Trinity University

Digital Commons @ Trinity

Physics & Astronomy Honors Theses

Physics and Astronomy Department

5-2021

Investigating the Design, Construction, and Performance of a π^0 Spectrometer

Thomas M. Baer

Trinity University, thehungarybaer@gmail.com

Follow this and additional works at: https://digitalcommons.trinity.edu/physics_honors

(Investigating the Design, Construction, and Performance of a π^0 Spectrometer)

(Thomas Baer)

A departmental senior thesis submitted to the Department of Physics and Astronomy at Trinity University in partial fulfillment of the requirements for graduation with departmental honors.

(April 22nd, 2021)

_____   _____

Thesis Advisor   Department Chair

_____

Michael Soto, AVPAA

Student Agreement

I grant Trinity University ("Institution"), my academic department ("Department"), and the Texas Digital Library ("TDL") the non-exclusive rights to copy, display, perform, distribute and publish the content I submit to this repository (hereafter called "Work") and to make the Work available in any format in perpetuity as part of a TDL, digital preservation program, Institution or Department repository communication or distribution effort.

I understand that once the Work is submitted, a bibliographic citation to the Work can remain visible in perpetuity, even if the Work is updated or removed.

I understand that the Work's copyright owner(s) will continue to own copyright outside these non-exclusive granted rights.

I warrant that:

  1) I am the copyright owner of the Work, or
  2) I am one of the copyright owners and have permission from the other owners to submit the Work, or
  3) My Institution or Department is the copyright owner and I have permission to submit the Work, or
  4) Another party is the copyright owner and I have permission to submit the Work.

Based on this, I further warrant to my knowledge:

  1) The Work does not infringe any copyright, patent, or trade secrets of any third party,
  2) The Work does not contain any libelous matter, nor invade the privacy of any person or third party, and
  3) That no right in the Work has been sold, mortgaged, or otherwise disposed of, and is free from all claims.

I agree to hold TDL, DPN, Institution, Department, and their agents harmless for any liability arising from any breach of the above warranties or any claim of intellectual property infringement arising from the exercise of these non-exclusive granted rights."

I choose the following option for sharing my thesis (required):

[X] Open Access (full-text discoverable via search engines)
[ ] Restricted to campus viewing only (allow access only on the Trinity University campus via digitalcommons.trinity.edu)

I choose to append the following Creative Commons license (optional):

Investigating the Design, Construction, and Performance of a π^0 Spectrometer

by

Thomas M. Baer

An honors thesis submitted to the Department of Physics & Astronomy at

TRINITY UNIVERSITY

in partial fulfillment of the requirements for the
Bachelor of Science in Physics
May 2021

Accepted by ....
Prof. Kelvin Cheng

Accepted by ....
Prof. Nirav Mehta

Accepted by ....
Prof. David Pooley

Accepted by ....
Prof. Orrin Shindell

Accepted by ....
Prof. Jennifer Steele, Chair

Accepted by ....
Prof. Niescja Turner

Accepted by ....
Prof. Dennis Ugolini

Investigating the Design, Construction, and Performance of a π^0 Spectrometer

by

Thomas M. Baer

Submitted to the Department of Physics & Astronomy
on April 22, 2021, in partial fulfillment of the
requirements for the Bachelor of Science in Physics

Abstract

A π^0 spectrometer constructed at the Los Alamos Meson Physics Facility in 1977 by H. W. Baer is investigated using a historical and physical approach. A history is given of the spectrometer, including the circumstances which gave rise to its construction and its eventually decommissioning. This is supported by a theoretical description of the spectrometer operation, as well as current Monte Carlo simulations which aimed to reproduce the behavior of the original π^0 spectrometer.

Thesis Supervisor: Dr. Orrin Shindell

Acknowledgments

Foremost I would like to thank Dr. Orrin Shindell for all the support and guidance he has given me throughout this project, and throughout my time as an undergraduate as a whole. I would also like to thank the Department of Physics and Astronomy for guiding me through the joy that is learning physics, and for introducing me to my classmates with whom I have been able to share this journey. And I would especially like to thank my grandfather H. W. Baer, with whom I can share the joys of physics despite his absence.

Contents

Chapter 1

Motivation

Because this project is of a more historical nature, in both a general and personal sense, and lacks the possibility of discovering anything new physics-wise, I believe that it is important to provide a motivation for why I pursued a topic of this nature for a senior thesis. Although physics is inherently a progress oriented field, always searching for new hypotheses to test and new theories to explain natural phenomena, improving, revising, and building off of the previous work of countless individuals, there is real value in studying the history of the advancements we have made. The background and context, be that political, economical, or the prevailing worldview, in which scientific research is conducted, heavily influence the approaches, the resources, and the mental frameworks with which science is done. In history, because it is further away from our own time, we may more easily perceive how these contexts influence and inform scientific research. It is my hope then, that by studying the history of science and investigating the contexts in which science has been done in the past, we may gain new insight and perspective in to how our own scientific research is influenced by the context of our times.

This project is also significant as a personal history for me. My grandfather H. W. Baer was a nuclear physicist at Los Alamos National Laboratory from the 1970s until he died in 1991. It had always been a wish of mine to learn more about him and to understand the research that he conducted. The countless boxes he left behind in his shed, full of his correspondences, lecture notes, and published papers unintelligible to the uninitiated, gnawed on my curiosity since childhood. And he himself, having died before I was born, became a larger than life figure to me, on one hand a motivating example of an intelligent and hardworking man, and on the other a companion with whom I knew I shared the joy of learning physics. This project is a fulfillment of that wish.

Throughout my time working on this project I have poured over his letters with friends and colleagues, discussing new theoretical developments in nuclear physics, design challenges of the π^0 spectrometer, art, and everything in between. I have been amazed at the way physics was conducted in his time looking through hand-drawn graphs, detailed blueprints and schematics of every component in the spectrometer, and photographs of the spectrometer at every stage of construction. In addition I have spent countless hours working through his published papers, trying to understand the questions he was probing with his neutral pion spectrometer. I also contacted his old friends and was able to ask them about the work they did

together with my grandfather, as well as the results of their investigations. Inadvertently I also discovered some memorabilia of my grandfather tucked away in these boxes, pictures of himself with the design team and engineers, and a framed sheet of paper with Maxwell's equations written on it with photograph of himself in the corner, which I cherish especially. All of this has been an extremely rewarding experience, and one which on its own would justify the slightly unusual nature of this project. Although this might not convince some physicists that this topic makes for a good senior thesis, my hope is that in a profession which is defined by insatiable curiosity, the reader will forgive me for pursuing a curiosity of my own.

Finally, this thesis topic was also an opportunity for me to explore a sub-field of physics to which I had had very little exposure, and to develop skills which I can use throughout my graduate studies. Throughout my time working on this project I have learned how to use Geant4, a Monte Carlo simulation program frequently used in particle physics today. Through researching the physics behind π^0 physics I have been exposed to the literature of contemporary particle physics and have come into contact with current particle physicists at the University of Michigan. I can also credit this thesis project with singlehandedly turning my professional interests towards particle physics, which I will pursue in graduate school. So although this project is primarily concerned with things that happened in the past, it still points towards the future.

Chapter 2

Introduction

The content of this project consists of two major sections. The first is a history of the π^0 spectrometer including nuclear physics at the time of construction, the construction of the Los Alamos Meson Facility and the π^0 spectrometer itself, some of the discoveries made by the spectrometer, and the final fate of the spectrometer. The second portion is the construction of a model of the π^0 spectrometer in Geant4, and a series of simulations which aim to show that the model accurately represents the spectrometer.

The goal of the history project is to present a complete narrative of the π^0 spectrometer from construction to decommission using details from papers, progress reports, and personal notes from that time. Much of the physics present in these sections is from the standpoint of the prevailing thought at the time, not from a modern perspective. I believe that this will most honestly represent the motivations of the the physicists who participated in these projects, without imposing any of the arrogance which often accompanies hindsight. This is especially applicable to this endeavour, since many of the limitations of the π^0 spectrometer inhibited its ability to contribute to the field as a whole.

Since no new physics will be explored in the simulation portion of this experiment, the goal of that portion of the thesis will be different than usual. The purpose of the Geant4 application is to build an accurate representation of the π^0 spectrometer which will be demonstrated through simulations of the detector in various configurations. This will demonstrate technical ability in the use of Geant4 and of building Geant4 applications and it will demonstrate a knowledge of the physical processes by which the π^0 spectrometer detects and measures the energy of neutral pions.

Chapter 3

History

3.1 Background

In the wake of the stupendous accomplishments of nuclear physics in the 1940s and 50s, there was great excitement and hope for a new age of cheap energy production stemming from the promise of the hidden power of the atom. Nuclear physics had become woven into many layers of the fabric of society, especially in defense, industry, and medicine. It helped end a great war, yet it had also uncorked the potential for wars far deadlier than any before, challenging world leaders to create a lasting peace. Los Alamos was at the center of this narrative, and throughout the 1960s under the leadership of Norris Bradbury it sought to diversify its nuclear physics research while still maintaining its national security capabilities. [1]

In the 1960s, the foundation of nuclear physics was based mainly on quantum theory and Yukawa's theory of nuclear interactions. Particle physics had just seen the introduction of the widely successful Eightfold Way [2]. And the primary model of the nucleus was that of a liquid drop, filled with a Fermi liquid of nucleons governed by the Pauli exclusion principle, as described by Carl Friedrich von Weizsäcker in 1935 [3]. In this model, the nucleus is a collection of nucleons, each moving independently, with an internal repulsive electrostatic force proportional to the number of protons. In this case, the nucleonic degrees of freedom were the prime factor in exploring nuclear structure.

Yukawa's pion had been discovered, and it appeared to fit well with his theory. However some uncertainty still remained, and it was postulated that pionic degrees of freedom must be a part of any comprehensive description of nuclei and nuclear reactions [1]. Since the pion was the force carrier of the strong nuclear force, it seemed to be an ideal probe of nuclear structure. It is in this context that meson factories were constructed around the globe, with the intention of using mesons, and pions in particular, as tools for nuclear research.

The meson factory constructed at Los Alamos was the Los Alamos Meson Physics Facility (LAMPF), and its construction had been proposed in 1962. In October of 1968, the first funds were allocated for its construction, and within four years the first full powered proton beam was observed at the switchyard. This beam was then diverted for various experimental facilities: a biomedical facility, a high-resolution proton

spectrometer, the Los Alamos Neutron Scattering Facility (which still operates today), a high energy pion channel, and a low energy pion channel (LEP).

