

EFFICIENCY OPTIMIZATION OF A HPGE CLOVER
ARRAY FOR IMPLANTATION DECAY
EXPERIMENTS AT FRIB

by
Hannah N. Grover

© Copyright by Hannah N. Grover, 2018

All Rights Reserved

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Nuclear Engineering).

Golden, Colorado

Date _____

Signed: _____

Hannah N. Grover

Signed: _____

Dr. Kyle Leach

Thesis Advisor

Golden, Colorado

Date _____

Signed: _____

Dr. Mark Jensen

Professor and Head

Department of Nuclear Engineering

ABSTRACT

The Facility for Rare Isotope Beams is a new state-of-the-art facility being built at Michigan State University in order to better understand the the fundamental building blocks of the universe. The decay station group hopes to combine a clover style gamma ray detector array with a small, planar HPGe detector to study some of the most exotic nuclei. This project focused on ensuring that this combination would maintain or improve the photon efficiency of a clover style array on its own, as well as optimizing the size of the planar HPGe detectors in the center of the array configuration. The results from simulations done in Geant4 show that the efficiency of this combination of detectors is higher than just a clover array, and even significantly high for low energy gamma rays. In addition, the optimum radius of the planar HPGe detectors is 3.5 cm when considering both the cost and efficiency for the total configuration.

TABLE OF CONTENTS

ABSTRACT	iii
LIST OF FIGURES AND TABLES	vi
LIST OF ABBREVIATIONS	ix
ACKNOWLEDGMENTS	x
CHAPTER 1 INTRODUCTION	1
1.1 Radioactive Decay	2
1.2 Beta Decay	2
1.3 Gamma Ray Emission	4
1.4 Production of Rare Isotopes	6
1.5 Facility for Rare Isotope Beams	6
1.6 The FRIB Decay Station	8
CHAPTER 2 GAMMA RAY DETECTOR TECHNOLOGY	10
2.1 High Purity Germanium Detectors	11
2.2 Photon Interactions in HPGe Detectors	12
2.3 Relevant Detector Designs	14
CHAPTER 3 MODELING GERMANIUM DOUBLE SIDED STRIP DETECTORS	17
3.1 Geant4 Simulation Environment	17
3.2 Creating the Geometry File for the Germanium Double Sided Strip Detectors	19
3.3 Incorporating the GeDSSD into the Simulation Package	22
3.4 Running Simulation with GeDSSDs	26

3.5 Automating Simulations	30
CHAPTER 4 RESULTS AND ANALYSIS	33
4.1 Automatic Analysis of Raw Output Data	33
4.2 Generating Efficiency Curves	36
CHAPTER 5 SUMMARY AND CONCLUSIONS	45
REFERENCES CITED	48
APPENDIX GEOMETRY FILE FOR THE GEDSSDS	51

LIST OF FIGURES AND TABLES

Figure 1.1	The nuclear chart shows all of the isotopes that have been discovered with the number of neutrons on the x-axis and number of protons on the y-axis.	1
Figure 1.2	This illustration shows that nuclei with higher masses (and thus lower binding energies) will move toward stability by emitting radiation.	3
Figure 1.3	An example of a decay scheme that can be constructed by studying the decay of nuclei and their daughter products.	4
Figure 1.4	A schematic layout of the FRIB facility.	7
Figure 1.5	A design rendering of HPGe detectors combined with a new neutron array (NEXT) for the FRIB decay station.	9
Figure 2.1	A schematic of TESSA, the original gamma ray detector array.	11
Figure 2.2	Cross sections for photoelectric absorption, Compton scattering, and pair production in germanium.	13
Figure 2.3	Half of the GRIFFIN array is shown on the left hand side of the configuration, with the clovers closely surrounding the center sphere.	14
Figure 2.4	The germanium double sided strip detector created by PhD students at UT Knoxville.	15
Figure 2.5	Schematic of double sided strip detector showing the orthogonal nature of the germanium strips.	15
Figure 3.1	A single clover with 4 HPGe crystals from the GRIFFIN array modeled in Geant4.	18
Figure 3.2	This graphic depicts the layout of the two GeDSSDs inside the cryogenic chamber with associated dimensions.	21
Figure 3.3	The DetectorConstruction.cc file tells the environment to call the functions in the geometry file that builds and places the GeDSSSD setup.	23
Figure 3.4	The lines in DetectorMessenger.cc allow the simulation environment to communicate with the runtime macro that the user inputs.	23

Figure 3.5	The fill function in the EventAction.cc file for the GeDSSDs attributes every event recorded in the simulation to a histogram for data output.	24
Figure 3.6	This code shows the initialization of the vectors that track particle interactions in the GeDSSDs while the simulation is running.	24
Figure 3.7	The HistoManager.cc file gives each detector type a histogram name and index that the EventAction.cc file uses and assigns information to, shown here with the GeDSSDs.	25
Figure 3.8	The SteppingAction.cc file defines the identifier values for particles, interaction processes, and detectors. Only detectors with systemID values will have their interactions recorded.	26
Figure 3.9	The run macro used to generate the efficiency curves needed for this project. .	28
Figure 3.10	This image rendered in Blender shows the thick GeDSSD inside of the cryogenic chamber inside of half of the GRIFFIN array.	29
Figure 3.11	These variables are necessary to run the automatic simulation file.	31
Figure 3.12	This function changes the energy of the beam in the run.mac file and then executes it in the Griffin simulation environment.	31
Figure 3.13	This function is responsible for running all raw output files through ntuple to create files for analysis.	32
Figure 4.1	The fit of the photopeak centered at 500 keV for a GeDSSD radius of 4.5 cm. .	34
Figure 4.2	An example of the fitted peak for the 500 keV simulation data created by using the TPeak function in GRSISort.	35
Figure 4.3	The systemIDs correspond to a detector, showing the number of interactions recorded in each during the simulation.	36
Figure 4.4	An efficiency curve previously established for a GeDSSD through experimental data.	37
Figure 4.5	This is the efficiency curve created from the simulations for the thin GeDSSD with a radius of 4 cm. Data points were taken for every 5 keV increment from 15-1000 keV.	38
Figure 4.6	The individual and total efficiency curves for the GeDSSD radius of 0.5 cm. .	39
Figure 4.7	The individual and total efficiency curves for the GeDSSD radius of 1.0 cm. .	40

Figure 4.8	The individual and total efficiency curves for the GeDSSD radius of 1.5 cm.	41
Figure 4.9	The individual and total efficiency curves for the GeDSSD radius of 2.0 cm.	42
Figure 4.10	The individual and total efficiency curves for the GeDSSD radius of 2.5 cm.	42
Figure 4.11	The individual and total efficiency curves for the GeDSSD radius of 3.0 cm.	43
Figure 4.12	The individual and total efficiency curves for the GeDSSD radius of 3.5 cm.	43
Figure 4.13	The individual and total efficiency curves for the GeDSSD radius of 4.0 cm.	44
Figure 4.14	The individual and total efficiency curves for the GeDSSD radius of 4.5 cm.	44
Figure 5.1	The total efficiencies for each GeDSSD radius value is shown here, with the smallest radius farthest to the left.	46
Figure A.1	The first third of the GeDSSD geometry file in the Geant4 simulation package.	52
Figure A.2	The middle third of the GeDSSD geometry file in the Geant4 simulation package.	53
Figure A.3	The final third of the GeDSSD geometry file in the Geant4 simulation package.	54
Table 3.1	This table displays all of the dimensions used to create the geometry file for the GeDSSDs and associated cryogenic chamber.	19
Table 3.2	The particle types used in simulations have assigned identifier values used for sorting output data.	25

LIST OF ABBREVIATIONS

Geometry and Tracking, version 4	GEANT4
Facility for Rare Isotope Beams	FRIB
Michigan State University	MSU
Department of Energy	DOE
Department of Energy Office of Science	DOE-SC
National Superconducting Cyclotron Laboratory	NSCL
Gamma-Ray Infrastructure For Fundamental Investigations of Nuclei	GRIFFIN
Germanium Double Sided Strip Detector	GeDSSD
Isotope Separation On-line	ISOL
Advanced Rare Isotope Laboratory	ARIEL

ACKNOWLEDGMENTS

My experience at the Colorado School of Mines has been absolutely incredible, and there are so many people that I'd like to thank for helping me achieve my goals while I've been here.

I must start by thanking my advisor Dr. Kyle Leach, who encouraged me from the very start to go after what I wanted and pursue this thesis. Whenever I had setbacks and frustrations, he never once doubted my ability to complete this work and assured me that if I just kept trying I would figure it out. Without his wisdom, knowledge, and support, I don't think I would have made it to this point. I also must thank Connor Natzke, who helped me overcome many of my early setbacks when I was trying to learn something totally foreign to me. In addition, Connor Bray provided help that was invaluable to the success of this thesis. He helped me with the analysis of the simulation results and was extremely patient while explaining his thoughts. I was lucky enough to be a part of a wonderful research group that was supportive, curious, and fun.

My family has and always will be my rock, and I would not be where or who I am today without them. My mother Dana has patiently listened and encouraged me through countless moments of doubt and fear. My father David has shown me what it means to believe in yourself and has always supported my decisions, even when I was unsure. And my sister Paige keeps me laughing and understands me better than anyone in the world. She stands up for herself and everyone around with a courage I admire and strive to have in my own life. Finally, my grandparents have all shown me endless love and support, for which I will be eternally grateful.

Finally, I want to thank my friends and wonderful partner Luke. I have met so many wonderful people at Mines that have piqued my curiosity and encouraged me to ask questions about everything around me. Rilee was my roommate for 3 years and Lydia was there for my first and last year in school. Luke, we always managed to find our way to each other in my first couple of years here, and I am so glad we did because the last three years have been better than I could have imaged. Your patience and support during this time has given me the courage to keep going and chase my

dreams knowing you'll be there beside me. And to my puppy Luna, gave me plenty of snuggles when I needed them most and taught me that love should never be conditional.

CHAPTER 1

INTRODUCTION

The atomic nucleus is composed of protons and neutrons which, when in a bound nuclear system, exhibit a complex structure. This structure varies depending on the combination of protons (Z) and neutrons (N) that are allowed. All of the nuclides that have been discovered to date can be seen in the nuclide chart, Figure 1.1. Around 3000 different nuclides have been discovered so far, but more than 7000 are expected to exist [1].

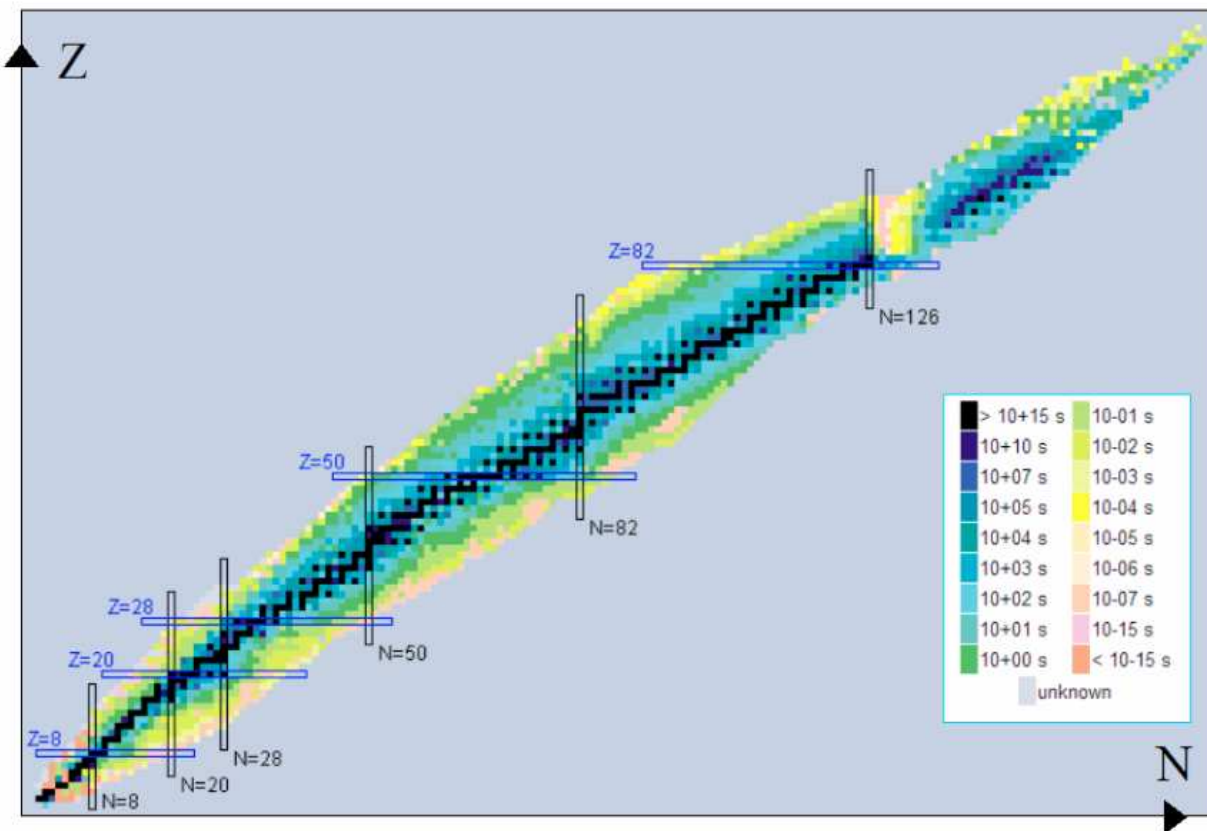


Figure 1.1: The nuclear chart shows all of the isotopes that have been discovered with the number of neutrons on the x-axis and number of protons on the y-axis. [2]

The valley of β stability is represented in Figure 1.1 by the squares in black along the center of the chart. The “driplines” (represented in Figure 1.1 by the left and right limits of the chart) are defined as the point past which no bound systems can exist.

1.1 Radioactive Decay

Radioactive decay is the process of an unstable nuclide moving toward stability by emitting various forms of radiation or undergoing conversion. Some of the most common types of decay are alpha decay, beta minus (plus) decay, gamma emission, electron capture, and spontaneous fission. When a nucleus decays, it is referred to as the parent nucleus, and the initial product nucleus of that decay is called the daughter nucleus. For each nucleus, the method in which it decays varies, and each mode has a specific probability. By studying nuclei and their daughter products, decay schemes can be constructed. One of the best ways to construct these decay schemes is by studying the decay characteristics such as types of decay, rate of decay, and probabilities of different forms of decay.

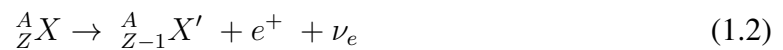
1.2 Beta Decay

Beta decay consists of either a proton converting to a neutron, or a neutron converting to a proton. The weak force allows for this transition through the exchange of a W boson and emission of either a positron and a neutrino, or an electron and an antineutrino.

- β^- decay



- β^+ decay



The decay times of nuclei that undergo beta decay can be fractions of a second up to thousands of years and depends on the binding energy of the nucleus. The binding energy of a nucleus is equal to its mass defect, which is defined as the amount of energy required to hold together a bound system of particles and is the amount of energy released when this bound system is created. This means that a bound system will have a smaller mass than the sum of its constituents in an unbound state. If a newly formed nucleus enters an excited energy level, and retains that mass

equivalence, it will need to be emitted in some form of radiation in order to return to the ground state. Until this energy is emitted, it will not have a mass deficit. This emission of energy can take the form of emitted electromagnetic waves (i.e. γ decay), the energy of an ejected particle (internal conversion), or the rest mass of an emitted particle (beta decay).

As a nucleus moves further away from stability, the nucleus' binding energy will decrease and thus drive the particle to move to a higher binding energy state, or stability, as shown in Figure 1.2. Imagining these parabolas all along the valley of stability in Figure 1.1, it becomes clear why nuclei on the edges of the chart are likely to decay, often extremely quickly, in order to come closer to stability.

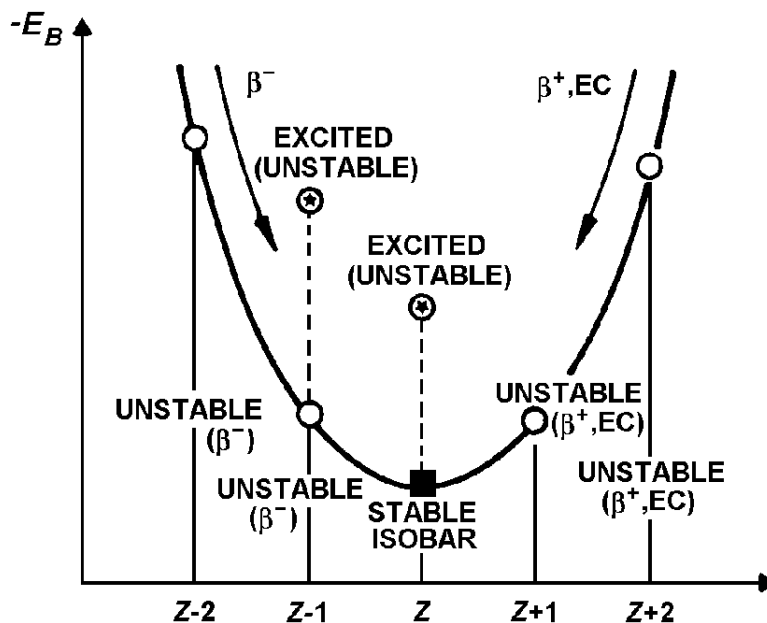


Figure 1.2: This illustration shows that nuclei with higher masses (and thus lower binding energies) will move toward stability by emitting radiation. [4]

It is common for nuclei to have more than one way to decay, sometimes even through the same type of radiation emitted, but at different energies. This results in the daughter nuclei being in an excited state, which is shown in Figure 1.3. Each branch has an associated probability, which is obtained experimentally, that determines how many of the decay events will occur through that pathway. These daughter nuclei will often be in an excited state, which results in further radiation

of gamma rays so the nuclei can return to a ground state.

