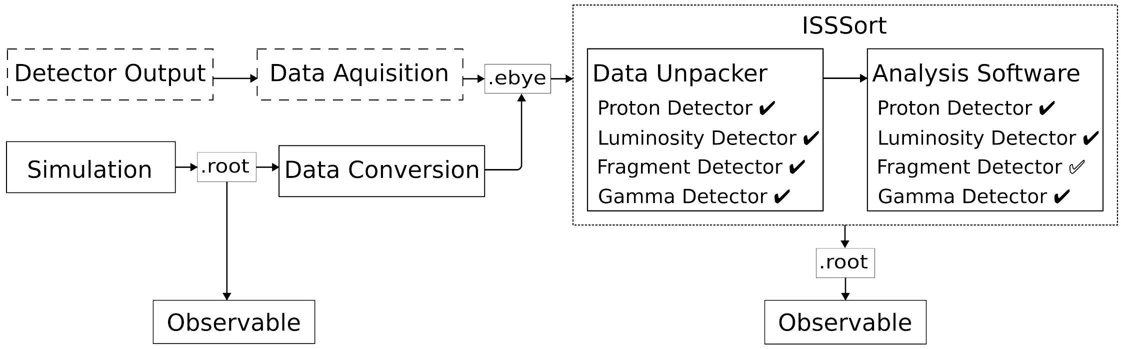


Figure 1.2: Data flow from detector output to analysis software. By using simulated data to emulate detector outputs it is possible to test the analysis by comparing the data before and after the data has been processed and without the need for experimental data. A filled check mark indicates that the corresponding detector is fully implemented, while a hollow check mark indicates the detector is partially integrated at the time of writing. For details of the satus before this thesis, see Figure 4.1.

1.3 Thesis outline

This thesis focuses on developing analysis routines for the experiment IS739 by using simulated data for tests. The current analysis software used at ISS is the program `ISSSort` which is the software that I have contributed to. Due to time constraints I have focused on energy reconstruction where each simulated event is separated by time to avoid the possibility of events being too close together and thus cannot be separated by the analysis software.

Chapter 2 describes some of the underlying physics for the fission experiment. It also describes how to measure and detect fission events during the experiment. The simulation and conversion software used in this thesis are described Chapter 3 and finally, in Chapter 4, upgrades to the analysis software are presented.

2

Fission in solenoidal spectrometers

In order to interpret simulations and verify if their outcome is reasonable, it is helpful to understand some of the underlying physics. This chapter describes the chain of events to obtain the desired information on fission of the exotic ^{230}Ac, illustraded in Figure 2.1, and also on Rutherford scattering which is typically used for normalisation when determining absolute cross sections. Special consideration is needed for kinematic correlations inside solenoidal spectrometers where helical trajectories in a magnetic field pose both unique challenges as well as unique possibilities.

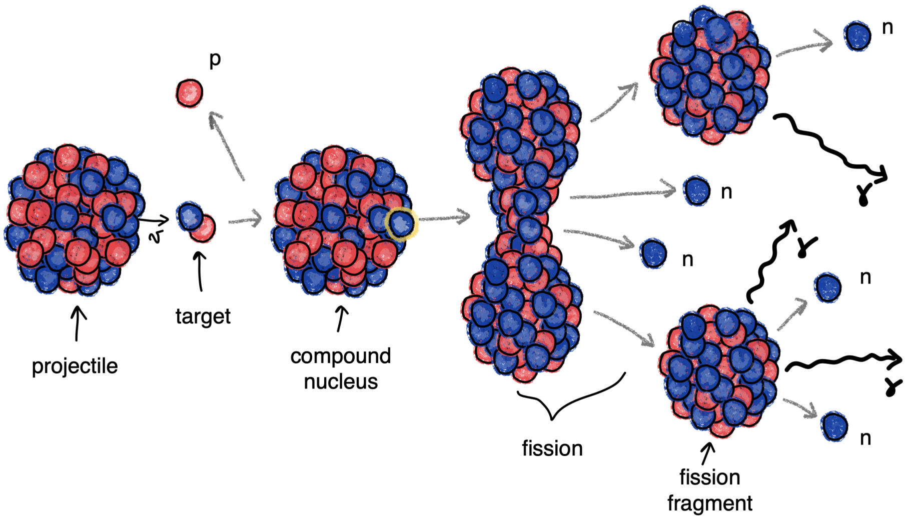


Figure 2.1: Illustration of fission events following a (d,p) reaction in inverse kinematics. The incoming projectile picks up a neutron from a deuterium target, forming a compound nucleus and ejects a proton. The compound nucleus continues in the forward direction and may undergo fission. Figure reprinted with permission from Ref. [11].

2.1 Nuclear reactions

The isotope ^{230}Ac has a half life of 122 ± 3 s [6] which makes it hard to study before it decays. In order to measure the fission probability of ^{230}Ac the first step is to accelerate ions of ^{229}Ac onto a target consisting partly of deuterium. When the incoming projectile interacts with a deuterium, there is a probability for a (d,p) reaction where a neutron from the deuterium transfers to the projectile, producing ^{230}Ac in an excited state and ejecting the remaining proton. The proton energy and momentum allow to reconstruct the excitation energy of the state populated in ^{230}Ac. The formed compound nucleus will try to reach a more stable state by the emission of light particles, e.g. neutrons, by the emission of γ-radiation, or, if the compound nucleus has an excitation energy above the fission barrier, it might undergo fission. This chain of events is depicted in Figure 2.1 and the goal of the experiment is to measure at which exitation energy of ^{230}Ac fission becomes a possible decay.

2.1.1 Inverse kinematics

A straightforward method to investigate nuclei is to accelerate a beam of light particles towards a target that contains nuclei from an isotope of interest. Experiments that study highly unstable isotopes cannot use this approach if the half life of the isotope is too short to produce and transport the target to the experimental setup and perform the experiment before a significant fraction of the isotope of interest decays. Given that targets with a short half-life are also highly radioactive, radiation safety also limits the activity of targets that can be used in such an experiment. If this so-called *direct reaction* approach is not possible, unstable isotopes of interest are instead produced near the experimental setup by a nuclear reaction and then accelerated towards a target containing the light particles. This is refered to as *inverse kinematics*.

2.1.2 Kinematics in solenoidal spectrometers

At the ISS, collisions of heavy ions with light nuclei occur inside a strong, homogenous, magnetic field which causes charged particles to move in a spiral trajectory. First, let us consider a stationary source of particles. The radius r of a trajectory that a charged particle travels from a stationary source can be described as

$$r = \frac{mv_\perp}{qeB}, \tag{2.1}$$

where m is the mass of the particle, $v_\perp$ is the velocity perpendicular to the magnetic field, qe is the charge of the particle in units of the elementary charge, and B is the magnetic field. The time it takes for the particle to make one turn is given by the cyclotron period

$$T_{\text{cyc}} = \frac{2\pi r}{v_{\perp}} = \frac{2\pi}{B}\frac{m}{qe}, \tag{2.2}$$

where equation (2.1) is used at the second equal sign. The cyclotron period depends only on the mass-to-charge ratio and the strength of the magnetic field. Thus, particles can often be identified by their single-turn time of flight [12].

The distance z a particle travels during one complete cycle is dependent on the initial velocity along the beam axis as

$$z = T_{\text{cyc}} v_z. \tag{2.3}$$

By measureing how far a particle, a proton for example, travels after one turn and its energy, we can extract the initial momentum of the particle.

2.1.3 Extracting excitation energy of compound nuclei

Because of the short half life of ^{230}Ac, it has to be produced close to the experimental setup. This is achieved by creating ^{229}Ac in a spallation reaction and accelerating it to a deuterium rich target at the experimental setup. During the collision, a neutron from the deuterium may be transferred to form an excited nucleus, ^{230}Ac*, by a (d,p) reaction while a proton is ejected. In a homogenous magnetic field, the ejected proton will move along a spiral, having traveled a distance z according to equation 2.3 after one full turn.