This channel was designed to have good resolution for pion-nucleus scattering experiments. For general utility requirements, it was deemed that pion flux be maximized, that the position and angle of the exit beam should be independent of momentum, that the beam should be short to minimize pion decays in the beam, that the beam spot size should be variable, and that the pion beam should be parallel to the floor [4]. It is in this channel that we find the location of the π^0 spectrometer.

3.2   Construction of π^0 Spectrometer

On the newly constructed LEP channel at LAMPF, various experiments and detectors were proposed, vying for a spot on the beamline. Among these was the π^0 spectrometer, which was proposed to fill a gap in standard pion physics experiments. Neutral pion detection efficiency and resolutions had improved and neutral pion physics was just now becoming viable for nuclear physics experiments, where higher resolution was required, (although they had been taking place in sub-atomic physics for some time).

The π^0 spectrometer was motivated by a lack of neutral pion experiments in pion-nucleus interactions. At the time many charged-pion spectrometers had been built at all meson factories, yet neutral-pion spectrometers were rare, since it was difficult to get good energy resolution because of the difficulty of measuring their kinematic properties. However, after experiments in the late 1970s had detected neutral pions from charge-exchange reactions (π^+, π^0), my grandfather along with his colleagues at Los Alamos, Case Western Reserve, and Tel-Aviv University endeavoured to build a high-resolution π^0 spectrometer at LAMPF.

The novel design of the spectrometer could achieve an energy resolution of 2-5MeV (fwhm) in detecting pions at high efficiency between 40-500MeV [5]. The double armed spectrometer design had a good enough resolution to study pion-charge-exchange reactions, π^0 production, and kaon strangeness-exchange reactions, although they struggled with achieving this theoretical resolution for several years [6]. The earliest physics proposals on the spectrometer were to study charge-exchange reactions $\pi^\pm + A \rightarrow A + \pi^0$ on various nuclei. The expected energy resolution was determined to be < 2MeV (fwhm) if the detector arms were placed sufficiently far away from the target, increasing the angular resolution but lowering the count rate [7]. The expected angular resolutions of 2-4° (fwhm) were also thought to be adequate for measuring diffraction-like angular distributions. This would be the first time that these sorts of experiments were to be conducted on nuclei with A > 3.

Later on, the experiments conducted would be focused primarily on exploring the Giant Dipole Resonance [8], a collective oscillation of the atomic nuclei, and isovecter excitations [9], in which nuclei and their spins are collectively excited. These experiments would take advantage of the new avenues opened up in charge-exchange reactions because of the zero spin of the neutral pions.

3.3 Discoveries Made by The π^0 Spectrometer

Among the work done by the π^0 spectrometer, a large share was dedicated to exploring the Giant Dipole resonance. Some of the earliest work done in this area was on ^{12}C and ^{40}Ca. It was found that the π^0 were very sensitive to this phenomenon, and that there was a peak in pion energies corresponding to GDR. These experiments confirmed the newer developing model of the nucleus not as a liquid drop, in which these sorts of collective motions were not expected, but instead as independent particles. In accounting for the strange behavior of the strong force, which is attractive at farther ranges, and repulsive at very-close range, a theoretical understanding of these phenomena was formed.

A second feature of the data collected on GDR experiments was the angle-dependent broadening of the GDR peak. These again seemed to have straightforward theoretical interpretations, and since the data from experiments had very clear data, there was much optimism for using pion-charge-exchange reactions to further study GDR [8]. In fact in the following years a great number of the GDR experiments would be conducted on the π^0 spectrometer by a large number of users [6].

Because the pion has 0 spin, it is an especially well suited tool for studying certain nuclear excitiations. In particular isovector excitations were of great interest at the time, as isovector monopole excitations had not yet been observed in nuclei. Using the spin-isospin properties of pions, the first identification of the giant isovector-monopole resonance (IVMR) was observed using the π^0 spectrometer. This excitation is a "collective motion in which the protons and neutrons oscillate out of phase along the nuclear radius" [1], helpfully described as the protons breathing in and out while the neutrons breathe out and in. This was able to be observed since the 0 spin of the pion limits the amount of spin-flip states excited, resulting in more isospin excitation.

At the time these were considered to be great achievements of this novel experimental method, one which easily justified the construction of the π^0 spectrometer to the administrators at LAMPF. Yet after these successes, the amount of productive work left for the spectrometer to partake in began to diminish, and over time many of the original designers and builders of this detector, including my grandfather, moved on to other projects, especially to high-energy physics at Fermilab.

3.4 Fate of the π^0 Spectrometer

As the field of nuclear physics and particle physics advanced, it became increasingly clear that low and medium energy physics was inherently limited in its ability to investigate the strong force. At these energies, the quarks remain bound and their interactions cannot be observed directly. As theory increasingly pointed towards the existence of quarks and gluons within nucleons, high energy physics took up the mantle of exploring the nuclear forces. A race to build increasingly powerful particle accelerators began culminating in the Large Hadron Collider at CERN, where previously medium and low energy meson factories had reigned supreme.

As for the π^0 spectrometer it remained in service for some time afterwards. In the late 80s there were proposals submitted to the LAMPF administrators and the Department of Energy asking for funding for a

new π^0 spectrometer which would continue work on pion-charge-exchange reactions at higher energies and with better resolution, and some correspondence remains between the proposal writers and my grandfather about the new π^0 spectrometer. However this proposal was never approved and the new spectrometer was not built.

Up until 1995 LAMPF continued to receive funding for meson physics from the DOE, however much of the research had shifted towards neutron physics, with a heavy emphasis on weapons development. At this point it was renamed to the Los Alamos Neutron Science Center (LANSCE) to reflect the internal changes at Los Alamos, and LANSCE remains in operation to this day. The meson factory at Los Alamos began as an attempt to diversify and de-weaponize the national labratory, but as the opportunities to make discoveries in nuclear physics using mesons began to dry up, Los Alamos lost funding for its non-weapons research and returned once more to being primarily a weapons lab.

While it is clear to us now that low energy pion physics could not hope to describe nuclear forces in their entirety, these experiments and spectrometers were still worthwhile to conduct and build. Not all physics experiments are going to yield massive breakthroughs in our understanding, yet in the pursuit of truth all avenues must be explored fully. After all we do not know ahead of time where we will find the next unexpected finding which undermines our existing theories and challenges us to reform our understanding of the universe.

Chapter 4

Physics

4.1 Introduction

In the physics portion of this thesis there are two main goals. One is to understand and describe in detail how the π^0 spectrometer operated, the other is to build a model of the spectrometer in Geant4 that can accurately simulate the operation of the spectrometer. In achieving these goals no experiments were conducted on untested hypotheses. Instead, simulations were run using the Geant4 model of the spectrometer in order to test how well the model represented the spectrometer.

In achieving the first of these goals, understanding the operation of the π^0 spectrometer, I relied upon published articles of my grandfather, especially *Design, Construction, and Performance of a High-Resolution π^0 Spectrometer for Nuclear Physics Experiments* [5], and the progress reports he presented to the LAMPF committees. In some cases I was also able to reference the original blueprints, although those that remained were mostly limited to mechanisms which enabled the vertical and lateral movement of the spectrometer arms. Throughout this process I moved from the theoretical calculations which premise double arm spectrometry for neutral pions to a description of how each component of the detector served to detect and record the π^0 decay.

The information collected about the π^0 spectrometer was then directly used to implement a Monte Carlo simulation in Geant4. Although many of the properties of the detector arms are presented quite clearly in the published articles and notes that my grandfather left behind, implementation of these elements in the Geant4 software is far from straightforward. Geant4 is written in C++ and it provides base classes with which the simulations are conducted. These include the hits, tracking, and geometry. It also includes a large number of prefabricated physics processes lists which it references for simulation, however these are not part of the core Geant4 software. Different materials are also appended in a large database to the application in the NIST material manager. However, building a simulation amounts to writing an object oriented application in C++ from scratch, while including the provided Geant4 base classes and databases.

Additionally, there is a large amount of prerequisite software which is required for installation and visualization of the simulations. These include CMake, a C++ compiler, Qt5, and X11 OpenGL visualization,

among others. Acquiring, installing, and ensuring the compatibility of all the supporting software from source is a large task and one that is prone to error. Fortunately, most users of Geant4 are able to avoid this entire process by accessing Tier 3 and Tier 2 CERN computing facilities, which have Geant4 already configured.

As for building the applications themselves, my initial inexperience with object oriented C caused me much consternation and confusion. Much of my time building the π^0 spectrometer simulation application was spent troubleshooting bugs which could have easily been avoided with more experience. To overcome these issues I relied heavily upon CERN's *Book For Application Developers* manual [10] but I also referenced many other sources to help with details more specific to the detector, especially the Geant4 resources provided by SLAC [11] [12].

In lieu of describing in detail the necessary steps required to operate Geant4 and build applications I will assume henceforth that the reader knows how to use Geant4 and build applications with it, or that she will be able to figure it out with sufficient time and effort. This means that while describing the constructing of the π^0 spectrometer simulation I will limit myself to details concerning the construction of this specific application. This I believe will provide sufficient material with which the Geant4 application could be reproduced.

4.2 Description of the π^0 Spectrometer

The motivation behind behind pion physics in the late 70s and 80s was to study nuclear structure through pion-nucleus interactions. At the scale of the atomic nucleus, mesons are the carriers of the strong force and as such it seemed natural to use mesons such as pions to probe nuclear strong force interactions. In labs around the United States, so-called meson factories were constructed to this end. One of these was a π^+ beam that was built in Los Alamos to be used in a variety of nuclear physics experiments at the Clint P. Anderson facility.

During this time there was an opportunity to study pion-charge-exchange reactions (π^+,π^0) by observing the π^0 produced in these reactions. In order to observe these neutral pions, a π^0 spectrometer was constructed at the Los Alamos by a collaboration of LAMPF, Tel-Aviv University, and Case Western Reserve University.

In simple terms, the reaction occurred as follows. A π^+ would strike a target, whose nuclear structure was to be analyzed, and produce π^0 in a pion-charge-exchange reaction $\pi^+ n \rightarrow \pi^0 p$. This short lived pion would be observed by the spectrometer via the dominant (98.8%) decay mode $\pi^0 \rightarrow \gamma\gamma$ [5]. The spectrometer was double armed and symmetrical about the target in order to select for π^0 which decay in to γ with similar energies. The spectrometer would then measure the direction and energy of the incident γ rays. This would then be used to determine the total energy of the π^0 in the following way. Let 2η be the laboratory frame angle between the two decay photons, E_1 and E_2 be the laboratory frame photon energies, W_{π^0} the total energy of the pion, m_0 the rest energy of the pion, and γ the ration between them.

Then

$$W_{\pi^0}^2 = \gamma^2 m_0^2 = \frac{2m_0^2}{(1 - \cos \eta)(1 - X^2)}, \tag{4.1}$$

where

$$X = \frac{E_1 - E_2}{E_1 + E_2}. \tag{4.2}$$

Thus if the energies of the photons are close to one another X^2 will be small and a measurement of η can be used to determine W_{π^0} [5].

The two arms of the spectrometer are position sensitive, high-energy gamma ray detectors that record the energy and direction of the incoming photons in the following way. First, incoming gamma rays will pass through a thin plastic cover on the face of the detector, which is used to block stray charged particles from interfering with the experiment. Then, the gamma rays pass through three sets of lead glass and multiwire proportional chambers (MWPC). The photons convert into positron-electron pairs in one of the three slices of lead glass, and the MWPCs record the conversion coordinates of the charged particles passing through them. Thus it is known in which of the three lead glass slices the gamma ray was converted and where. After this, the charged particles enter 15 blocks of Pb-glass in which they emit Cerenkov radiation proportional to the energy of the incoming photons. Finally the Cerenkov radiation is recorded by photomultiplier tubes located at the rear of the Pb-glass blocks. Thus the energy of the photons can be recorded which in combination with the angle between the photons (2η), determined by the angular separation of the detector arms, can be used to measure the total energy of the π^0 (W_{π^0}). A diagram of the detector is included below in Figure 4-1.
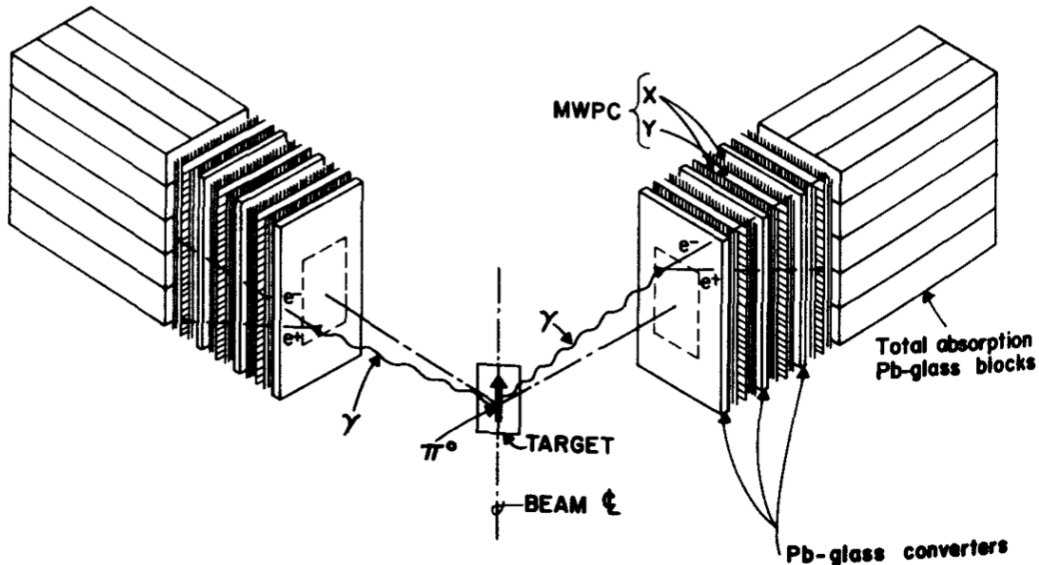

Figure 4-1: A schematic diagram of the detector. The scattering target, as well as the two detector arms are indicated. [5]

The resolution of the ΔW_{π^0} is dependent on the energy resolution of the incident gamma rays ΔE_γ and

the resolution of the opening angle $\Delta\eta$.

$$\Delta W_{\pi^0} = \left[\left(\frac{\partial W_{\pi^0}}{\partial E_\gamma} \Delta E_\gamma \right)^2 + \left(\frac{\partial W_{\pi^0}}{\partial \eta} \Delta \eta \right)^2 \right]^{1/2} \tag{4.3}$$

Thus the most important factor in determining the type of Pb-glass used in the spectrometer arms is their energy resolution of the gamma rays. Between the types of lead glass considered for the detector, SF5, SF2, F2, and LF5, it was found that the glass with less lead content had better energy resolution. The tests to determine this were conducted by T. Mast at the Mark III accelerator at Stanford University [13]. This is because the Pb-glass with less lead content has better transmittance for wavelengths between 300nm-700nm [14] [15].

The other factor which influences the π^0 energy resolution is the angular resolution $\Delta\eta$. The arms of the spectrometer cover too much solid angle to provide sufficiently precise angular resolution. For this reason the MWPCs are employed to provide a more precise measurement of the angle of the incident gamma rays. Especially important is good resolution in the x-direction (this is the lateral direction), while y-direction resolution is less important (this is the vertical direction). When an incoming photon was converted in one of the three Pb-glass slices, the MWPC module directly behind determined the x and y coordinate of its conversion. Each MWPC module consisted of three planes, two which determined the x-coordinate, and one which determined the y-coordinate. This can be observed on Figure 4-1. Unfortunately I have not been able to find any record of the algorithms used to determine this information from the three MWPC planes.

4.3 Geant4 π^0 Application

4.3.1 Introduction

Having thoroughly explored the design of the π^0 spectrometer, the way in which it would detect π^0 produced in charge-exchange reactions, as well as factors which influence its resolution, it is time to use the information gathered to build an accurate model of the spectrometer in order to run Monte Carlo simulations of its operation. The software chosen for this task was Geant4. Geant4 is a toolkit developed to simulate the passage of particles through matter using Monte Carlo methods [16]. Geant4 is maintained by a world-wide collaboration of physicists and software engineers, and is heavily supported by the CERN collaboration [17].

I chose Geant4 because of its important role and relevance throughout particle physics, and because it came highly recommended from the faculty at the University of Michigan with whom I had the privilege of working with in the summer of 2020. A major goal of this project was to develop these relevant skills in Monte Carlo simulation, and so a major portion of my work was dedicated to the construction of this Geant4 application, and to test its accuracy. To this end I will spend some time detailing the building of the application, which I think it necessary to demonstrate the work done in this thesis project.

In this section I will describe the methods used in order to construct and simulate an accurate model of

the π^0 spectromter. This will include a description of how to build a detector, initiate applicable physics processes, how to generate primary particles, as well as the initialization of the required classes necessary to run any Geant4 application, and command-based scoring to record hits on the detector. As stated previously, this section will assume prior knowledge of the installation and operation Geant4, as well as familiarity with object oriented C++. The version of Geant used in this application was 4.10.07.p1, and all relevant documentation can be found at [18] [19].

The information used for the following sections was gathered from the historical research done on the π^0 spectrometer, especially from *Design, Construction, and Performance of a High-Resolution π^0 Spectrometer* [5]. In particular this description will focus on the three required User Initialization Classes in a Geant4 application, `G4VUserDetectorConstruction`, `G4VUserPhysicsList`, and `G4VUserActionInitialization`, which initialize the simulation and are managed by the `G4ProcessManager` in the `main.cc`.

4.3.2 Geometry

The geometry of the π^0 spectrometer was taken from "Design, Construction, and Performance of A High-Resolution π^0 Spectrometer for Nuclear Physics Experiments" [5]. The spectrometer consists of two detector arms with dimensions of 93.3cm×45cm×75cm. The two arms are positioned in symmetrically about the target, where the π^0 come into existence. First, the geometry of the world volume must be specified, which in this instance was done with a `G4Box`. This is necessary since all the other volumes and primary particles must be placed with respect to its enclosing volume. Then, the solid volume of the detector arm was specified in another `G4Box`. Then, to place the detector arms in the world volume, a physical volume must be defined, which takes the solid volume and locates it relative to the world volume. This allows the detector to take physical form, in which physical processes can be simulated.

The location is defined by its radial distance from the target, and the angular separation between the arms. The radius is defined in the User Initialization Class `G4VUserDetectorConstruction`. In order to modify the radius it is necessary that the application be rebuilt and the geometry reconstructed from scratch. This was acceptable since each simulation run was held at a constant radius, while the angular separation varied. In order to vary the angular separation between the detector arms in the middle of a run, it was necessary to define a `G4GenericMessanger` class, which would allow the detector to geometry to be modified and reconstructed without the need to rebuild the application. After invoking the messenger class, a command must be defined which modifies the physical volume of the detector arms, and then modifies the geometry of those specific elements in the application before another event is simulated.

4.3.3 Materials

In order to have the detector interact with the particles in the simulation, a logical volume must be defined, which includes information about the detector's material. The detector consists of two different materials, one is the lead glass which is used for both conversion and for the total-absorption Cerenkov counters and the multiwire-proportional chambers which locate the conversion point of the incoming photons. For the purposes of the examining the energy resolution of the detector it is necessary to include the three sets of multi-wire proportional chambers (MWPC) on the front of the detector. However, since the the algorithms
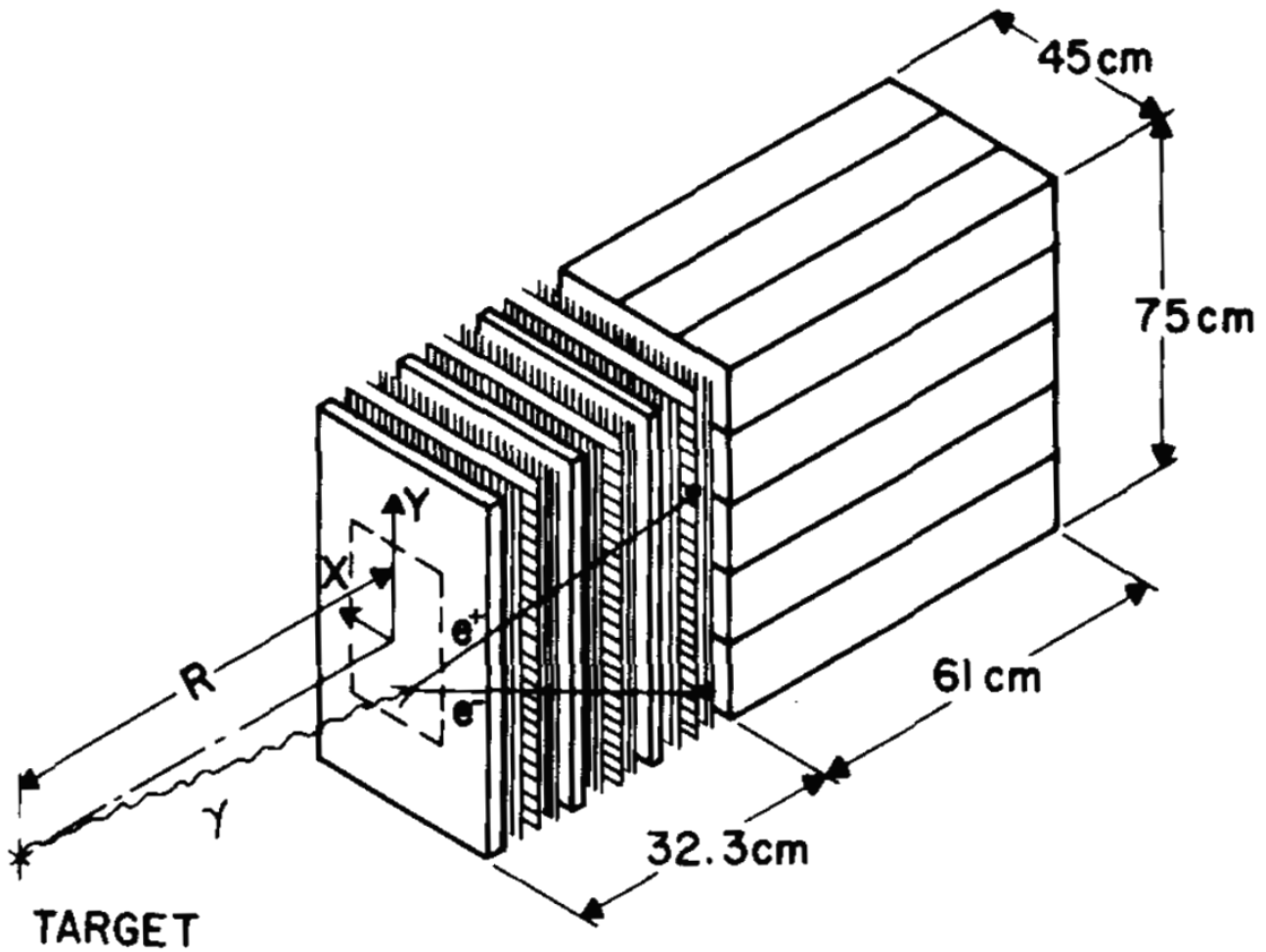
Figure 4-2: Figure 1: A schematic diagram of an arm. Three sets of converter MWPCs, as well as the array of Pb-glass total absorption Cerenkov counters can be seen. [5]

used to determine this are lost, and because the scope of this project is less ambitious, it was decided that it would be sufficient to determine the optimal radius and angular separation of the detector arms.

For this reason it was determined that it was sufficient to fill the entire detector arm with lead glass, which would be able to simulate both the conversion of the gamma rays into a charged particle shower, and the emission of Cerenkov radiation from that particle shower. The lead glass material used was taken from a predefined `G4Material` using the NIST Material Data Base, `G4_GLASS_LEAD`. The lead glass is defined as a partial fraction of Oxygen, Silicon, Titanium, Arsenic, and Lead [20].

In order to simulate Cerenkov radiation inside of the detector arms it is also necessary to provide the optical properties of the lead glass. The essential optical properties for Cerenkov simulation are the refractive index and the absorption depth. The data for the lead glass used in the application was taken for from Schott's LF5 Pb-glass data sheet [14] to match the optical properties with the original π^0 spectrometer for wavelengths 334nm-2325nm.

4.3.4 Physics Processes

The primary physics processes are defined by the `FTFP_BERT` physics list, which is a combination of the Fritiof Model and Bertini Cascade Model and includes a large range of hadronic and electromagnetic processes. This list is a common standard in Geant4 simulations. In addition to this, the optical processes are added explicitly through `G4OpticalPhysics.cc`, and a step limiter is added through `G4StepLimiterPhysics.cc` [21] [22]. The optical processes are only active in the regions of the world volume where optical properties are assigned to the materials. In this case of this application the optical processes are limited to the lead glass detector arms. The combination of these physics processes is then added to the `main.cc` as one of the necessary User Initialization Classes, the `G4VUserPhysicsList`.

4.3.5 Primary Generator Action

The `G4VUserActionInitialization` is a mandatory User Initialization class and requires at least one User Action Class, the `G4VUserPrimaryGeneratorAction`. The primary generator action defines the initial state of the primary event. In this case this is the production of a π^0 at the location of the target, which then quickly decays into two gamma rays. The primary action generation used in this simulation is the simplest possible. It is a point source of π^0 defined by `G4ParticleGun` with a single energy, which was then varied throughout the simulations from 40MeV to 400MeV, as was measured in *Design, Construction, and Performance of a High-Resolution π^0 Spectrometer* [5]. In order to vary the energy, the application must be altered and then rebuilt, similarly to altering the radius of the detector arms.

4.3.6 Scoring

In order to record the hits on the detector arms, it is necessary to include scoring in the application. Geant4 does its physics simulation silently, and in order to retrieve information, special scoring volumes and scoring commands must be implemented [11]. To avoid creating additional classes, I opted to use Command-based scoring to record the hit information. Command-based scoring defines a `G4MultiFunctionalDetector` to a volume, which allows a logical volume to act as either a scoring mesh or a scoring probe, which can be defined after initiating a `G4ScoringManager` in the `main.cc` file. I decided to use the total-absorption lead glass blocks as the scoring volume, and defined a scoring mesh inside of the block. Although a more accurate representation of the spectrometer would be to include photo-multiplier tubes at the rear of the detector arms to act as a Geant4 sensitive detector. This would then be able to record and report data as the spectrometer itself would. However, for the purposes of this simulation it was sufficient to gather the hit data from the detector arms themselves, thus assuming maximum efficiency of the Cerenkov counters.

Once a scoring mesh is defined, any number of primitive scorers can be defined to that detector [12]. The primitive scorer used in this simulation is the population scoring method. The population scoring method is activated if the scoring volume detects particles of a particular type, in this case gamma rays, and then counts the number of photons which enter the scoring region. In this way we can simply determine if both detectors were struck by photons simultaneously and thus can be classified as a hit.

4.4 Methods

The purpose of this simulation is to determine where the optimal location of the detector arms are so that the most number of hits are recorded. Since the detector arms individually only cover a small amount of solid angle, even when close to the detector, much effort was spent optimizing the recording capabilities of the detector. To simulate this, we placed the detector arms in a number of different locations, and simulated the amount of hits recorded at each location.

To observe the performance of the π^0 spectrometer at different angular and radial locations, a series of simulations were run at various energies. The simulations were run at four different energies, in order to encompass the energy range for which the detector was designed. The energies chosen were 40Mev, 100Mev, 200Mev, and 400Mev. Since the creation of the π^0 in the primary generator action was for a single energy, the application needed to be rebuilt for the simulations run at each of these energies.

For each energy chosen, simulations were run for radial distances from the target from 1.00m to 3.00m in 25cm increments. Just as for the energies, the application was rebuilt for each of these radii. At each radius, 1000 events were simulated per each angular separation, ranging from 10 to 80 degrees in 5 degree increments, excluding those angular separations which would result in the detector arms overlapping with one another. In total, this resulted in 496,000 events being simulated.

For each experimental run, at a constant energy and radius, a list of commands is fed to the Geant4 to execute. After initializing and building the detector, the two detector arms are each separately initialized as multifunctional detectors capable of scoring. Then, after a scoring mesh is created in the detector arms, the primitive scoring capability is added. With the goal of determining when each of the arms had scored a hit, the population scoring capability is added to the scoring mesh, which would only activate upon receiving Cerenkov radiation from the charged particle shower. After this the detector arm angle is set via command, and 1000 events are simulated.

At each event, the data from the scoring meshes is read out of the system, and are compared to one another. If both detector arms experienced Cerenkov radiation, then a hit is recorded. After 1000 events the detector arm angles are changed via command and another 1000 events are run. All the applicable angles for a given energy and radius are thus simulated, and their hit number recorded.

4.5 Observations and Analysis

The results of these simulations are reported in the following figures. For each figure, some of the features are fairly intuitive. At lower radii the hit count is always higher than at further radii, simply because the detector arms cover a larger solid angle when they are close to the detector. For each energy, there is also an optimal angle at which more hits are detected. For the 40MeV pion this appears to be 50 deg, for the 100MeV pions this is around 37 deg, for the 200MeV pions this is around 23 deg, and for the 400MeV pions it is a very small angle which can not be clearly ascertained. This angle is where the photons have nearly identical energy and they decay symmetrically. This is especially useful if the radius of the detector arms from the target is increased in order to achieve better angular resolution.

Some artifacts are the result of systemic error. At smaller angles more hits are counted than at larger angles, although their distance from the optimal may be identical. This is caused when gamma rays which strike only one arm scatter and enter into the second arm, causing a smaller but still significant Cerenkov signature. This could be remedied by taking into the account the amount of energy deposited in the detector arms, and only triggering the detector when the energies of the photons are similar.

Other factors such as the blank boxes are caused due to detector limitations. No events could be simulated for these regions since the detector arms would overlap at these locations. The error for each box is 3.16%, which could be improved upon by increasing the number of events simulated for each box.
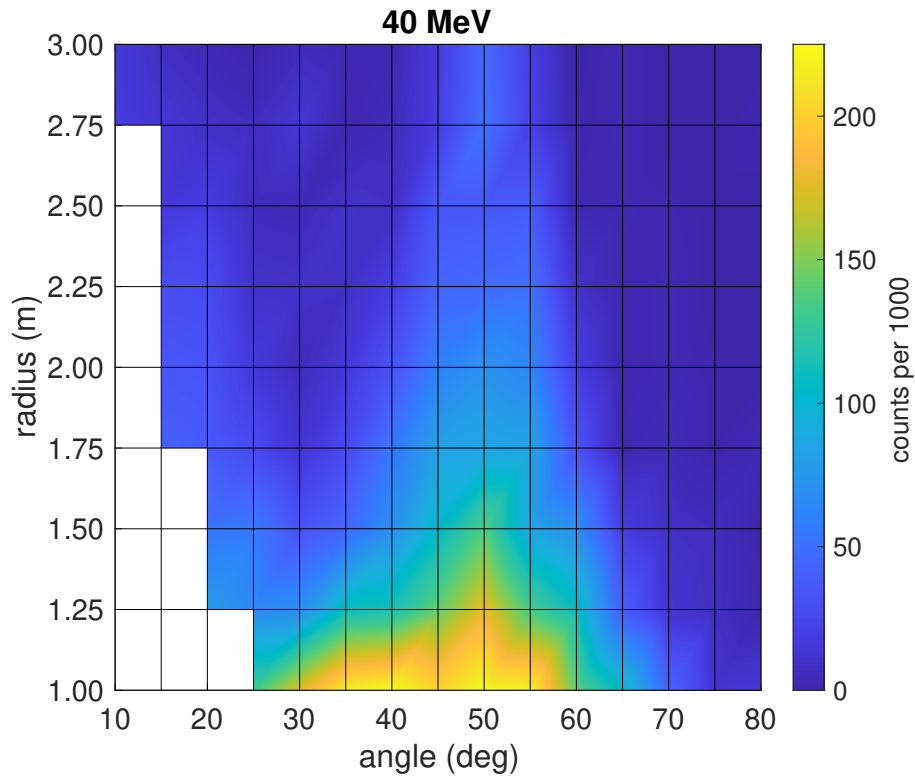

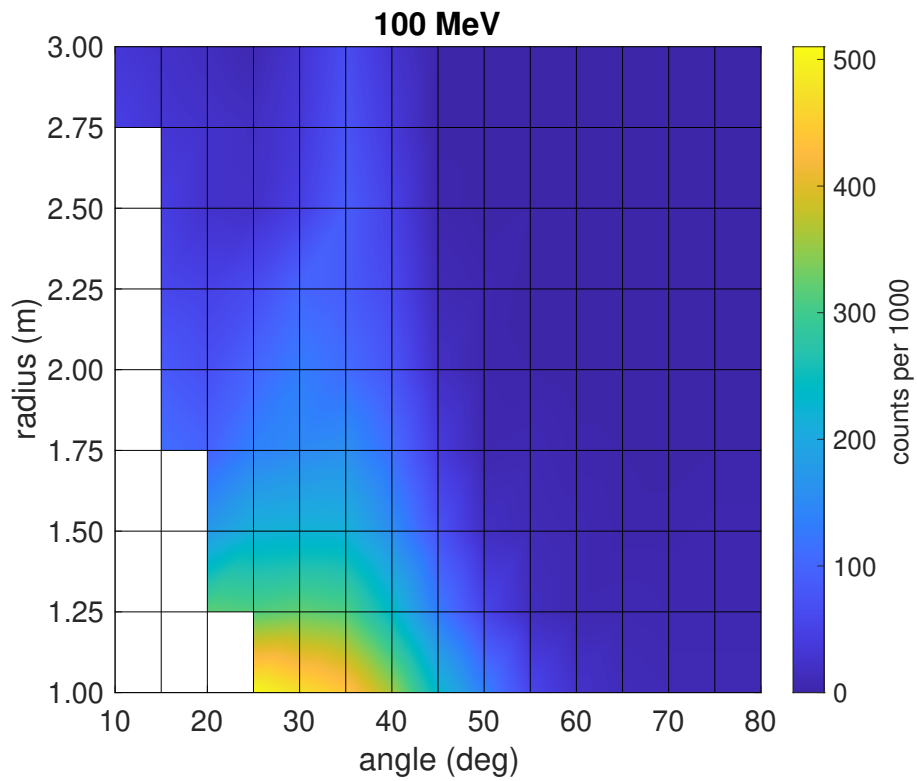Figure 4-3: Heat map of hit number at 40MeV at various radii and angles

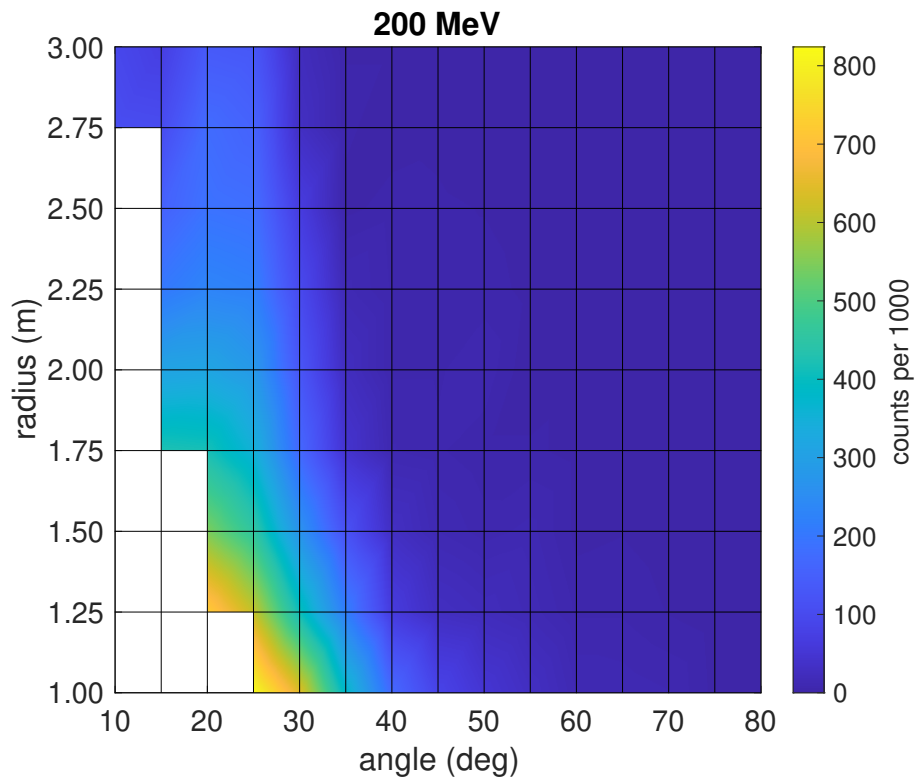Figure 4-4: Heat map of hit number at 100MeV at various radii and angles

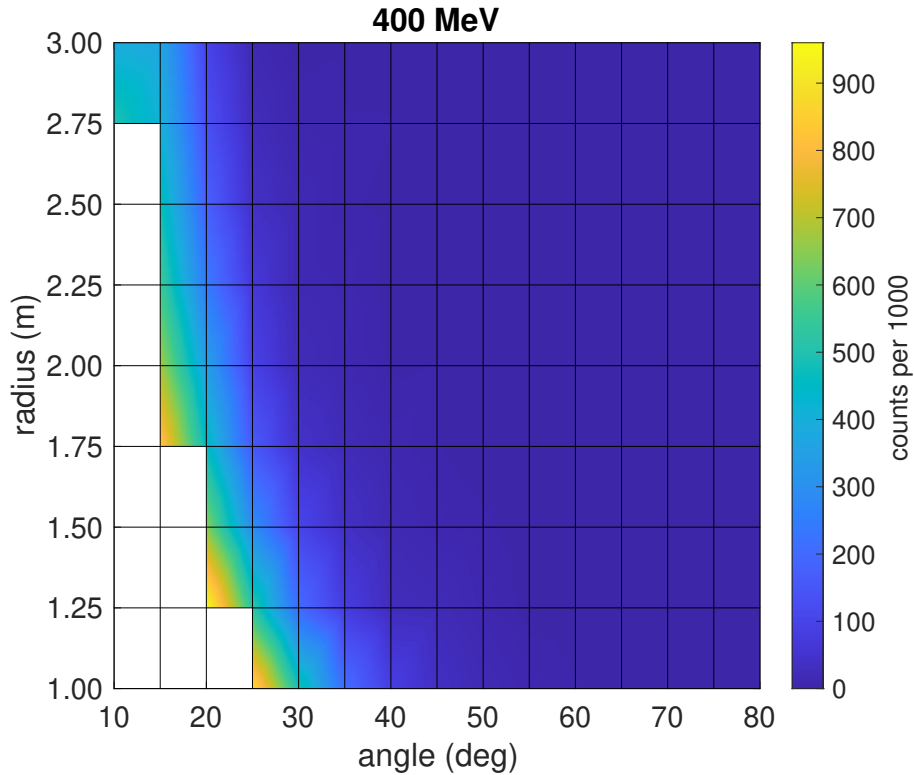Figure 4-5: Heat map of hit number at 200MeV at various radii and angles

Figure 4-6: Heat map of hit number at 400MeV at various radii and angles

4.6   Results

What I found is that these results generally agree with many of the observations made with the π^0 spectrometer, but with a few caveats. One is that the systemic error causing the amount of events to be detected at low angles is not present on the spectrometer, since shielding was put into place to counteract this. Secondly, while the energy of the incoming π^+ could be controlled, this did not necessarily correspond to a single π^0 energy, and usually there was a distribution of energies with a half width of 50MeV. Furthermore, the momentum of the π^0 was not necessarily fixed to be straight down the middle between the two detector arms, and there was often an angle at which the π^0 scattered off the target. Neither of these considerations were simulated.

In addition, the π^0s were not produced in the same location, since the charge-exchange reaction could occur at any point in the width of the target. Generally the π^0 that were produced at the rear of the target had lower energy, since the π^+ needed to pass through some material, slowing it down. This lead to a greater opening angle between the decay gamma rays, somewhat counteracting the smaller radius from the detector. Including all these additional considerations, the optimal positioning of the detector arms required much more nuance, something not present in the current simulations.

Overall however, the Geant4 application of the π^0 spectrometer provides a solid foundation, which can accurately simulate the behaviour of the spectrometer as expected. If this model is to be improved all that is necessary is to include more of the nuanced physics which is present. Since the application itself is a

solid, albeit simplistic, model of the spectrometer, it can simply be added to and need not be discarded and reworked from scratch. In as far as this is concerned, the simulation is a success.

Chapter 5

Conclusion

I believe that the historical work present in this thesis provides a good insight into a previous era of nuclear physics, and that it has a good narrative of how physics research is conducted, applicable even to the modern day. Understanding the history of physics research provides valuable context, which can be used to better orient ourselves in our own current research, even if the topics themselves are no active areas of interest. In addition, no physical phenomenon is explored in vain, and in fact much of the work done on pion-nucles interactions still appears to this day as a background process in particle physics experiments such as the Muon to Electron experiment [23].

As for the personal goals I set for myself when beginning this thesis, in that respect it has been a complete success. I have been able to understand for the first time the research that my grandfather conducted, which had been a long standing ambition of mine. Although there is always more I could uncover digging through boxes upon boxes of his notes, I am at least able to put into context his work and achievements. And besides, I can say with some confidence that I understand the work that he did on this π^0 spectrometer, which his colleagues and friends had considered his most important work before he died.

I have also been able to gain experience using particle physics tools such as Geant4, something that I believe will be an invaluable asset to me as I continue my studies in graduate school. So that even if my project does not contribute to furthering our physical understanding, it has prepared me to do so in the future.

The physics portion of this project also achieved its aims of producing an accurate model of the π^0 spectrometer which can be simulated to test various characteristics of the detector. The biggest limitation of this portion of the project are the simplifications made to the design as respect to the original spectrometer. However with further time, these limitations could be overcome by changing portions of the current model individually.

Appendix A

Code Listing

```
1  pi0Spectrometer/
2  |  build/
3  |  include/
4  |  |  pi0SpecActionInitialization.hh
5  |  |  pi0SpecDetectorConstruction.hh
6  |  |  pi0SpecPrimaryGeneratorAction.hh
7  |  src/
8  |  |  pi0SpecActionInitialization.cc
9  |  |  pi0SpecDetectorConstruction.cc
10 |  |  pi0SpecPrimaryGeneratorAction.cc
11 |  CMakeLists.txt
12 |  GNUmakefile
13 |  init.mac
14 |  init_vis.mac
15 |  pi0Spec.cc
16 |  run.mac
17 |  vis.mac
```

Listing A.1: Directory and File structure for the π^0 spectrometer application

```cpp
1  // file pi0SpecActionInitialization.hh
2  // brief Definition of the pi0SpecActionInitialization class
3
4  #ifndef pi0SpecActionInitialization_h
5  #define pi0SpecActionInitialization_h 1
6
7  #include "G4VUserActionInitialization.hh"
8
9  /// Action initialization class.
10
11 class pi0SpecActionInitialization : public G4VUserActionInitialization
12 {
13   public:
14     pi0SpecActionInitialization();
15     virtual ~pi0SpecActionInitialization();
```

```
16
17     virtual void BuildForMaster() const;
18     virtual void Build() const;
19
20 };
21
22 // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
23
24 #endif
```

Listing A.2: Content of `pi0SpecActionInitilization.hh`

```
1  // file pi0SpecDetectorConstruction.hh
2  // brief Definition of the pi0SpecDetectorConstruction class
3
4  #ifndef pi0SpecDetectorConstruction_h
5  #define pi0SpecDetectorConstruction_h 1
6
7  #include "globals.hh"
8  #include "G4VUserDetectorConstruction.hh"
9  #include "G4RotationMatrix.hh"
10
11 #include <vector>
12
13 class G4VPhysicalVolume;
14 class G4LogicalVolume;
15 class G4Material;
16
17 class G4GenericMessenger;
18
19 /// Detector construction
20
21 class pi0SpecDetectorConstruction : public G4VUserDetectorConstruction
22 {
23   public:
24     pi0SpecDetectorConstruction();
25     virtual ~pi0SpecDetectorConstruction();
26
27     virtual G4VPhysicalVolume* Construct();
28
29     void SetArmAngle(G4double val);
30     G4double GetArmAngle() { return fArmAngle; }
31
32     void ConstructMaterials();
33
34   private:
35     void DefineCommands();
36
37     G4GenericMessenger* fMessenger;
38
39     G4double fArmAngle;
```

```
40
41    G4RotationMatrix* fRightArmRotation;
42    G4VPhysicalVolume* fRightDetArmPhys;
43
44    G4RotationMatrix* fLeftArmRotation;
45    G4VPhysicalVolume* fLeftDetArmPhys;
46  };
47
48  // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
49
50  #endif
```

Listing A.3: Content of `pi0SpecDetectorConstruction.hh`

```
1   // The primary generator action class with particle gun.
2   ///
3   /// The default kinematic is a 40 MeV pi0, starting at (0,0,0).
4
5   class pi0SpecPrimaryGeneratorAction : public G4VUserPrimaryGeneratorAction
6   {
7     public:
8       pi0SpecPrimaryGeneratorAction();
9       virtual ~pi0SpecPrimaryGeneratorAction();
10
11      // method from the base class
12      virtual void GeneratePrimaries(G4Event*);
13
14      // method to access particle gun
15      const G4ParticleGun* GetParticleGun() const { return fParticleGun; }
16
17    private:
18      G4ParticleGun*  fParticleGun; // pointer a to G4 gun class
19  };
20
21  // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
22
23  #endif
```

Listing A.4: Content of `pi0SpecPrimaryGeneratorAction`

```
1   // file pi0SpecActionInitialization.cc
2   // brief Implementation of the pi0SpecActionInitialization class
3
4   #include "pi0SpecActionInitialization.hh"
5   #include "pi0SpecPrimaryGeneratorAction.hh"
6
7   // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
8
9   pi0SpecActionInitialization::pi0SpecActionInitialization()
10   : G4VUserActionInitialization()
11  {}
```

```
12
13  // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
14
15  pi0SpecActionInitialization::~pi0SpecActionInitialization()
16  {}
17
18  // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
19
20  void pi0SpecActionInitialization::BuildForMaster() const
21  {
22  }
23
24  // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
25
26  void pi0SpecActionInitialization::Build() const
27  {
28    SetUserAction(new pi0SpecPrimaryGeneratorAction);
29  }
30
31  // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
```

Listing A.5: Content of `pi0SpecActionInitialization.cc`

```
1   // file pi0SpecDetectorConstruction.cc
2   // brief Implementation of the pi0SpecDetectorConstruction class
3
4   #include "pi0SpecDetectorConstruction.hh"
5
6   #include "G4RunManager.hh"
7   #include "G4GenericMessenger.hh"
8
9   #include "G4NistManager.hh"
10  #include "G4Material.hh"
11
12  #include "G4Box.hh"
13  #include "G4LogicalVolume.hh"
14  #include "G4PVPlacement.hh"
15  #include "G4VPhysicalVolume.hh"
16
17  #include "G4SystemOfUnits.hh"
18
19  // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
20
21  pi0SpecDetectorConstruction::pi0SpecDetectorConstruction()
22  : G4VUserDetectorConstruction(),
23    fMessenger(nullptr), fArmAngle(30.*deg), //fScoringVolume(0),
24    fRightArmRotation(nullptr), fLeftArmRotation(nullptr)
25  {
26    fRightArmRotation = new G4RotationMatrix();
27    fRightArmRotation->rotateY(fArmAngle);
28
```

```
29    fLeftArmRotation = new G4RotationMatrix();
30    fLeftArmRotation->rotateY(-fArmAngle);
31
32    // Define commands for this class
33    DefineCommands();
34  }
35
36  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo........oooOO0OOooo......
37
38  pi0SpecDetectorConstruction::~pi0SpecDetectorConstruction()
39  {
40    delete fRightArmRotation;
41    delete fLeftArmRotation;
42    delete fMessenger;
43  }
44
45  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo........oooOO0OOooo......
46
47  G4VPhysicalVolume* pi0SpecDetectorConstruction::Construct()
48  {
49
50    //Construct Materials
51    ConstructMaterials();
52    auto air = G4Material::GetMaterial("G4_AIR");
53    auto leadglass = G4Material::GetMaterial("G4_GLASS_LEAD");
54
55      // ——————————— Generate & Add Material Properties Table ———————————
56    //
57    std::vector<G4double> photonEnergy = {
58      2.034 * eV,  2.068 * eV,  2.103 * eV,  2.139 * eV,  2.177 * eV,  2.216 * eV,
59      2.256 * eV,  2.298 * eV,  2.341 * eV,  2.386 * eV,  2.433 * eV,  2.481 * eV,
60      2.532 * eV,  2.585 * eV,  2.640 * eV,  2.697 * eV,  2.757 * eV,  2.820 * eV,
61      2.885 * eV,  2.954 * eV,  3.026 * eV,  3.102 * eV,  3.181 * eV,  3.265 * eV,
62      3.353 * eV,  3.446 * eV,  3.545 * eV,  3.649 * eV,  3.760 * eV,  3.877 * eV,
63      4.002 * eV,  4.136 * eV
64    };
65
66
67    //
68    std::vector<G4double> refractiveIndex1 = {
69      1.3435, 1.344,  1.3445, 1.345,  1.3455, 1.346,  1.3465, 1.347,
70      1.3475, 1.348,  1.3485, 1.3492, 1.35,   1.3505, 1.351,  1.3518,
71      1.3522, 1.3530, 1.3535, 1.354,  1.3545, 1.355,  1.3555, 1.356,
72      1.3568, 1.3572, 1.358,  1.3585, 1.359,  1.3595, 1.36,   1.3608
73    };
74
75    std::vector<G4double> absorption = {
76      3.448 * m,   4.082 * m,   6.329 * m,   9.174 * m,   12.346 * m, 13.889 * m,
77      15.152 * m,  17.241 * m,  18.868 * m,  20.000 * m,  26.316 * m, 35.714 * m,
78      45.455 * m,  47.619 * m,  52.632 * m,  52.632 * m,  55.556 * m, 52.632 * m,
```

```
79      52.632 * m, 47.619 * m, 45.455 * m, 41.667 * m, 37.037 * m, 33.333 * m,
80      30.000 * m, 28.500 * m, 27.000 * m, 24.500 * m, 22.000 * m, 19.500 * m,
81      17.500 * m, 14.500 * m
82    };
83
84    G4MaterialPropertiesTable* myMPT1 = new G4MaterialPropertiesTable();
85
86    myMPT1->AddProperty("RINDEX", photonEnergy, refractiveIndex1)
87      ->SetSpline(true);
88    myMPT1->AddProperty("ABSLENGTH", photonEnergy, absorption)->SetSpline(true);
89
90    std::vector<G4double> energy_leadglass = {
91      1.56962 * eV, 1.58974 * eV, 1.61039 * eV, 1.63157 * eV, 1.65333 * eV,
92      1.67567 * eV, 1.69863 * eV, 1.72222 * eV, 1.74647 * eV, 1.77142 * eV,
93      1.7971 * eV,  1.82352 * eV, 1.85074 * eV, 1.87878 * eV, 1.90769 * eV,
94      1.93749 * eV, 1.96825 * eV, 1.99999 * eV, 2.03278 * eV, 2.06666 * eV,
95      2.10169 * eV, 2.13793 * eV, 2.17543 * eV, 2.21428 * eV, 2.25454 * eV,
96      2.29629 * eV, 2.33962 * eV, 2.38461 * eV, 2.43137 * eV, 2.47999 * eV,
97      2.53061 * eV, 2.58333 * eV, 2.63829 * eV, 2.69565 * eV, 2.75555 * eV,
98      2.81817 * eV, 2.88371 * eV, 2.95237 * eV, 3.02438 * eV, 3.09999 * eV,
99      3.17948 * eV, 3.26315 * eV, 3.35134 * eV, 3.44444 * eV, 3.54285 * eV,
100     3.64705 * eV, 3.75757 * eV, 3.87499 * eV, 3.99999 * eV, 4.13332 * eV,
101     4.27585 * eV, 4.42856 * eV, 4.59258 * eV, 4.76922 * eV, 4.95999 * eV,
102     5.16665 * eV, 5.39129 * eV, 5.63635 * eV, 5.90475 * eV, 6.19998 * eV
103   };
104
105   leadglass->SetMaterialPropertiesTable(myMPT1);
106
107   // Option to switch on/off checking of volumes overlaps
108   G4bool checkOverlaps = true;
109
110   // geometries ————————————————————————————————————————————————————
111   // World Volume
112   G4double world_hx = 8.0*m;
113   G4double world_hy = 2.0*m;
114   G4double world_hz = 8.0*m;
115
116   auto worldSolid
117     = new G4Box("worldSolid", world_hx, world_hy, world_hz);
118   auto worldLogical
119     = new G4LogicalVolume(worldSolid, air, "worldLogical");
120   auto worldPhysical
121     = new G4PVPlacement(0,G4ThreeVector(),worldLogical,"worldPhysical",0,
122                         false,0,checkOverlaps);
123
124   //Detector Arm
125   auto detArmSolid
126     = new G4Box("detectorArmSolid",45*cm,75*cm,93.3*cm);
127   auto detArmLogicalRight
128     = new G4LogicalVolume(detArmSolid,leadglass,"rightDetectorArmLogical");
```

```
129    auto detArmLogicalLeft
130      = new G4LogicalVolume(detArmSolid, leadglass, "leftDetectorArmLogical");
131    auto r = 1.5*m + 93.3*cm;
132    auto z = r * std::cos(fArmAngle) - 5.*m;
133
134    //Placement of Detector Arms
135    //This is a function so that it can be variable even after construction
136
137    //Right arm
138    auto x_right = -r * std::sin(fArmAngle);
139    fRightDetArmPhys
140      = new G4PVPlacement(fRightArmRotation, G4ThreeVector(x_right, 0., z), detArmLogicalRight,
141                          "rightDetectorArmPhysical", worldLogical, false,
142                          0, checkOverlaps);
143    //Left arm
144    auto x_left = r * std::sin(fArmAngle);
145    fLeftDetArmPhys
146      = new G4PVPlacement(fLeftArmRotation, G4ThreeVector(x_left, 0., z), detArmLogicalLeft,
147                          "leftDetectorArmPhysical", worldLogical, false,
148                          0, checkOverlaps);
149
150    //set the detector arm as the scoring volume ——————————————————————————————
151
152    //fScoringVolume = detArmLogical;
153
154    // return the world physical volume ——————————————————————————————————————
155    return worldPhysical;
156  }
157
158 // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
159
160 void pi0SpecDetectorConstruction::ConstructMaterials()
161 {
162    auto nistManager = G4NistManager::Instance();
163
164    //Air
165    nistManager->FindOrBuildMaterial("G4_AIR");
166    //Lead Glass
167    nistManager->FindOrBuildMaterial("G4_GLASS_LEAD");
168
169    // I don't know what this does
170    G4cout << G4endl << "The materials defined are : " << G4endl << G4endl;
171    G4cout << *(G4Material::GetMaterialTable()) << G4endl;
172 }
173
174 // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
175
176 void pi0SpecDetectorConstruction::SetArmAngle(G4double val)
177 {
178    if (!fRightDetArmPhys & !fLeftDetArmPhys) {
```

```
179        G4cerr << "Detector has not yet been constructed." << G4endl;
180        return;
181    }
182
183    fArmAngle = val;
184    //right arm
185    *fRightArmRotation = G4RotationMatrix();
186    fRightArmRotation->rotateY(fArmAngle);
187    auto r = 1.5*m + 93.3*cm;
188    auto z = r * std::cos(fArmAngle) - 5.*m;
189    auto x_right = -r * std::sin(fArmAngle);
190    fRightDetArmPhys->SetTranslation(G4ThreeVector(x_right,0.,z));
191
192    //left arm
193    *fLeftArmRotation = G4RotationMatrix();
194    fLeftArmRotation->rotateY(-fArmAngle);
195    auto x_left = r * std::sin(fArmAngle);
196    fLeftDetArmPhys->SetTranslation(G4ThreeVector(x_left,0.,z));
197
198    // tell G4RunManager that we change the geometry
199    G4RunManager::GetRunManager()->GeometryHasBeenModified();
200 }
201
202  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo........oooOO0OOooo......
203
204 void pi0SpecDetectorConstruction::DefineCommands()
205 {
206    // Define /pi0Spec/detector command directory using generic messenger class
207    fMessenger = new G4GenericMessenger(this,"/pi0Spec/detector/","Detector control");
208
209    //ArmAngle Command
210    auto& armAngleCmd
211      = fMessenger->DeclareMethodWithUnit("armAngle","deg",
212                                    &pi0SpecDetectorConstruction::SetArmAngle,
213                                    "Set rotation angle of the second arm.");
214    armAngleCmd.SetParameterName("angle", true);
215    armAngleCmd.SetRange("angle>=0. && angle<=90.");
216    armAngleCmd.SetDefaultValue("30.");
217 }
218
219 //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo........oooOO0OOooo......
```

Listing A.6: Content of `pi0SpecDetectorConstruction.cc`

```
1 // file pi0SpecPrimaryGeneratorAction.cc
2 // brief Implementation of the pi0SpecPrimaryGeneratorAction class
3
4 #include "pi0SpecPrimaryGeneratorAction.hh"
5
6 #include "G4LogicalVolumeStore.hh"
7 #include "G4LogicalVolume.hh"
```

```
8  #include "G4Box.hh"
9  #include "G4RunManager.hh"
10 #include "G4ParticleGun.hh"
11 #include "G4ParticleTable.hh"
12 #include "G4ParticleDefinition.hh"
13 #include "G4SystemOfUnits.hh"
14 #include "Randomize.hh"
15
16 // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
17
18 pi0SpecPrimaryGeneratorAction::pi0SpecPrimaryGeneratorAction()
19 : G4VUserPrimaryGeneratorAction(),
20   fParticleGun(nullptr)
21 {
22   G4int n_particle = 1;
23   fParticleGun  = new G4ParticleGun(n_particle);
24
25   // default particle kinematic
26   G4ParticleTable* particleTable = G4ParticleTable::GetParticleTable();
27   G4String particleName;
28   G4ParticleDefinition* particle
29     = particleTable->FindParticle(particleName="pi0");
30   fParticleGun->SetParticleDefinition(particle);
31   fParticleGun->SetParticleMomentumDirection(G4ThreeVector(0.,0.,1.));
32   fParticleGun->SetParticleEnergy(140.*MeV);
33 }
34
35 // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
36
37 pi0SpecPrimaryGeneratorAction::~pi0SpecPrimaryGeneratorAction()
38 {
39   delete fParticleGun;
40 }
41
42 // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
43
44 void pi0SpecPrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)
45 {
46   fParticleGun->SetParticlePosition(G4ThreeVector(0,0,-5.*m));
47
48   fParticleGun->GeneratePrimaryVertex(anEvent);
49 }
50
51 // .... oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ........ oooOO0OOooo ......
```

Listing A.7: Content of `pi0SpecPrimaryGeneratorAction.cc`

```
1 #----------------------------------------------------------------------------
2 # Setup the project
3 cmake_minimum_required(VERSION 3.8...3.18)
4 if(${CMAKE_VERSION} VERSION_LESS 3.12)
```

```cmake
 5    cmake_policy(VERSION ${CMAKE_MAJOR_VERSION}.${CMAKE_MINOR_VERSION})
 6  endif()
 7  project(Pi0Spectrometer)
 8
 9  #----------------------------------------------------------------------------
10  # Find Geant4 package, activating all available UI and Vis drivers by default
11  # You can set WITH_GEANT4_UIVIS to OFF via the command line or ccmake/cmake-gui
12  # to build a batch mode only executable
13  #
14  option(WITH_GEANT4_UIVIS "Build example with Geant4 UI and Vis drivers" ON)
15  if(WITH_GEANT4_UIVIS)
16    find_package(Geant4 REQUIRED ui_all vis_all)
17  else()
18    find_package(Geant4 REQUIRED)
19  endif()
20
21  #----------------------------------------------------------------------------
22  # Setup Geant4 include directories and compile definitions
23  # Setup include directory for this project
24  #
25  include(${Geant4_USE_FILE})
26
27  #----------------------------------------------------------------------------
28  # Locate sources and headers for this project
29  # NB: headers are included so they will show up in IDEs
30  #
31  include_directories(${CMAKE_CURRENT_SOURCE_DIR}/include
32                      ${Geant4_INCLUDE_DIR})
33  file(GLOB sources ${PROJECT_SOURCE_DIR}/src/*.cc)
34  file(GLOB headers ${PROJECT_SOURCE_DIR}/include/*.hh)
35
36  #----------------------------------------------------------------------------
37  # Add the executable, and link it to the Geant4 libraries
38  #
39  add_executable(pi0Spec pi0Spec.cc ${sources} ${headers})
40  target_link_libraries(pi0Spec ${Geant4_LIBRARIES})
41
42  #----------------------------------------------------------------------------
43  # Copy all scripts to the build directory, i.e. the directory in which we
44  # build B5. This is so that we can run the executable directly because it
45  # relies on these scripts being in the current working directory.
46  #
47  set(PI0SPEC_SCRIPTS
48    gui.mac
49    init.mac
50    init_vis.mac
51    vis.mac
52    run.mac
53    )
54
```

```
55  foreach( _script ${PI0SPEC_SCRIPTS})
56    configure_file(
57      ${PROJECT_SOURCE_DIR}/${_script}
58      ${PROJECT_BINARY_DIR}/${_script}
59      COPYONLY
60      )
61  endforeach()
62
63  #---------------------------------------------------------------------------------
64  # Add program to the project targets
65  # (this avoids the need of typing the program name after make)
66  #
67  add_custom_target(Pi0Spectrometer DEPENDS pi0Spec)
68
69  #---------------------------------------------------------------------------------
70  # Install the executable to 'bin' directory under CMAKE_INSTALL_PREFIX
71  #
72  install(TARGETS pi0Spec DESTINATION bin)
```

Listing A.8: Content of `CMakeLists.txt`

```
1  # -------------------------------------------------------------------------
2  # GNUmakefile for pi0 Spectrometer.   Thomas Baer, 02/07/21.
3  # -------------------------------------------------------------------------
4
5  name := pi0Spec
6  G4TARGET := $(name)
7  G4EXLIB := true
8
9  ifndef G4INSTALL
10    G4INSTALL = ../../..
11  endif
12
13  .PHONY: all
14  all: lib bin
15
16  include $(G4INSTALL)/config/binmake.gmk
17
18  visclean:
19    rm -f g4*.prim g4*.eps g4*.wrl
20    rm -f .DAWN_*
```

Listing A.9: Content of `GNUmakefile`

```
1  # Macro file for the initialization of example B1
2  # when running in interactive mode without visualization
3  #
4  # Set some default verbose
5  /control/verbose 2
6  /control/saveHistory
7  /run/verbose 2
```

```
 8 #
 9 # Change  the  default  number  of  threads  (in  multi−threaded  mode)
10 #/run/numberOfThreads 4
11 #
12 # Initialize  kernel
13 /run/initialize
14 # Set  up  population  count  scoring
15 /score/create/realWorldLogVol rightDetectorArmLogical
16 /score/quantity/population gamma
17 /score/close
18 #
19 /score/create/realWorldLogVol leftDetectorArmLogical
20 /score/quantity/population gamma
21 /score/close
22 #
23 # Start  Run
24 /pi0Spec/detector/armAngle 25
25 /control/loop run.mac num 1 1000
26 /pi0Spec/detector/armAngle 30
27 /control/loop run.mac num 1 1000
28 /pi0Spec/detector/armAngle 35
29 /control/loop run.mac num 1 1000
30 /pi0Spec/detector/armAngle 40
31 /control/loop run.mac num 1 1000
32 /pi0Spec/detector/armAngle 45
33 /control/loop run.mac num 1 1000
34 /pi0Spec/detector/armAngle 50
35 /control/loop run.mac num 1 1000
36 /pi0Spec/detector/armAngle 55
37 /control/loop run.mac num 1 1000
38 /pi0Spec/detector/armAngle 60
39 /control/loop run.mac num 1 1000
40 /pi0Spec/detector/armAngle 65
41 /control/loop run.mac num 1 1000
42 /pi0Spec/detector/armAngle 70
43 /control/loop run.mac num 1 1000
44 /pi0Spec/detector/armAngle 75
45 /control/loop run.mac num 1 1000
46 /pi0Spec/detector/armAngle 80
47 /control/loop run.mac num 1 1000
```

Listing A.10: Content of `init.mac`

```
1 # Macro  file  for  the  initialization  phase  of  pi0  Spectrometer
2 # when  running  in  interactive  mode  with  visualization
3 #
4 # Set  some  default  verbose
5 #
6 /control/verbose 2
7 /control/saveHistory
8 /run/verbose 2
```

```
 9  #
10  # Change the default number of threads (in multi-threaded mode)
11  #/run/numberOfThreads 4
12  #
13  # Initialize kernel
14  /run/initialize
15  #
16  # Visualization setting
17  /control/execute vis.mac
18  #
19  # Set up population count scoring
20  /score/create/realWorldLogVol rightDetectorArmLogical
21  /score/quantity/population gamma
22  /score/close
23  #
24  /score/create/realWorldLogVol leftDetectorArmLogical
25  /score/quantity/population gamma
26  /score/close
```

Listing A.11: Content of `init_vis.mac`

```
 1  //main program of first draft of pi0 Spectrometer
 2
 3  //Mandatory User Initialization Classes
 4  #include "pi0SpecDetectorConstruction.hh"
 5  #include "pi0SpecActionInitialization.hh"
 6
 7  //Include the Run manager
 8  #include "G4RunManagerFactory.hh"
 9  #include "G4ScoringManager.hh"
10
11  //Include the UI manager
12  #include "G4UImanager.hh"
13
14  //Include the physics lists
15  #include "FTFP_BERT.hh"
16  #include "G4EmStandardPhysics_option4.hh"
17  #include "G4OpticalPhysics.hh"
18  #include "G4StepLimiterPhysics.hh"
19
20  //Include UI and Visualization intercoms
21  #include "G4VisExecutive.hh"
22  #include "G4UIExecutive.hh"
23
24  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo........oooOO0OOooo......
25
26  int main(int argc, char** argv)
27  {
28      // Detect interactive mode (if no arguments) and define UI session
29      //
30      G4UIExecutive* ui = 0;
```

```cpp
31    if ( argc == 1 ) {
32      ui = new G4UIExecutive(argc, argv);
33    }
34
35    // Construct the default run manager
36    //
37    auto* runManager =
38      G4RunManagerFactory::CreateRunManager(G4RunManagerType::Default);
39
40    // Activate command-based scorer
41    G4ScoringManager::GetScoringManager();
42
43    // Mandatory user initialization classes
44    runManager->SetUserInitialization(new pi0SpecDetectorConstruction);
45
46    //Physics List
47    G4VModularPhysicsList* physicsList = new FTFP_BERT;
48    physicsList->ReplacePhysics(new G4EmStandardPhysics_option4());
49    G4OpticalPhysics* opticalPhysics = new G4OpticalPhysics();
50    physicsList->RegisterPhysics(opticalPhysics);
51    physicsList->RegisterPhysics(new G4StepLimiterPhysics());
52    runManager->SetUserInitialization(physicsList);
53
54    // User action initialization
55    runManager->SetUserInitialization(new pi0SpecActionInitialization());
56
57    // Visualization manager construction
58    auto visManager = new G4VisExecutive;
59    // G4VisExecutive can take a verbosity argument - see /vis/verbose guidance.
60    // G4VisManager* visManager = new G4VisExecutive("Quiet");
61    visManager->Initialize();
62
63    // Get the pointer to the User Interface manager
64    auto UImanager = G4UImanager::GetUIpointer();
65
66    if ( !ui ) {
67      // execute an argument macro file if exist
68      G4String command = "/control/execute ";
69      G4String fileName = argv[1];
70      UImanager->ApplyCommand(command+fileName);
71    }
72    else {
73      UImanager->ApplyCommand("/control/execute init_vis.mac");
74      if (ui->IsGUI()) {
75          UImanager->ApplyCommand("/control/execute gui.mac");
76      }
77      // start interactive session
78      ui->SessionStart();
79      delete ui;
80    }
```

```
81
82    // Job termination
83    // Free the store: user actions, physics_list and detector_description are
84    // owned and deleted by the run manager, so they should not be deleted
85    // in the main() program !
86
87    delete visManager;
88    delete runManager;
89 }
90
91 //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo........oooOO0OOooo......
```

Listing A.12: Content of `pi0spec.cc`

```
1 #Macro file for turning on the beam and simulating a single run.
2 /run/beamOn 1
3 #Dump data into temporary file
4 /score/dumpAllQuantitiesToFile rightDetectorArmLogical rightDataTmp.csv
5 /score/dumpAllQuantitiesToFile leftDetectorArmLogical leftDataTmp.csv
6 #Append temporary data onto main file
7 /control/shell cat rightDataTmp.csv >> rightData.csv
8 /control/shell cat leftDataTmp.csv >> leftData.csv
```

Listing A.13: Content of `run.mac`

```
1 #
2 # Macro file for pi0Spec
3 #
4 # Set verbosity
5 #
6 /control/verbose 2
7 #
8 # Use this open statement to create an OpenGL view:
9 /vis/open OGL 600x600−0+0
10 #
11 # Disable auto refresh and quieten vis messages whilst scene and
12 # trajectories are established:
13 /vis/viewer/set/autoRefresh false
14 /vis/verbose errors
15 #
16 # Draw geometry:
17 /vis/drawVolume
18 #
19 # Set visualization to accumulate events into one picture
20 /vis/scene/endOfEventAction accumulate
```

Listing A.14: Content of `vis.mac`

Bibliography

[1] Louis Rosen. *LAMPF - Its Origin, History, and Accomplishments*. Kyoto, Japan, August 1985.

[2] David Griffiths. *Introduction to Elementary Particles*. 2nd, revised. 2008.

[3] C. F. von Weizsäcker. "Zur Theorie der Kernmassen. (German) [on the theory of nuclear masses]". In: *Zeitschrift für Physik* 90 (1935), pp. 431–458.

[4] Burman R L. and Fulton R.L. "Designing of The LAMPF Low-Energy Pion Channel". In: *Nuclear Instruments and Methods* 131 (1975), pp. 29–38.

[5] H. W. Baer, R. D. Bolton, and J. D. Bowman. "Design, Construction, and Performance of A High-Resolution π^0 Spectrometer for Nuclear Physics Experiments". In: *Nuclear Instruments and Methods* 180 (1981), pp. 445–459.

[6] H. W. Baer. *Personal Notes and Correspondence*. 1975-1991.

[7] Bowman et al. J. D. "A High Resolution π^0 Spectrometer for Nuclear Physics Studies". In: *IEEE Transactions on Nuclear Science* NS-25.1 (1978), pp. 344–346.

[8] H. W. Baer and J. D. Bowman. *LAMPF Progress Report*. 1981.

[9] Jonas Alster. *Isovector Excitations in Nuclei and Other New Results from the π^0-Spectrometer*. 1981.

[10] Geant4 Collaboration. *Book For Application Developers*. URL: https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/html/index.html#book-for-application-developers.

[11] Makoto Asai. *Scoring I*. URL: https://www.slac.stanford.edu/xorg/geant4/SLACTutorial14/Scoring1.pdf.

[12] Makoto Asai. *Scoring Functionalities*. URL: https://indico.cern.ch/event/746466/contributions/3345729/attachments/1833373/3003047/Scoring.pdf.

[13] T. Mast. private communication. Berkeley, 1975.

[14] Schott AG. *LF5 Data Sheet*. URL: https://www.spacematdb.com/spacemat/manudatasheets/schott-datasheet-lf5.

[15] Schott AG. *SF5 Data Sheet*. URL: https://www.sydor.com/wp-content/uploads/2019/06/SCHOTT-N-SF5-Optical-Glass.pdf.

[16] Geant4 Collaboration. *Overview*. URL: https://geant4.web.cern.ch.

[17] Geant4 Collaboration. *Collaboration*. URL: https://geant4.web.cern.ch/collaboration.

[18] Geant4 Collaboration. *User Documentation*. URL: https://geant4.web.cern.ch/support/user_documentation.

[19] Geant4 Collaboration. *Installation Guide*. URL: https://geant4-userdoc.web.cern.ch/UsersGuides/InstallationGuide/html/index.html.

[20] Geant4 Collaboration. *Material Database*. URL: https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/html/Appendix/materialNames.html.

[21] Beamline for Schools. *Project technical report*. Tech. rep. URL: https://twiki.cern.ch/twiki/pub/BL4S/BL4SDataAnalysis/technicalreport2015.pdf.

[22] Andreas Schälicke. *Optical Physics*. URL: https://indico.cern.ch/event/58317/contributions/2047457/attachments/992304/1411067/OpticalPhysics.pdf.

[23] Jacob Colston. *A study of the radiative pion capture process as a background to the search for muon to electron conversion with the mu2e experiment*. 2016.