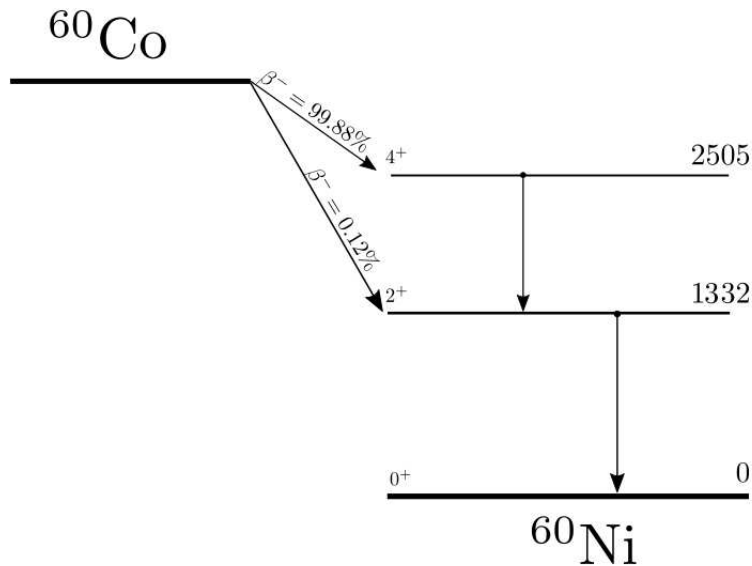


Figure 1.3: An example of a decay scheme that can be constructed by studying the decay of nuclei and their daughter products. [3]

1.3 Gamma Ray Emission

A nucleus in an excited state will attempt to return to stability through photon emission, or the emission of energy "packets" in the form of electromagnetic radiation [5]. When a photon is emitted, it will carry a discrete energy value equal to the difference between excitation states of the nucleus. Excited states of nuclei exist at distinct energy levels and nuclear configurations, and photon emission can be measured to characterize several nuclear properties and decay characteristics. In addition, studying gamma rays allows for probing the nucleus without perturbing it. By utilizing the spectroscopy of γ rays as they leave excited states of nuclei, knowledge of these excitation modes can be constructed [6].

For a single photon emission, there cannot be a change in momentum of zero during a single γ transition. A photon's multipolarity (ℓ) is the amount of angular momentum carried away by the photon, and has a minimum value equal to the difference between the angular momentum of the final and initial states ($\ell = \Delta I\hbar = |I_i - I_f|\hbar$). It has a maximum value equal to the combined

angular momentum of the two states ($\ell = (I_f + I_i)\hbar$). This means that the angular momentum carried by any photon must fit the condition shown below [7].

$$|I_i - I_f|\hbar \leq \ell \leq (I_i + I_f)\hbar$$

The other consideration that dictates the characteristics of the emitted photon is the conservation of parity. When a nucleus de-excites, the distribution of both matter and charge will change. If there is a change in the orbital or distribution of charge, this results in an electric field, but if there is a change in direction of the proton orbital a magnetic field will arise. The type of transition along with the angular momentum dictates the parity of the photon [7]. The change in parity, typically denoted by $\Delta\pi$, is the same as the change in whether the spin is positive or negative. The change in parity is different for electric and magnetic emissions, and can be calculated by the following [8]:

1. Electric: $\Delta\pi = (-1)^\ell$
2. Magnetic: $\Delta\pi = (-1)^{\ell+1}$

So a photon type of E1 would be electric with an ℓ value of 1, meaning it would have a change in parity. So this type is only possible if all of those requirements are met by the change in states of the nucleus. In general, the type of photon emitted depends on whether or not there is a change in parity and what ℓ values are possible for the change in state [8]. All possible forms of emission will occur at different rates as the nucleus de-excites.

The study of correlated beta and gamma decay using high resolution spectroscopic tools provides a powerful experimental avenue for obtaining fundamental quantities related to the structure of nuclear matter. This thesis presents an optimization study of decay experiments with a high efficiency HPGe clover array at FRIB.

1.4 Production of Rare Isotopes

In order to study the decay modes discussed in the previous sections, the radioactive ions to be studied must be produced and delivered to detection systems. The two main ways that experimental facilities achieve this is through:

1. Isotope Separation On-line (ISOL)
2. In-Flight Fragmentation

For the purpose of this thesis, only the fragmentation technique will be discussed.

In-flight fragmentation is used in several facilities around the world, including the National Superconducting Cyclotron Facility located on the campus of Michigan State University [9]. This method consists of creating a fast, heavy nucleus primary beam, which impinges on a target of light nuclei and "fragments" into smaller radioactive nuclides. These fragments maintain up to 90% of their initial kinetic energy [10]. The fragments are referred to as the "secondary beam", which have a wide range of energies and angular spreads that must be separated out to be delivered to experiments. Magnets set in an arc separate out the fragments in the secondary beam based on the mass of each ion. This production method is fast and able to produce very short-lived (even un-bound) nuclei [11]. However, the energies these beams are delivered at are quite high, around 200 MeV per nucleon, so not all detection systems are suitable for this method [12].

1.5 Facility for Rare Isotope Beams

The Facility for Rare Isotope Beams (FRIB) is a state-of-the-art facility currently being constructed at Michigan State University for the Office of Nuclear Physics in the U.S. Department of Energy Office of Science. The goal of the facility is to provide scientists with the equipment and opportunity to study rare and exotic isotopes that are typically only found in the most extreme conditions in the universe. The facility will be able to produce these rare isotopes in a timely and efficient manner, maximizing the discoveries that can be made. These discoveries are vital because they provide insights and advancements in a variety of fields including nuclear astrophysics,

fundamental interactions, medicine, and homeland security [13].

The goal of FRIB is to look beyond the previously observed nuclei, and gain information on nuclei that have never been created or studied before. New and innovative technology needs to be developed to perform these next-generation experiments. A schematic of the basic layout of the FRIB facility can be seen in Figure 1.4.

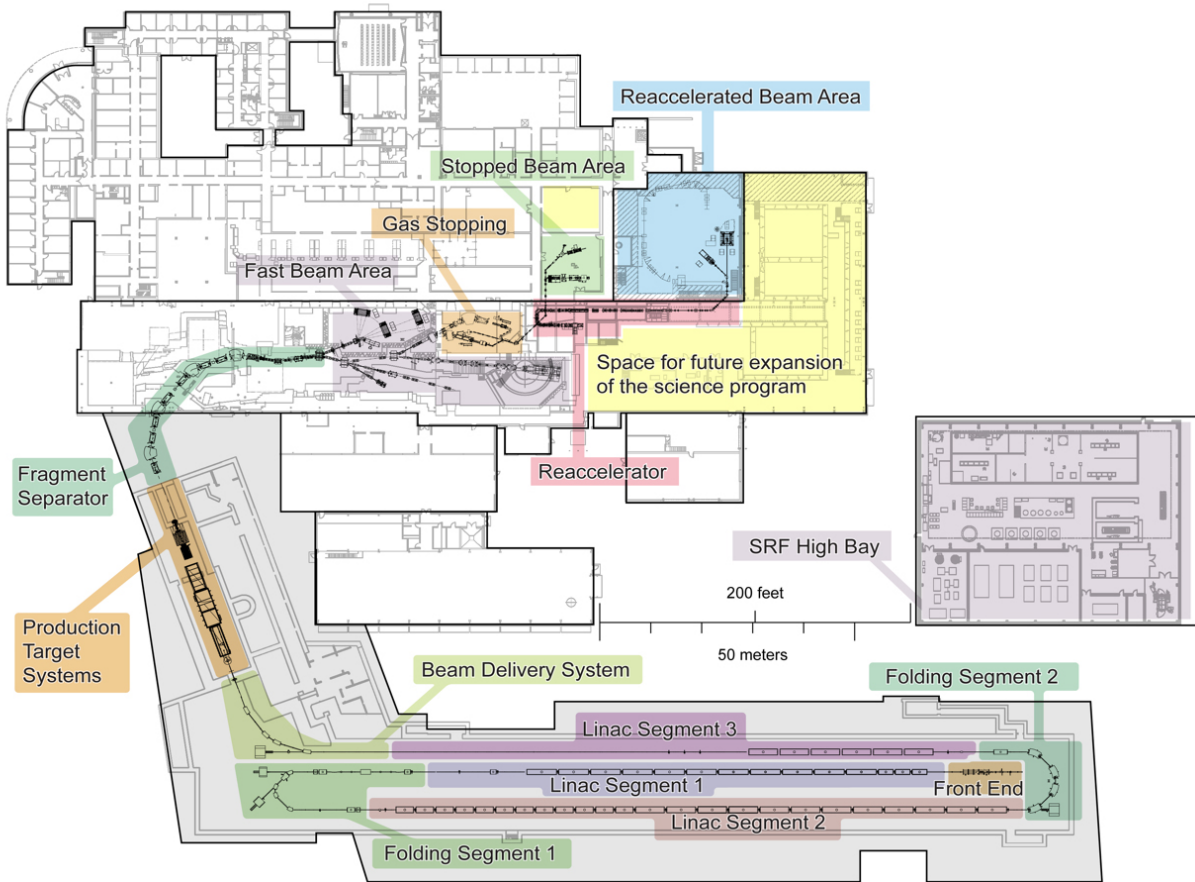


Figure 1.4: A schematic layout of the FRIB facility. [13]

The beam is produced using a linac driver system, which meets the goal of providing consistent and stable beams, which will then be delivered to the production target. This is where the rare isotopes will be produced and separated by in-flight fragmentation. [11]. For the purpose of decay studies, this beam of rare isotopes can then be transported to one of three different rooms, with associated energy ranges [14]:

1. The fast beam area - 200 MeV/u
2. The stopped beam area - 60 keV/q
3. The reaccelerated beam area - 0.3 - 20 MeV/u

The focus of the work is on fast beams, which will be directly implanted into a detector. Direct implantation is more efficient since the isotopes will be short-lived, and many targets typically used in decay experiments are too thin to use effectively with beams at such high energy and power [15].

FRIB will be able to accelerate all stable ions in the primary beam to at least 200 MeV/u with a power of 400 kW at the target surface, with an energy spread of 1% [12]. The ion species here will also range from protons to ^{238}U . To meet the demand of steady, reliable beams at FRIB, continuous wave (CW) beams are used to produce the radioactive ions. Due to the continuous wave, the power demand of 400 kW can be met with a current of less than 2 mA [12].

1.6 The FRIB Decay Station

The FRIB decay station is an experimental area where a suite of main and auxiliary detectors can be deployed in the fast or stopped beam rooms. The goals of the setup are to characterize nuclei through spectroscopy to understand fundamental symmetries, probe the edges of the nuclide chart, and test the standard model. To take complete measurements of decay properties to characterize these exotic nuclei, innovative detection technology that has high efficiency and resolution is critical.

The exotic nuclei that will be studied have decay half-lives longer than 100 ns [16]. In addition, multiple nuclei will be measured in a short period of time, requiring the utilization of the different areas of FRIB available to users. Due to the very low production rates of the radioactive ions, the detectors being used must be highly efficient, or the experiments would take an unreasonable amount of time before any meaningful amount of data could be taken.

The decay station at FRIB hopes to maximize its discovery potential by combining existing and new detector technologies (Figure 1.5). For example, combining detector types that detect

efficiently both low- and high-energy gamma rays is one approach to being able to run a broader range of experiments in one setting. This could be accomplished using detector arrays in combination with newer target detector technology.

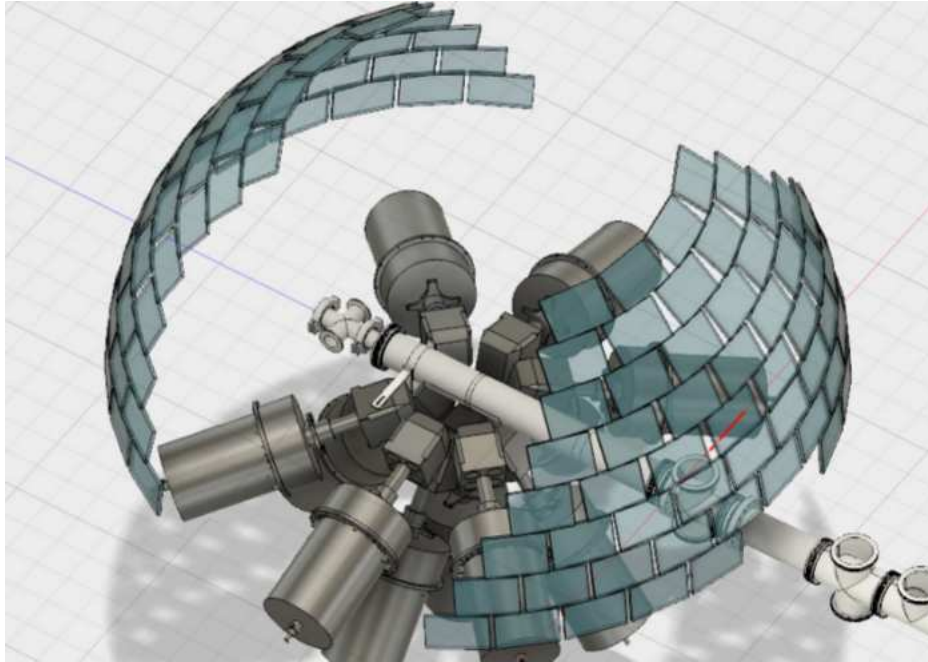


Figure 1.5: A design rendering of HPGe detectors combined with a new neutron array (NEXT) for the FRIB decay station. [17]

The goal of this work is to examine a specific example of two detector types being combined into a new configuration. This new combination of detectors, explained further in the next chapter, will lead to better discovery potential by combining high efficiency detection of both β and γ decay using high purity germanium detectors. This work also optimizes the size of two of these detector types in order to achieve the highest gamma ray detection efficiency possible from this configuration.

CHAPTER 2

GAMMA RAY DETECTOR TECHNOLOGY

Gamma rays are a deeply penetrating radiation, and carry no intrinsic charge, so they are difficult to detect using conventional detector methods. Instead of typical gas or liquid scintillators, high Z, solid materials are desirable to increase the likelihood of a photon interaction. This section will cover some of the solid state γ ray detector technologies that have been used, focusing on the high purity germanium (HPGe) detectors.

The two most useful gamma spectroscopy detector categories are inorganic scintillators and germanium semiconductors [18]. The most common inorganic detector type is the thallium-doped sodium iodide NaI(Tl) scintillator and is economically available in large quantities. These detectors have a high efficiency, but have poor energy resolution at 5-10%, making them ineffective for resolving complex γ -ray spectra. In contrast, germanium detectors have a low absolute efficiency, but much sharper energy resolution at a few tenths of a percent [18].

Cesium iodide (CsI) scintillators have higher efficiency and light yield than NaI(Tl) scintillators, with thallium doped CsI scintillators yielding 65,000 photons/MeV compared to the 38,000 for NaI [19]. However, they are still inorganic scintillators and thus have a lower resolution than germanium detectors.

The process of lithium drifting is used in both silicon and germanium crystals to increase the depletion depth that can be achieved. High depletion depths, or active volumes, are imperative for detecting the more penetrating radiation of γ rays. For silicon of high purity, a depletion depth of 1-2 mm is possible, but depletion depths of up to 5-10 mm are achievable after lithium drifting is performed. Due to the low atomic number of silicon, Si(Li) detectors are typically only used for very-low-energy gamma ray detection instead of general γ -ray spectroscopy [18].

For a couple of decades, the process of lithium drifting was also widely used with germanium detectors, since it was the only way large active volumes could be generated. Ge(Li) detectors were

used to generate the first gamma spectroscopy array, The Escape Suppressed Spectrometer Array (TESSA) in Riso, Denmark [6]. The development of this configuration opened the possibility for coincidence studies to be undertaken because of the use of multiple detectors around the target, as can be seen in Figure 2.1.