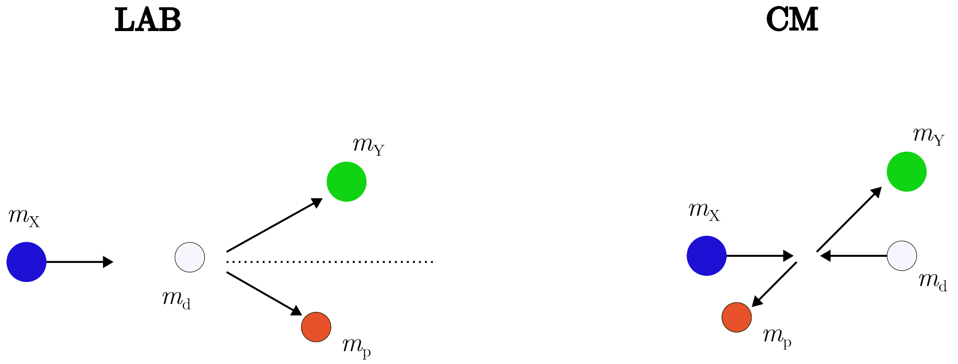


Figure 2.2: Two-body scattering of two incoming particles m_X and m_d and two outgoing particles, m_Y and m_p in both the laboratory frame (left) and the center-of-mass frame (right).

By measuring the energy of the proton E at the position z from the target and knowing the energy of the incoming beam, it is possible to calculate the excitation energy E^* of the compound nucleus by exploiting energy and momentum conservation. We are also interested in the emission angle in the center-of-mass (CM) frame, θ_{cm}. This means that the quantities z and E which are measured in the laboratory frame, need to be converted to the CM frame. Considering a two-body scattering as in the (d,p) reaction above, the incoming ion with a mass m_X reacts with a deuterium target with mass m_d and the outcoming ion m_Y and proton m_p are illustrated in both the laboratory frame and the CM frame in Figure 2.2.

The equations to extract E^* and θ_{cm} from E and z are

$$E^* = -m_Y + \sqrt{M_c^2 + m_p^2 - 2\gamma M_c(E - \alpha\beta z)} \tag{2.4}$$

and

$$\cos\theta_{cm} = \frac{\gamma(E\beta - \alpha z)}{\sqrt{\gamma^2(E - \alpha\beta z)^2 - m_p^2}}, \tag{2.5}$$

where M_c is the mass of the total system [11, 13]. With these equations and measurements of z and the proton energy, it is possible to reconstruct the excitation energy of the compound nucleus, in this case ^{230}Ac*. To measure the fission probability for a certain excitation energy we also need to detect the fission fragments, which is possible with a CD-shaped ΔE-E telescope detector.

2.1.4 Rutherford scattering as a diagnostic tool

Another reaction that takes place in the target is Rutherford scattering, which occurs when the incoming particle has an impact parameter larger than the nucleon-nucleon interaction. For such collisions, the reactions is purely of electromagnetic character. These scattering processes are very well understood and are a good way to determine the number of incoming beam particles per unit time during an experiment. In inverse kinematics, the target nuclei are scattered out of the target due to the high energy of the incoming beam by electromagnetic collisions.

Given an incoming beam intensity, I_0, traveling through a target density of N (cm^{-2}), the number of scattered particles J can be calculated as

$$J = I_0 N \sigma \tag{2.6}$$

where σ is the scattering cross-section. With a detector covering a certain solid angle dΩ, the differential cross-section $\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega}(\theta, \phi)$ is useful to get the probability for a particle being scattered with a certain angle (θ, ϕ) corresponding to a path towards the position of the detector. Measuring a yield Y in the detector, the experimental luminosity $I_0 N$ can be calulated as

$$I_0 N = \frac{Y}{\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega}\mathrm{d}\Omega\varepsilon_\pi\varepsilon_\delta} \tag{2.7}$$

if the beam purity ε_π and the absolute detector efficiency are known. The luminosity is easy experimentally determine, as it is difficult to separate I_0 and N from an experiment [13].

2.2 Measuring fission events

Oftentimes, we are interested in properties, which allow us to understand the structure of a nucleus but we cannot measure those properties directly. Instead, we measure the energy of particles emitted as a nucleus decays from one state to another, or the particles emitted after a decay process.

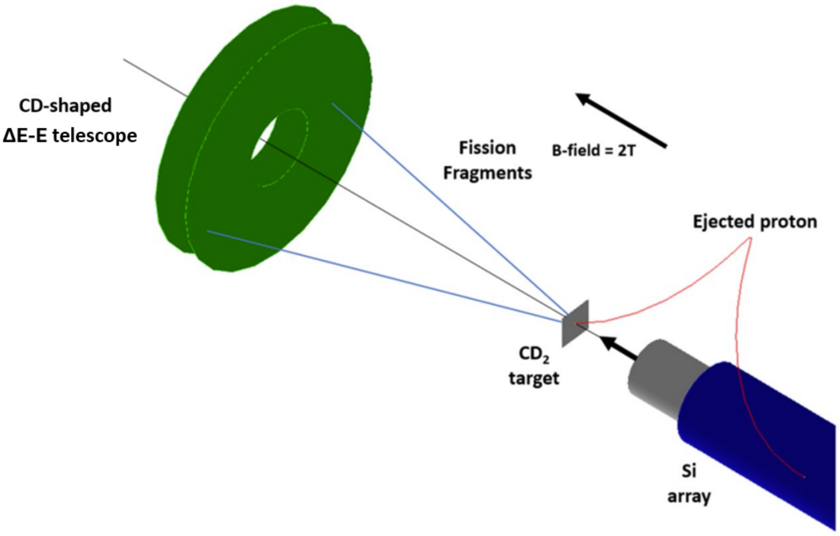
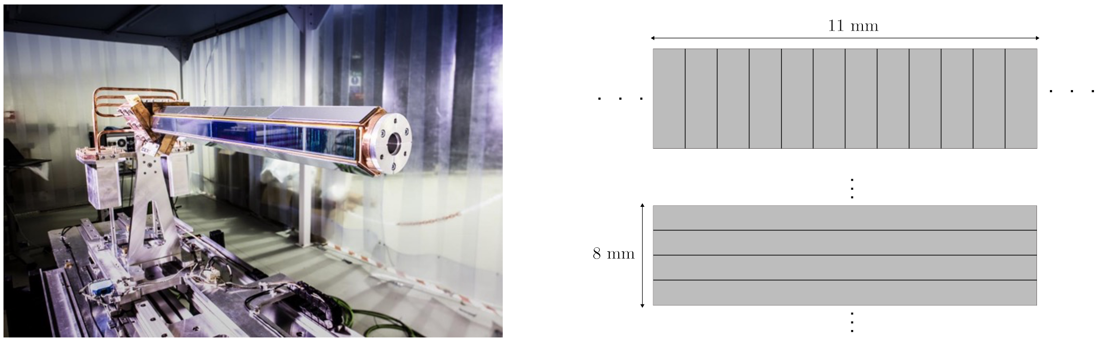


Figure 2.3: To detect fission events, a CD-shaped ΔE-E telescope is placed downstream from the target to measure the two fission fragments. Upstreams from the target sits a silicon array which detects ejected protons moving in the backwards direction. The array is hollow to let the incoming beam pass through to hit the target. Figure reprinted from Ref. [13].

When a particle impinges on a semiconducting detector, it creates electron hole pairs and this current is propagated to (at least) one side where a readout channel sits. Each readout channel is connected to a data aquision system (DAQ) via one or more electronic modules needed for collecting, shaping and digitising the detector signals. The DAQ either collects data if a certain threshold is exceeded to store only relevant data or simply collects all data it recieves in a “free running” mode. A triggered DAQ system only collects data when it recieves a trigger signal, for example a signal from a specific detector channel. For the upcoming fission experiment, the DAQ will be used in a “free running” mode.

Two detectors are crucial to measure a fission event through a (d,p) reaction in inverse kinematics. One detector is placed *downstream* from the target measuring the two fission fragments and a detector to measure the ejected protons from the (d,p) reactions is required to know that a formation of ^{230}Ac has occured. More detectors mean that we can obtain additional information to improve the outcome of the experiment. The fission fragment detector that will be used is a CD shaped ΔE-E silicon detector and a silicon array to measure ejected protons. It will be placed *downstream* from the target to avoid blocking trajectories of the fission fragments. Figure 2.3 illustrates the position of the detectors. In addition, four LUME detectors will be used to measure the luminosity and an arrangement of $CeBr_3$ crystals, mounted around the mentioned detectors, to measure γ-radiation from the fission fragments.

2.2.1 Liverpool on-axis array



(a) Liverpool an-axis array. Reprinted with permission from [14]. Copyright 2019 Science and Technology Facilities Council (STFC).

(b) Segmentation of a DSSD

Figure 2.4: A double sided silicon strip detector (DSSD) is split horizontally on one side and vertically on the other. Each strip is connected to a readout channel. A hit on the detector will produce a signal on both sides of the detector and by matching a signal from a horizontal strip with a vertical, the position of the hit can be determined. Each DSSD on the array has an active region of 121.5 mm × 21.9 mm [15].

When studying (d,p) reactions in inverse kinematics, we can extract the excitation energy of the resulting nuclei of interest by measuring the emitted protons. This can be achieved with a hollow array of silicon detectors which is placed along the beam axis in the *upstream direction*. The beam passes through the hollow center of the array before impinging on the target. An emitted proton will travel along its helix trajectory, hitting the detector array. The deposited energy of the proton and where it hits the detector, together with the energy of the incoming beam provides sufficient information to reconstruct the excitation energy of the resulting nucleus and the emitted angle of the proton.

The Liverpool on-axis array (ISS array) consists out of 24 double sided silicon strip detectors (DSSD) arranged in a hexagonal shape with four detectors per side. Each DSSD is divided into 128 vertical segments on one side and 11 horizontal segments on the other. Each segment is connected to a readout channel and a position can be infered from the channels that delivers a signal. The segmentation of a DSSD is illustrated in Figure 2.4.

2.2.2 Fission fragment detector

To measure fission fragments, a ΔE-E telescope detector will be used. The ΔE-E detector consists of two CD-shaped silicon detectors (CD) and apart from measuring the energy of a projectile, it can be used to identify charged particles. Similar to a DSSD from the ISS array, a CD detector is segmented to give a position as well as an energy deposit. The CD detectors which will be used in the experiment, detectors of type S3 by Micron Semiconductor Ltd [16], are segmented in 24 rings on one side of the detectors and 32 sectors on the other. A schematic illustration of such a detector is shown in Figure 2.5.

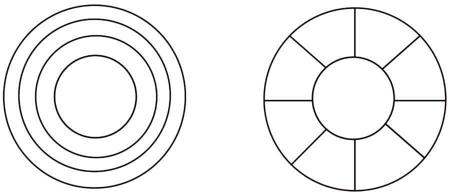


Figure 2.5: A CD shaped DSSD has one side split into rings and the other side split into sectors. Each ring and sector is connected to a readout channel. Matching a signal from a sector with a signal from a ring makes it possible to determine where a projectile has hit the detector.

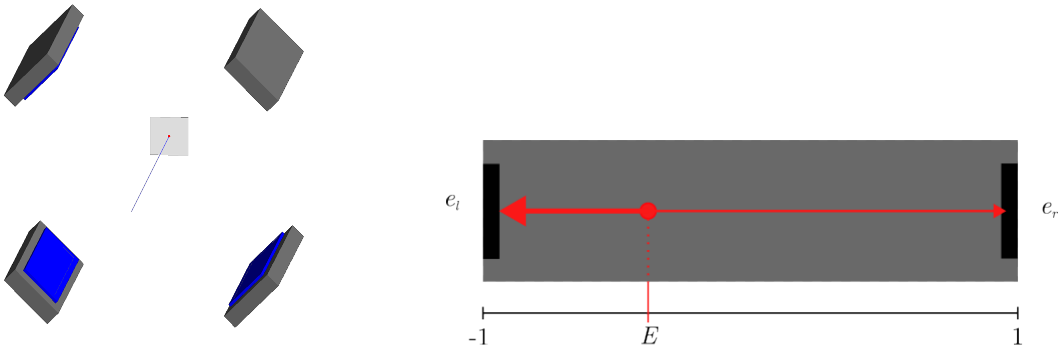
2.2.3 Luminosity detectors

The beam luminosity, described in Section 2.1.4, is measured by four luminosity detectors (LUME) placed *downstream*, parallel to and at a distance from the beam axis to not block the trajectories of the fission fragments, see Figure 2.6a. Each

LUME is a position-sensitive silicon detector (PSD) with three readout channels, one on the back side and two channels on the front side. The two channels on the front side are placed on opposite ends of the detector. When a projectile hits the detector, the signal on the front side is split between the two channels, e_l and e_r, see Figure 2.6b. If the projectile hits the center of the detector, the signal is divided by half. The relative position u on the detector can be calulated as

$$u = \frac{e_r - e_l}{e_r + e_l}. \tag{2.8}$$

Signals from the channel on the back side corresponds to the total energy deposited.



(a) Orientaion of four PSDs around the beam axis.

(b) Schematic of a PSD.

Figure 2.6: A position sensitive detector (PSD) has three readout channels, two on the front side and one on the back. The signal from a projectile on the front side of a PSD is split into two, proportionally to where on the detector it hits. The amplitude of the two signals e_l and e_r can be used to calculate the position of the hit while the back side reads the total energy deposited. Four PSD detectors are placed around the beam axis to avoid blocking trajectories of fission fragments.

With this short description of the detectors, the next step is to simulate these detectors and use those simulations to test the analysis software.

3

From simulations to analysis

In order to test analysis software without the need for measured data, an alternative is needed. Simulated energy deposits converted to a raw data format which the analysis software expects, makes it possible to conduct tests without even modifying the analysis software. A key feature to note is that each program or subroutine, either it being part of simulation, conversion, unpacking or analysis, transforms data from one structure to another. As long as the data structures remain unchanged, each program and subroutine can be considered independently. Furthermore, each data structure in the chain can be considered and visualised which makes it easier to test which part of the larger software stack may be causing a problem. A more detailed flow chart expanded from Figure 1.2 is shown in Figure 3.1, together with graphs of CD energy versus channel number at various stages. The leftmost plot shows simulated data, the middle plots shows the data after being converted to raw data, emulating individual channel gains. Finally, the rightmost plot shows the same data after being calibrated by `ISSSort`.

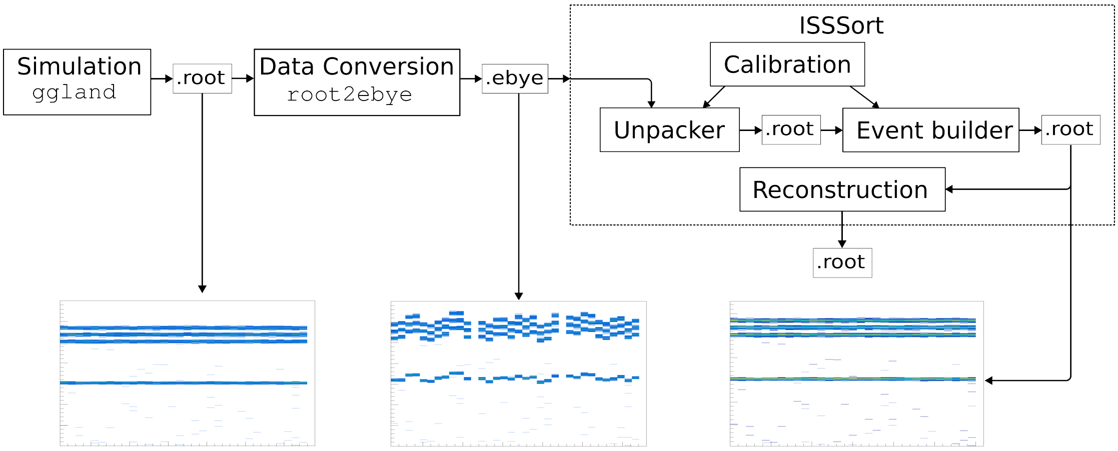


Figure 3.1: A detailed flow chart depicting the data flow which is used to test `ISSSort` with simulations. A graph depicting energy versus CD channels of simulated alpha data is shown before the conversion step, after the conversion and a third time after being calibrated by `ISSSort`.