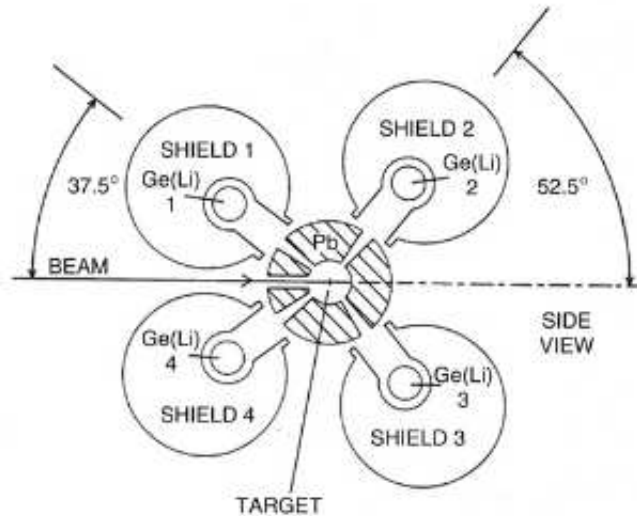


Figure 2.1: A schematic of TESSA, the original gamma ray detector array. [6]

2.1 High Purity Germanium Detectors

As detector technology progressed, large volumes of high purity germanium (HPGe) crystals were made that could compete in size with Ge(Li) detectors, making Ge(Li) unpopular in modern detection. In addition, the impurity concentration went from 10^{13} atoms/cm⁻³ to 10^{10} atoms/cm⁻³ [20], eliminating the need to introduce Li into the detectors. This also meant that larger crystals could be made, which leads to higher energy efficiency, since more charges can be collected in a single detector.

The incident γ rays ionizes the detector material, which results in charges that are directly collected in the semiconductor material, where an external electric field is applied. When the ionizing radiation penetrates the detector, an electron-hole pair is created. Due to the electric field, the electrons (negative charge) and holes (positive charge) will move to opposite electrodes and generate

the pulse that is measured [21].

HPGe detectors have depletion depths of several centimeters, making them more efficient at detecting γ rays [18]. The energy resolution is highest in these detectors, and is determined by three factors:

1. The inherent statistical spread in the number of charge carriers
2. Variations in the charge collection efficiency
3. Contributions of electric noise

One way the maximization of efficiency is achieved is by tailoring the size of the detector to the needs of the experiment. Larger volumes have more electric noise and magnified carrier loss, so they should only be used for high-energy experiments. Smaller volume, planar HPGe detectors are the best choice for low-energy γ rays, since the active volume does not need to be as large, and the background noise from electronics is minimized [18].

2.2 Photon Interactions in HPGe Detectors

There are three main types of interactions between γ rays and HPGe detectors, and they are photoelectric absorption, Compton scattering, and pair production. Each type of interaction will have a different result in the energy recorded by the detector, so it's important to understand the differences between them and how they will be represented in the work shown later.

Figure 2.2 shows the cross sections for the different potential photon interactions that take place in germanium detectors. The photoelectric effect is the dominant process for γ rays at 0-150 keV, and occurs when a photon transfers all of its energy to a recoil electron. This recoil electron is ejected from the atom and creates that electron-hole pair that results in an output pulse proportional to the incoming energy [21]. This type of interaction will contribute to the full-energy photopeak in the energy spectrum of interest.

Compton scattering is a process where the incoming gamma ray collides with an electron and imparts part, but not all, of its energy. Then, the electron becomes energized enough to leave and

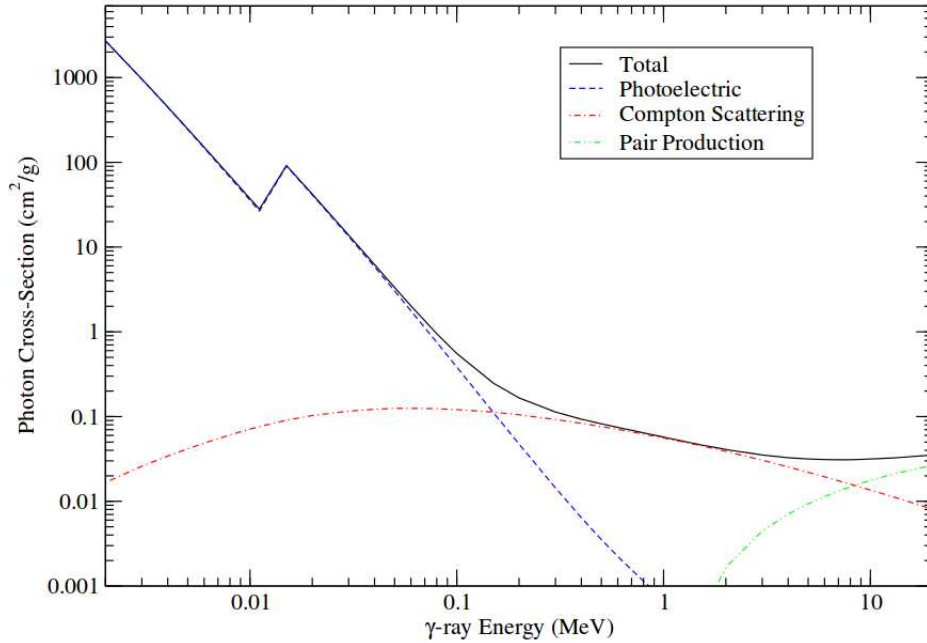


Figure 2.2: Cross sections for photoelectric absorption, Compton scattering, and pair production in germanium. [18]

ionize the atom, while the photon scatters and moves along a different path. This process can occur multiple times in detectors with large volumes as the photon collides with electrons in the material, until the photon loses all of its energy or leaves the detector. This process dominates for incoming γ rays with energies $150 \text{ keV} \leq E_\gamma \leq 8.5 \text{ MeV}$. This process will create a Compton continuum below the photopeak energy if the photon leaves the detector without imparting all of its energy. It will instead result in a full-energy pulse if there are multiple Compton events that all occur inside of the HPGe crystal and the final event is a photoelectric interaction [21].

The final interaction type is pair production, or the spontaneous creation of an electron-positron pair. For this process to occur, the incoming energy must be greater than the energy of the combined rest mass of this pair, or $2m_0c^2$ [22]. For both the electron and positron, the rest mass, m_0 , is $511 \text{ keV}/c^2$. This means the incident photon must have a minimum energy of 1022 keV for this process to occur.

2.3 Relevant Detector Designs

The two most relevant detector types for this work are small, planar HPGe detectors and an array of large HPGe crystals. The Gamma-Ray Infrastructure For Fundamental Investigations of Nuclei (GRIFFIN) array, currently housed at TRIUMF National Lab in Vancouver, Canada, is of particular interest (Figure 2.3). This array consists of 16 clover style configurations of HPGe detectors, with each clover housing 4 crystals [23]. This is in part because the array design allows more coverage of the volume around the source. This design also allows for detection of coincident photon emission, which is a characteristic beneficial to the experiments that will be run with this new detector configuration at FRIB. There will not be an exact copy of the GRIFFIN array made, but it was used for the purposes of this study because it has similar characteristics to what will eventually be used at FRIB and already had extensive coding infrastructure in place, which will be discussed in the next chapter.

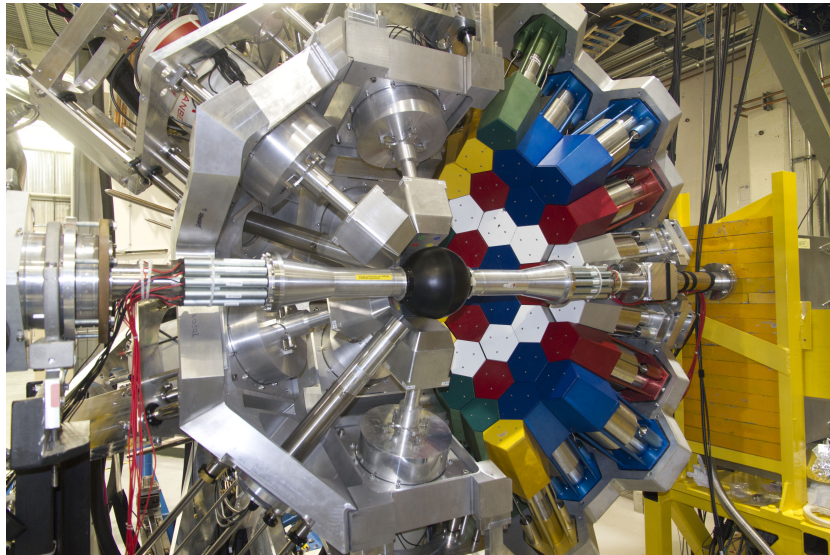


Figure 2.3: Half of the GRIFFIN array is shown on the left hand side of the configuration, with the clovers closely surrounding the center sphere. [23]

The main detector type of interest for this work is the germanium double sided strip detector (GeDSSD), which can be seen in Figure 2.4. The GeDSSD has high efficiency and energy resolution for photons, as well as for charged particles [24], separating it from the clover style array

previously discussed. Orthogonal strips of germanium overlapped allow for precise determination of hit position, as seen in Figure 2.5. This high precision tracking, in addition to the ability to detect charged particles, allows for a wide range of experiments to be run, particularly the study of nuclei that emit a beta particle and photon almost simultaneously. If there is a beta decay in the GeDSSD, and then the surrounding clover array can make a temporal match to a detected photon, then it can be determined those two decays were emitted at essentially the same time. This is the type of study and information that allows further understanding of nuclear structure and properties.

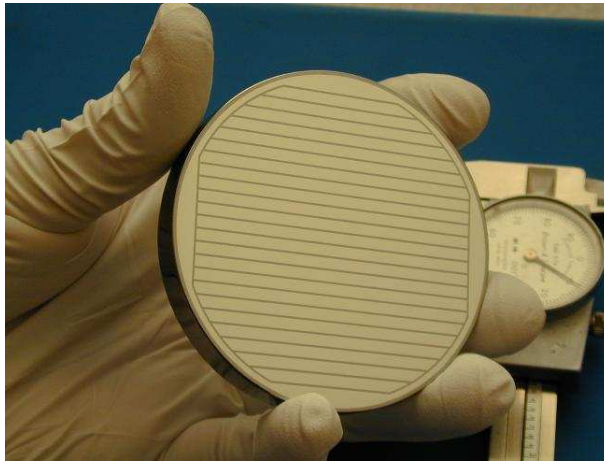


Figure 2.4: The germanium double sided strip detector created by PhD students at UT Knoxville. [25]

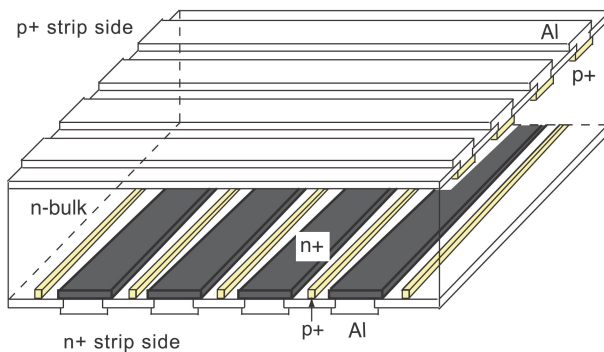


Figure 2.5: The strip detector consists of orthogonal layers of positive p-type and negative n-type highly doped silicon strips on either side of the detector. [26]

The goal of the work is to be able to show that the combination of these two detector types would allow for the study of the most exotic nuclei. The high degree of position tracking of both

charged particles and gamma rays in the GeDSSDs combined with the high efficiency of a full clover style gamma detector array appears to be a viable candidate for this purpose. The rest of this thesis will discuss the results of simulating the combination of the GRIFFIN array and two GeDSSDs of different thicknesses.

CHAPTER 3

MODELING GERMANIUM DOUBLE SIDED STRIP DETECTORS

In addition to studying the viability of the combination of GeDSSDs with a clover style gamma detector array, the project aimed to optimize the radius of the GeDSSDs. This required the development of a simulation that would emulate the real-world outcomes of experiments. The rest of this chapter will explain the process used to generate simulation results and create efficiency curves.

3.1 Geant4 Simulation Environment

To meet the goals of the project, Geometry And Tracking v.4 (Geant4) was chosen as the programming toolkit because it already had the necessary physics embedded as well as the necessary modeling capabilities. The program's base function is to simulate the passage of particles through matter, which can be applied to a wide range of fields, such as medical and nuclear physics [27]. This toolkit was developed over many years by software engineers and physicists all over the world to provide a way to accurately model real world phenomena in a computational environment.

Geant4 provides the basics for defining an environment to simulate, including the ability to create complex geometries and add materials with all relevant information already programmed. In addition, a more comprehensive simulation package that works in the Geant4 toolkit environment allowed for repeated simulations to be done with accuracy and consistency. A research group at the University of Guelph in Ontario, Canada created a collection of files called "detectorSimulations_v10" that worked together to create an environment where the GRIFFIN array and its auxiliary detectors could be modeled to a high degree of precision. This simulation package was used because GRIFFIN serves as a suitable candidate for the HPGe clover style gamma detector array, and then the GeDSSDs could be integrated to run accurate simulations. Every detector type in the suite has an associated geometry file that translates the real-world dimensions into a computational model. There are then several files that dictate how those detectors interact with each other and the simulation environment, as well as how data recorded during the simulation is output.

One important quality of both Geant4 and the simulation package is the ability to visualize the detectors modeled in the GRIFFIN suite to ensure there are no overlapping volumes. In addition, the geometry can be visualized to ensure it is being modeled as expected. An example of this visualization can be seen in Figure 3.1, where a single clover from GRIFFIN is simulated with one source event.

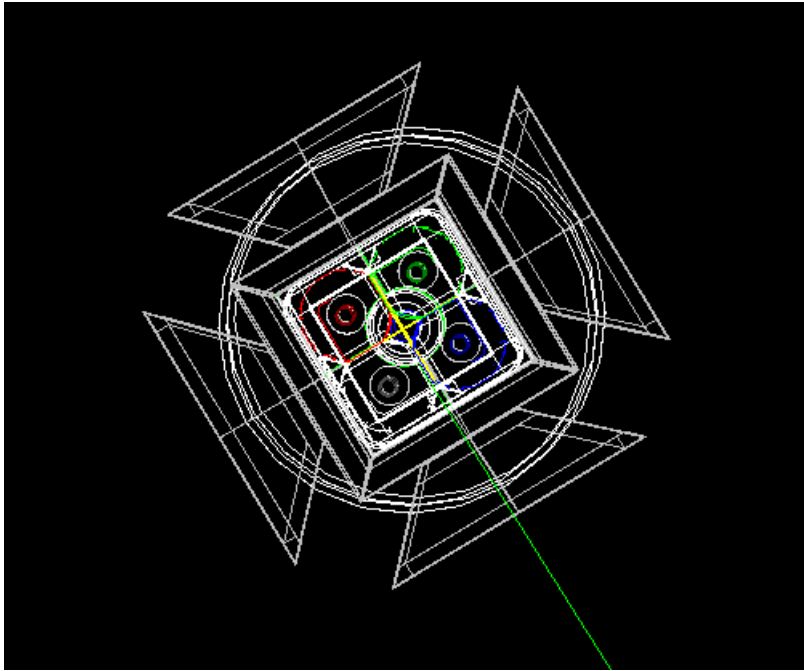


Figure 3.1: A single clover in the GRIFFIN array is shown here being modeled in Geant4 with the simulation program used for this project. This complicated clover shows 4 crystals in different colors with associated geometries.

Before any detector can be modeled, a geometry file has to exist and be incorporated into the simulation package. The GeDSSDs had to have a geometry file written for them based on their previously recorded dimensions. All of the information needed to create a new geometry, or detector, is the dimensions and material of the object. Materials and their associated physical and nuclear properties are included in the Geant4 physics libraries. The geometry can be made from the predetermined, simple shapes or more complicated, unique shapes that the user defines. The following sections discuss how the file was created for the GeDSSDs and then integrated into the remaining simulation files.

3.2 Creating the Geometry File for the Germanium Double Sided Strip Detectors

To study the effectiveness of the GeDSSDs in conjunction with the GRIFFIN array, a new geometry file was developed. The dimensions were provided by Professor Sean Liddick at Michigan State University from an initial attempt at modeling the detectors. The geometry was a simple construction of cylinders and included a thin GeDSSD, a thick GeDSSD, and a cryogenic chamber with end caps that housed the detectors to keep them cool during experimentation. To create this, first the materials were defined, then the shapes, and finally made logical volume and placed in the environment.

The materials used were elemental aluminum and germanium. The cryogenic chamber was composed of aluminum, and though this is a simplification, the aluminum is the major element and is not going to effect the results of the simulation. The thin GeDSSD had a thickness of 1.5 mm and the thick GeDSSD had a thickness of 1 cm. The original radius of both of the detectors was 4.5 cm. All of the dimensions for the detectors and associated cryogenic chamber can be found in Table 3.1. One of the main goals of the project was to optimize the radius of the GeDSSDs, so simulations were done for radii values between 0.5 cm - 4.5 cm, at 0.5 cm intervals for a total of 9 radii values to be tested.

Table 3.1: This table displays all of the dimensions used to create the geometry file for the GeDSSDs and associated cryogenic chamber.

Part	Dimension	Value
Thin GeDSSD	Thickness	1.5 mm
	Radius	0.5 - 4.5 cm
Thick GeDSSD	Thickness	1 cm
	Radius	0.5 - 4.5 cm
Cryogenic Chamber	Length	8.89 cm
	Inner Radius	6.6 cm
	Outer Radius	9.6393
Chamber End Caps	Thickness	1.143 mm
	Radius	6.6 cm

A physical volume takes the real-world dimensions and makes the necessary shapes out of them. In this case, all of the volumes were simple cylinders constructed as G4Tubs, an object identifier in the Geant4 program. A physical volume must be made into a logical volume before it is able to interact with the particles in a simulation. In making the physical volume a logical one, the placement and rotation were defined. The directional vector defines the axis along which the shape lies and all cylinders were oriented parallel to the z axis, resulting in a G4ThreeVector with the coordinates (0., 0., 1.) for all 4 volumes. Then, the position vector, another G4ThreeVector, was defined for every object. At FRIB, the beam will impinge on the thin GeDSSD, so it was placed directly at the origin of the simulation environment. With high energy gamma rays, it is possible that the incoming beam will not directly implant in the thin GeDSSD, but move through it and implant in the thick GeDSSD. However, for the purpose of this thesis, all gamma rays were simulated as starting in the center of the thin GeDSSD. The thick detector was placed with 5 mm of space in between the two faces of the detectors. This parameter could be changed in the future to determine its effect on the efficiency of the entire detector configuration. The cryogenic chamber is an annulus centered around the origin of the simulation environment, so the thin detector is in the middle of the chamber and the thick detector is slightly off center. The end caps are situated as they would be in real life, sitting directly on the ends of the chamber. As seen in Table 3.1, the radius of the end caps is the same as the inner radius of the chamber so that it creates a seal. After all of these volumes were defined for the GeDSSD, they were placed into a group so they could be called and built as one detector. The entire geometry file can be seen in Appendix A, and a graphic of the GeDSSD configuration can be seen in Figure 3.2.

In Figure 3.2, the two outer discs represent the ends of the cryogenic chamber, where the endcaps are not differentiated since they sit flush to the chamber. The yellow disc is set in the center of the cryogenic chamber and represents the thin GeDSSD where the experimental beam would implant in FRIB. There is then a gap between the thin and thick GeDSSD, represented by the light red rectangle. This figure is exactly replicated in the detectorSimulations package so that simulations could be run in the environment. First, however, several other files needed to be altered

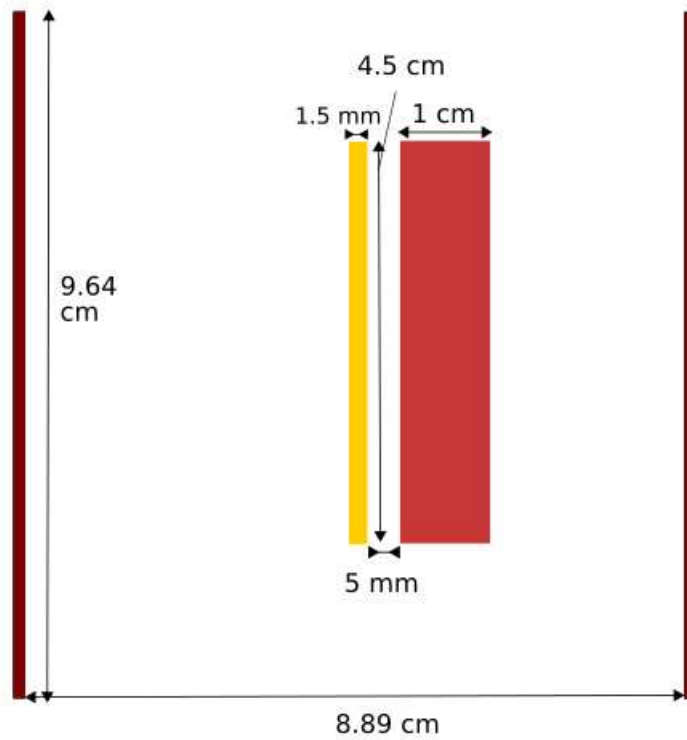


Figure 3.2: This graphic depicts the layout of the two GeDSSDs inside the cryogenic chamber with associated dimensions.

to incorporate this GeDSSD configuration.

3.3 Incorporating the GeDSSD into the Simulation Package

There are several files in the detectorSimulations_v10 package that integrate all of the detector geometry files so they work together. Some depend on the actual detector types being written in them, while others work without directly mentioning the detector types. The files that needed to be modified to incorporate the GeDSSDs can be seen below:

- DetectorConstruction
- DetectorMessenger
- EventAction
- HistoManager
- SteppingAction

All of the files have both header (.hh) and C++ source code (.cc) file types, just as the geometry files. Only the source code files will be discussed since the header files only contain definitions and function names called in the .cc files. To include the GeDSSDs in the simulation environment, new code was added to the above files that depended on what function that file accomplished.

The DetectorConstruction.cc file is responsible for calling the functions in the geometry file described above to build and place the detectors in the simulation environment. As seen in Figure 3.3, the file creates a new logical world, builds the detector, places the detector, and then creates a histogram for that detector to save the data that is recorded. This logic is the same for all other detectors that can be called in this environment, though the specific functions called may vary.

The DetectorMessenger.cc file facilitates communication between the runtime macro file that the user creates and the geometry files in the package. The lines shown in Figure 3.4 determine the syntax of the input line used to place the GeDSSDs and cryogenic chamber in the simulation

```

void DetectorConstruction::AddDetectionSystemGedssd(G4int ndet) {
    if(fLogicWorld == NULL) {
        Construct();
    }

    DetectionSystemGedssd* pGedssd = new DetectionSystemGedssd();
    pGedssd->Build();
    pGedssd->PlaceDetector( fLogicWorld, ndet );

    HistoManager::Instance().Gedssd(true);
}

```

Figure 3.3: The DetectorConstruction.cc file tells the environment to call the functions in the geometry file that builds and places the GeDSSSD setup.

```

fAddDetectionSystemGedssdCmd = new G4UICmdWithAnInteger("/DetSys/det/addGedssd",this);
fAddDetectionSystemGedssdCmd->SetGuidance("Add Detection System Gedssd");
fAddDetectionSystemGedssdCmd->AvailableForStates(G4State_PreInit,G4State_Idle);

```

Figure 3.4: The lines in DetectorMessenger.cc allow the simulation environment to communicate with the runtime macro that the user inputs.

environment. These lines are used for every detector, where the specific syntax of the input line varies.

The next file, EventAction.cc, tracks the hits and steps (or events) that occur during the simulation and allocates them to specific detectors. It also creates the categories used in the ntuple TTree that will be outlined later. The file starts by calling the "fill" functions for all detectors (Figure 3.5) which is how the histogram is filled out with data for the GeDSSD. As the simulation tracks the particle interactions, it also tracks their position and energy. There is a minimum energy threshold, set in the header file, that the particle interaction must have to be recorded. The first for loop goes through both of the GeDSSD detectors, and then enters the if loops that records the hits in the histogram. If the energy is above the minimum energy threshold, then it fills the histogram and adds to the sum of hits recorded. There are two detectors for the GeDSSD, which is defined as the variable "MAXNUMDETGEDSSD" seen in both Figure 3.5 and Figure 3.6. The second nested if loop fills the histogram that tracks the entire energy spectrum recorded for both detectors. The for loop in Figure 3.6 shows the initialization of the vectors that keep track of the hits recorded in the simulation.

```

void EventAction::FillGedssd() {
    G4double energySumDet = 0;
    for (G4int j=0; j < MAXNUMDETGEDSSD; j++) {
        if(fGedssdEnergyDet[j] > MINENERGYTHRES) {
            if(WRITEEDEPHISTOS) HistoManager::Instance().FillHisto(kGedssdEdep, fGedssdEnergyDet[j]);
            if(WRITEEDEPHISTOS) HistoManager::Instance().FillHisto(kGedssdEdepDet0+j, fGedssdEnergyDet[j]);
            energySumDet += fGedssdEnergyDet[j];
        }
    }
    if(energySumDet > MINENERGYTHRES) {
        if(WRITEEDEPHISTOS) HistoManager::Instance().FillHisto(kGedssdEdepSum, energySumDet);
    }
}

```

Figure 3.5: The fill function in the EventAction.cc file for the GeDSSDs attributes every event recorded in the simulation to a histogram for data output.

```

for (G4int ii = 0; ii < MAXNUMDETGEDSSD; ii++) {
    fGedssdEnergyDet[ii] = 0;
    fGedssdTrackDet[ii] = 0;
}

```

Figure 3.6: This code shows the initialization of the vectors that track particle interactions in the GeDSSDs while the simulation is running.

The HistoManager.cc file that is referenced in the EventAction.cc file creates the histograms filled during simulations. The file outlines the general format for the histograms that will be made for the simulation and is filled when a specific detector type is called by the user. Figure 3.7 shows the code for the GeDSSD that modify the general histogram format to be specific to those detectors.

The final file that was modified to incorporate the GeDSSDs was the SteppingAction.cc file that defines the identifiers for both particle types and detector types. The particle types recognized by the file and their assigned values are presented in Table 3.2.

If the particle type is not recognized, it will be assigned a value of zero. For the purposes of this project, only particles of type 1 (gamma rays) were considered. This file is critical for communicating to the simulation program that a logical volume needs to have the interactions that occur in it recorded. Figure 3.8 shows that both the thin and thick detectors are assigned a specific systemID value that is used when creating histograms and tracking particle interactions. This is how information can be extracted for individual detectors and analyzed after the simulation is run.

```

if(fGedssd) { //gedssd detector

    name = "gedssd_crystal_edep";
    MakeHisto(analysisManager, name, title, xmin, xmax, nbins);
    fGedssdHistNumbers[0]=fMakeHistoIndex;

    name = "gedssd_crystal_edep_sum";
    MakeHisto(analysisManager, name, title, xmin, xmax, nbins);
    fGedssdHistNumbers[1]=fMakeHistoIndex;

    for(G4int i=0; i < MAXNUMDETGEDSSD; i++) {
        detString = G4intToG4String(i);

        name = "gedssd_crystal_edep_det" + detString;
        MakeHisto(analysisManager, name, title, xmin, xmax, nbins);
        fGedssdHistNumbers[i+2]=fMakeHistoIndex;
    }
} //if(fGedssd)

```

Figure 3.7: The HistoManager.cc file gives each detector type a histogram name and index that the EventAction.cc file uses and assigns information to, shown here with the GeDSSDs.

These files all work together to communicate to the Geant4 toolkit what geometries to build, where to build them in the environment, and what processes will be simulated and how they are recorded. Creating total efficiency curves for the detector configuration with the GeDSSDs and the GRIFFIN array was the primary goal of the project, and the incorporation of the GeDSSDs with the detectorSimulations_v10 simulation package resulted in the ability to achieve this. The thin and thick detectors were placed in the middle of the GRIFFIN array so their response to a monoenergetic γ ray beam could be studied.

Table 3.2: The particle types used in simulations have assigned identifier values used for sorting output data.

Particle Type	Assigned Value
gamma	1
e ⁻	2
e ⁺	3
proton	4
neutron	5
deuteron	6
C12	7

```

found = volname.find("GeThinDetector");
if (edep !=0 && found!=G4String::npos) {
  SetDetNumberForGenericDetector(volname);
  fEventAction->GedssdDet(edep, stepl, fDet-1);
  mnemonic.replace(0,3,"GTN");
  mnemonic.replace(3,2,G4intToG4String(fDet));
  mnemonic.replace(5,1,GetCrystalColour(fCry));
  systemID = 9500;
  fEventAction->AddHitTracker(mnemonic, evntNb, trackID, parentID, fStepNumber, particleType,
processType, systemID, fCry-1, fDet-1, edep, pos2.x(), pos2.y(), pos2.z(), time2, targetZ);
}

found = volname.find("GeThickDetector");
if (edep !=0 && found!=G4String::npos) {
  SetDetNumberForGenericDetector(volname);
  fEventAction->GedssdDet(edep, stepl, fDet-1);
  mnemonic.replace(0,3,"GTK");
  mnemonic.replace(3,2,G4intToG4String(fDet));
  mnemonic.replace(5,1,GetCrystalColour(fCry));
  systemID = 9600;
  fEventAction->AddHitTracker(mnemonic, evntNb, trackID, parentID, fStepNumber, particleType,
processType, systemID, fCry-1, fDet-1, edep, pos2.x(), pos2.y(), pos2.z(), time2, targetZ);
}

```

Figure 3.8: The SteppingAction.cc file defines the identifier values for particles, interaction processes, and detectors. Only detectors with systemID values will have their interactions recorded.

3.4 Running Simulation with GeDSSDs

In order to perform a simulation using the GeDSSDs, a macroscopic file that indicates which detectors will be used and what type of experiment or interaction will be simulated was written and will be referred to as a run macro. Once the new detector was incorporated into the simulation package and compiled successfully, a full simulation was run and visualized to ensure they were modeled as expected.

A run macro has several key features to run a simulation, and then there are options to fit the specific needs of the simulated experiment. The first few lines of the code initialize the simulation and specify the physics library, which can be seen in Figure 3.9. To add detectors to a simulation, the user includes the line defined for the detector in the DetectorConstruction.cc file, seen in Figure 3.3. The GRIFFIN detector is added one crystal at a time, so the number of crystals can be modified to fit the needs of the simulation. Every crystal actually requires 2 input lines, which is different than the other detectors in the suite that require only 1. To add the GeDSSDs, the line used in the run macro was:

```
/DetSys/det/addGedssd 2
```

The 2 after the text is the second input, and indicates that 2 detectors total are being added. The next major component of the run macro is setting the runtime beam. First, the type of particle in the beam is set, which can be a single particle type or an ion. If it is an ion, such as ^{60}Co , the nucleus limits are set in the next line and a beam of those ions will be produced and decay to create interactions in the detectors. The beam for this work was a monoenergetic γ ray beam at various energies. Once the particle is set, the energy is set in the next line. Finally, the beam is "turned on" by telling the program how many particles of the type described above are to be fired. For this project, 1,000,000 particles were fired for every simulation. All of the lines described above can be seen in the full run macro shown in Figure 3.9.

To execute the run macro given, the input line (if in the directory with the GRIFFIN executable)

```
./Griffinv10 run.mac
```

is given in the terminal environment. This executes the run macro in the simulation package and will begin to run the simulation and collect results. Once finished, these results will be output in the file

```
g4out.root
```

This file contains all of the hit and step information recorded in histograms as was outlined by the supporting files in the simulation package. There is also an option that can be included that will pull up a visualization of the detectors and simulation. The line tells the program to execute another macro specifically made for visualizing in this simulation package. Again, this macro can be found in the GitHub repository mentioned previously. The GeDSSD combination was visualized to ensure that it was being modeled as expected, and the result of this visualization is seen as a rendered version in Figure 3.10.

```

/DetSys/phys/SelectPhysics emlivermore
/run/initialize
/process/em/fluo true
/process/em/auger true
/process/em/pixe true

/DetSys/det/SetCustomShieldsPresent 0
/DetSys/det/SetCustomRadialDistance 11 cm
/DetSys/det/SetCustomExtensionSuppressorLocation 0
/DetSys/det/includeGriffinHevimet 0

/DetSys/det/SetCustomDeadLayer 1 1 0
/DetSys/det/addGriffinCustomDetector 1
/DetSys/det/SetCustomDeadLayer 2 2 0
/DetSys/det/addGriffinCustomDetector 2
/DetSys/det/SetCustomDeadLayer 3 3 0
/DetSys/det/addGriffinCustomDetector 3
/DetSys/det/SetCustomDeadLayer 4 4 0
/DetSys/det/addGriffinCustomDetector 4
/DetSys/det/SetCustomDeadLayer 5 5 0
/DetSys/det/addGriffinCustomDetector 5
/DetSys/det/SetCustomDeadLayer 6 6 0
/DetSys/det/addGriffinCustomDetector 6
/DetSys/det/SetCustomDeadLayer 7 7 0
/DetSys/det/addGriffinCustomDetector 7
/DetSys/det/SetCustomDeadLayer 8 8 0
/DetSys/det/addGriffinCustomDetector 8
/DetSys/det/SetCustomDeadLayer 9 9 0
/DetSys/det/addGriffinCustomDetector 9
/DetSys/det/SetCustomDeadLayer 10 10 0
/DetSys/det/addGriffinCustomDetector 10
/DetSys/det/SetCustomDeadLayer 11 11 0
/DetSys/det/addGriffinCustomDetector 11
/DetSys/det/SetCustomDeadLayer 12 12 0
/DetSys/det/addGriffinCustomDetector 12
/DetSys/det/SetCustomDeadLayer 13 13 0
/DetSys/det/addGriffinCustomDetector 13
/DetSys/det/SetCustomDeadLayer 14 14 0
/DetSys/det/addGriffinCustomDetector 14
/DetSys/det/SetCustomDeadLayer 15 15 0
/DetSys/det/addGriffinCustomDetector 15
/DetSys/det/SetCustomDeadLayer 16 16 0
/DetSys/det/addGriffinCustomDetector 16
/DetSys/det/addGedssd 2

/DetSys/app/addGriffinStructure 0
/control/verbose 1
/run/verbose 1
/event/verbose 0
/tracking/verbose 0

/gun/particle gamma
/DetSys/gun/efficiencyEnergy 1000 keV
/run/beamOn 1000000

```

Figure 3.9: The run macro used to generate the efficiency curves needed for this project.

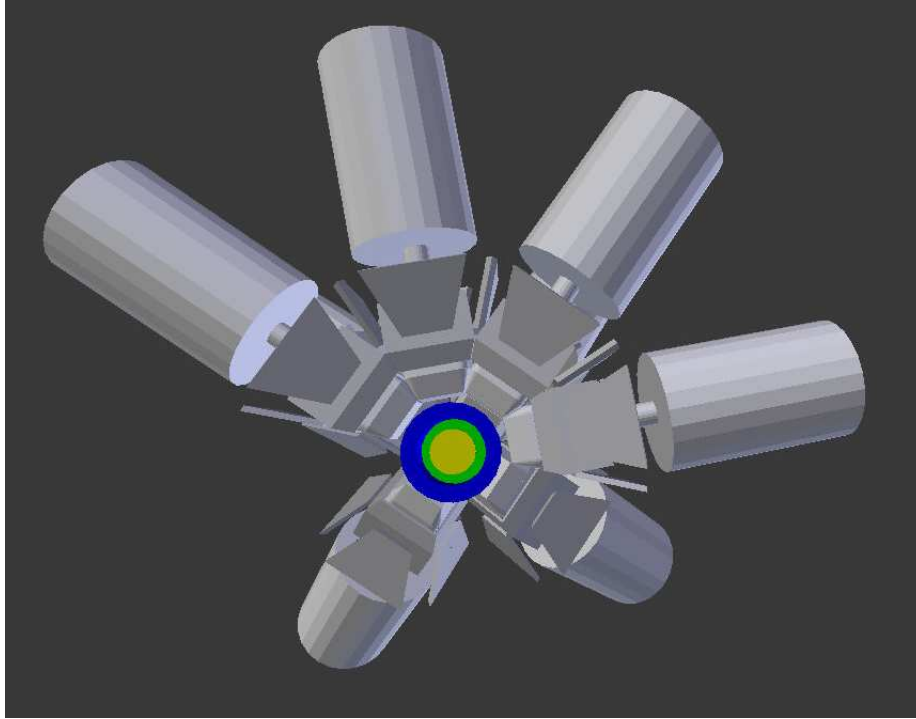


Figure 3.10: This image rendered in Blender shows the thick GeDSSD inside of the cryogenic chamber inside of half of the GRIFFIN array.

The data file output from the simulation will not resemble data that would be collected in a real-life experiment, which is not suitable when simulating what would happen in an experiment. The program NTuple fixes this problem by taking a root output file and smearing the data in a way that resembles how events would actually appear in an experiment. This program needed no changes made to it in order to work with the data from the GeDSSDs. To run NTuple, there needs to be an input file (original root file) and a settings file. The settings file was also not altered in any way for the GeDSSDs. To run a file through NTuple, the input command is as follows:

```
/path/to/NTuple -v1 1 -if /path/to/inputFile.root -sf /path/to/Settings.dat
```

where the /path/to changes depending on where those files are stored in the users directory. All of these above commands result in a data file that can be analyzed to extract information on how the experiment would look if run in a real-world setting. Firing a specified number of monoenergetic γ rays at the GeDSSDs and GRIFFIN array allowed for an efficiency to be calculated for that energy

by dividing the number of detected photons by the number fired. To meet the goal of the project, this process had to be done many times over to create the efficiency curves desired.

3.5 Automating Simulations

An efficiency curve shows how many particles interact with the detector compared to how many source particles were emitted in a given time frame, for every particle energy level. For example, if 100 gamma rays at 500 keV were emitted from the source in a given time frame, but only 50 were detected, then the absolute efficiency of the detector would be 50%. This information for detectors is important because in an experimental setting, the known efficiency of the detector is used to back-calculate the actual number of decay particles being emitted from the source. To understand the detection capabilities and viability of this new detector configuration, total efficiency curves needed to be constructed. To do this, monoenergetic gamma ray beams were generated in the middle of the thin GeDSSD at energies from 15-1000 keV in 5 keV increments. Then, the interactions detected were divided by the number of gamma rays initially simulated as being emitted from the source to gather the individual contributions to total efficiency from each detector, resulting finally in total efficiency.

The first step in proving the viability of the suggested combination of detectors is showing that there would not be a decrease in overall detector efficiency by comparing the curves from the new combination to ones for individual detectors. Since efficiency curves require simulations with energy levels at regular intervals, and only monoenergetic γ ray beams were being simulated each run, a way to automate this simulation process was required to maximize the amount of data that could be collected.

The automation of the simulations required the energy level in the run macro to be changed, then the simulation run at each energy level, the output file processed through NTuple, and all output files labeled by energy to distinguish them. An executable file was created called `efficiencyCurves.cpp` to complete this process. The file begins by defining the 5 variables that are necessary to meet all of the simulation needs, which can be seen in Figure 3.11.

```

string runMacroFileName = "/home/hgrover/geant/FilesNeeded_Automatic/run.mac"; // Full or relative path to run macro
string unit = " keV"; // Beam energy unit. HARDCODED
string settings = "/home/hgrover/geant/FilesNeeded_Automatic/Settings.dat"; // Full file path for ntuple settings file. Changeable on command line.
string sim = "/home/hgrover/geant/detectorSimulations_v10-0.5/Griffinv10"; //Executable simulation program file path. Changeable on command line.
string energyFile = "empty"; // Give whitespace separated list of energies. REQUIRED ON COMMAND LINE
//string macro = "empty"; // Give full file path for the root macro, if used. REQUIRED ON COMMAND LINE
mutex readfile; // For threadsafe file reading

```

Figure 3.11: These variables are necessary to run the automatic simulation file.

The first variable is the path to the run macro that has been previously described. The second variable is the unit of the energy that will be used in the run macro. The settings file is used for NTuple, and is given a full path as well. The sim variable is the path to the executable Griffin simulation environment. Finally, the energy file tells the program what energy to input in the run macro, and can be changed to fit the needs of the automated simulation. The tasks of changing the energy level and then running the simulation were met in the function shown in Figure 3.12.

```

void runSimulation(string sim, double energy, string macro){
    // Rewrite Griffin macro file with different energies
    readfile.lock();
    int status = replaceLineInFile("/DetSys/gun/efficiencyEnergy 1000 keV","/DetSys/gun/efficiencyEnergy "+to_string(energy)+unit,macro,"run"+to_string(energy)+".mac");
    readfile.unlock();
    if(status) {
        cout << "Error with replacing in file for energy "+to_string(energy)+unit+"\n";
        returnVal = status;
        return;
    }

    // Run simulation
    status = bash(sim+" run"+to_string(energy)+".mac "+to_string(energy)+" &> /dev/null",11); // Run simulations in bash.
    if(status) {
        cout << "Error running simulation energy "+to_string(energy)+unit+"\n";
        returnVal = status;
        return;
    }
    numThreads--;
    return;
}

```

Figure 3.12: This function changes the energy of the beam in the run.mac file and then executes it in the Griffin simulation environment.

This function starts by reading the run.mac file and replacing the line that defines the beam energy with a line that is identical, but with a different energy level. Then, the run.mac file is executed as seen before, but with an additional input that serves to name the output file. For example, if the energy that was being run was 500 keV, the input line would read

/path/to/Griffinv10 run500.mac 500

which would result in an output file of "g4out500.0000.root". This prevents the output file from being overwritten every time as new energy levels are run. Once this is done, the program fills the directory it was run in with raw output files to be processed through NTuple. This is accomplished in the function seen in Figure 3.13.

```
void runNTuple(string sim, double energy, string settings, string macro){
    //Run root file through ntuple
    string ntuple = "nt -vl 1 -if g4out"+to_string(energy)+".root -of converted"+to_string(energy)+".root -sf " + settings;
    int status = bash(ntuple, 11);
    if(status) {
        cout << "Error running ntuple for energy "+to_string(energy)+unit+"\n";
        returnVal = status;
        return;
    }
    numThreads--;
    return;
}
```

Figure 3.13: This function is responsible for running all raw output files through ntuple to create files for analysis.

Here, the input line starts with the abbreviation "nt", which is a shortcut for the full file path to NTuple. It also specifies the name of the input file by energy, and gives it an output file name also tagged by the energy value. In the main function of the file, the method for inputting the file paths for required or changeable inputs is established, and then the two functions for running the simulation and NTuple are called. This must be run in an empty directory, and once the executable file has finished running, the directory will be filled with both g4out(energy).root and converted(energy).root files that can be analyzed.

CHAPTER 4

RESULTS AND ANALYSIS

In order to draw conclusions about the viability of the combination of the GeDSSD with a clover style HPGe array and the optimum radius for the GeDSSDs, the output data from the simulations was processed and analyzed. Due to the large volume of data, the analysis process was also automated. A curve showing the contribution to the total detector configuration efficiency was generated for each individual detector, at every radius value, to compare. The individual contribution to efficiency curves, as well as the total efficiency curve, show that the overall gamma ray efficiency of the detector configuration is greater than the individual detectors would be if not in this particular configuration.

4.1 Automatic Analysis of Raw Output Data

The information in the raw output data is presented as an aggregate of all interactions broken up into several categories. The converted data files that result from running the raw output through NTuple show the number of interactions that occur at each energy value for the individual detectors.

To analyze the interaction results for the GRIFFIN array, GRSISort, an analysis program developed by the same research team at the University of Guelph, was used to fit the photopeaks of interest. This was chosen because of the function embedded in the program called TPeak, which takes a given range of values and creates the best fit for the given graph. The basic structure can be seen below,

$$\text{TPeak* peak} = \text{new TPeak}(\text{PeakEnergy}, \text{LowEnergy}, \text{HighEnergy}),$$

where the PeakEnergy is the data files' associated monoenergetic beam, and the Low- and HighEnergy inputs define the range of the peak. For the purposes of this analysis, the low energy was 5 keV (or bins) below the centroid and the high energy was 5 above. The fit that resulted

from this process resembles the one seen in Figure 4.1, with a zoomed in look at the fit seen in Figure 4.2.

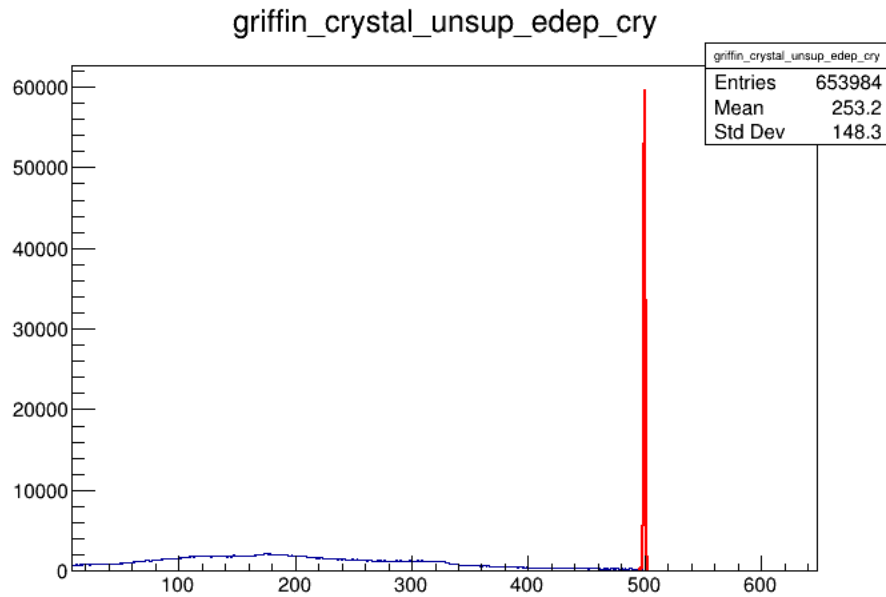


Figure 4.1: The fit of the photopeak centered at 500 keV for a GeDSSD radius of 4.5 cm.

Due to the number of files that needed to be analyzed, a C++ file was written to automate the analysis process. Every output file resulted from a monoenergetic photon source beam, so the analysis program needed the converted input file and associated beam energy. To find the individual contribution to the total configuration efficiency, the area of the photopeak was needed. After fitting the photopeak, the area under the curve was extracted using the following line:

```
peak->GetArea()
```

The analysis program only analyzed one file at a time, so a bash script was created to run through all of the converted output files in the current directory. Each radius value for the GeDSSD had its own directory with raw and converted output files, so the bash script was run in each of those 9 directories. The bash script called all of the converted files, then parsed the title to extract the energy value associated with the file. Then, the file and energy were used as the required inputs for the analysis script. The resultant area value was then output with the associated energy and

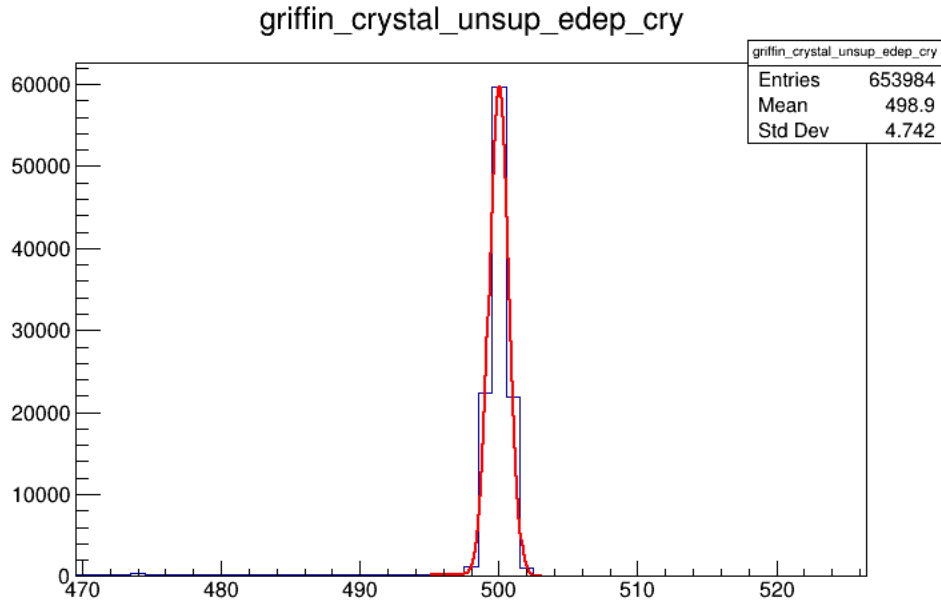


Figure 4.2: An example of the fitted peak for the 500 keV simulation data created by using the TPeak function in GRSISort.

appended to a text file. After all of the data files were run through the analysis program, the text file contained all of the count values needed for creating efficiency curves.

This process did not apply to the GeDSSDs because the interaction counts at the photopeak were all concentrated in one energy bin. The TPeak function was not necessary, so ROOT was used to run the analysis script instead of GRSISort. The C++ source file code was similar to the one used for the GRIFFIN data, but simply output the height of the bin corresponding to the energy of the monoenergetic source beam for the file instead of the area under a fitted curve. The bash script for analyzing these two detectors was the same structure, but called the new analysis script. After the bash scripts were run in each of the directories corresponding to the different GeDSSD radius values, each directory had 3 text files with count data, resulting in 21 total files with data to begin creating curves depicting the individual contributions to the total efficiency of the configuration.

When analyzing the heights of the bins output for the thin GeDSSD, the low value energies (≤ 150 keV) were showing counts of over 1 million. This was of course not possible, since that's

how many gamma rays were emitted from the simulated source. This occurred because the counts were high enough that NTuple was not a suitable processing program. Instead, the raw output data files were used to get more accurate count data.

In the raw output file, interaction information is separated by several categories, one of them being systemID, or individual detector. However, the interactions are not separated by energy, so all interactions that occur in the detector are presented in one bin, which can be seen in Figure 4.3. The heights of these bins were still used because at these energy values, the photoelectric effect heavily dominates the interaction types, so there are few interactions at energies lower than the photopeak. This was done for all data points below the at 150 keV for both the thin and thick GeDSSDs.

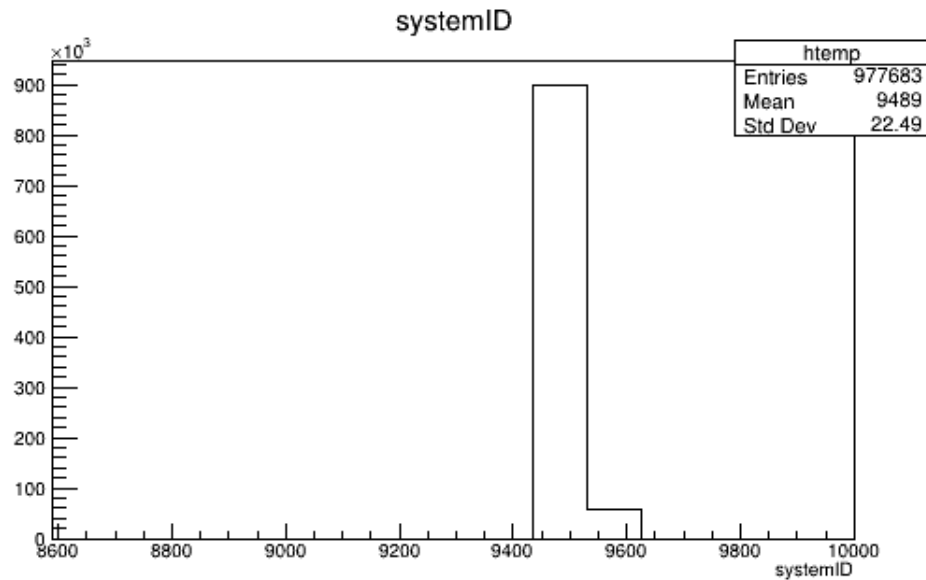


Figure 4.3: The systemIDs correspond to a detector, showing the number of interactions recorded in each during the simulation.

4.2 Generating Efficiency Curves

Each of the text files generated by the automated analysis scripts were used to make curves showing the detectors' individual contributions to efficiency to validate the process for all three detectors. In addition, the total efficiency of the configuration was produced to ensure that the

efficiency of the configuration was physically possible. To ensure that the results were realistic, the individual curves were compared to previously established curves. An efficiency curve for a GeDSSD can be seen in Figure 4.4, and it resembles the curve generated for the thin GeDSSD from simulations, seen in Figure 4.5. The thicker GeDSSD does not have the same curve because it is not where the beam is originally implanted. The comparison of these curves confirms that the simulation worked as expected.

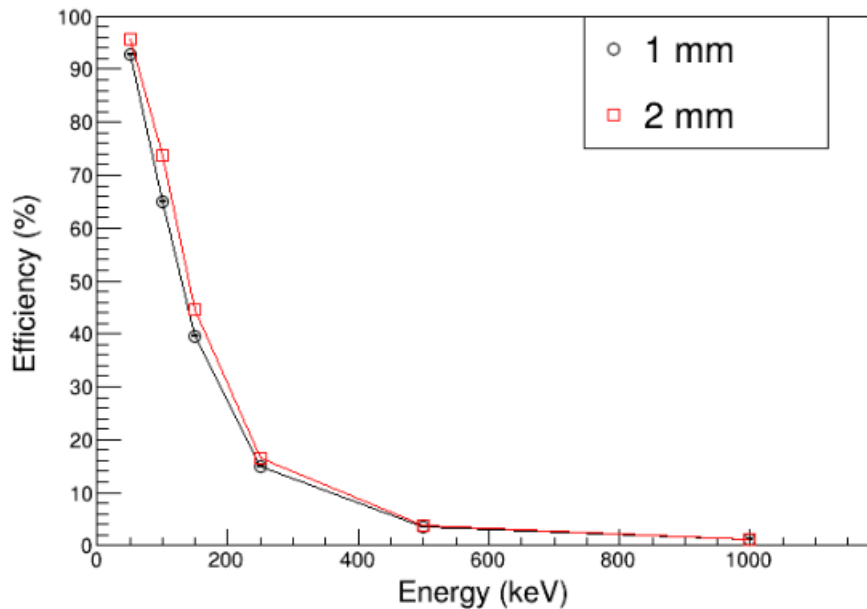


Figure 4.4: This is the efficiency curve for a GeDSSD presented at an FRIB meeting update. The 1 mm and 2 mm curves refer to the implantation depths of the particles. [28]

Once the validity of the simulation was confirmed, the total efficiency curves, as well as the individual contributions from each detector, were created for every radius of the GeDSSDs. The curves for the 0.5 cm radius can be seen in Figure 4.6. Since the simulation has the beam generating in the center of the thin GeDSSD, the detector shows that it detects 100% of fired particles at 15 and 20 keV. It then has a steep drop off in efficiency from 30-200 keV, until it has less than 1% efficiency from about 650 keV and greater. The thick GeDSSD does not show this same pattern because most of the low energy gamma rays are absorbed in the thin GeDSSD before they even make it to the thick detector. This means the beginning of the curve more closely resembles the

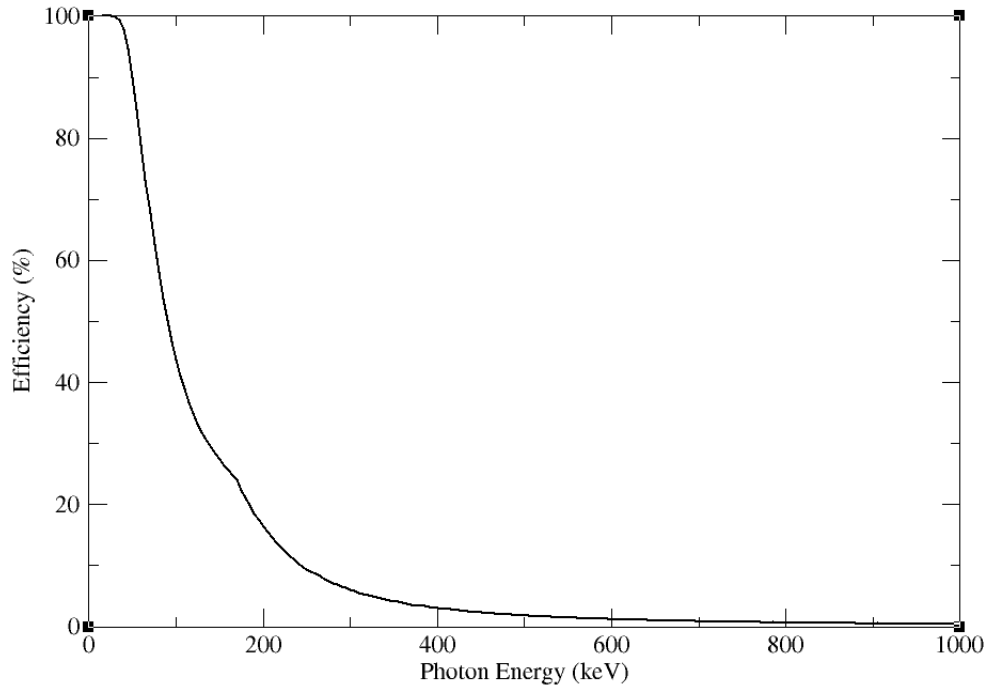


Figure 4.5: This is the efficiency curve created from the simulations for the thin GeDSSD with a radius of 4 cm. Data points were taken for every 5 keV increment from 15-1000 keV.

GRIFFIN array, with a peak in the low energy range. However, it eventually reaches an energy value where the gamma rays begin to move through the detector more often than being absorbed, leading to the decrease in efficiency that closely resembles the curve of the thin GeDSSD, but with more counts due to the larger volume. Due to its crystal and total size, the GRIFFIN array still has a higher contribution to total efficiency at high energies.

The next radius, 1.0 cm, shows a marked increase in efficiency for the thick GeDSSD at low energies, thereby increasing the total efficiency of the array in the same energy range, seen in Figure 4.7. The next graph, Figure 4.8, shows yet another increase in total energy at low energies. For the 0.5 cm radius, the total configuration efficiency at 200 keV was about 46%, but by increasing the radius to 1.5 cm, the efficiency at the same energy jumped to about 58%. At the 1.5 cm radius, the efficiency for the total configuration is above 90% for energies ≤ 100 keV.

The efficiency curves seen in Figure 4.9 and Figure 4.10 show that the negative slope in efficiency in the low energy range is less steep as radius increases. This makes sense as the larger

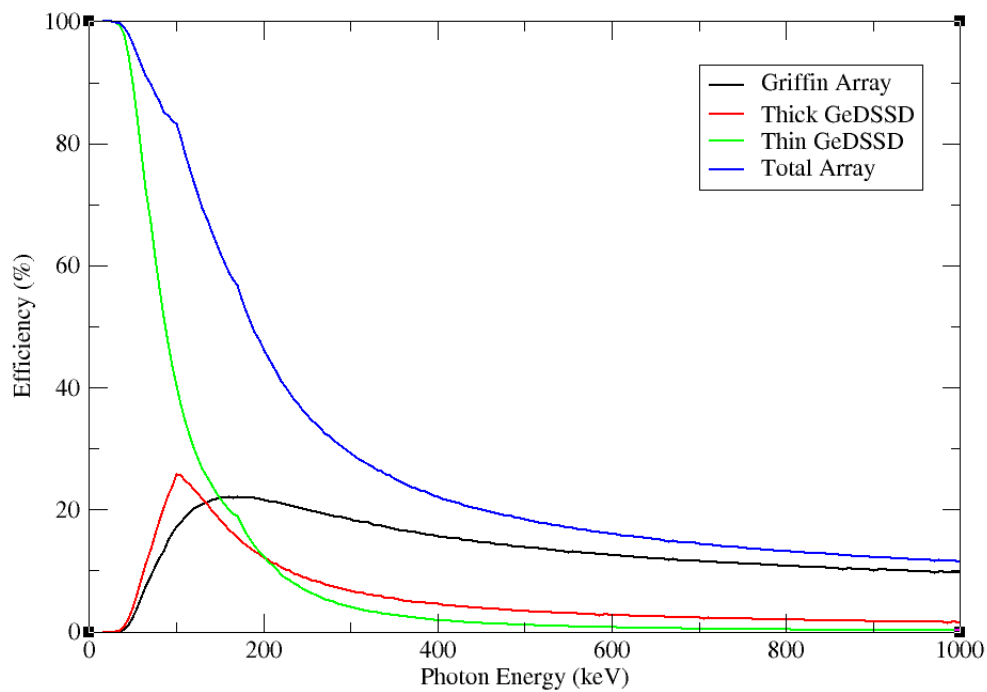


Figure 4.6: The individual and total efficiency curves for the GeDSSD radius of 0.5 cm.

radius allows more volume for detecting those lower energy photons. In addition, it is important to note that the efficiency of GRIFFIN, or any surrounding clover array, will have lower efficiency than usual at low energies since many of those gamma rays are being absorbed in the GeDSSDs. In addition, the efficiency for the thick GeDSSD and GRIFFIN array crosses over at higher photon energy levels as the radius increases.

The efficiency continues to increase as the size increases, but in smaller increments. The total array efficiency at 200 keV for the radius of 3.0 cm is at 69%, with over 90% efficiency until 130 keV as seen in Figure 4.11. The GeDSSDs still have very low efficiency at high energies due to their smaller overall size, so most of the efficiency is attributed to the GRIFFIN array at high energies. The 3.5 cm radius efficiencies seen in Figure 4.12 are very similar, increasing as expected. However, this increase in radius is where the smallest change in efficiency is seen, indicating this might be the optimum value for the radius. As the radius increase for the next two test cases, seen in Figure 4.13 and Figure 4.14, there is an increase in efficiency but only by at most

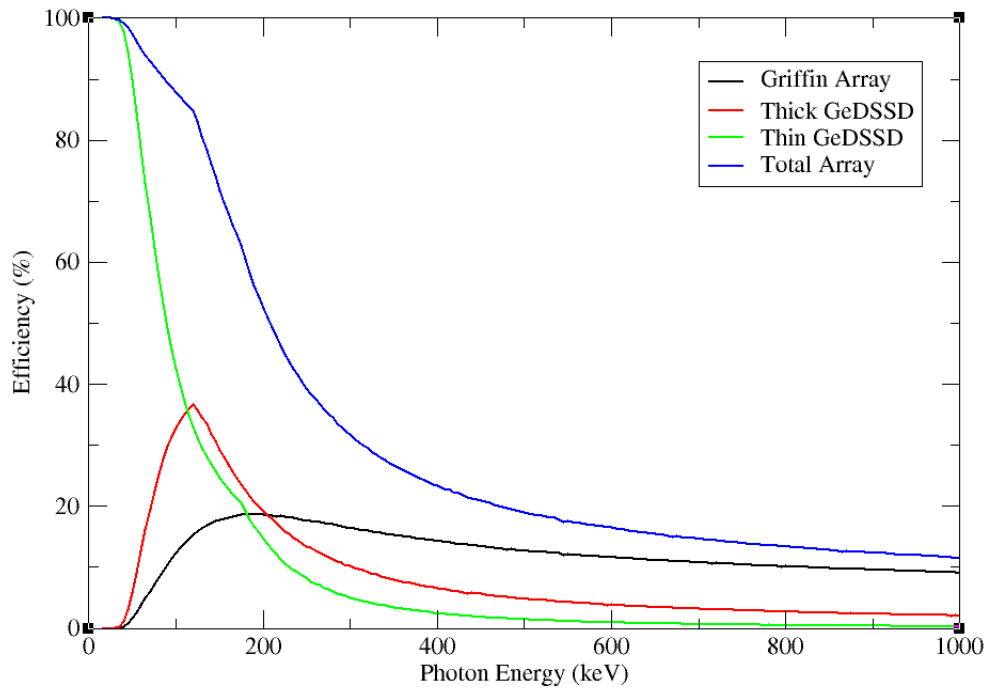


Figure 4.7: The individual and total efficiency curves for the GeDSSD radius of 1.0 cm.

2% percentage points at any given energy. At 100 keV, there is no significant increase, and only a 4% increase at 200 keV. This results in about a 1% increase for each 0.5 cm increase in radius for both the thin and thick GeDSSDs. It is also important to keep in mind the range from 200-300 keV is where the largest change in efficiency is seen, so all other energies see a difference in efficiency of less than 2%. The final two radii seen in Figure 4.13 and Figure 4.14 show this

The results from the simulations support that the combination of the thin and thick GeDSSDs with a clover style array, like GRIFFIN, would increase total gamma ray detection efficiency. The work suggests that the radius that provides the highest value, or most efficiency for the cost, is 3.5 cm. This seems to be the point of diminishing returns for increasing the radius. At this radius, the efficiency is over 90% for 135 keV and below. The lowest efficiency at 1 MeV is still at 13.75%, meaning that experiments with both high and low energy photon emissions can be run with this detector configuration. There are several other parameters that could be varied with this code in the future to optimize the size and position, such as the spacing between the GeDSSDs and

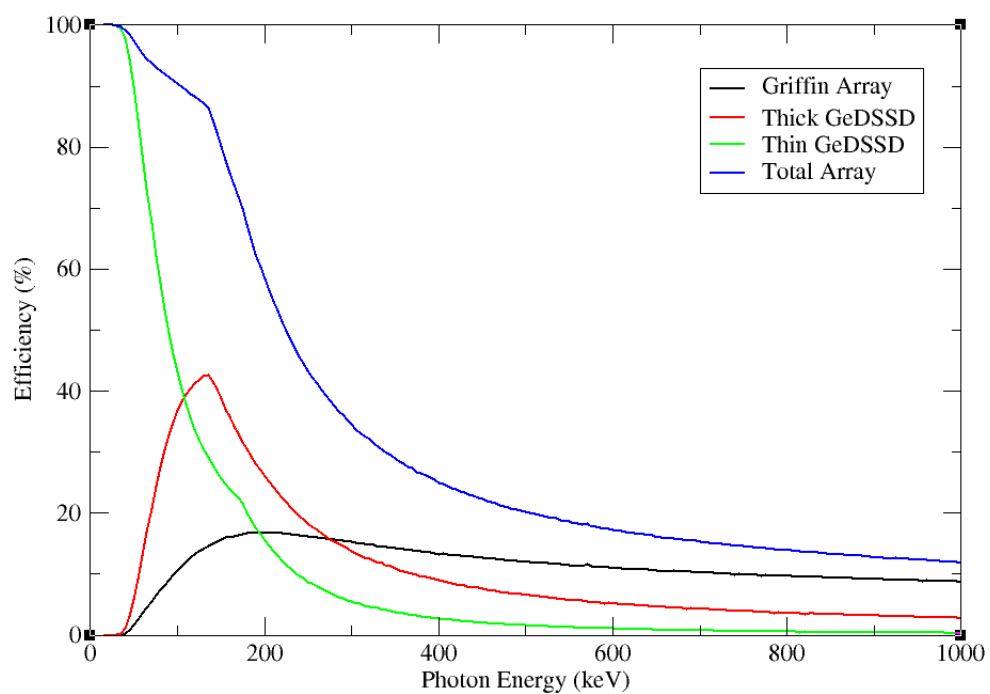


Figure 4.8: The individual and total efficiency curves for the GeDSSD radius of 1.5 cm.

thicknesses of them both. Overall, the high efficiency at low photon energy with the beta decay detection ability in the GeDSSDs indicates that this novel detector configuration would be highly beneficial to the FRIB decay station's goal of studying the most exotic nuclei.

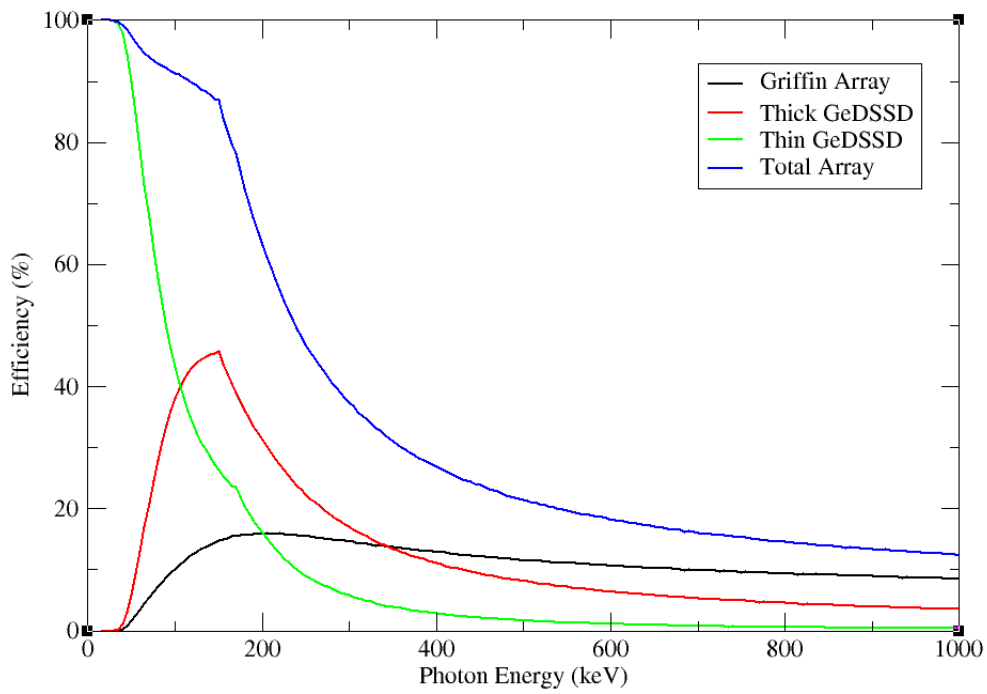


Figure 4.9: The individual and total efficiency curves for the GeDSSD radius of 2.0 cm.

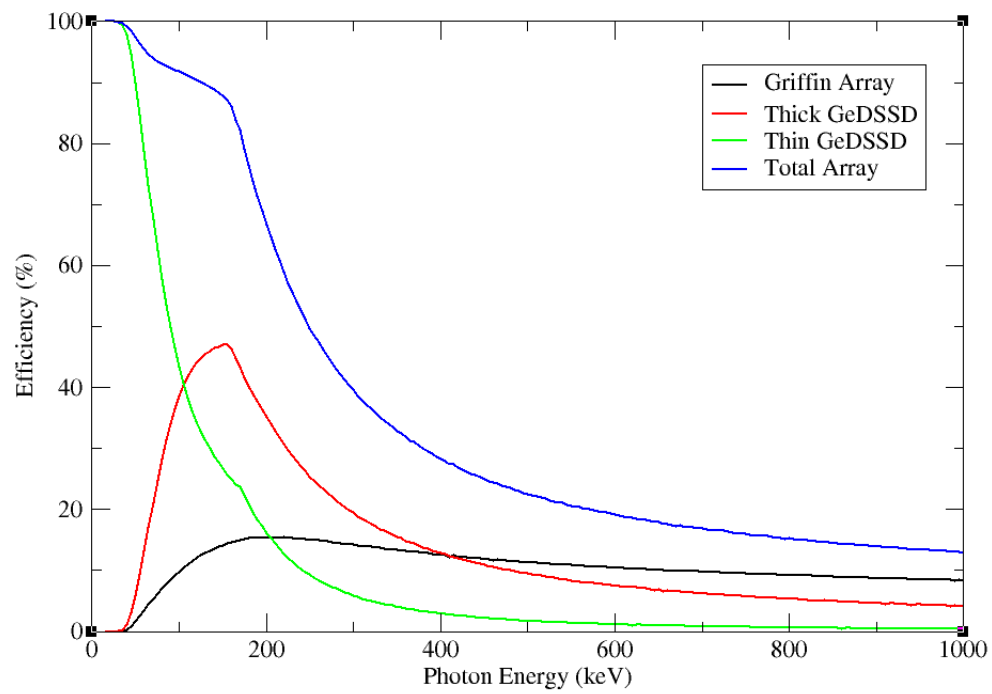


Figure 4.10: The individual and total efficiency curves for the GeDSSD radius of 2.5 cm.

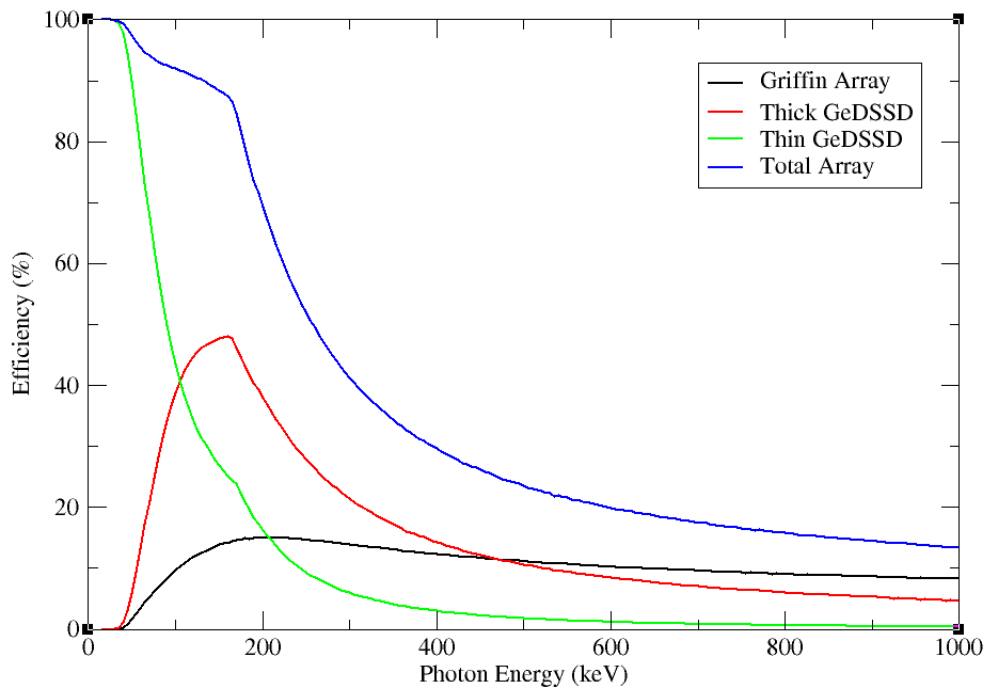


Figure 4.11: The individual and total efficiency curves for the GeDSSD radius of 3.0 cm.

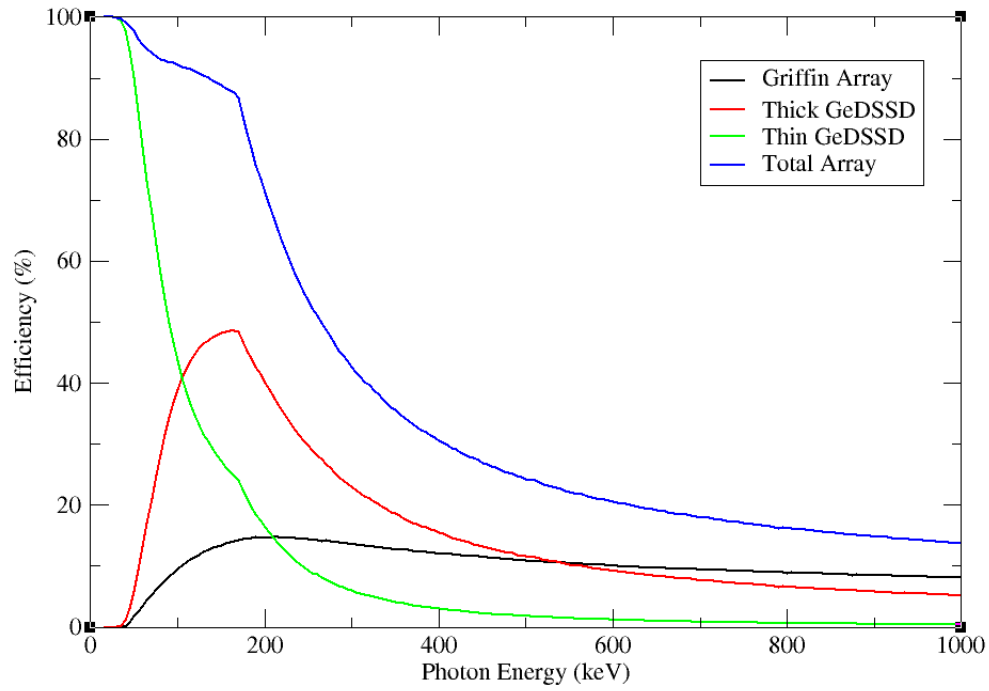


Figure 4.12: The individual and total efficiency curves for the GeDSSD radius of 3.5 cm.

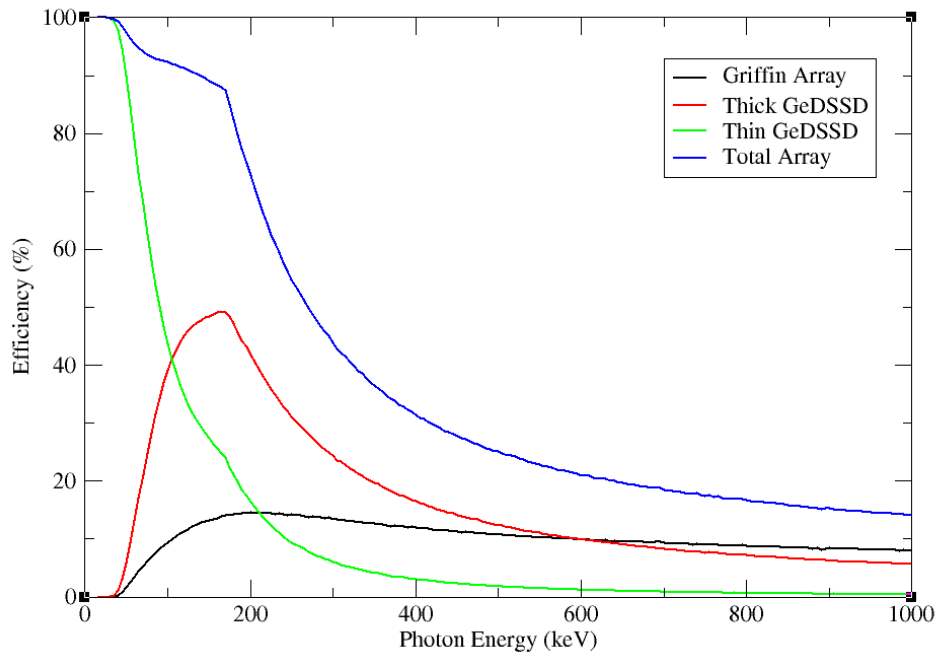


Figure 4.13: The individual and total efficiency curves for the GeDSSD radius of 4.0 cm.

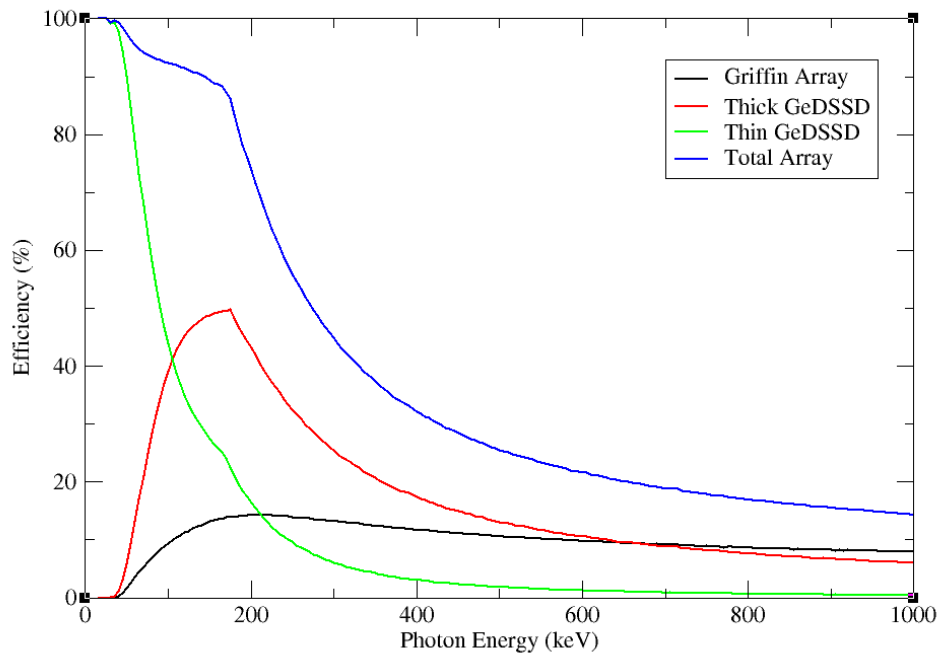


Figure 4.14: The individual and total efficiency curves for the GeDSSD radius of 4.5 cm.

CHAPTER 5

SUMMARY AND CONCLUSIONS

The study of nuclear structure is critical to understanding the fundamental forces of nature and the laws that govern them. There are facilities around the world that are working toward this goal by using high end, versatile equipment to produce and detect as many different nuclei as possible. There is still much to be learned about the exotic nuclei that are not typically found on Earth, and we are approaching a time in science where we have the technology to explore the reaches of the nuclide chart. One of the main ways to begin making these discoveries is through the building and use of the Facility for Rare Isotope Beams.

FRIB will be a state-of-the-art facility with production of nuclei that have never been studied before. The FRIB decay station will utilize these advancements and create a suite of detectors that will be able to perform a wide range of experiments efficiently and with good resolution. One of the ways this will be done is through combining detectors that have different strengths to maximize discovery potential. The project this thesis has focused on is the integration of germanium double sided strip detectors and a gamma ray detector array such as GRIFFIN.

Geant4 was used to simulate a real-world response to a monoenergetic gamma ray beam, so efficiency curves could be generated. The GeDSSDs were integrated into a simulation package in Geant4 using dimensions from previous detectors. The radii were varied from 0.5 to 4.5 cm to optimize the size of the detectors in this type of configuration. The data from the simulations then generated curves showing the individual contributions and total efficiencies of the new detector configuration. Figure 5.1 shows each of the total efficiency curves generated for every GeDSSD radius value. The results shown by the curves show that while the total efficiency does continue to increase all the way up to a 4.5 cm radius, there is a point where an increase in radius results in a smaller increase in efficiency. After 3.5 cm, the efficiency increases by no more than 3%, which happens in the 200 keV energy range. For this reason, the optimum GeDSSD radius for

both efficiency and cost is 3.5 cm.

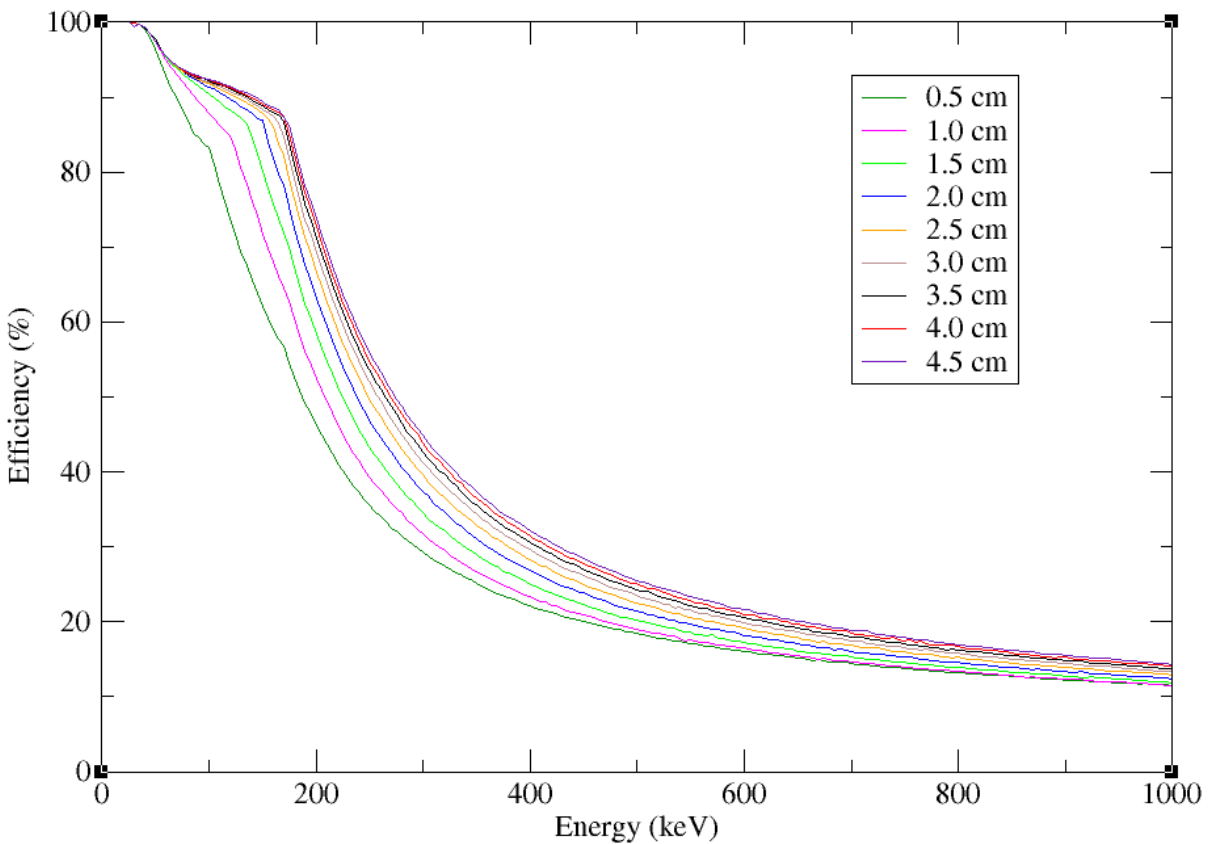


Figure 5.1: The total efficiencies for each GeDSSD radius value is shown here, with the smallest radius farthest to the left.

The results of this project confirmed that the combination of a thin and thick germanium double sided strip detector with a clover style gamma ray detector array improves efficiency while also allowing for the simultaneous detection of beta particles and photons. Now that the GeDSSDs have been incorporated into the simulation package in Geant4, future simulations can be run, varying different parameters. One potential study is on the effect of the spacing between the two GeDSSDs on the photon and charged particle efficiency of the total detector configuration. In addition, the same analysis programs could even be used to obtain efficiency curves. There could also be studies done with more realistic source particles to simulate potential results from implantation experiments that are going to be run at FRIB. To make the simulations even more accurate, the

exact clover configuration that will be combined with the GeDSSDs could be incorporated into the simulation package. At that point, positioning and relative spacing between the GeDSSDs and the clover array could be varied as well. This work was the first step in confirming that the combination of GeDSSDs and a clover style gamma detector array is beneficial, and future work can continue to optimize the parameters to make it as efficient and versatile as possible to maximize discovery potential at FRIB.

REFERENCES CITED

- [1] Alex C. Mueller. An overview of radioactive beam facilities. In *European Particle Accelerator Conference*, pages 73–77, Vienna, Austria, June 2000. Institute of High Energy Physics of the Austrian Academy of Sciences,.
- [2] Alejandro Sonzogni. NNDC chart of nuclides. In O. Bersillon, F. Gunsing, E. Bauge, R. Jacqnrnin, and S. Leray, editors, *International Conference on Nuclear Data for Science and Technology 2007*, pages 105–06, Nice, France, 2007. EDP Sciences. doi: 10.1051/ndata:07530.
- [3] Connor Natzke. Private communication.
- [4] Gregoary Chopping, Jan Rydberg, and Jan-Olov Liljenzin. *Radiochemistry and Nuclear Chemistry*. Elsevier, 2002.
- [5] Sharon Bewick, Richard Parsons, Therese Forsythe, Shonna Robinson, and Jean Dupon. Types of radioactivity: Alpha, beta, and gamma decay, August 2017. URL [https://chem.libretexts.org/Textbook_Maps/Introductory_Chemistry_Textbook_Maps/Map%3A_Introductory_Chemistry_\(Tro\)/17%3A_Radioactivity_and_Nuclear_Chemistry/17.03%3A_Types_of_Radioactivity%3A_Alpha%2C_Beta%2C_and_Gamma_Decay](https://chem.libretexts.org/Textbook_Maps/Introductory_Chemistry_Textbook_Maps/Map%3A_Introductory_Chemistry_(Tro)/17%3A_Radioactivity_and_Nuclear_Chemistry/17.03%3A_Types_of_Radioactivity%3A_Alpha%2C_Beta%2C_and_Gamma_Decay).
- [6] J. Eberth and J. Simpson. From ge(li) detectors to gamma-ray tracking arrays50 years of gamma spectroscopy with germanium detectors. *Progress in Particle and Nuclear Physics*, 60(2):283–337, 2008. doi: <https://doi.org/10.1016/j.pnpnp.2007.09.001>. URL <http://www.sciencedirect.com/science/article/pii/S0146641007000828>.
- [7] Oregon State University. Chapter 9 - gamma ray decay. . URL <http://oregonstate.edu/instruct/ch374/ch418518/CHAPTER%209%20GAMMA%20RAY%20DECAY-rev.pdf>.
- [8] Alison Bruce. Gamma-ray spectroscopy [presentation].
- [9] Bradley M. Sherrill. Exotic beam summer school: Accelerators and beams. July 2011. URL <https://people.nscl.msu.edu/~zegers/ebss2011/sherrill.pdf>.
- [10] Brad M. Morrissey, David J. and Sherrill. *In-Flight Separation of Projectile Fragments*, pages 113–135. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-44490-9. doi: 10.1007/978-3-540-44490-9_4. URL https://doi.org/10.1007/978-3-540-44490-9_4.

- [11] Toni Feder. Rare isotopes will be used to probe mysteries, yield applications. *Physics Today*, 68(2):23, 2015. URL <https://doi.org/10.1063/PT.3.2682>.
- [12] Jyunlin Wei, D Arenius, E Bernard, Nathan Bultman, F Casagrande, S Chouhan, Cyle Compton, K Davidson, A Facco, V Ganni, P Gibson, T Glasmacher, Lee Harle, K Holland, M Johnson, S Jones, Daniela Leitner, M Leitner, Guillaume Machicoane, and Qs Zhao. The frib project - accelerator challenges and progress. In *International Conference on Heavy Ion Accelerator Technology*, pages 8–19, Chicago, IL, 2012.
- [13] Michigan State University. Facility for rare isotope beams: About frib. online, . URL <https://www.frib.msu.edu/about/index.html>.
- [14] Brad Sherrill. Frib experimental overview. PowerPoint, October 2014.
- [15] JA Lettry. Exotic ion-beams, targets and sources. *arXiv preprint physics/0009036*, 2000.
- [16] Robert Grzywacz. Developing the tools for discovery. Dec 2016.
- [17] Robert Grzywacz. Private communication.
- [18] Glenn F. Knoll. *Radiation Detection and Measurement*. John Wiley and Sons, Inc., 4 edition, 2010.
- [19] P. A. Russo and D. T. Vo. Gamma-ray detectors for non destructive analysis. Online. URL <http://www.lanl.gov/orgs/n/n1/panda/1.%20Gamma-ray%20Detectors%20LAUR.pdf>.
- [20] R. N. Hall and T. J. Soltys. High purity germanium for detector fabrication. *IEEE Transactions on Nuclear Science*, 18(1):160–165, Feb 1971. ISSN 0018-9499. doi: 10.1109/TNS.1971.4325857.
- [21] Mayeen Khandaker. High purity germanium detector in gamma-ray spectrometry. *International Journal of Fundamental Physical Sciences*, 1:42–46, 06 2011.
- [22] Ian Rittersdorf. Gamma ray spectroscopy. March 2007.
- [23] The University of Guelph. Griffin. Online. URL <https://www.physics.uoguelph.ca/Nucweb/griffin.html>.

- [24] Shinichiro Takeda, Shin Watanabe, Takaaki Tanaka, Kazuhiro Nakazawa, Tadayuki Takahashi, Yasushi Fukazawa, Hajimu Yasuda, Hiroyasu Tajima, Yoshikatsu Kuroda, Mitsunobu Onishi, and Kei Genba. Development of double-sided silicon strip detectors (dssd) for a compton telescope. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 579(2):859 – 865, 2007. doi: <https://doi.org/10.1016/j.nima.2007.05.305>. URL <http://www.sciencedirect.com/science/article/pii/S0168900207011965>. Proceedings of the 6th "Hiroshima" Symposium on the Development and Application of Semiconductor Detectors.
- [25] Knoxville The University of Tennessee. Decay spectroscopy. Online. URL <http://www.phys.utk.edu/expnuclear/decspec.html>.
- [26] Shin'Ichiro Takea, Tadayuki Takahashi, Shin Watanabe, Hiroyasu Tajima, Takaaki Tanaka, Kazuhiro Nakazawa, and Yasushi Fukazawa. Double-sided silicon strip detector for x-ray imaging. *SPIE*, February 2008. doi: 10.1117/2.1200802.0889.
- [27] J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce Dubois, M. Asai, G. Barrand, R. Capra, S. Chauvie, R. Chytrcek, G. A. P. Cirrone, G. Cooperman, G. Cosmo, G. Cuttone, G. G. Daquino, M. Donszelmann, M. Dressel, G. Folger, F. Foppiano, J. Generowicz, V. Grichine, S. Guatelli, P. Gumplinger, A. Heikkinen, I. Hrivnacova, A. Howard, S. Incerti, V. Ivanchenko, T. Johnson, F. Jones, T. Koi, R. Kokoulin, M. Kossov, H. Kurashige, V. Lara, S. Larsson, F. Lei, O. Link, F. Longo, M. Maire, A. Mantero, B. Mascialino, I. McLaren, P. Mendez Lorenzo, K. Minamimoto, K. Murakami, P. Nieminen, L. Pandola, S. Parlati, L. Peralta, J. Perl, A. Pfeiffer, M. G. Pia, A. Ribon, P. Rodrigues, G. Russo, S. Sadilov, G. Santin, T. Sasaki, D. Smith, N. Starkov, S. Tanaka, E. Tcherniaev, B. Tome, A. Trindade, P. Truscott, L. Urban, M. Verderi, A. Walkden, J. P. Wellisch, D. C. Williams, D. Wright, and H. Yoshida. Geant4 developments and applications. *IEEE Transactions on Nuclear Science*, 53(1):270–278, Feb 2006. ISSN 0018-9499. doi: 10.1109/TNS.2006.869826.
- [28] Nicole Larson. Development of a planar germanium double sided strip detector for beta-decay spectroscopy. PowerPoint, April 2014.

APPENDIX

GEOMETRY FILE FOR THE GEDSSDS

The geometry file mentioned in the document is shown here in three separate figures. Figure A.1 shows the beginning of the document with variable definitions and initializations. Figure A.2 shows the part of the code responsible for material and visual specifications. Finally, Figure A.3 shows the end of the code where the physical volumes are oriented and made into logical volumes so that they could interact with particles during simulations.

```

DetectionSystemGedssd::DetectionSystemGedssd() :
    //Logical Volumes
    fThickDetectorLog(0),
    fThinDetectorLog(0),
    fThickDetectorCryoLog(0),
    fCryoEndCapLog(0)
{
    //Dimensions for Gedssd

    fConvert = 0.0254*m; //Needed to convert inches to meters
    fGeThickDetectorThickness = 1.0*cm; // 2.0cm//Hiro's detector
    fGeThinDetectorThickness = 1.5*mm;
    fGeThickDetectorRadius = 0.5*cm; // 3.0 * cm; //3.0*cm;Hiro's detector
    fGeThinDetectorRadius = 0.5*cm;
    fGeThickDetectorCryoInnRad = 6.6*cm;
    fGeThickDetectorCryoOutRad = 9.6393*cm;//From Ethan
    fGeThickDetectorCryoThickness = 8.89*cm;//measured with tape measurer
    fGeCryoEndCapOutRad = fGeThickDetectorCryoInnRad;
    fGeCryoEndCapThickness = 1.143*mm; //use 0.1 cm for radioxenon
    fSpacing = 5*mm;
    pi = 3.14159265358979323846;

}

DetectionSystemGedssd::~~DetectionSystemGedssd() {
    //Deleting logical volumes
    delete fThickDetectorLog;
    delete fThinDetectorLog;
    delete fThickDetectorCryoLog;
    delete fCryoEndCapLog;
}

G4int DetectionSystemGedssd::Build() {
    fAssemblyGedssd = new G4AssemblyVolume();

    BuildGedssd();

    return 1;
}

G4int DetectionSystemGedssd::PlaceDetector(G4LogicalVolume* expHallLog, G4int detectorNumber) {

    G4RotationMatrix* rotateNull = new G4RotationMatrix;
    G4ThreeVector moveNull(0., 0., 0.);

    for(G4int j=1; j<2; j++) {
        move = moveNull;
        fAssemblyGedssd->MakeImprint(expHallLog, move, rotateNull);
    }

    return 1;
}

```

Figure A.1: This part of the code shows the initialization of the files, along with the previously established dimensions.

```

G4int DetectionSystemGedssd::BuildGedssd() {
    G4ThreeVector move, direction;
    G4RotationMatrix* rotate;

    //Materials for the Gedssd
    G4Material* fThinMaterial = G4Material::GetMaterial("Germanium");
    if( !fThinMaterial ) {
        G4cout << "----> Material " << "Germanium" << " not found, cannot build! " << G4endl;
        return 0;
    }

    G4Material* fThickMaterial = G4Material::GetMaterial("Germanium");
    if( !fThickMaterial ) {
        G4cout << "----> Material " << "Germanium" << " not found, cannot build! " <<G4endl;
        return 0;
    }

    G4Material* fCryoMaterial = G4Material::GetMaterial("Aluminum");
    if( !fCryoMaterial ) {
        G4cout << "----> Material Aluminum not found, cannot build! " << G4endl;
        return 0;
    }

    G4Material* fEndCapMaterial = G4Material::GetMaterial("Aluminum");
    if( !fEndCapMaterial ) {
        G4cout << "----> Material Aluminum not found, cannot build! " <<G4endl;
        return 0;
    }

    //Building the Thin Detector
    G4Tubs* ThinDetector = new G4Tubs("GeThinDetector",0.,fGeThinDetectorRadius, fGeThinDetectorThickness/2.,0.,2*pi)
;

    //Building the Thick Detector
    G4Tubs* ThickDetector = new G4Tubs("GeThickDetector",0.,fGeThickDetectorRadius, fGeThickDetectorThickness/2.,0.,2
*pi);

    //Building the Thick Detector Cryo Chamber
    G4Tubs* CryoChamber = new G4Tubs("GeThickDetectorCryo",fGeThickDetectorCryoInnRad, fGeThickDetectorCryoOutRad, fG
eThickDetectorCryoThickness/2.,0.,2*pi);

    //Building the Cryo End Caps
    G4Tubs* EndCaps = new G4Tubs("GeCryoEndCaps",0.,fGeCryoEndCapOutRad, fGeCryoEndCapThickness/2.,0.,2*pi);

    //The visual attributes for all components
    G4VisAttributes* ThinDetectorVisAtt = new G4VisAttributes(G4Colour::Red());
    ThinDetectorVisAtt->SetVisibility(true);
    G4VisAttributes* ThickDetectorVisAtt = new G4VisAttributes(G4Colour(1.0,1.0,0.));
    ThickDetectorVisAtt->SetVisibility(true);
    G4VisAttributes* ThickDetectorCryoVisAtt = new G4VisAttributes(G4Colour::Blue());
    ThickDetectorCryoVisAtt->SetVisibility(true);
    G4VisAttributes* CryoEndCapVisAtt = new G4VisAttributes(G4Colour::Green());
    CryoEndCapVisAtt->SetVisibility(true);
}

```

Figure A.2: The middle part of the document shows the material specification as well as the visual attributes for the simulations being set. In addition, it shows the physical volumes being made, which are all G4Tubs, or simple cylinders.

```

//Define the rotation and placement for the thin detector
thinDirection =G4ThreeVector(0.,0.,1.);
//thinPosition =G4ThreeVector(0.,0.,-(fGeThickDetectorThickness/2.)-(fGeThinDetectorThickness/2.))-fSpacing);
thinPosition =G4ThreeVector(0.,0.,0.);
thinRotation =new G4RotationMatrix;

//Define Logical Volume for Thin Detector
if(fThinDetectorLog == NULL) {
    fThinDetectorLog = new G4LogicalVolume(ThinDetector, fThinMaterial, "GeThinDetector", 0,0,0);
    fThinDetectorLog->SetVisAttributes(ThinDetectorVisAtt);
}
fAssemblyGedssd->AddPlacedVolume(fThinDetectorLog, thinPosition, thinRotation);

//Define rotation and placement for the thick detector
thickDirection =G4ThreeVector(0.,0.,1.);
//thickPosition =G4ThreeVector(0.,0.,0.);
thickPosition =G4ThreeVector(0.,0.,-(fGeThickDetectorThickness/2.)-(fGeThinDetectorThickness/2.))-fSpacing);
thickRotation =new G4RotationMatrix;

//Define logical volume for thick detector
if(fThickDetectorLog == NULL) {
    fThickDetectorLog = new G4LogicalVolume(ThickDetector, fThickMaterial, "GeThickDetector",0,0,0);
    fThickDetectorLog->SetVisAttributes(ThickDetectorVisAtt);
}
fAssemblyGedssd->AddPlacedVolume(fThickDetectorLog, thickPosition, thickRotation);

//Define rotation and placement for cryo chamber
cryoDirection =G4ThreeVector(0.,0.,1.);
cryoPosition =G4ThreeVector(0.,0.,0.);
cryoRotation =new G4RotationMatrix;

//Define logical volume for cryo chamber
if(fThickDetectorCryoLog == NULL) {
    fThickDetectorCryoLog = new G4LogicalVolume(CryoChamber, fCryoMaterial, "GeThickDetectorCryo", 0,0,0);
    fThickDetectorCryoLog->SetVisAttributes(ThickDetectorCryoVisAtt);
}
fAssemblyGedssd->AddPlacedVolume(fThickDetectorCryoLog, cryoPosition, cryoRotation);

//Define rotation and placement for end caps
capDirection =G4ThreeVector(0.,0.,1.);
cap1Position =G4ThreeVector(0.,0.,fGeThickDetectorCryoThickness/2. - fGeCryoEndCapThickness/2.);
cap2Position =G4ThreeVector(0.,0.,-fGeThickDetectorCryoThickness/2. + fGeCryoEndCapThickness/2.);
capRotation =new G4RotationMatrix;

//Define logical volume for end caps
if(fCryoEndCapLog == NULL) {
    fCryoEndCapLog = new G4LogicalVolume(EndCaps, fEndCapMaterial, "GeCryoEndCap", 0,0,0);
    fCryoEndCapLog->SetVisAttributes(CryoEndCapVisAtt);
}

fAssemblyGedssd->AddPlacedVolume(fCryoEndCapLog, cap1Position, capRotation);
fAssemblyGedssd->AddPlacedVolume(fCryoEndCapLog, cap2Position, capRotation);

```

Figure A.3: The final part of the file is where the orientation for each object is set, and then the physical volumes are made into logical volumes and placed into the logical group to be called during a simulation.