3.1 Simulations with a command-line wrapper

The simulations are performed with `ggland` [17], a command-line wrapper for the physics simulation toolkits `GEANT3` [18] and `Geant4` [19]. With a few command line arguments from the user, `ggland` sets up the desired geometry and particle generation for a simulation, making it easy to utilise the `GEANT3` and `Geant4` libraries without the user needing to write any code. The wrapper has a few predefined detectors, among them the four detector types that will be used in IS739, each with their own command. I have contributed several of those before this thesis.

The particle generator, called *gun*, in `ggland` requires a particle type and an energy to be specified as a minimum. It is versatile with many options and can also simulate Rutherford scattering in inverse kinematics by passing a few arguments on the command line. There is also a possibility to have multiple guns in layers to simulate decay chains and reactions.

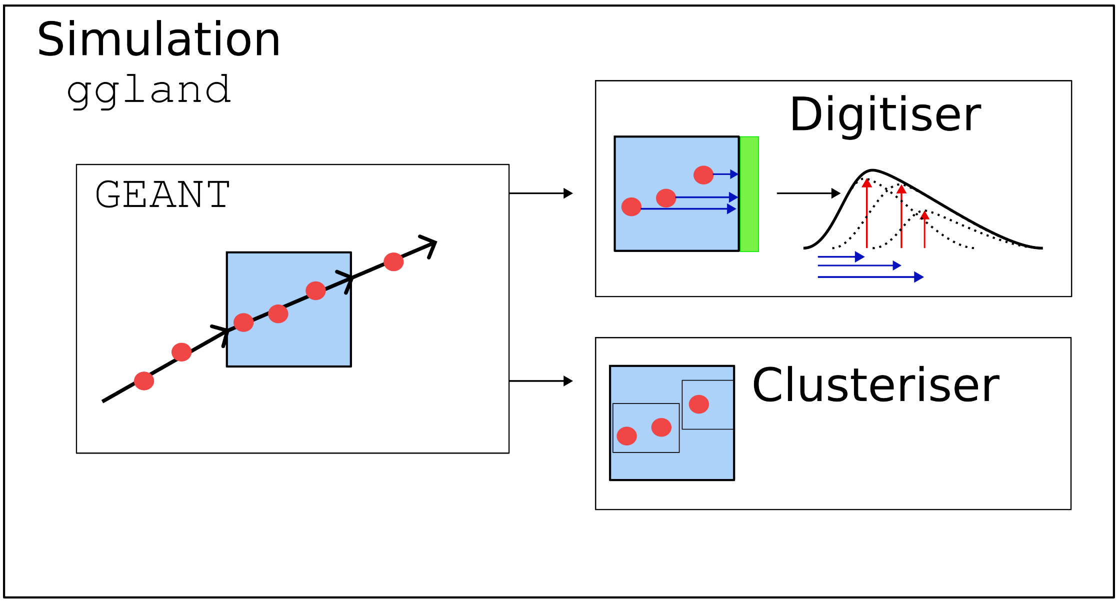


Figure 3.2: The concept of the clusteriser and digitiser routines of `ggland`. The clusteriser groups energy deposits (red) in an active volume delivered by a `GEANT` simulation. In contrast, the digitiser propagates the energy deposits to the readout side (shown in green) of the detector and further processes these amplitudes to a pulse shape from which the routine delivers the resulting amplitude together with a timestamp as the output.

When `ggland` has passed the setup conditions from the user to `Geant4`, a Monte Carlo simulation will start and particles will move through the geometry depositing energy in mediums according to their material properties. A particle can deposit energy in the same medium multiple times. If the current medium is an active part of a detector, the deposited energy along with time and position information is returned to the wrapper. Then, `ggland` processes the returned data in one of two ways. The first is to cluster the energy depositions from a detector within a certain window in space and time, where each cluster is intended to correspond to one hit. This data have more information than a detector would deliver in reality and may be considered to be like reconstructed data.

The second way is to deliver the results from the Monte Carlo simulations in a "digitised" version. Instead of clustering the depositions together, `ggland` propagates the deposited energies to the readout sides of the detectors. These energy amplitudes are given an analytic pulse shape and are then added together. The peak of the combined pulse is now the total energy for the hit and to deliver a time, a constant fraction descriminator (CFD)[1] is emulated. This energy and time pair thus is more akin to and closely resembles actual detector outputs, as stored by the DAQ. Illustrations of both the clusteriser and digitiser routines are shown in Figure 3.2.

The output from the simulations can be stored as `ROOT` [21] files in a tree structure. The digitised data resembles unpacked and calibrated data from detectors. To test analysis software with these simulations, one way would be to extend the analysis software so it can parse the `ROOT` files directly. But that would circumvent the data unpacking step and forcing the analysis software to handle the data structure from simulations separately. Converting the simulated data to the same data format as the DAQ output makes it possible to test the whole analysis chain, without any special cases.

3.2 From simulations to binary data

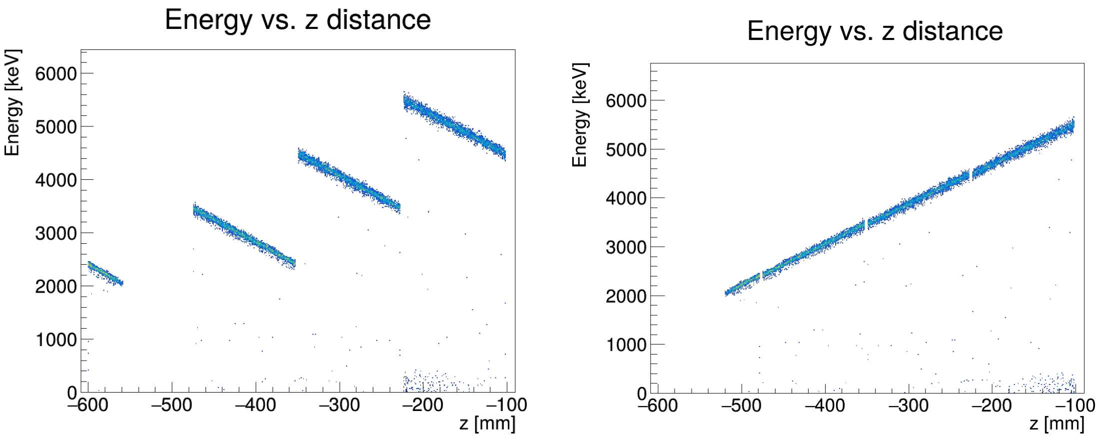
In order to test the analysis chain with simulations we need a software tool to convert our simulated data to the same format as the DAQ delivers. Having a separate program to convert the data offers a few advantages compared to extending either `ggland` to have a new output format or `ISSSort` to process another data format. Since the data output from `ggland` can be analysed directly, it is possible to compare the data before conversion and after unpacking, which is illustrated in Figure 3.1. If the data quality after has changed considerably, then either the conversion or the analysis software must be the reason. In a similar way, if `ISSSort` is extended to handle simulated data separately from the DAQ output parts of the analysis chain cannot be tested. Instead, a small program `root2ebye` (a part of the

---

[1] CFDs are electronic modules that produce logic timing signals that ideally have no time dependency on the variation of the input pulse amplitude [20].

`egmwsort` repository) [22], is introduced which takes the output from simulations with the digitiser routine of `ggland` and converts it to the same data structure which `ISSSort` expects as input. Having smaller programs which do one thing well can be easier to build and maintain rather than extending and complicating already existing programs [23].

When a simulation file is passed to `root2ebye`, the program starts by scanning the file, detecting which detectors are present in the simulation. Using a pseudo random number generator, `xorshift64*` [24], gain-offset parameters for all available detectors are prepared and by specifying a seed value from the command line, this procedure becomes deterministic. This makes it possible to have different simulations with the same gain-offset parameters which opens up the possibility to test calibration routines with an alpha or gamma source, then using the calibration for different simulations. For this reason, `root2ebye` has been used with the seed 3[2] throughout this project.



(a) Incorrect channel mapping. **(b)** correct channel mapping.

Figure 3.3: Artefacts appearing during event reconstruction (left) due to incorrect channel mapping of the ISS array in `root2ebye`. The plot to the right shows the same simulation with correct mapping.

In the next step, `root2ebye` maps the detector channels from the simulation to the corresponding electronics module and channel numbers for every channel fired in each event in the simulation file. If the mapping is incorrect, strange artefacts may arise during the analysis, as can be seen for the ISS array in Figure 3.3. At the same time as the channel mapping, floating point energy values, E, from the simulation is converted to integer values e using the randomised gain g and offset

---

[2] The seed was chosen at random.

m parameters as

$$e = \frac{E}{g} + m,$$

while also taking overflow and saturation into account. The simulated times from each event are converted while also a larger time offset is being added to each event, separating the events from one another. This is done to prevent events being too close together in time so that they can be clearly separated by the analys software. The issue when events cannot be separated is called *event mixing*. Finally, the converted values are written to a .ebye buffer.

The conversion software is not limited to this experiment and `ISSSort`. In fact, `root2ebye` can write another output data format, used by the ISOLDE Decay Station (IDS) collaboration. Furthermore, `root2ebye` could be extended to convert to other data formats as well and use it to test other analysis softwares. At the end of this chapter is a small guide on how to use `root2ebye`, as well as `ggland` and `ISSSort`.

3.3 From binary data back to ROOT

With generated data files mimicing DAQ output, it is time to consider the analysis software. `ISSSort` analyses data in three steps. In the first step, an unpacker routine converts the data to a .root file and creates some histograms for diagnostic purposes. The user can give a settings file as input to specify the confiugration of the detectors and electronics used. If the user has provided a calibration file, the unpacker will also convert the ADC data to keV. After the unpacker has converted the data to .root format, the data is sorted according to time which uses a lot of I/O operations. Thus, this can take some time if a lot of the data is out of order.

The next step is the event builder. It takes the time-sorted data file from the unpacker step and recontructs hits on the detectors as specified by the settings file (or default values if no file is provided). The reconstruction is done by matching data between detector channels, depending on the detector type, within a time window. The output from the event builder is written to a new .root file. If the data were not calibrated during the unpacker step, it is calibrated during this step.

The last step combines detector data to reconstruct physics events or filtering events from a detector by requiring a hit in another detector within a given time window. For example, a fission fragment hit is only considered if there is a coincident proton hit on the ISS array, which is needed in order to extract the excitation energy of the compound nucleus as an example. These results are written to a .root file as histograms with an option to save the reconstructed data in a tree structure.

Running the simulation chain

An example simulation with `ggland`

As a starting point, `ggland` has a few predefined decay chains, among them ^{148}Gd, which decays by emitting an α-particle with an energy of 3.183 MeV. The ISS array is placed 4 cm from the source and we will write the result of the simulation to a ROOT file. The command line

```
./land_geant4 --gun=decay-Gd-148 --isia=z0=-35cm \
              --world=type=vacuum --fieldbox=d=200cm,Bz=2T \
              --events=100000 --np --tree=digi,alpha.root
```

will simulate 100000 events with one α-particle emitted isotropically per event in a “world” of vacuum. A uniform magnetic field of 2 T covers a cubic volume with a side length of 2 meters. All objects have their origin in the center of their top level volume, so in order to move the ISS array 4 cm upstream of the source in addition we have to move the detector by half its length. (The array is a bit longer than 62 cm.)
The `--np` argument runs multple simulation processes in parallell and in the `--tree` argument we specify a digi output, which the next program is looking for. Run `ggland` with `--help` for more explanations of the possible arguments to use.

Packing and unpacking

The next step is to convert the data generated with ggland:

```
./root2ebye alpha.root alpha.ebye
```

with the first argument being the input file and the second is the output file. It takes care of all the mapping internally and by adding a specific seed N with `--seed=N` the same gain-offset parameters would be used every run.
Finally, we pass the converted alpha.ebye file as input to `ISSSort`. In general we want to pass a settings and calibration file as well and example files exists in the repository, but `ISSSort` expects the array as default, so we can simply write

```
./iss_sort -i alpha.ebye
```

to obtain the ROOT files in a directoy called *sorted*.

4

Software upgrades

With three new detector types at the ISS setup, upgrades to the analysis software are crucial in order to monitor and extract information from these detectors during data-taking. Having data files with simulated nuclear reactions help during development to test the software and see if it behaves as expected.

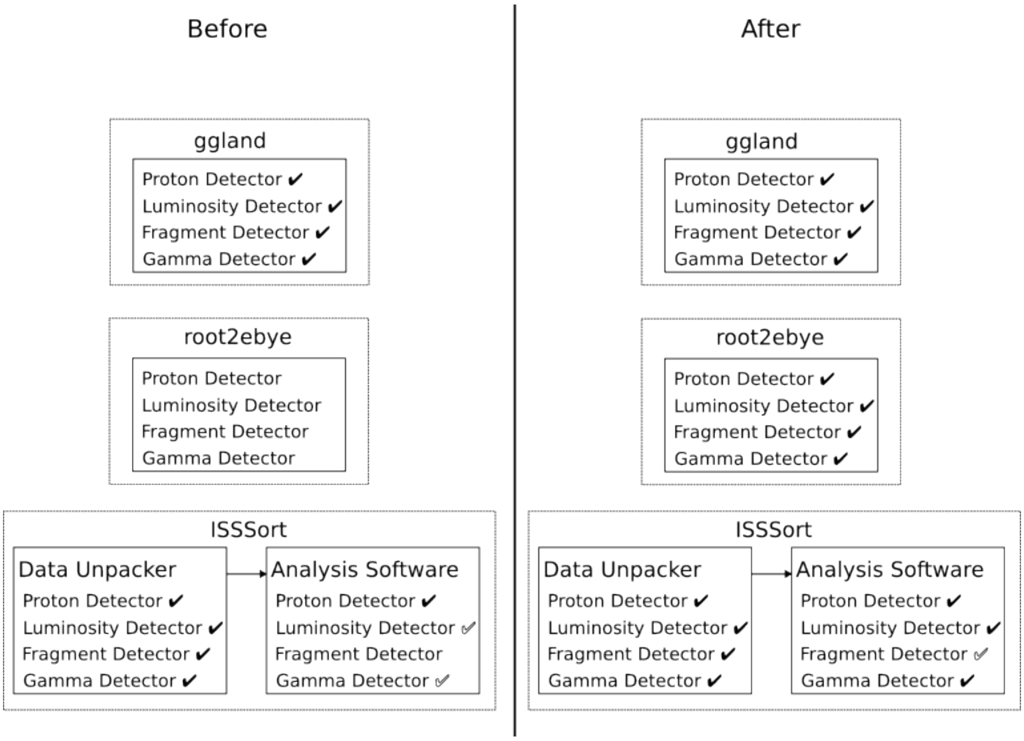


Figure 4.1: The status of the detector implementations for different softwares at the start of this thesis and at the time of writing. The next step is an analysis routine to match signals from the front of the fission fragment detectors with signals from the back. The step after that is to be able to analyse data from multiple detectors in coincidence[1].

---

[1] Meanwhile, these routines have already been implemented.

At the start of this thesis, only the ISS array was fully integrated in `ISSSort`. Of the new detectors, LUME and CeBr$_3$ detectors were partially integrated while unpacking for CD detectors was started but nothing further implemented. At the time of writing, the LUME detectors are implemented. The CD and CeBr$_3$ also allow monitoring, while some functionalities for more advanced analysis are left to do. Work is also ongoing to implement analysis routines to exctract observables by combining data from multiple detector types. Figure 4.1 gives an overview of the state of the softwares before and at the end of this thesis.

4.1 Two CD detectors for fission fragments

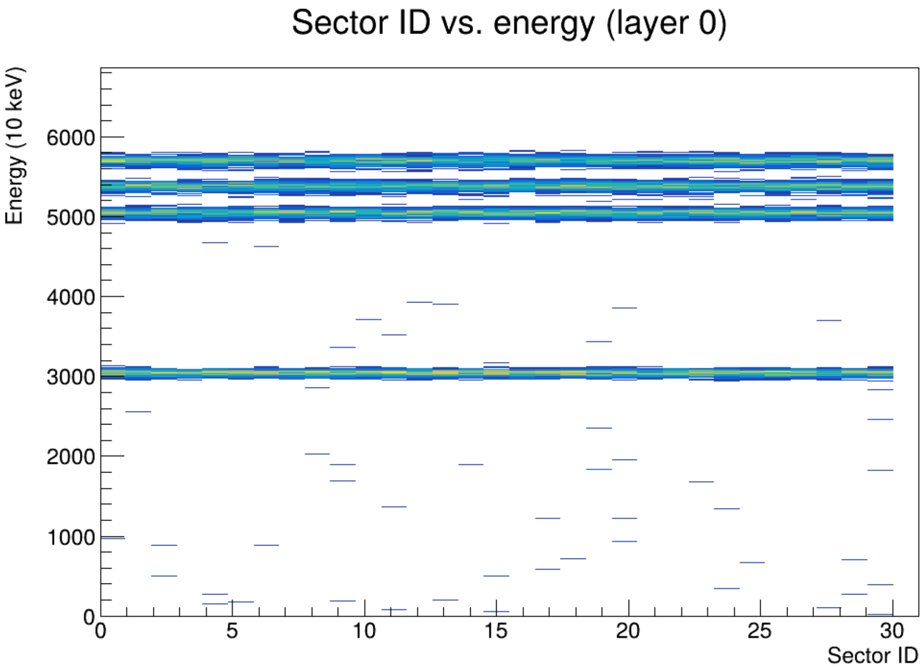
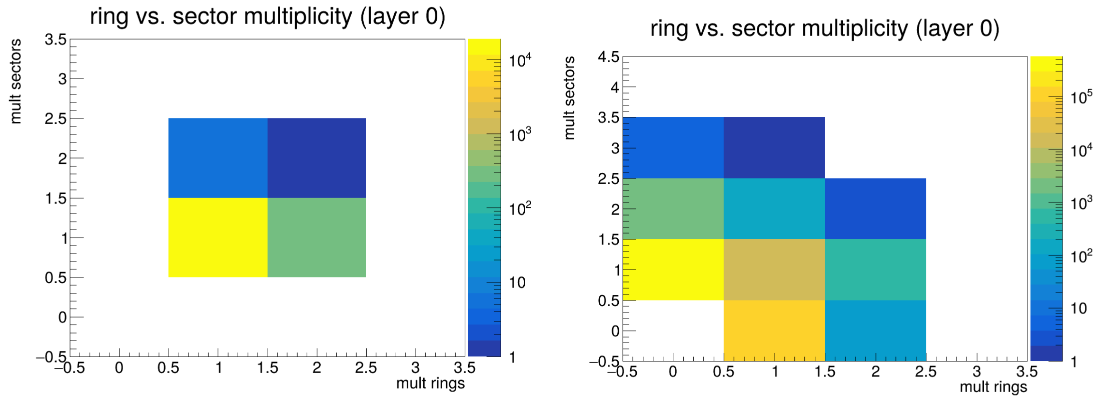


Figure 4.2: Calibrated data from a simulated alpha source typically used at ISS. This is an easy way to see that all sector channels from a CD detector are calibrated as the energies for each channel should line up. This simulation corresponds to a 50 s measurement of a source with an activity of 2 kBq placed 2 cm from the detector.

The CD detectors will be used to measure fission fragments. It is a requirement that they work in order to detect fission events. As a start, the analysis needs to operate with calibrated data. To obtain a quick overview of how well calibrated a detector with many readout channels is, one way is to plot energy versus the corresponding channel ID from an irradiation with particles of distinct energies. If the calibration is good, the energy peaks from each channel should line up and form lines at the energies corresponding to, for example, an alpha source, as is the case in Figure 4.2. The simulation corresponds to 50 s of measuring an alpha source, placed at a distance of 2 cm from the detector, with an activity of 2 kBq, which is the activity of the corresponding alpha source used at ISS. The raw data peaks need to be fitted in order to extract the calibration parameters however, either manually which quickly becomes cumbersome with many channels to calibrate, or with a routine which automatically does the fitting and calculates the gains and offsets. An automatic calibration routine is currently not available for the CD detectors, but it is something which could be developed before the experiment.



(a) Simulated alpha source. **(b)** An alpha source measurement.

Figure 4.3: Number of hits on the ring side and sector side of a CD detector from a simulation (left) and a measurement (right). The simulation is a test to see that the software is able to determine the multiplicity correctly while reality has a tendency to be more complex and nuanced.

Another useful tool for a CD detector is to look at the multiplicity, the number of hits, on one side of the detector versus the other side of the detector. Ideally, the detector would register one energy deposit on each side per particle in an event. However, a particle may hit the detector at the edge between two sectors, having its energy shared between the two, while still hitting only one ring or vice versa. This is known as charge sharing and needs to be taken into account when matching hits between the front and back side of a detector. If we consider the same alpha source as before, centered in front of the detector, most of the alpha particles deposit their energies only in one segment and one ring, while more particles hit two rings and one segment than two segments and one ring, which is seen in Figure 4.3a. The reason is that alpha particles that hit the detector are emitted at an angle with respect to the detector plane, which increases the likelihood of passing through multiple rings. This would not be the case if the alpha particles were traveling perpendicular to the detector surface. The analysed result in Fig 4.3a of the simulation shows this difference between rings and sectors and thus suggests that `ISSSort` counts the number of hits per event correctly when the simulated events are separated in time to avoid event mixing.

The same claim is harder to make when looking at measured data where faulty wires or other artifacts caused by the hardware can mix with the hits from the alpha particles, see Figure 4.3b. Since the analysis passes the test of the simulated data, the other multiplicites in the measured data not present in the simulated ditto are likely caused by the hardware or possibly that times between hits are too close and are mixed together during hit reconstruction.

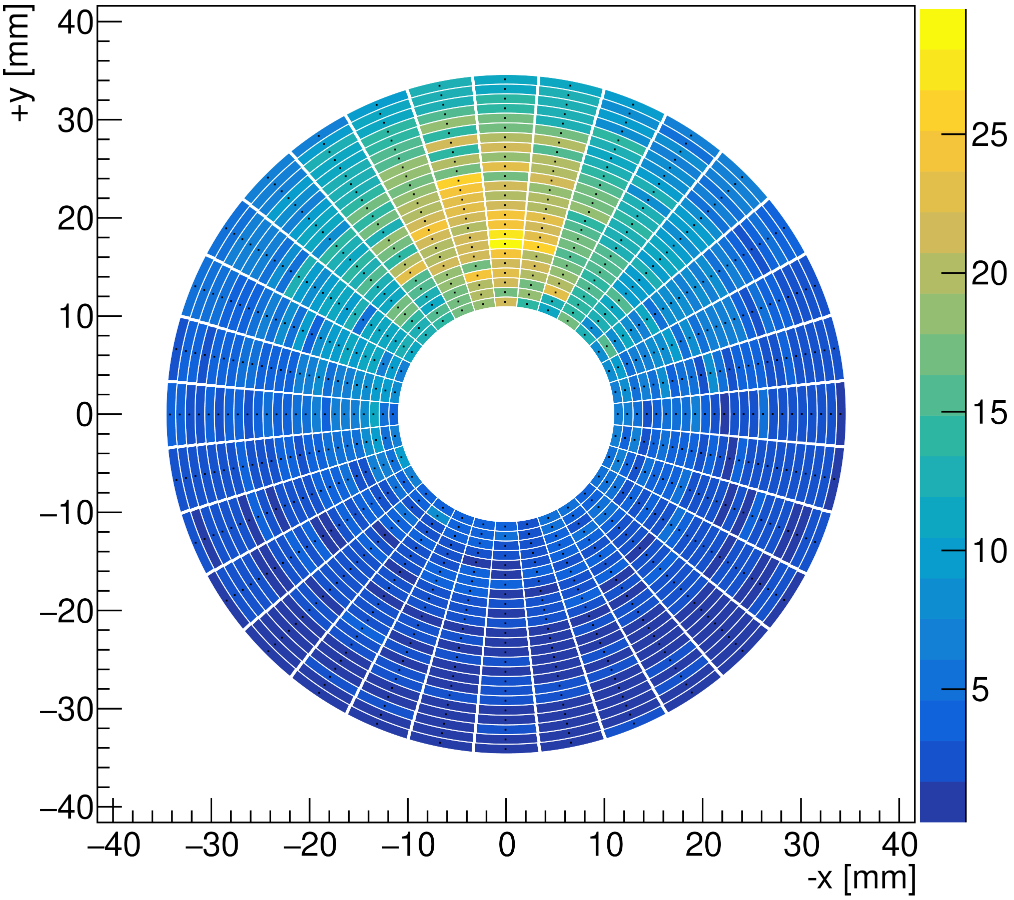


Figure 4.4: A 2D hit pattern with simulated data of an alpha source which is located 2 cm above the center point. Each pixel represents a sector and its overlap with a corresponding ring.

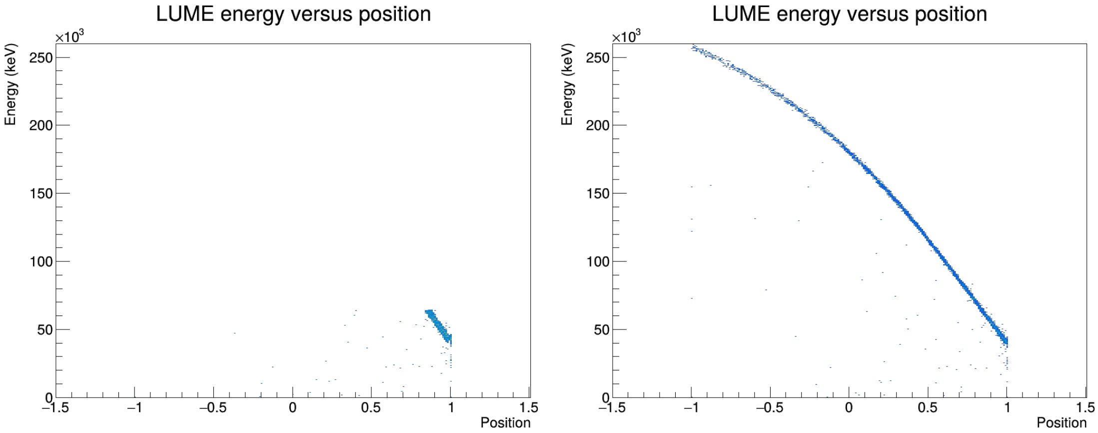
With an appropriate routine to match signals from both sides of the detector it is possible to plot a hit pattern to visualise where on the detector the particles hit. It can be used to see if the incoming beam or a stationary source is off-centered, which would have an impact when reconstructing angles. A hit pattern is developed as part of the preparation, see Figure 4.4 [25]. It is a simple task with `ggland` to simulate an alpha source which is displaced from the center to test if the software generating the hit pattern has mapped the segments and rings of the detector correctly. A simulation with an alpha source located 2 cm above the center point is clearly reflected in the figure. Each "pixel" represents a coincidence between a sector segment on one side with the corresponding ring on the other.

The main thing left for the implementation of CD detectors is to test the routine which matches signals from the front and back sides of the detector in order to reconstruct hits on the detector. For the fission experiment, two fragments are expected and these fragments would have kinetic energies around 1 GeV given the incoming beam energy of 8 MeV/u. With such a high beam energy of a very heavy projectile, other reactio channels open, which could produce more than two particles. Those need to be studied as well, since they are likely to contribute to the background.

4.2 Four LUME detectors to diagnose the beam

Apart from hit patterns such as the one in Figure 4.4, multiple LUME detectors can also be used to check if the beam is off-centered. As described in Chapter 2, four LUME detectors will be mounted parallel to the beam, mainly to detect recoiling deuterium and carbon when ^{229}Ac is elastically scattered. The scattered and recoiling particles have no preferred direction in the xy-plane so the four detectors should have an equal amount of statistics if the beam is parallel and centered relative the detectors. When a particle hits the LUME, the deposited energy will be split between the two readout channels at the front side on the d etector and a relative position can be calculated by Eq. (2.8). If the energy is too low, only one side will have detected a signal as the other signal falls below the threshold at the other side. This results in a hit reconstructed at either 1 or -1.

A strength with simulations is the possibility to test the software with events that otherwise require an accelerator complex. A simulation of ^{229}Ac with an incoming energy of 8 MeV/u scattered on a carbon target, causing the carbon to be ejected from the target and possibly hitting one of the detectors is one such example. This simulation can test the ability of the analysis software to reconstruct higher energies from the carbon projectiles and their positions on a LUME. When testing `ISSSort` with this simulation, an overflow issue was spotted, see Figure 4.5a. This was caused by `root2ebye` adding a large gain, causing energies higher than 60 keV being converted to a larger number than can be stored in the data format and thus set to the highest value possible by `root2ebye`. This caused a discovery of a suboptimal overflow handling in `ISSSort` which caused some overflow events to end up among the low energy noise. After an update it is now easier to spot overflow from a detector and after setting the gain in `root2ebye` to a lower value, the complete carbon line can be seen in Figure 4.5b and it corresponds to 16 min of data taking considering a beam energy of 8 MeV/u with an estimated intensity of $4.2 \cdot 10^5$ ions per second [13].



(a) Overflow caused by high gain. **(b)** Data retained with low gain.

Figure 4.5: Simulated Rutherford scattering of ^{229}Ac on carbon. A conversion with a high enough gain causes the energy to be larger than the maximum value the data format can store and data is lost. The same simulation with a lower gain retains the data quality.

5

Conclusion and outlook

The status of the analysis software for the upcoming fission experiment has come a long way in the past few months. With the ability to test the software with simulations, the capabilities and deficiences of the analysis algorithms can be evaluated before the experiment. These tests make it easier to spot errors or software bugs in the analysis routines without confusing them with artefacts produced by the hardware itself.

The main missing items are calibration routines for the newly added detectors to reduce the workload of having to do them manually. In addition, analysis routines which reconstruct physical observables with data from multiple detectors are needed and such capabilities are being developed as I am concluding this thesis.

While primarily being written for the fission experiment, the conversion program `root2ebye` can very well be used for other experiments with other data formats. In fact, a few detectors have been mapped to the data format used at the ISOLDE Decay Station collaboration also to show the versatility of the program and to use it to prepare experiments that are forseen for the late summer or autumn.

Bibliography

[1] S. Giraud et al. “Simulations and analysis tools for charge-exchange (d, 2He) reactions in inverse kinematics with the AT-TPC”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1051 (2023), p. 168213. ISSN: 0168-9002. DOI: `https://doi.org/10.1016/j.nima.2023.168213`.

[2] Björn Jonson. *Private communication.*

[3] CMG Lee. *Nucleosynthesis periodic table.* Online; accessed 2025-05-03. 2017.

[4] Belczynski, K. et al. “The origin of the first neutron star – neutron star merger”. In: *AA* 615 (2018), A91. DOI: `10.1051/0004-6361/201732428`.

[5] E. Margaret Burbidge et al. “Synthesis of the Elements in Stars”. In: *Rev. Mod. Phys.* 29 (4 Oct. 1957), pp. 547–650. DOI: `10.1103/RevModPhys.29.547`.

[6] C. Morse. In: *Nucl. Data Sheets 197, 259* (2024).

[7] T. L Tang et al. “First Exploration of Neutron Shell Structure below Lead and beyond N = 126”. In: *Physical Review Letters* 124 (Feb. 2020). DOI: `10.1103/PhysRevLett.124.062502`.

[8] P. MacGregor et al. “Evolution of single-particle structure near the N = 20 island of inversion”. In: *Physical Review C* 104 (Nov. 2021). DOI: `10.1103/PhysRevC.104.L051301`.

[9] Kristian Haverson et al. “Searching for particle-hole cluster bands in Be using the ISOLDE Solenoidal Spectrometer”. In: *EPJ Web of Conferences* 311 (Oct. 2024). DOI: `10.1051/epjconf/202431100012`.

[10] Liam P. Gaffney et al. *ISOLDESolenoidalSpectrometer/ISSSort.* Feb. 2024. DOI: `10.5281/zenodo.6366317`.

[11] Anna Kawęcka. *To fission or not to fission:* ^{230}Ac. Chalmers University of Technology, 2024.

[12] J. C. Lighthall et al. “Commissioning of the HELIOS spectrometer”. English. In: *"Nuclear Instruments & Methods in Physics Research. Section A: Accelerators, Spectrometers, Detectors, and Associated Equipment"* 622.1 (Oct. 2010), pp. 97–106. ISSN: 0168-9002. DOI: `10.1016/j.nima.2010.06.220`.

[13] Maria Vittoria Managlia. *Fission: impossible?* Chalmers University of Technology, 2025.

[14] *Si-detector array.* `https://isolde-solenoidal-spectrometer.web.cern.ch`.

[15] *Si-detector array.* `https://isolde-solenoidal-spectrometer.web.cern.ch/setup/si-detector-array`.

[16] Micron Semiconductor Ltd. *S3.* `https://www.micronsemiconductor.co.uk/product/s3/`.

[17] H. T. Johansson et al. “GGLAND - command line simulations”. In: *Scientific Report 2013.* Vol. 2014-1. GSI Report. Darmstadt: GSI Helmholtzzentrum für Schwerionenforschung, 2014, 154 p. DOI: `10.15120/GR-2014-1-NUSTAR-KR-15`.

[18] R. Brun et al. “GEANT3”. In: (Sept. 1987).

[19] S. Agostinelli et al. “Geant4—a simulation toolkit”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003), pp. 250–303. ISSN: 0168-9002. DOI: `https://doi.org/10.1016/S0168-9002(03)01368-8`.

[20] Adithya Thaduri et al. “Reliability prediction of constant fraction discriminator using modified PoF approach”. In: *2013 Proceedings Annual Reliability and Maintainability Symposium (RAMS).* 2013, pp. 1–7. DOI: `10.1109/RAMS.2013.6517746`.

[21] Rene Brun and Fons Rademakers. “ROOT — An object oriented data analysis framework”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 389.1 (1997). New Computing Techniques in Physics Research V, pp. 81–86. ISSN: 0168-9002. DOI: `https://doi.org/10.1016/S0168-9002(97)00048-X`.

[22] H. T. Johansson. *Miniball (FEBEX) / ISS (ASIC/CAEN) unpacker.* `https://git.chalmers.se/expsubphys/egmwsort`. 2024.

[23] M. D. McIlroy, E. N. Pinson, and B. A. Tague. “UNIX Time-Sharing System: Foreword”. In: *The Bell System Technical Journal* (1978).

[24] George Marsaglia. “Xorshift RNGs”. In: *Journal of Statistical Software* 8.14 (2003), pp. 1–6. DOI: `10.18637/jss.v008.i14`.

[25] Erik A. M. Jensen. *CD matcher for IS739*. `https://git.chalmers.se/expsubphys/erik/is739`. 2025.

A

Code segments for ggland

For the curious reader, a more detailed overview of the simulations used for this thesis is presented. Some code segments for the particle generator, *gun*, are shown and explained in Section A.1, while code segments for different detectors are shown in Section A.2.

A.1 The particle generator

To simulate an alpha source like the one used for Figure 4.2, the following command line can be used:

```
1  ./land_geant4 \
2      --world=type=vacuum \
3      --gun=dummy,feed=source \
4      --gun=from=source:0.25,decay-Gd-148 \
5      --gun=from=source:0.25,decay-Am-241 \
6      --gun=from=source:0.25,decay-Pu-239 \
7      --gun=from=source:0.25,decay-Cm-244
```

The purpose of each line:

1: Is the executable file of `ggland`.

2: Set the *world* environment to vacuum.

3: Declare an origin, *source*, which subsequent guns can depend on. It does not generate a particle itself.

4–7: Each line specifies an α particle with given energies from predefined decay chains. One of the α particles will be randomly generated each event at the *source* location, each with a probability of 25 %.

To split the command line into multiple lines, e.g. in a script, a backslash character is added at the end.

To simulate inverse Rutherford scattering as in Figure 4.5, the following lines can be used:

```
1    --test=type=C2D4,dmz=1000ug/cm2,target \
2    --gun=H2:target,Ac229,T/u=8MeV/u,rutherford,dtheta=recoil=10:85deg \
3    --incoming=1000 \
4    --fieldbox=d=200cm,Bz=-2T
```

The purpose of each line:

1: Define a target volume of CD_2 with a thickness of 1000 μg/cm^2.

2: Specify inverse Rutherford scattering of ^{229}Ac with an incoming energy of 8 MeV/u impinging on deuterium inside the target volume. The differential cross section is calculated for the recoil (d) angular range 10° – 85°.

3: Together with the differential cross section from the gun above, `ggland` calculates how many scattering events to simulate in order to correspond to 1000 incoming particles, given the target thickness on line 1.

4: Define a homogenous magnetic field cube with a side length of 2 m. The field strength is 2 T in the z-direction.

A.2 Detectors

Multiple detectors are predefined in `ggland` and can be used with a single command line argument. For example, the line

```
1  --isia
```

will set up the geometry for the ISS array (see Section 2.2.1) in the center of the world space. It is also possible to rotate and move detectors (and guns) with command line options by extending the corresponding argument. The command line arguments to simulate the ΔE-E telescope for Figure 4.2 are:

```
1  --cd-01=z0=2cm,sigma_e=0.005
2  --cd-02=z0=3cm,sigma_e=0.005
```

The purpose of each line:

1: Place a CD shaped detector 2 cm from the origin in the z-direction. Output data will be smeared by an energy resolution of 0.005 MeV. The cd-01 is a uniqe ID for the detector and `root2ebye` expects the ΔE detector to have this ID.

2: Place a detector, identical to the first one, but 3 cm from the origin. Its unique ID is cd-02 and `root2ebye` interprets this ID as the E detector.

It gets a bit more complicated to place one LUME detector together with a backing panel of aluminium to shield the rear side of the detector from incoming particles. The following command line:

```
1  --sistrip-01=roty=90deg,rotz=45deg,\\ \
2    z0=4cm,x0=-2.69cm,y0=-2.69cm,\\     \
3    tree-name=LUME01,\\                  \
4    sigma_e=0.005                        \
5  --test-1=dx=6.6cm,dy=1.6cm,dz=4.1mm,\\ \
6    roty=90deg,rotz=45deg,z0=4cm,\\      \
7    x0=-2.88cm,y0=-2.88cm,\\             \
8    type=Al
```

places one out of four LUME detectors used for Figure 4.5. The purpose of each line:

1: Rotate a PSD detector by 90° around the y-axis followed by a rotation around the z-axis by 45°. Each rotation and placement are done in the order they appear on the command line.

Adding a backslash to the argument, written as two backslashes on the command line, makes it possible to write a single argument across multiple lines with indentation.

2: Dislace the PSD 4 cm from the origin and 3.8 cm from the z-axis.

3: Give the PSD the name LUME01 in the output root tree. The name is needed for `root2ebye`.

4: Finally, add a resolution of 0.005 MeV.

5: Create a volume which is 6.6 cm long, 1.6 cm wide and 4.1 mm thick.

6–7: Rotate the volume and place it behind the LUME detector.

8: The volume is made out of aluminium.

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY