Development and Optimization of Simulation Software tools for the Coordinator Detector in Jefferson Lab's Hall A

by

Nandhu Sridhar

A Thesis Submitted to
Saint Mary's University, Halifax, Nova Scotia
in Partial Fulfilment of the Requirements for
the Degree of Master of Science in Applied Science

May 3, 2024, Halifax, Nova Scotia

Approved: Dr. Adam Sarty
Co-supervisor

Approved: Dr. Edward Brash
Christopher Newport University
Co-supervisor

Approved: Dr. David Hornidge
Mount Allison University
External Examiner

Approved: Dr. Rituparna Kanungo
Examiner

Approved: Dr. Robert Singer
Examiner

Date: April 29, 2024

Abstract

Development and Optimization of Simulation Software tools for the Coordinator Detector in Jefferson Lab's Hall A

by *Nandhu Sridhar*

submitted on May 3, 2024:

The Coordinate Detector is an arrangement of 2352 plastic scintillators that detects recoil electrons, or other charged particles, originating from collisions of a high-energy electron beam on a hydrogen target within Hall A of Jefferson Lab in Newport News, Virginia, USA. Exploiting the Coordinate Detector's arrangement of scintillator detectors to study nucleon structure through electron-scattering requires modelling the response of the entire detector to the desired scattered particles, alongside other charged particle sources. This thesis details the optimization of existing simulation software for the Coordinate Detector, focusing on construction, data generation and analysis algorithms, including new algorithms. Four projects were undertaken: expanding the simulation from one subunit to the entire detector; optimizing and expanding the scope of simulated particle generation; using scintillator timing information to determine the horizontal angle of incident particles; and, building an event display to provide an intuitive view of the particle hit-patterns across the detector.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Physics Motivation

In the fields of nuclear and particle physics, one of the primary areas of interest is the structure of nucleons (protons and neutrons). As nucleons are generally very stable particles, when found in stable nuclei, the principal experimental technique for studying their internal structure is the scattering of a beam of medium to high energy probe particles from target nuclei and then observing the results of this interaction; that is, we detect either the scattered incident probe particle, or scattered target remnants, or both. Because they interact weakly with target nuclei, and because they are, according to our current knowledge, point particles, electrons are an ideal probe particle for this purpose. Electrons have no internal structure of their own, and so interpretation of the scattering data is more straightforward, in this sense. Moreover, electrons interact with the target nuclei electromagnetically, and this interaction is extremely well understood within the context of Quantum ElectroDynamics (QED). To put it simply, we can use a probe which is extremely well understood to interact with target nuclei via an interaction which is extremely well understood to gain insight into something that is currently not as well understood - the structure of the nucleons themselves.

As electrons interact electromagnetically, they inherently probe the charge and current distributions within nucleons. These internal charge and current distributions arise, at least primarily, from the moving charged quarks inside of protons and neutrons. At the same time, the energy of the electron probe particles determines the distance scale at which we are able to probe the nucleons; as the energy of the electron increases, we are able probe smaller and smaller distance scales within the nucleon. While the large distance structure of protons and neutrons is relatively well understood, a full understanding of the small distance scale structure remains a topic of great current interest.

When an electron with high energy (on the order of 1 GeV or higher) interacts with a nucleon (within a target nucleus), the most likely scenario is that the nucleon does not remain intact. That is, additional scattered particles are created in a (potentially quite complicated) inelastic process. However, very occasionally, the target nucleon scatters elastically (and no additional scattered particles are produced). Because the electron interacts (at first order) with only a single quark within the target nucleon and transfers a large amount of energy and momentum to that quark, it must therefore follow that in elastic scattering there must be, in turn, a redistribution or subsequent ”sharing” of this energy and momentum amongst the other valence quarks in the nucleon, in order for the nucleon to remain intact. This occurs though secondary gluon exchange between the quarks. Thus, while at first glance it may appear that elastic scattering is a ”simple” interaction, when viewed in this way it becomes clear that it is an interesting and informative reaction for understanding the strong interaction between quarks inside the nucleon. Moreover, these rare elastic collisions are of particular interest because it turns out that there are a wealth of solid and fundamental theoretical predictions that we can compare the data to in order to gain further insight.

1.2 Elastic Nucleon Form Factors

For the case of elastic scattering of electrons from a nucleon target, we can use fundamental symmetry arguments to show that all of the nucleon structure information is contained within a set of mathematical functions called elastic nucleon Electromagnetic Form Factors (EMFFs)[5]. These form factors embody all of the details of the spatial distributions of charge and magnetization in a nucleon. As noted above, the primary charge carriers within a nucleon are quarks, and thus knowledge of the form factors of a nucleon can provide significant insight into a nucleon's quark substructure[5]. The quark substructure of nucleons is still not well understood, especially at small distance scales, and further investigation into nucleon EMFFs would be beneficial for this field of research[7, 28]. In the non-relativistic picture, nucleon Form Factors are Fourier Transforms of the charge/magnetization distribution they represent[17].

Again, for the case of elastic scattering where the incident electron interacts via the exchange of a single virtual photon, there are two nucleon form factors, $\mathrm{G_E(Q^2)}$ (electric form factor) and $\mathrm{G_M(Q^2)}$ (magnetic form factor), each of which is a function of the four-momentum (Q^2) that is transferred from the electron to the nucleon in the scattering interaction. Further, each different type of nucleon will have a distinct $\mathrm{G_E(Q^2)}$ and $\mathrm{G_M(Q^2)}$. These nucleon EMFFs can, in principle, be measured through elastic electron-nucleon scattering. Most importantly for our purposes, by varying the kinematics of the scattering, or by using/measuring the polarization of the incoming electrons and/or of the scattered protons, one can measure the dependence of the form factors on Q^2 [18].

There are two types of nucleons — protons and neutrons — and thus we have, in fact, four distinct elastic EMFFs: $\mathrm{G_E^p(Q^2)}$, $\mathrm{G_E^n(Q^2)}$, $\mathrm{G_M^p(Q^2)}$ and $\mathrm{G_M^n(Q^2)}$ [5]. An extremely important point is that through the measurement of $all\ four$ EMFFs (in a series of different experiments), it could be possible to analyze flavour separation in Quantum Chromodynamics (QCD) [12].

A wealth of testable theoretical predictions exist for the Q^2 evolution of the nucleon form factors. These calculations are carried out using the framework of the strong interaction (QCD) either in traditional QCD-inspired models or more recently using the more fundamental but admittedly more computationally intensive methods of Lattice QCD. In particular, it is the large Q^2 behaviour of the form factors that is most interesting, in the sense that it is in this region where models begin to diverge from one another, owing to the lack of previous experimental data to constrain the models in this region. The main point is that testing these theoretical predictions helps test our understanding of the dynamics and symmetry properties of QCD, as well as our understanding of the nature and role of gluons in nucleon structure [20].

Previous experiments (at Jefferson Lab, SLAC, Mainz, MIT-Bates, for example) measuring EMFFs at low to intermediate Q^2 have enabled further understanding and analysis of nucleon quark structure; in particular, data collected on $\mathrm{G_E^p}$, $\mathrm{G_E^n}$, $\mathrm{G_M^p}$ and $\mathrm{G_M^n}$ in Hall A at

Jefferson Lab have allowed researchers to analyze the up and down quark contributions to the elastic nucleon form factors [9].

As noted above, elastic electron-nucleon scattering at large momentum transfer is a rare process (meaning, the process has a low cross-section, thus a low probability). Thus, there are a number of background processes that are far more probable, and thus can make measurements of elastic scattering more difficult. These processes include elastic electron-electron interactions and processes from other (inelastic) electron-nucleon or electron-nucleus interactions.

The competing electron-nucleon interactions are Bremsstrahlung processes and inelastic meson production, both of which have cross sections similar to elastic nucleon-electron scattering at these energies[16]. One of the central and crucial experimental requirements in the form factor experiments at Jefferson Lab is the identification and isolation of elastic scattering events from within a large background of inelastic events. As we will see in the following sections, we have a number of techniques that we can employ to aid us in this effort.

1.3 Jefferson Lab

To measure the elastic form factors at the momentum transfers of interest ($Q^2 \approx 1 - 10\ GeV^2$), there are two principal requirements for the electron beam. The first is that it be of sufficient energy, and the second that it be of sufficient intensity (as again, we are dealing with a rare process)[15]. Electron accelerators such as the Continuous Electron Beam Accelerator Facility (CEBAF) accelerator at Jefferson Lab (JLab) [13] and the Mainz Microtron (MAMI) [19] accelerator provide the facilities to make measurements at high energies and with a beam of sufficient intensity to make measurements feasible.

JLab is the premiere facility in the world for these measurements, due to having the highest energy electron beam amongst comparable institutions [26] and being able to provide luminosities up to 10^{38} $cm^{-2}s^{-1}$ for liquid hydrogen crytargets in Hall A (See Figure 1.1). Jefferson Lab also has the ability to produce a *polarized* electron beam with sufficient degree of polarization to be able to isolate the electric and magnetic form factors from one another. This is accomplished by utilizing longitudinally polarized electron beams and either polarized nucleon targets or measuring the polarization of scattered recoil nucleons [5].

The primary end-station for scattering measurements for nucleon form factors at large Q^2 at JLab is Hall A [2] in which the experiments proposed by the Hall A Collaboration take place.

1.4 The GEp and GMn Experiments at JLab

Two of the principal experiments in Hall A at Jefferson Lab whose aim is to further our understanding of nucleon structure through the measurement of elastic electromagnetic form factors are the so-called G_{Ep} and G_{Mn} experiments (JLab experiment numbers E12-07-109 [28] and E12-09-019 [7], respectively).

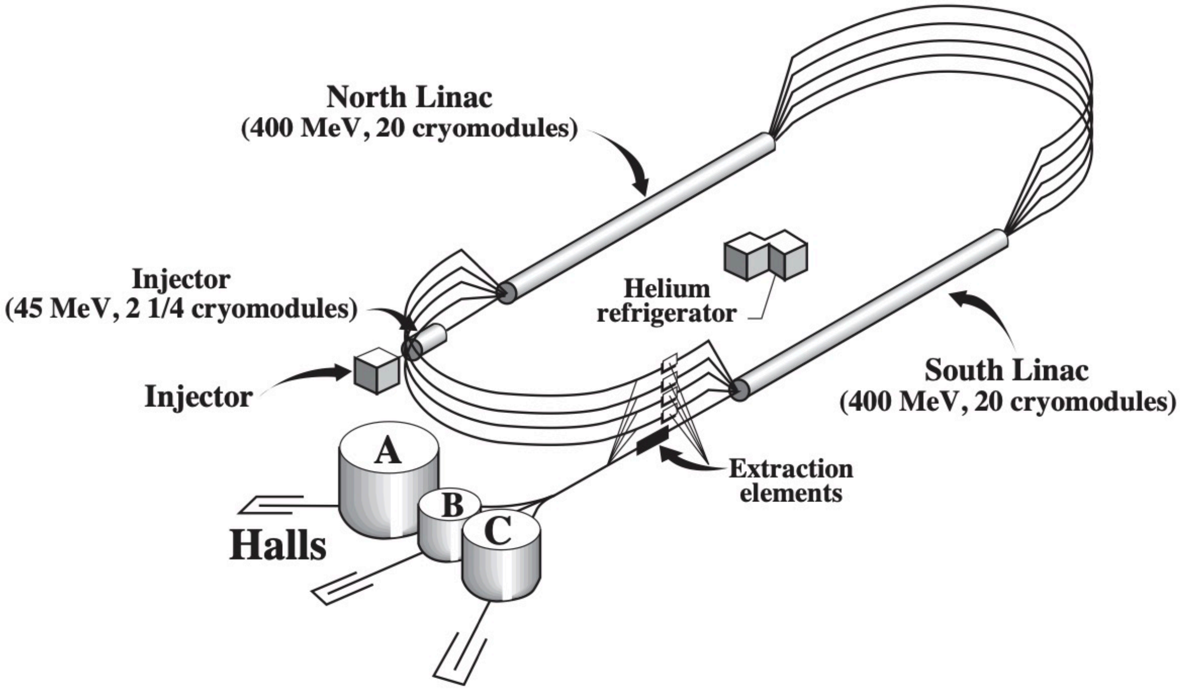

Figure 1.1: An Overview Schematic of the CEBAF Accelerator Facility at Jefferson Lab (Reprinted from [2])

These experiments, along with several others, will employ a newly-built spectrometer/detector system, known as the Super BigBite Spectrometer (SBS) to detect elastically scattered electrons in coincidence with either a scattered proton (for G_{Ep}) or a scattered neutron (for G_{Mn}). Unlike most spectrometers, SBS is modular and can be rearranged for different experiments. It is a high-resolution spectrometer with a large acceptance. SBS is designed for an experimental program that will use the JLab 12 GeV beam to explore elastic scattering at large four-momentum-transfer-squared (Q^2), a region in which most processes are deep inelastic scattering. This means the rates for elastic scattering are very low, so the large acceptance is required in order to increase the number of detected events.

SBS has two detector "arms". One arm detects hadrons, and the other detects electrons. There are two main configurations of the arms: one for experiments which will detect

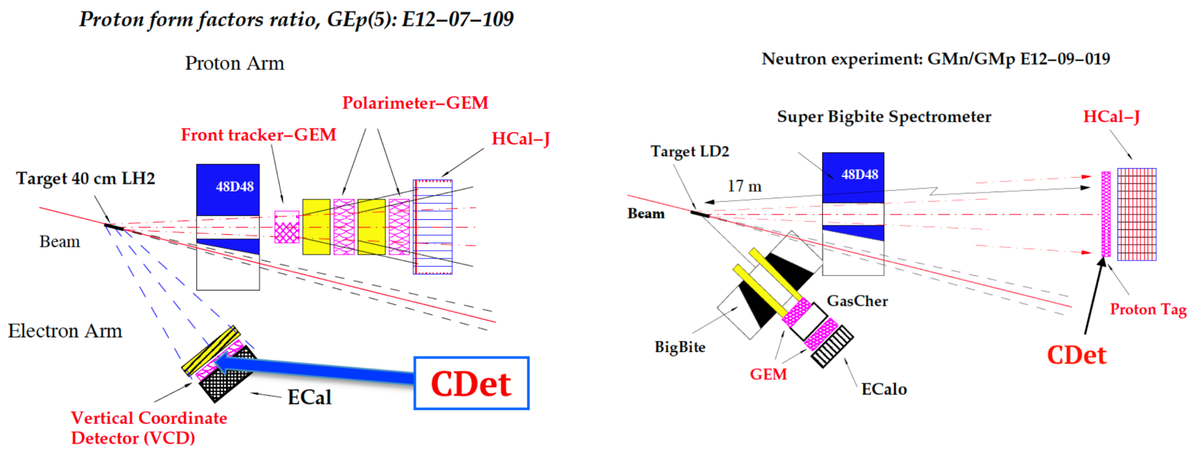

Figure 1.2: Schematic Setup for the G_{Ep} (left) and G_{Mn} (right) Experiments at JLab

scattered protons, and one for experiments which will detect scattered neutrons. In both configurations, there is a large dipole magnet at the front of the hadron arm and a hadron calorimeter at the end of the hadron arm. There is an electromagnetic calorimeter at the end of the electron arm. In the proton configuration (shown in the left panel of Fig. 1.2), a tracker and polarimeter setup is placed between the magnet and the hadron calorimeter. A supplemental coordinate detector (CDet) is placed in front of the electromagnetic calorimeter. In the neutron configuration (shown in the right panel of Fig. 1.2), only the coordinate detector is placed in front of the hadron calorimeter. On the electron arm, the smaller BigBite spectrometer is placed in front of the electromagnetic calorimeter along with additional trackers.

1.5 Observing Scattered Particles

The two-body scattering process requires that both recoil particles, which in the case of electron-nucleon scattering are electrons and either protons or neutrons, need to be measured simultaneously at a high precision (meaning, precise track/angle and momentum measurements of the particles). This has to be done to isolate elastic electron scattering from more probable reactions, as some reactions are inelastic and can create side products, such as pions, or contaminate the measurement in various other ways [24]. The ability to use double-polarization techniques to further reduce error from two-photon production mechanisms in the interaction with the nucleon (as opposed to the assumed simpler direct interaction referred to earlier) help increase the accuracy of extracting nucleon form factors further [4].

It is important to ensure that the energy of both the incident and scattered particles, the angle of both scattered particles, and potentially the polarization of the target and/or scattered nucleons, can be measured to a reasonably high precision. An example of this could include HRS (the High Resolution Spectrometer) in Hall A to detect scattered particles, achieving momentum measurement precision to a few parts in ten thousand, and being able to measure angle within 1 mr [2]. Using a combination of various different types of radiation detectors assembled in layers, while also using magnets to bend scattered particles according to momentum (thus sorting them) can provide the best way of extracting the most information out of any scattering, as this type of configuration allows one to use the strengths of multiple types of detectors to the best of their ability.

Admittedly, the momentum resolution for SBS in Hall A (either for electrons or nucleons) is significantly worse that for the HRS spectrometer mentioned above. One must make a trade-off, typically, between momentum resolution and angular acceptance. In the design of SBS, the need for large acceptance outweighed the need for momentum resolution. At the same time, we still need to identify elastic scattering events in many SBS experiments, and thus we must use the capabilities of other detectors within the SBS detector stack to achieve this.

1.6 The Coordinate Detector (CDet)

The main underlying physics principle that is used to identify elastic scattering events is the angular correlation between the scattered electron and nucleon, which exists because the scattering is a two-body reaction. Thus, having a detector with SBS that has excellent position resolution (which can then be used to determine scattering angles) is crucial.

The coordinate detector (CDet) is a multipurpose scintillator-based detector. In proton experiments, it serves as a supplemental tracking detector for charged particles (electrons). It is constructed with stacks of narrow plastic scintillator paddles to detect the vertical (z) coordinate of a scattered particle. The vertical position then provides the out-of-plane angle of the particle's trajectory with excellent angular resolution. In neutron experiments, CDet serves as a charged particle veto to distinguish neutrons from other charged particles. The position of CDet within the SBS system is highlighted in Fig. 1.2 above.

1.7 Scintillation

As noted, CDet is a scintillator-based detector. Scintillation is a process in which a small flash of light is emitted shortly after an ionizing particle passes through a scintillating, semi-transparent material. Although it appears superficially similar to Cerenkov radiation, scintillation is more similar to fluorescence. Unlike fluorescence, however, which excites atoms by absorbing a quantum of light and then de-excites to release light, scintillation causes the excitation of atoms through the passage of light, rather than through absorption. The atoms then de-excite promptly, releasing light [8].

Electrons moving through matter lose energy through multiple means, with inelastic collisions with electrons and nuclei having the largest cross section, with Bremsstrahlung being significant at higher energies. At high energies, energy lost due to Bremsstrahlung is proportional to Z^2[16].

These collisions cause the electron to be deflected in lower energies, and would result in a path significantly larger than a straight line. However, at higher energies, these deflections have little impact on the path of the electron. Path lengths are also proportional to the Z of the scintillator medium.

These inelastic collisions, primary with electrons, results in ionization and excitation of atoms in the scintillation medium.

The primary mechanism for the emission of light following excitation is fluorescence. Fluorescence occurs when an excited electron in the S_1 shell deexcites to the S_0 ground state. This radiative transition has a lifetime of approximately 10^{-9} seconds to 10^{-8} seconds, which is quite rapid in experimental terms[8].

The prompt response of a scintillating material to ionising radiation makes scintillation especially useful for precise timing. Using some of the same techniques that are used in making a Cerenkov counter (Total internal reflection, photo-multiplier tube signal amplification) enables a Scintillator counter (See Section 1.7.2) to be made for the purposes of timing and producing energy/path information [21].

1.7.1 Scintillators

A scintillator is a material from which scintillation can be detected. To make a scintillator, it is important to note that the material must be transparent in order for the scintillation light to reach a detector. If the material is opaque, the light would be absorbed by the material, and would make it difficult to measure scintillation. Additionally, the incident ray would lose significant amounts of energy and may be slowed down completely, limiting the ability of the material to detect the total energy of the particle and its ability to be further analysed by other downstream detectors.

Secondly, materials made solely out of scintillating compounds would make it hard to detect the scintillation light as well, as the energy of the scintillation light corresponds exactly to the energy required to make an electron jump from the ground state to the excited state of the atom. As the concentration of such atoms would be very high, this means that the timing advantages of scintillators would be lost, as the information would have to travel through a series of excitations and de-excitations.

Therefore, most scintillators are built from a transparent material doped with scintillator compounds. As the medium is transparent and the concentration of scintillator material is relatively low, both of these risks can be mitigated.

Scintillators may be made out of multiple different materials, such as inorganic crystals, organic crystals and plastic scintillators.

1.7.2 Scintillation Counters

A traditional scintillation counter is a device used to detect and measure ionizing radiation. Counters typically consist of a scintillator and a Photo-Multiplier Tube (PMT) that measures the light amount of light produced in the scintillation process. The light is typically transported to the PMT through a clear plastic lightguide. As the scintillation light generated in the scintillation process is often of a different wavelength than what the PMT is most sensitive to, sometimes Wavelength Shifters (WLS) are used to shift the wavelength of the scintillation light prior to transporting the light to the PMT. Wavelength shifting is a complex process that involves absorption of the initially produced scintillation light, followed by re-emission at a different wavelength.

The primary mechanism to ensure the scintillation light does not escape the system is to have an air gap outside the scintillator. As air has a lower and a significantly different refractive index from the scintillator (1.0 compared to about 1.6), this induces Total Internal Reflection (TIR) in the scintillator, preventing the light from leaving the system. Additionally, a reflective material (e.g. aluminum foil) is often used to wrap the scintillator outside of the air gap, reflecting back any of the light that does escape. In addition, this reflective wrapping acts as an opaque agent to prevent outside light from entering the scintillator (but note that all scintillator counters also have an additional black, opaque wrapping material

- such as black plastic, or electrical tape - around the reflective wrapping to ensure ”light tightness”).

Lightguides, if used, are attached to the scintillator from either one or both ends, without any air gap to prevent total internal reflection of the light back into the scintillator. These lightguides are often made out of materials of the same refractive index as the scintillator for this reason. Lightguides are designed to smoothly change shape to guide the internal scintillation light into the photocathode of the PMT, again without an air gap to prevent TIR.

1.8 Photo-Multiplier Tubes

PMTs are devices used to amplify the signal given by a faint light source into a signal strong enough to be picked up by an analog circuit. It is important that the PMT's response is proportional to the number of input photons. To accomplish this, a photocathode at the receptor end of the PMT releases electrons in response to incident photons via the photoelectric effect.

The electrons from the photocathode then travel to a dynode, which releases more electrons, in response to the incident electrons. These electrons then engage with a series of dynodes, repeating the process until a cascade of electrons is present at the other end of the PMT[16]. In this way, the final number of electrons produced, and thus the size of the electronic signal produced, is inherently proportional to the original number of incident photons impinging on the photocathode. The analog electronic signal is then integrated using

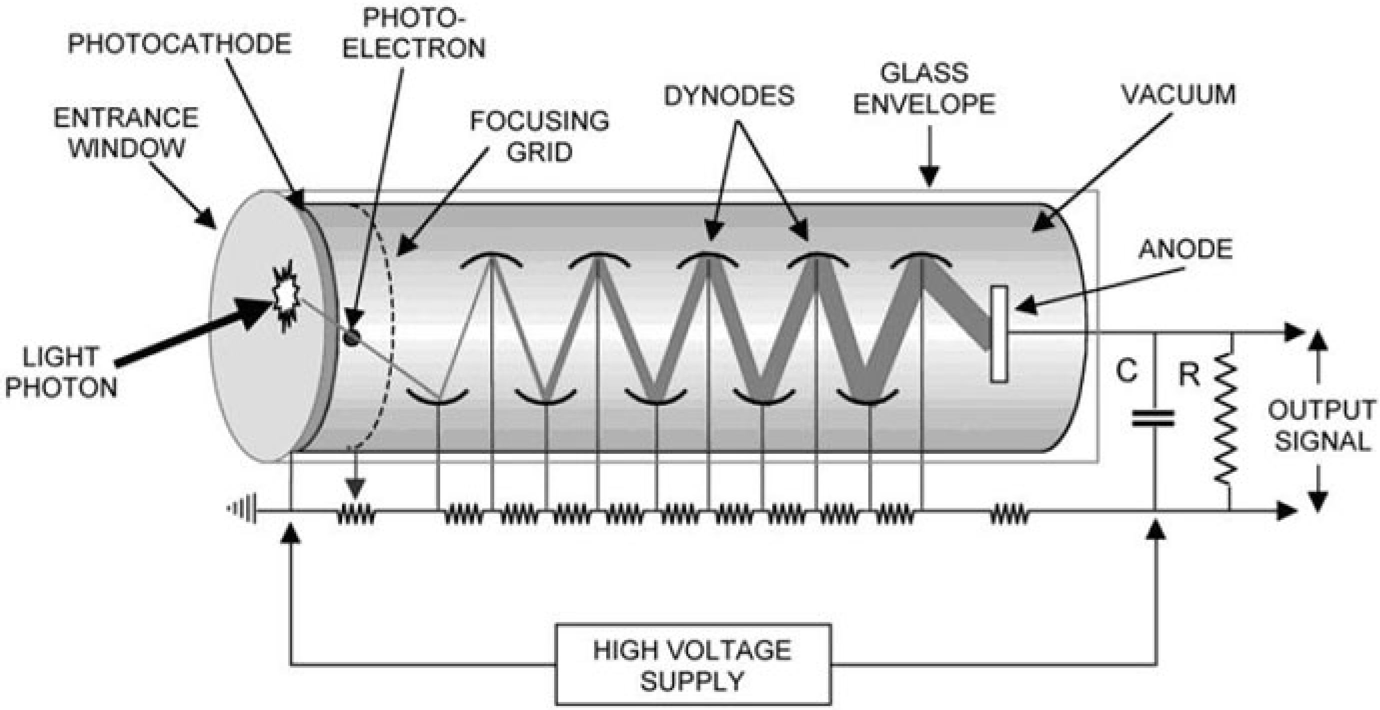

Figure 1.3: A representation of a Photo-Multiplier Tube. Reprinted from [22]

an Analog-Digital Converter (ADC) to finally produce a digital measure of the number of incident photons[16].

1.9 The Coordinate Detector (CDet) Structure

1.9.1 Material

The scintillators of CDet are made of extruded plastic scintillator material [3] with properties similar to BC 404 and BC 408 scintillators produced by Bicron [6] that have peak emission wavelengths between 408 and 425 nm. The index of refraction of these scintillators is 1.58.

Each individual plastic scintillator ”paddle” in CDet has a wavelength-shifting optical fibre inserted through its centre. These wavelength-shifting fibres are BCF-92 fast blue-green scintillators [25] which serve the dual purpose of serving as lightguides to direct generated scintillator light to the PMTs, as well as shifting the peak wavelength of scintillation light to match the sensitivity of PMTs. The paddles have a length of 50 cm to increase angular acceptance, a width of 0.5 cm to ensure high out-of-plane angle acceptance and a depth of 4 cm to ensure that relevant particles passing through are detected.

The PMT's are of the multi-anode variety, such that signals from up to 16 fibre-optic lightguides can be multiplexed to a single multi-anode PMT; note that CDet uses only 14 of these 16 PMT inputs for each ”bar” (defined in the next subsection), with the remaining 2 inputs being available for troubleshooting or repairs during experimental implementation.

1.9.2 Geometrical Structure

Paddle Arrangement

The coordinate detector is composed of two identical layers vertically stacked upon each other. Each layer is composed of three modules, each of which are composed of two symmetric half-modules. Half-modules consist of 14 bars, which are themselves composed of 14 scintillator paddles. This sums up to a total of 2352 paddles in CDet. Schematic diagrams of a CDet module, and a single ”bar” in CDet are shown in Fig. 1.4.

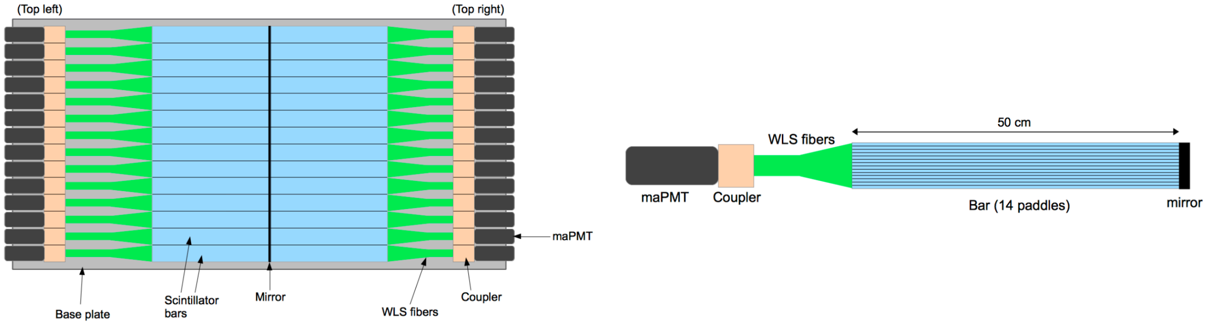

Figure 1.4: Schematic Diagram of a Single Module (left) and Single Bar (right) in CDet

1.9.3 Monte Carlo Simulation of CDet

G4SBS

The G4SBS simulation package is a group of simulations written in C++ using the GEANT4 framework [1] that simulates the entirety of the SBS system. It allows for the testing and development of analysis software prior to the deployment of the Detector, and to evaluate and optimize detector design and positioning to minimize the impact of expected high event rates in the system [11][14]. As one can imagine, G4SBS is a huge simulation package and is computationally intensive. As a result, an additional smaller ”standalone” GEANT4-based package, known as CDetMC, was developed to carry out focused studies of CDet.

1.9.4 Coordinate Detector Simulation (CDetMC)

Over the last several years, a Monte Carlo simulation has been built in GEANT4, and implements all of the materials, structure and physics explained in the previous sections.

An important part of any particle physics simulation is the event generator. That is, one must decide how to generate the initial ”primary” particles that one is going to simulate and track through the detectors being simulated. CDetMC uses output data from the G4SBS simulation as its event generator to generate input events, which are then tracked through CDet. The input data is in the form of a ROOT file[23] that contain the relevant information about the input particles, such as the position vector, directional derivative vector, particle type, and total energy of each particle. These events are then simulated, and an output ROOT file is produced by CDetMC for further analysis.

1.10 A description of the Following Chapters

The following chapters provide a detailed report on a series of projects that have been undertaken aimed at improving the CDetMC simulation and using that improved simulation to develop additional analysis algorithms and other tools useful for the SBS collaboration. Prospects for future developments are also discussed.

In Chapter 2, implementation of the full CDet detector within the CDetMC framework is discussed, together with a system that was developed to optimize simulation of large numbers of events. In Chapter 3, improvements to the CDetMC event generation scheme are presented and discussed. Chapter 4 discusses the implementation of algorithms designed to allow for determination of the in-plane hit position of incident particles. Chapter 5 will present details of a new event display tool that has been developed to aid in the simulation and analysis of noise hits (or ”background”) in CDet. Finally, in Chapter 6, conclusions and recommendations for future development are provided.

Chapter 2

Simulation of the Full CDet Detector

2.1 Background

Prior to this project, the CDetMC simulation was focused primarily on the simulation of an individual bar in CDet, with the aim of optimizing the simulation to reproduce cosmic ray data collected for recently constructed bars of the actual detector. However, when considering extending the simulation to include the full CDet detector configuration, the initial previously developed simulation was seen to be extremely inefficient, with unacceptably long processing times for each event that was simulated. This made it difficult to acquire a useful amount of simulated data from the entire simulated geometry of CDet.

Thus, this initial version of CDet simulation consisted solely of a single bar (14 paddles) with two smaller scintillation-based counters (labeled as finger counters) used to provide an event trigger (such as existed in the testing setup for the actual detector). This allowed the team to produce results in a reasonable amount of time, and allowed for the more rapid testing of optimized improvements. However, while this was useful for refining and optimizing the functioning of one bar in the simulation, it was less useful for fulfilling the simulation's goal of simulating the detector as a whole.

To simulate the full detector, it was necessary to optimize the simulation for one bar first (Section 2.2) to a reasonable degree, and then construct the full detector once the bar is optimized enough so that it becomes reasonable to simulate a much larger number of paddles (Section 2.3). As all the bars are identical in geometry and function, optimizations made to a single bar can be easily propagated to all.

2.2 Simulation Optimization for a Single Bar

A significant source of the excess processing leading to slow simulation speeds was the full simulation of optical photon production via direct simulation of the scintillation process. In the scintillation process (Section 1.7), a single high energy charged particle generates many photons, resulting in a significantly higher number of photons to be tracked in the simulation than any other type of particle. Obviously, photon generation is important in the simulation as ultimately they provide the signal required to detect particles in the scintillation counters. These photons are thus simulated in their entirety and recorded like any other particle after the event is complete. Therefore, the most potent method to obtain significant optimization improvements would be to optimize the handling of these photons.

Two methods were attempted to reduce the processing time associated with scintillation photons: not producing scintillation photons at all (and in turn modeling the number of photons produced via their energy deposition only) (Section 2.2.1) and separately not recording the full photon information in the output ROOT file (Section 2.2.2). The results arising from investigating these options are discussed in Section 2.4.1.

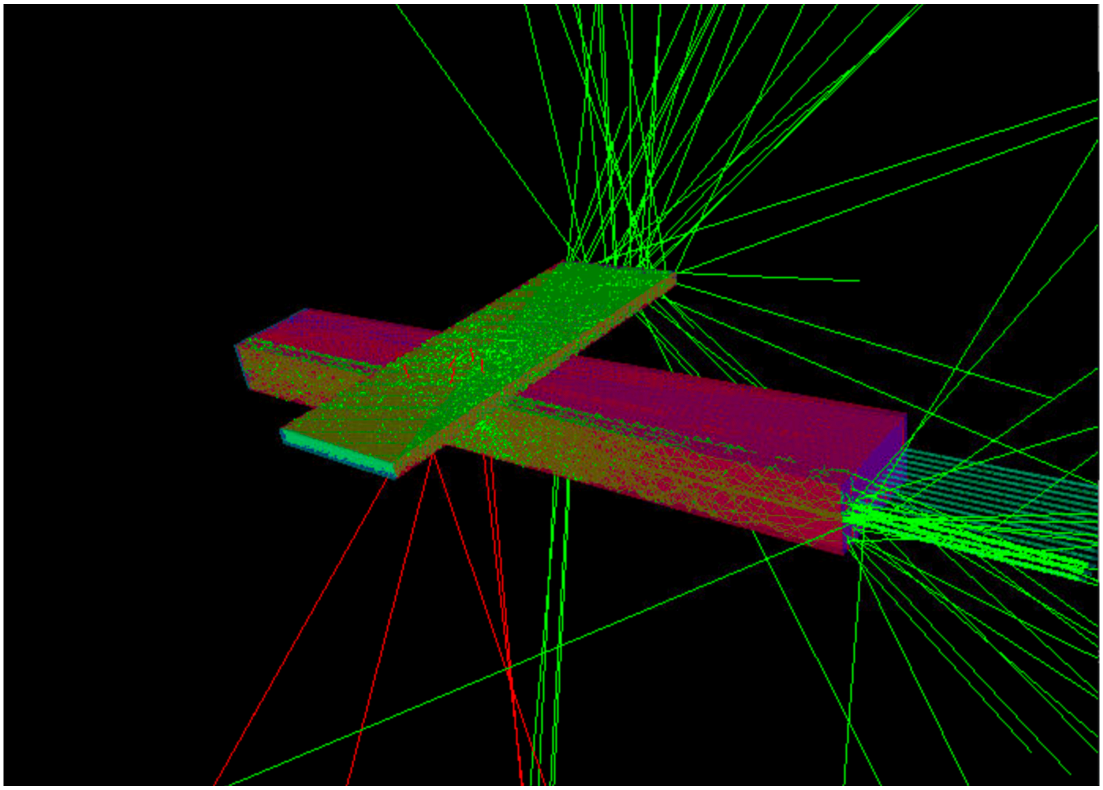

Figure 2.1: A GEANT4 visual representation of the Coordinate Detector simulation prior to improvements made that allowed all of the bars of the full detector to be simulated. In the image, there is a single CDet bar, containing 14 paddles, represented in red in the middle of the figure, along with a ”finger counter,” also a scintillator-based detector, above the CDet bar. When the PMT at the end of the finger counter detects scintillation, readout is triggered in the paddles of the CDet bar.

2.2.1 Modeling Photon Production via Energy Deposition

One method to reduce the inefficiency caused by fully simulating the large numbers of scintillation photons is to not generate them or their subsequent physics interactions altogether by using instead a model for photon generation based on the incident particle's energy deposition in the scintillating material of CDet.

As the incident particle moves through the scintillator material and deposits energy (through primarily electromagnetic interactions), optical photons are generated (Section 1.7). The rate of photon generation is approximately constant for every unit of distance the particle moves in the scintillator medium for minimum ionizing particles such as either cosmic rays or the particles that are produced in SBS experiments. As such, larger amounts of energy deposited results in proportionally more photons being generated. It is then expected that there is a direct correlation between energy deposition and photon generation. To understand this correlation, the simulation was run for many events using a single bar. The incident electrons were generated with a random distribution of incident angles and positions across the length of the bar, to ensure a variety of distances traversed by the electrons in a given paddle for different events; this makes it possible to create a phenomenological model of number of photons produced as a function of energy deposition. Moreover, we also simulated the quantum efficiency of the multi-anode PMT (maPMT), in order to produce photoelectrons in the maPMT photocathodes. The value of the energy deposited, E_{dep} and the number of photonelectrons generated, N_γ were recorded for each event. This plot was then fit with a least squares linear fit to find the constant of proportionality, k.

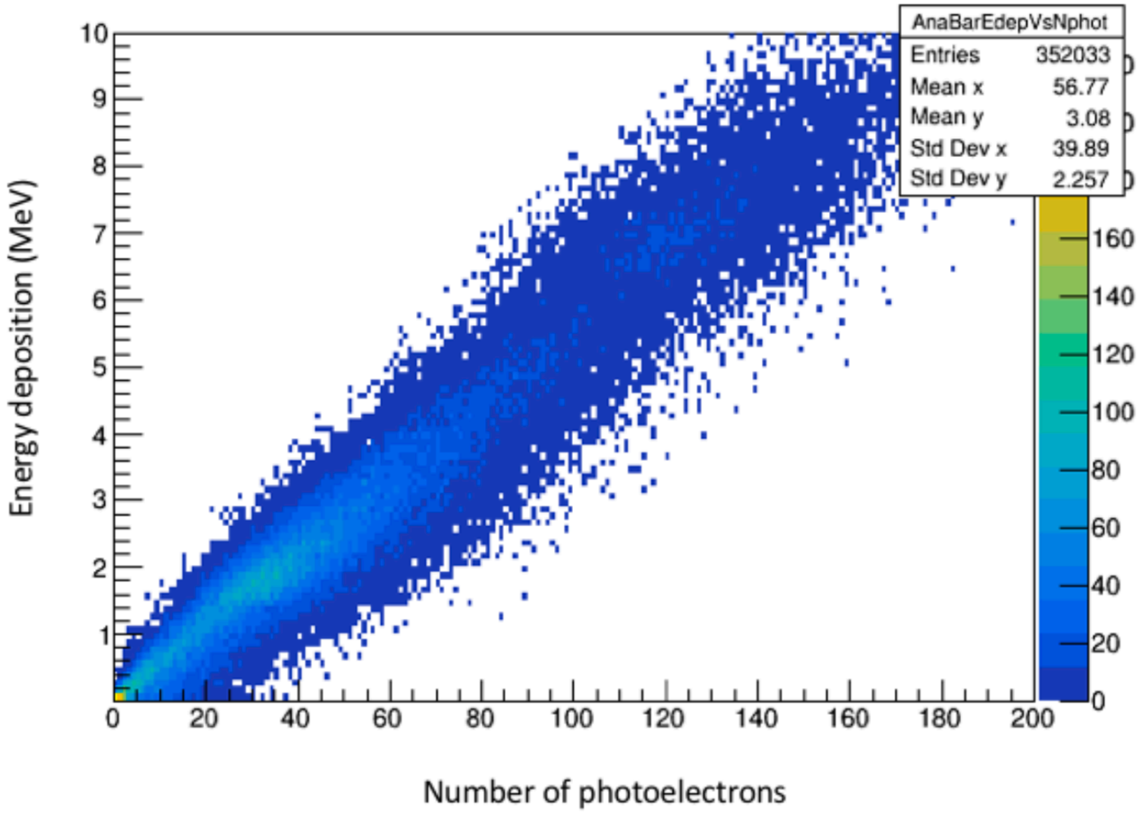

Figure 2.2: A graph representing the energy deposited, E_{dep} (in MeV) vs. the number of photoelectrons, N_γ, generated from this deposited energy. The constant of proportionality, k, is the inverse of the slope of this line.

The constant of proportionality, k was extracted from the slope of plotting E_{dep} against N_γ, as seen in Figure 2.2. k was found to be 22.0 MeV^{-1} .

GEANT4 is capable of simulating E_{dep} for a scintillator without having to fully simulate optical photons in the scintillation process. N_γ can then be found simply through a proportionality relationship:

$$N_\gamma = kE_{dep} \tag{2.1}$$

It is then trivial to only record E_{dep} and quickly calculate the N_γ output, without generating the photons in the first place. Implementing this resulted in a very large improvement to efficiency and processing time, as the vast majority of particles whose tracking takes up a very significant amount of valuable simulation time do not need to be simulated in the first place.

To test the efficiency gains of this method, the time taken to process the events needs to be recorded. Fortunately, this is made simple, as all one needs to do is to turn off scintillation photon generation and record the computation time taken, as the time taken to perform a single multiplication as in Equation (2.1) is negligible. The results of this are further discussed below.

2.2.2 Not Recording Scintillation Photon Information

A significant source inefficiency lies not in the generation or tracking of the scintillation photons, but rather in the recording of the photons' parameters (position/energy) for each step along their tracks to the output ROOT file. Recording these values for each photon is very I/O intensive, as the scintillation photons are much higher in number than any other particle in the simulation; this requires a relatively huge number of values to be stored for every event.

These individual photon energy and position values are not of significant use in the final analysis of the simulated data, as the output from the PMT is simply proportional to the number of photons.

Thus, for most situations, the simulation was modified so that the recording of individual photon position and energy data was turned off. At the same time, in simulation configurations where it was useful, the optical photons were still generated and tracked, allowing further analysis to take place, especially in regard to event timing, as explored in Section 4.1.

2.3 Full Detector Construction in CDetMC

With the single bar simulation having been improved and significantly optimized, as explained in the previous section, the simulation processed events rapidly, generating ROOT output files on reasonable timescales. This solved the efficiency barrier that had previously prevented the simulation of the full CDet detector. It was then possible to use these efficiency gains to simulate the entire detector in a much more reasonable amount of time.

To construct the detector, the geometry, as specified in GEANT4, was significantly modified. As all the bars are identical, and only differ by position, the optimized bars were replicated and were then placed into the geometry shown in Section 1.9.2. To fully replicate the geometry used in the team's cosmic ray test setup at JLab, the number of finger counters increased from two to four, and the sizes — as positions of these detectors — was modified to match that used in the test setup as well. This was done to produce output that can be used to compare simulation attempts to test data.

Each paddle (within each bar) was indexed with a unique identifying number, which allowed for the storage and identification of each individual paddle. The paddle index also corresponded to the index of the arrays used to store energy and timing information in the ROOT output file.

The positions and orientations of the individual paddles were calculated using an algorithm, and not hard coded. This allows for customization of certain quantities, such as the number of layers in the detector, or the number of paddles in a bar, making it possible to use this simulation for different configurations.

The procedural construction of the individual scintillation counters was done using the simple, cuboidal shape of the scintillator, as the length and width of each paddle is the same for all paddles. As the centroid coordinates of each paddle was procedurally generated, the coordinates of each corner can be quickly extrapolated by adding or subtracting half of the paddle dimensions in the relevant axis.

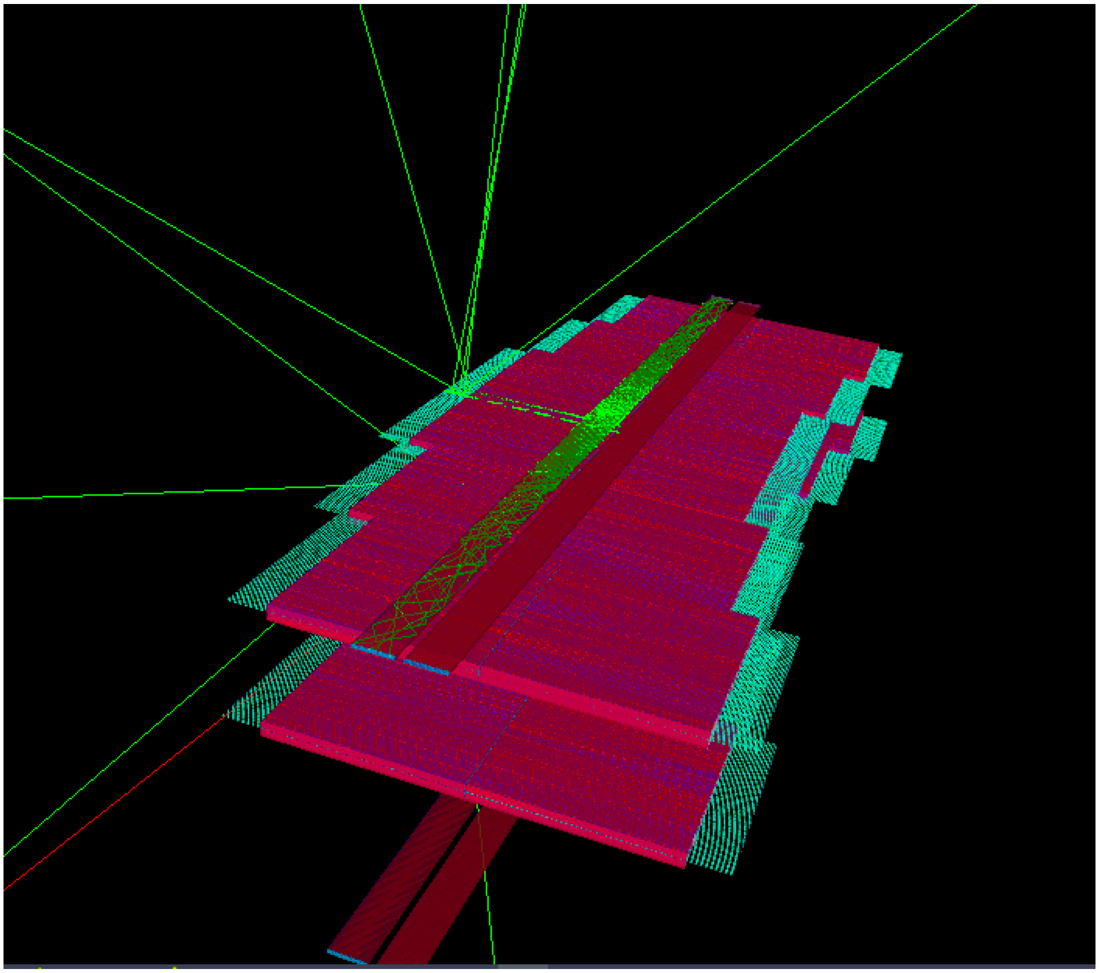

Figure 2.3: A GEANT4 visual representation of the CDet simulation after significant optimization, with every paddle in the detector now simulated. The two layers in the simulation are the layers of the Coordinate detector, the four finger paddles can also be seen above and below the layers.

2.3.1 Geometry MetaData

As the numerical index of each paddle is generated while the geometry is being constructed, it is saved in a vector variable. In addition, the x,y and z position of the centroid of each paddle are stored in companion vectors, which can be cross-referenced with each other as they have corresponding indices.

This data is saved as metadata in the header of the output ROOT file after running the simulation. This flexible approach allows for data analysis tools to adapt to different geometries of the simulation, in case any are chosen for testing.

2.4 Results and Discussion

2.4.1 Testing Optimization Strategies

Using the same multi-core CPU to make fair comparisons, the computation time taken for the simulation of a single bar for 1000 events was recorded. This was done for three versions of the simulation: the original with no optimization, which simulates all the physics and records all the particles; a version where optical photons were not generated at all, corresponding with Section 2.2.1 and a version for which optical photon data was not recorded at the end of an event while the physics of the photon production was unaltered, as outlined in Section 2.2.2. The results are shown in Tab. 2.1.

Optimization Method	**Computation Time (s)**	
	Trial 1 (s)	**Trial 2 (s)**
None	314	316
No Photons Generated	3	3
No Photons Recorded	286	280

Table 2.1: A table showing the amount of time taken for the simulation to process 1000 events for different optimization methods. The methods are: no optimization (None), not generating any photons during scintillation (No Photons Generated), and not recording photons after the event is over (No Photons Recorded). Times are shown to the nearest second.

The time difference between the non-optimized version of and the version without photon generation was the largest, with this method reducing the time by the order of 10^2. This extreme reduction in time taken helps confirm the hypothesis that the major driver of the inefficiency was the photons generated in the simulation.

Matching expectation, the time taken for the version without photon recording, with the photons still being simulated was in between the time taken for the other two versions.

In Fig. 2.4, we show another plot of energy deposition vs. number of photoelectrons generated at the maPMT, for a large number of simulated events, and in this case shown as a profile histogram, rather than a traditional scatter plot. This allows one to visualize the centroid and standard deviation of each photoelectron bin. It is clear from this plot that the relationship between energy deposition and number of photoelectrons is not strictly linear, as assumed in Section 2.2.1. These deviations result from finite and asymmetric geometry effects, due to the rectangular geometry of the paddles in CDet. At the same moment, while an interesting result of this work, the relatively small deviations from a linear relationship are not nearly large enough to have a major impact on any of the results of simulations that were carried out using the proportionality model elsewhere in this work.

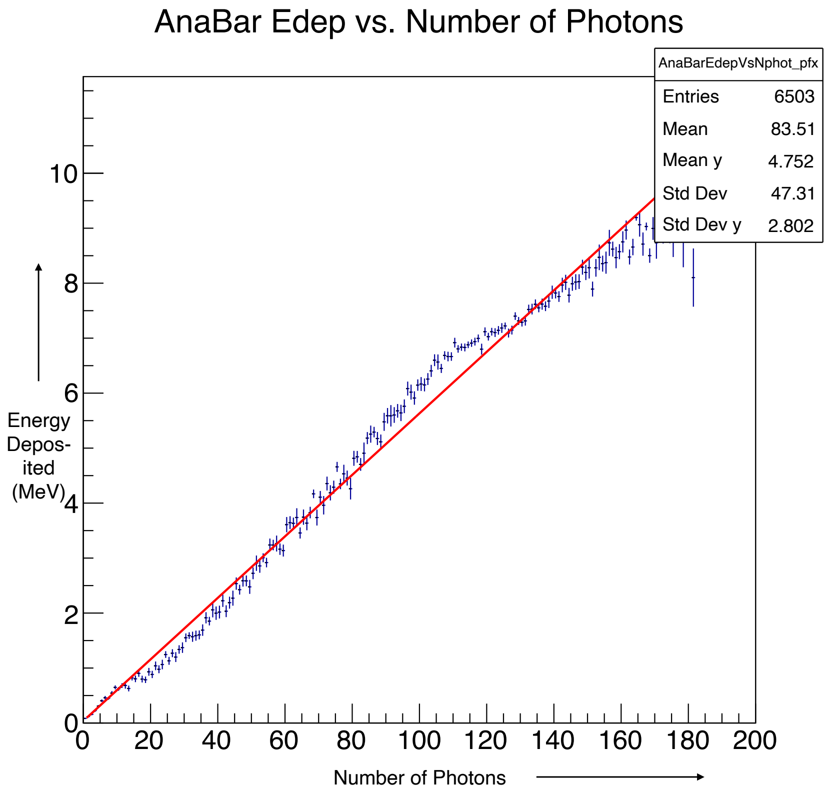

Figure 2.4: Correlation between Deposited Energy (in MeV) and Number of Photoelectrons Produced in CDet. The data is presented as a profile histogram. The data points correspond to the centroid and standard deviation of the data in each number of photoelectron bin (the photoelectron bin size is variable and is chosen within the plotting software for visual optimization).

Chapter 3

Event Generation

3.1 Background

As mentioned in the introductory chapter, a necessary and crucial component of any particle physics detector simulation is the event generator, in which one creates an event sample of incident particles whose starting position, type, momentum vector, and energy correspond as closely as possible to the expected distributions in the experiment under study. The event generator used prior to the project's commencement was fairly simplistic, in that for the most part was either a random distribution or a distribution that modeled cosmic ray muons. Moreover, changing parameters such as incident particle type, or a more realistic momentum distribution required the configurations files to be manually edited, and the entire simulation code to be recompiled. Each step in event generation was done manually, and required manual input for each individual task of generating a run list in which the number of events could be specified, generating incident particles from that run list, sending this input to the batch manager, and then adding all the output files run on different cores in the batch queue system together to generate a complete output.

Furthermore, the event generator was only able to simulate muons, and required heavy editing in order to simulate other particles necessary to make a better representation of the experiment. It was additionally difficult to specify more advanced event generation mechanisms.

The event generator in CDetMC was significantly improved by simplifying the steps needed to generate input particles, through automation of redundant steps. Additionally more flexibility in having additional parameters to control the generation of particle type now drastically increases the usefulness of the event generator, without having to hard-code desired changes to input particle types. This means that instead of just generating muons, the event generator now can be easily made to generate electrons, protons, and/or neutrons as one expects in the real experiments.

Additionally, the original event generator generated fairly simplistic particle momentum distributions, with an initial position and angle that could be specified in the code mostly for the sake of testing. However, this is not representative of the position and momentum distributions of scattered particles expected in the experiments, and the ability to simulate incident particle distributions that are more representative those from actual experiments has proven useful to the SBS collaboration, as a whole.

Section 3.2 documents the steps that were needed to generate simulated input particles in CDetMC simulation prior to automation and simplification of the event generation pipeline, providing necessary context for the improvements demonstrated in Section 3.4, which documents the changes made to these steps.

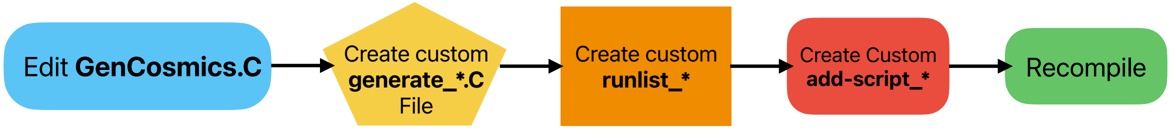

Figure 3.1: The steps required to edit the simulation's parameters. * refers to a number that distinguishes this set of run numbers from others. To work, all these files must share the same set of run numbers.

3.2 Manual steps for Event Generation

To initiate event generation, inputs to the simulation are required. This takes the form of ROOT files that contain the position and direction information of each particle that is then simulated, event by event, as one of the principle goals of these simulation studies is to simulate multiple particles in different configurations to better understand the detector's response to them.

In addition, a single run of the simulation can vary in the number of events being simulated, as the amount of computing time required in various configurations, together with the statistical precision required to understand a particular topic of interest, is variable and may require a different number of events. For example, it would be unnecessary to simulate an extremely large amount of events to generate enough data to simply debug code, whereas it might be necessary to generate a very large number of events for a final analysis, as a large amount of data ensures lower statistical uncertainty and aberrations in data can be noticed more easily.

In order to generate these event generator input ROOT files, users may create multiple scripts that are written in the format `generate_*.C`. For instance, `generate_smalltest.C`

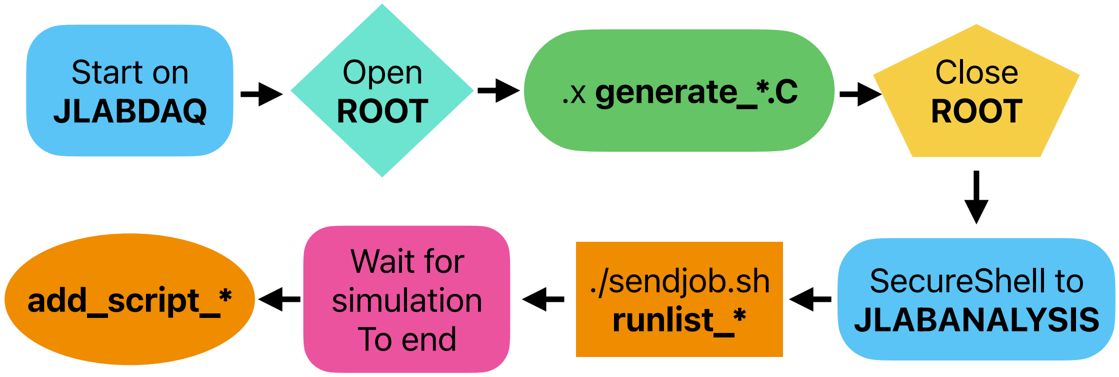

Figure 3.2: Previous steps to generate inputs and simulate CDetMC, follows from the steps in Figure 3.1

generates a relatively small number electrons starting above the finger counter, with a momentum distribution corresponding to cosmic rays. These user scripts all in turn call the `GenCosmics.C` script, which generates the input particle parameters, subject to certain parameters that are passed from the user script. For example, the number of events to be simulated is chosen in the user script, and then this parameter is passed into the `GenCosmics.C` script. One advantage of this approach is that for simulations that require massive number of events, the user can tailor the user script to run multiple smaller simulations across different cores of the batch queue system, thus decreasing the total run time.

The `GenCosmics.C` script generates particles with a predefined range of attributes, such as starting position, momentum vector and energy, to simulate a single event. Each call to the `GenCosmics.C` script requires the number of events this script processes and a run number, which helps organize the multi-threaded/multi-core operation of the simulation. As such, each run corresponds to a particular core, and each core handles multiple events of the same run number.

An important point to note is that to change the *physics* parameters of the input particles, the code of the `GenCosmics.C` script must be edited directly. From here, it is possible to change attributes of the input particles, such as position, angle and even particle type. However, these are hard-coded in the script and cannot be changed in any other way.

The code that sorts the different run numbers and assigns them to different cores is the shell script `sendjobs.sh`, which is also the script that initiates the GEANT4 simulation itself. To execute this code a relevant run list is provided, which allows `sendjobs.sh` to distinguish the relevant run numbers for a particular set of runs.

As the simulations are being carried out for the full list of run numbers provided, output ROOT files are being generated. This entire process results in multiple output ROOT files – one for each run number in the run list. While each of these output files contain complete and valid data for analysis, one must finally combine together all of the runs together into a final output ROOT file. As such, the script called `add_script` helps solve this problem.

3.3 Customization

As discussed in Section 3.2, the requirement to edit and create sets of entirely new user scripts for each simulation configuration posed a significant bottleneck to the efficient generation of events. To solve this problem, it was considered useful to have more generalizable functions where important input particle values can be altered by adding additional parameters to a more generalized user script that allows for the selection of input values, rather than having to make a different version of the script for every change.

To facilitate this, `GenCosmics.C` was generalized to have all of its functions be carried out by a new script, `GenParticles.C`, named because it could not only generate cosmic rays, but a much larger selection of particles.

For testing, it was necessary for the input particles to be generated from a variety of different positions and input parameters. It is also important for the collaboration to test the response of CDet to different types of particles, for potential use as a veto detector (for example, another variant of `GenCosmics.C` exists, called `GenNeutrons.C`, which only generates neutrons). As such, it is now possible to generate any type of particle from the Standard Model without having to create new files and subroutines for every possibility. This new script, `GenParticles.C`, was created to solve this issue.

GEANT4 is written to use, consistently, a numbering scheme created by the Particle Data Group (PDG) [29] to specify and identify the types of particle generated in any GEANT4 simulation. Each particle type has a unique integer value assigned to it, and then GEANT4 uses PDG database functions to access its attributes. In our new generation script, instead of having the particle type hard coded in `GenCosmics.C`, it is an optional parameter so that the particle type can be specified by the equivalent PDG number when called.

As mentioned above, particles were often generated with random starting positions over a specific physical range of CDet (for testing purposes, typically). It is often helpful to test different regions of a detector, to look for geometric dependencies. Previously, this had to be done by creating a new version of `GenCosmics.C` with different hard-coded values for the range of positions to be studies. In the updated script, the minimum and maximum values for this range are now passed as additional parameters to `GenParticles.C`.

Another significant issue was the need to create a multitude of different scripts for every permutation of different run numbers and number of events. This heavily limits testing, as the team is then forced to work with a small number of permutations, based on organizational capabilities. A solution to this was developed using a simplified simulation pipeline that is further explained in the next section.

3.4 Automation

With most customization issues in Section 3.2 solved in Section 3.3, there were still pertinent issues remaining with the pipeline, including time lost waiting for a process to end before typing the next command, alongside remembering the specific instructions needed to complete a certain run list. A small mistake, such as using the wrong `generate_*.C` file, `runlist` or `add_script`, leads to wasted resources, and perhaps more importantly, unusable simluated data. This section covers solutions aimed at streamlining the process, in an attempt to reduce the number of steps required to perform the simulation.

A shell script, `Automate.sh` was written which significantly reduces the number of steps to go from a set of input parameters to useful output files, and also to eliminate the need to create new files for every subset of events that need to be generated. The scripts simplifies the simulation pipeline, fulfilling the functions of all the steps in Section 3.2. Now, the user calls only `Automate.sh` without having to manually edit files and remember all of the distinct steps detailed previously.

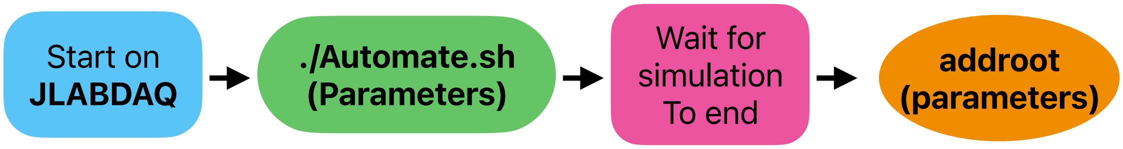

Figure 3.3: Updated steps to begin simulation, updated from the steps in Figure 3.2. Note that the steps in Figure 3.1 are made redundant in this update by the use of parameters.

The need to create new files for different sets of run numbers was eliminated, as the beginning and ending run numbers are now selected as parameters in `Automate.sh`. This eliminates the need for the `generate_*.C` scripts, as `Automate.sh` simply calls the `GenParticles.C` script for all of the required run numbers via an iteration loop. To maintain the customization ability of `GenParticles.C`, the particle type and initial position ranges are also parameters of `Automate.sh`.

Previously, the script `sendjobs.sh` required a user-edited run list file corresponding to the runs that were simulated. `Automate.sh` now handles the automatic generation of this run list file, eliminating the need for the user to provide it.

3.5 Event Generation using G4SBS Input Data

In order to generate input events for CDetMC with realistic particle distributions, we developed a pipeline based on G4SBS, the Hall A SBS large scale simulation framework. Using already existing SBS parameter files for the Gep-V experiment[10], a large sample (10^7 events) of scattered particles was generated. From that G4SBS output ROOT file, we extracted the information required for CDetMC input particle parameters (e.g. particle type, momentum vector, starting position vector, and total energy) and wrote this information to a plain text file. Then, a ROOT macro file was written to process this plain text file and create an input ROOT file in the format required by CDetMC.

This process is displayed schematically in the left panel of Fig. 3.4. While relatively straightforward, one complication that should be highlighted is that G4SBS and CDetMC use different coordinate systems; G4SBS uses the coordinate system of Hall A (with the target at the origin of the coordinate system), whereas CDetMC uses a ”local” coordinate system with the origin in the middle of the front face of the detector, and the z-axis along the central line of SBS. Thus, it was necessary in preparing the G4SBS-generated data for input into CDetMC that a coordinate transformation was applied. The two coordinate systems are illustrated in Fig. 3.6.

3.6 Event Generation using a Toy Model Monte Carlo

The main problem with the approach of using G4SBS to provide input event generation for CDetMC is that it requires the GEp-V[10] simulation to be run within G4SBS, which is very computationally intensive. For simulating only CDet, it is preferred to have a quick way to generate events without running the entire GEp-V simulation.

To that end, we developed a so-called ”toy model” event generator, which simulates elastic $e-p$ scattering in the kinematics of Gep-V[10]. Using the basic theory of conservation of four-momentum and total energy in elastic scattering (using relativistic kinematics), it

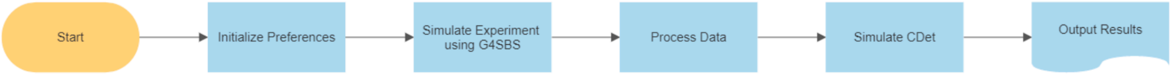

Figure 3.4: Schematic Diagram of G4SBS → CDetMC Event Generation Pipeline.

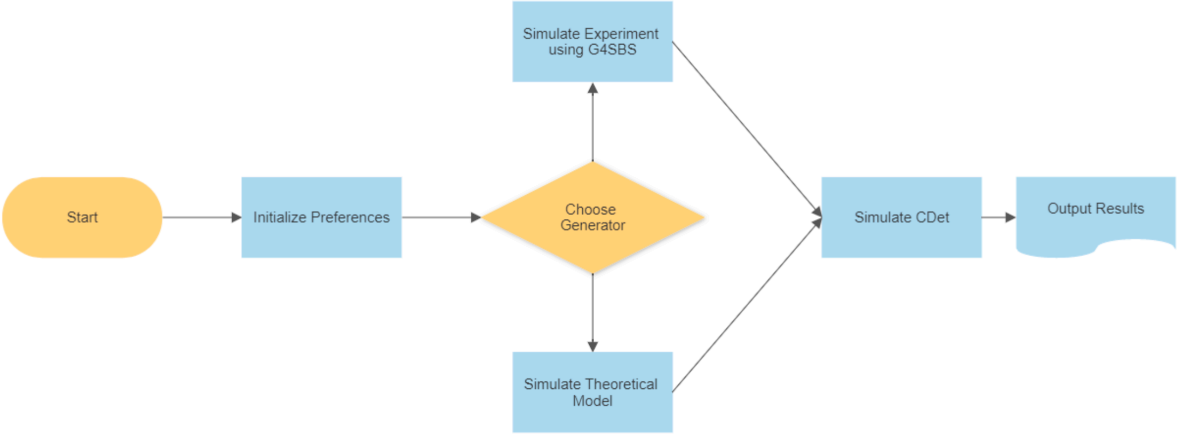

Figure 3.5: Schematic Diagram of Event Generation Pipeline for Multiple Generators

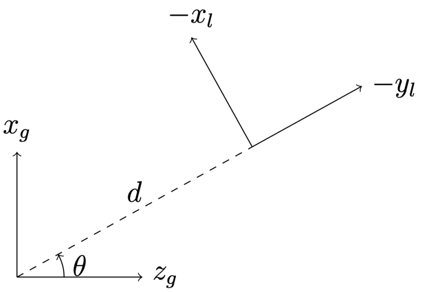

Figure 3.6: Illustration of the Global/Hall A Coordinate System and the CDetMC Coordinate System - the Global/Hall A system is labeled with x_g and z_g, with the CDetMC system labeled with x_l and y_l. Note that d represents the distance between the target centre and the front face of CDet, and θ is central angle setting of CDet.

is straightforward to show that one can determine a correlation between the position of a ”hit” on the face of CDet and the scattering angle of the scattered electron in elastic $e-p$ scattering, given by:

$$\theta = cos^{-1}\left(\frac{-xsin\theta_0 + zcos\theta_0}{\sqrt{x^2+y^2+x^2}}\right) \tag{3.1}$$

where θ_0 is the central scattering angle of the electron in the global Hall A coordinate system.

In addition, the energy of the scattered electron is given by:

$$E_{e'} = \frac{m_p E_e}{E_e(1-cos\theta)+m_p} \tag{3.2}$$

where m_p is the mass of the proton, and E_e is incident beam energy. Thus, we can use the two equations together to calculate the angle and energy (or momentum vector) of any electron hitting the face of CDet, based only on the position of that hit on the detector.

In our analysis of the input data generated using G4SBS, it was found that CDet was populated almost uniformly over its face. Because of this, a uniform random distribution of events over the face of CDet was used in this toy model simulation, and the momentum vector and energy for each event was then calculated based on hit position using the above equations.

The automation script described previously was modified to allow the user to specify which of the two event generation schemes are to be used. This is illustrated schematically in the right panel of Fig. 3.4.

In Figs. 3.7 and 3.8, simulated event distributions are shown for the momentum/position correlations exhibited in the input electron event distributions from the G4SBS event generator, and the toy model MC event generator, respectively. The qualitative agreement between the two generators is excellent, and sufficient to allow substitution of the toy model over the full G4SBS generator for purposes of modeling CDet response and performance. At the same time, the toy model generator requires significantly reduced computing resources, and so is an excellent alternative in this respect.

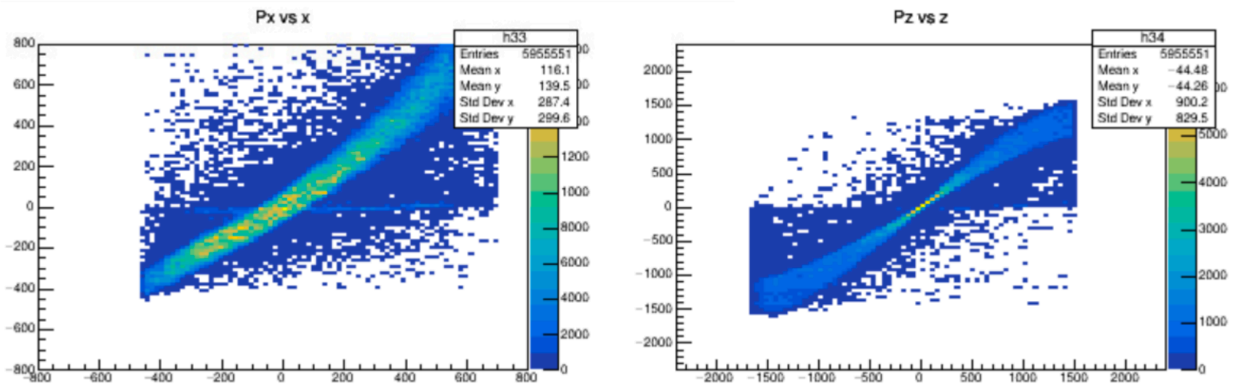

Figure 3.7: Horizontal Momentum vs. Horizontal Position (L) and Vertical Momentum vs. Vertical Position (R) from G4SBS Simulation

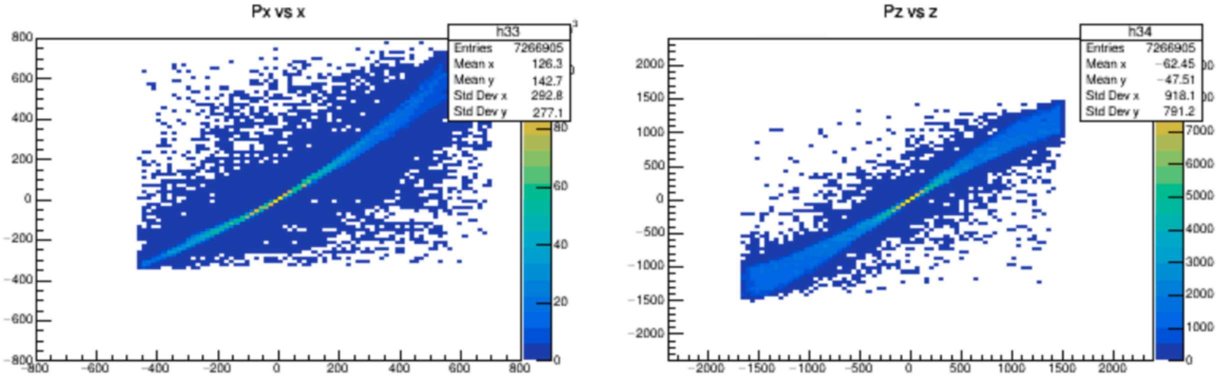

Figure 3.8: Horizontal Momentum vs. Horizontal Position (L) and Vertical Momentum vs. Vertical Position (R) from Toy Model Monte Carlo Simulation.

Chapter 4

Horizontal and Vertical Angle Calculations

One of the primary purposes of CDet is to identify charged particle tracks originating from the target in elastic $e-p$ scattering, as will be measured in the Gep-V experiment. One can use the horizontal (in-plane) and vertical (out-of-plane) angles of the tracks, as determined using the hit positions of tracks in the two independent layers of CDet, as a means of identifying the point of origin of the track. In addition, one can use correlations between these angles and those determined from an associated proton track in SBS to aid in identifying elastic $e-p$ scattering events. Thus, it is important to build tools that will quickly and accurately determine the scattering angles.

Because there are only two layers in CDet, these angle calculations are relatively simple, needing only the in-plane and out-of-plane positions of the particle in both layers of the detector. In the local coordinate system, The horizontal (in-plane) angle is given by:

$$\theta = tan^(- 1)\left(\frac{x_2 - x_1}{y}\right) \tag{4.1}$$

and the vertical (out-of-plane) angle is given by:

$$\phi = tan^(- 1)\left(\frac{z_2 - z_1}{y}\right)$$

In these equations, y is the distance between the two layers, (x_1,x_2) are the in-plane positions of the particle hits on both layers and (z_1,z_2) are the out-of-plane positions of the particle hits on both layers.

Determining ϕ is straightforward and relatively precise, as the narrow width of the scintillator paddles results in excellent out-of-plane position resolution, making it simple to use the out-of-plane centroids of the triggered scintillation paddles to reconstruct an accurate value of ϕ.

Determining θ is more complex because the in-plane-position determination (x) is much more involved. Indeed, CDet was not designed specifically for this purpose. However, due to the excellent timing resolution of the maPMTs used with CDet, it may be possible to achieve in-plane position resolution that is sufficient to make the in-plane-angle determination more useful that originally thought. It may also be possible to combine this information together with tracking information from other detectors in the electron detection arm in Gep-V.

Determining the in-plane position of a hit in CDet requires first to determine a time-to-distance calibration (Section 4.1). One can then use the timing information from the maPMT's of CDet to determine the horizontal position of events, relative the the center of a hit bar/paddle in CDet.

4.1 Timing to Distance Calibration

The horizontal (in-plane) direction in CDet is along the length of the scintillator paddles, and thus it is impossible to precisely measure in-plane angles with only the knowledge of which scintillator paddles were triggered in a single event.

It is important to note, however, that maPMT's used in CDet provide extremely good timing resolution (on the order of 100 ps). Thus, given the precise timing data that can be collected from the detector, together with the known index of refraction of the scintillation material, it is possible to use timing information to reconstruct the incident particle's position along the scintillator bar. This information can then be used to facilitate the reconstruction of the in-plane angles.

4.2 Implementation

In principle, the distance could simply be calculated as $d = \frac{c}{n}t$, where c is the speed of light, t is the time provided by the maPMT, and n is the index of refraction of the scintillator material (≈ 1.58).

However, there are a number of factors which complicate this naive approach. The first is that not all photons created in the scintillation process for a given incident particle arrive at the photocathode of the PMT at the same time; photons which happen to be emitted directly towards the PMT may arrive sooner, whereas photons emitted perpendicular to the scintillator bar may reflect dozens of times before reaching the PMT at a later time. Moreover, photons emitted away from the PMT may travel towards the center of CDet, strike the mirrors that are at the end of each bar, and then reflect back to the PMT eventually. All in all, the timing signal from the maPMT is generally not a narrow timing pulse, but instead is rather broad, and not practical to calculate from first principles. Moreover, the shape of the timing pulse is expected to vary from event to event, given the relatively random nature of the scintillation process itself.

Nonetheless, one of the features of the CDet read-out electronics design is that the ”time” of the signal from any maPMT is determined from the time of the leading edge of the electronic signal from the maPMT. In other words, this time is indicative of the arrival time of the first photons at the maPMT. As a result of this fact, we can, in CDetMC, use the average arrival time of these first photons to determine a maPMT time for a given event.

In addition, we have in CDetMC perfect information on the in-plane position of the hit (it is, after all, a simulation where we are generating the events and so these so-called ”truth” values of the track are known to us, unlike in a real experiment).

The process for determining the time-to-distance calibration was as follows:

1. A sample of input events was generated where all input particles in the sample struck the same horizontal position along a scintillator bar.
2. The leading edge time (statistical average) was determined for this event sample.
3. This process was repeated for a series of event samples at different horizontal positions along the scintillator bar.

In Fig. 4.1, the results of these simulations are shown. For each of the first and second layers of CDet, a best fit was obtained by fitting a quadratic function to the simulated data points. Shown in the figure are these best fit lines, together with lower and upper 95% confidence bounds on the fits. It is interesting to note that the fit, while quite reasonable for distances close to the end of the scintillator closest to the PMT (large positive X values), is poorer towards the mirror end of the scintillator (large negative X values). This is to be expected, as this is the region where the travel time of the photons from their generation point to the PMT has the largest amount of statistical fluctuation.

The results of the quadratic fit are

$$\begin{aligned} Time &= (-0.000374997 +/- 0.000117910)X^2 + && (4.2) \\ & (-0.051184439 +/- 0.001636894)X + && (4.3) \\ & (3.592988020 +/- 0.037587907) && (4.4) \end{aligned}$$

for the first layer of CDet, and

$$\begin{aligned} Time &= (-0.000409055 +/- 0.000141231)X^2 + && (4.5) \\ & (-0.050453363 +/- 0.001989061)X + && (4.6) \\ & (4.537007798 +/- 0.044768133) && (4.7) \end{aligned}$$

for the second layer, where $Time$ is the arrival time of the first photons at the maPMT photocathode, in ns, and X is the position of the hit in the scintillator bar, relative to the geometrical center of the bar, in cm.

In subsequent simulations, the mathematical inverse of these fits were used to calculate

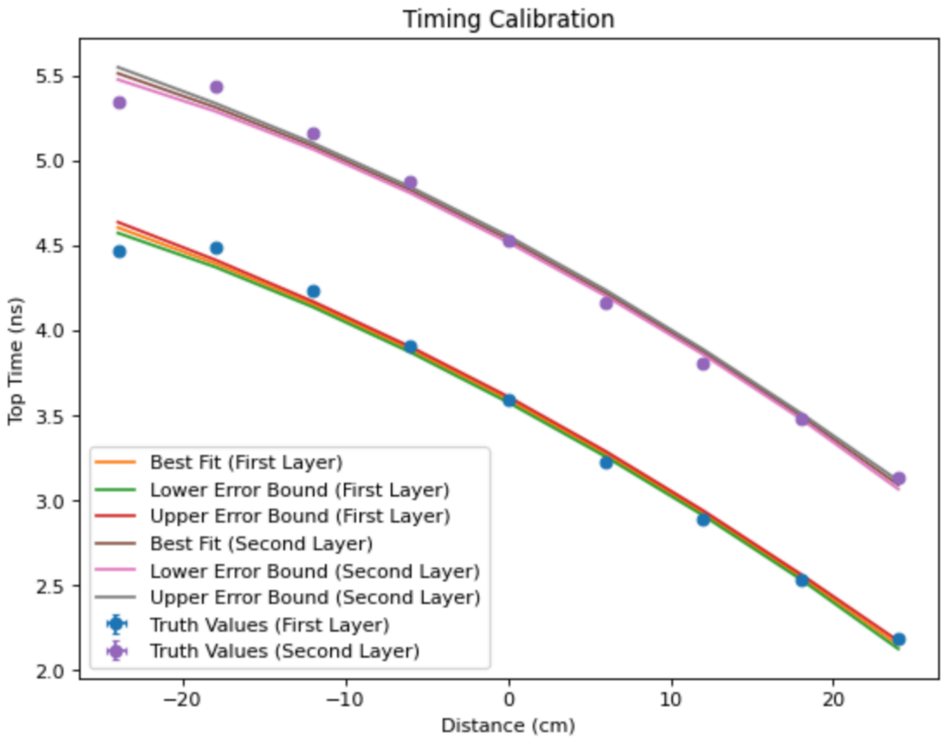

Figure 4.1: Timing-to-Distance Calibration Simulations

horizontal position offsets from the geometrical center of each hit paddle for each hit in each event of the simulation. The uncertainties in the fit parameters were used to generate an additional random offset to the calculated horizontal hit position. In this way, the expected horizontal position resolution in the experiments is also modeled.

In Fig. 4.2, the horizontal (top left) and vertical (top right) hit positions as known from the truth values in the simulation are plotted (y axis) versus those calculated based on the geometrical positions of the hit paddles together with the new time-to-distance calibration. This provides a measure of the expected horizontal and vertical reconstructed position resolutons of CDet. For events where the incident particle resulted in hits in adjacent paddles in a layer, a simple weighted average of the positions was calculated for that cluster of hits.

For the reconstructed hit positions, we see that the out-of-plane (Z) position resolution is excellent, as expected, owing to the thin segmentation of CDet in the out-of-plane direction. For the in-plane (X) positions, the resolution is moderate ($\approx 10cm$). When used along with horizontal position information from other detectors in the electron detection arm in Gep-V, it is hoped that the information from CDet may be useful. Moreover, it may be possible to use this horizontal position information to separate hits from multiple electron tracks, which are expected in the high luminosity environment of Gep-V.

Unfortunately, due to the relatively small distance that exists between the two layers of CDet, the in-plane momentum/angular resolution is seen to be extremely poor, and is thus probably not of great use in the planned experiments. At the same time, the out-of-plane momentum/angular resolution is still quite good, and it can be hoped that this will be of significant value in the analysis of Gep-V data.

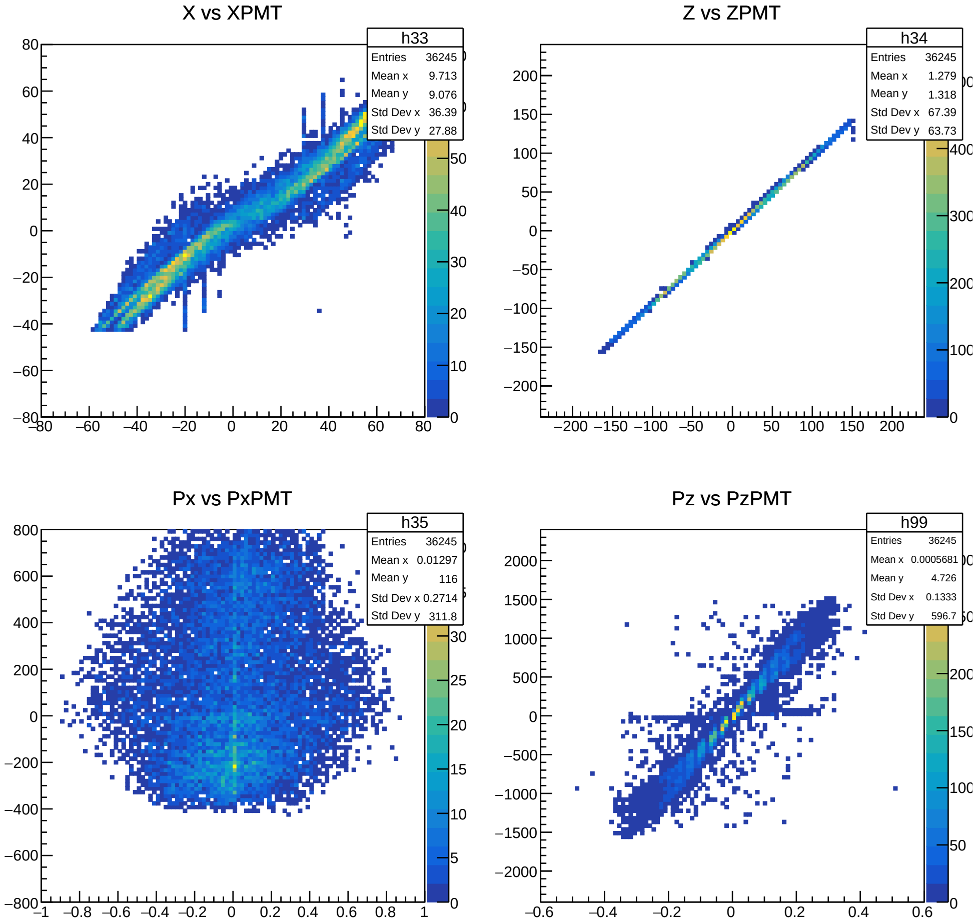

Figure 4.2: Comparison of Actual vs. Reconstructed Horizontal and Vertical Positions and Momenta in CDet. Top Row: Horizontal and Vertical Hit Positions, in *cm*. Bottom Row: Horizontal and Vertical Momentum Components - note that for the momenta, the y-axis is in MeV/c, and the x-axis is in GeV/c

Chapter 5

Development of an Event Display for CDet

Under the high rate conditions of Gep-V, it is expected that along with hits in CDet that come from the scattered electron in elastic $e - p$ scattering we may also have a significant number of extraneous hits associated with electromagnetic or hadronic background in Hall A, or from electronic ”noise” in the detector itself, or from both sources. In order to better understand how we might identify and eliminate these undesirable hits, additional software tools and analysis algorithms have been developed, and initial studies have been conducted using CDetMC together with a new event display tool. In this chapter, this event display will be discussed in detail.

5.1 Event Display

When creating and developing tools and analysis software for either simulation or data analysis of CDet, it is important for scientists to be able to visualize the effectiveness and impact of these tools and software. This is imperative in identifying errors in the software, initially, and then subsequently to ensure that the simulation and/or analysis software is running correctly, or to assist visualization of actual responses of the detector system for the real experiment. While the ROOT [23] analysis framework allows us display both raw and calculated data as histograms, and the GEANT4 simulation framework has the ability to display the physical detector model along with particle tracks in three dimensions, diagnosing problems can be further aided by the ability to display the event-by-event reconstruction parameters for tracks in a more intuitive fashion.

Further, such a tool will be beneficial to the SBS Collaboration at the beginning of experiments, as experimenters will have access to intuitive event-by-event visualizations in order to quickly identify and diagnose problems with the equipment in the experiment. It is also important for CDet data visualization to be able to see the results of a single event, in a format that closely mimics the spatial structure of CDet. To this end, an event display tool, based in the ROOT framework, has been developed for this purpose.

In Fig. 5.1, we show the new event display tool. For the event shown, the hit positions of the primary electron is shown. The geometry of each of the two layers is illustrated with a solid blue background. The hit paddles for the event are shown as horizontal lines at the geometrical position of the paddle, and the reconstructed horizontal hit position is shown as a solid black dot at the reconstructed position for each hit. Navigation between events is accomplished via the 'Previous' and 'Next' buttons at the bottom of the event display, with the display immediately updating to show that particular event. If viewing a particular event is desired, it can be entered in a text box between these buttons.

The event display allows the user to dynamically view any combination of the following:

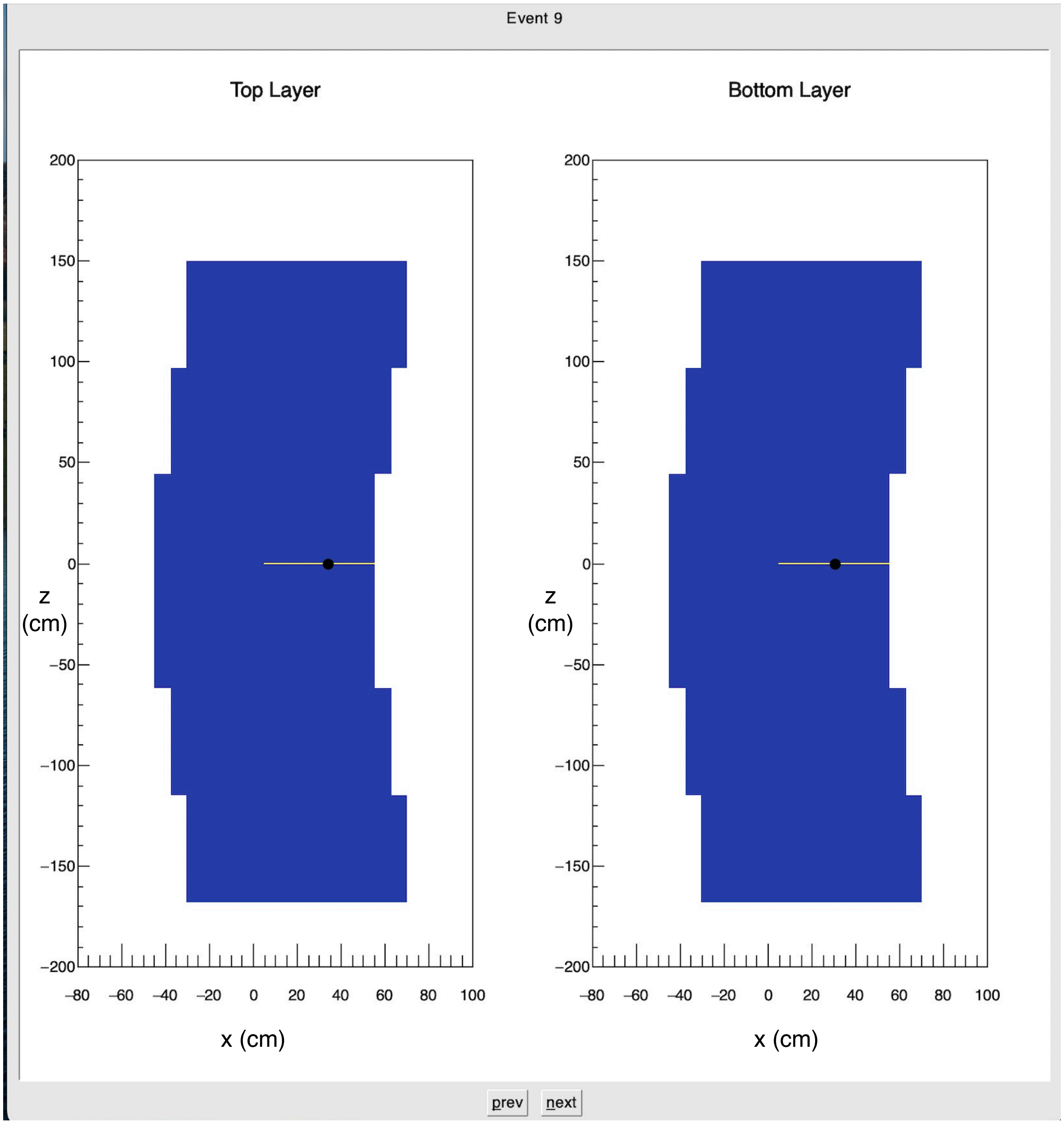

Figure 5.1: The Event Display, with only the option for the paddles triggered by the primary particle turned on. The triggered paddles are visible in the event display as orange lines overlaid on the detector's geometry in dark blue.

- The Paddles triggered by the particle (see Section 5.1.3)
- The Paddles triggered by noise (see Section 5.1.4)
- The Time-to-Distance Calibrated in-plane values (see Section 5.1.5)

The event display also allows the user to choose which datasets they would like to view by selecting checkboxes corresponding to these options. A legend at the bottom of the display screen shows the colour of the dataset on the display.

5.1.1 Implementation

The event display was implemented using ROOT's GUI library [23], and was designed to be as user-friendly as possible, have flexibility for use in different analysis software packages and to be easily updated.

The code for the event display object contains a variable for the event Number, which allows for navigation between different indexed events. Every time the event number is changed, the various objects containing display information for the event display are regenerated and the event display is refreshed to prevent data from other events contaminating the display of that new event.

For every event, the program finds all paddles that are above a certain photon number cutoff, and sorts them into two arrays based on whether they are real hits or noise. These arrays are used to calculate the in-plane position (see Section 4.1) of these particles for use in reconstructing in-plane angles (see Section 5.1.5).

However, the different objects containing the various options described in the previous subsection are not regenerated every time different options are selected. The display is updated to show different datasets; this is further elaborated on in the coming subsections: Geometry, Triggered Paddles, Noise Hits and Reconstructed In-Plane Positions.

A simple legend is constructed at the bottom, and this displays the colours that correspond to the various options chosen to ensure that clarity is present with regard to what each colour represents. This legend is also dynamic, as it only shows the legend for the data that is visible on the screen, corresponding to the chosen datasets.

5.1.2 Geometry

As the CDet detector is made of two layers of a flat set of paddles, it lends itself well to being displayed on a 2-Dimensional screen. The two layers are displayed side-by-side to ensure that it is easy to view the positions of paddles in relation to their counterparts on the other layer. As the layers are identical in the 2-Dimensional Plane, they are displayed at the same size.

A goal for the event display to be intuitively representative of the geometry of the CDet. However, as the modules are offset in the z-axis, it would not be a good representation to simply display them as a two-column table. To solve the problem of offset paddles that need to be represented by a single data structure for each layer, the `TH2Poly` [27] ROOT data structure was used. It is a ROOT two-Dimensional histogram, and can be used as such, however, each bin can have custom dimensions and coordinates when drawn as a two-dimensional Histogram, and the coordinates can be represented to scale in ROOT's `TCanvas` display. With the appropriate settings, each bin is separately displayed, with the colour of the bin corresponding to its contents when drawn.

As the geometry data from Section 2.3.1 is available from the header of every ROOT storage file, it is used to dynamically build the structure of the event display in a similar way to the the full detector in the simulation (Section 2.3).

To build the layer, a `TH2Poly` corresponding to the layer is constructed. Bins are added to this histogram for every paddle in the simulation. For each paddle, the dimensions of the bin are constructed appropriately by adding the half the paddle's length and width in the x and z coordinates from Section 1.9.2, to build a narrow rectangle. The paddle's x and z centroid position (from the geometry data) is added to these. These coordinates are used to construct the bin in the display, as a 2-Dimensional representative of the paddle. As the paddle's length is much larger than its width, the bins appear to be lines in the Event Display.

With all the bins constructed, the `TCanvas` is filled by a blue object with the same dimensions of a single layer of the Coordinate Detector, to scale. This particular histogram is never filled; the opaque histogram functions as a background that helps contextualize data that will be shown in the coming sections.

5.1.3 Triggered Paddles

A `TH2Poly` histogram identical to the one in Section 5.1.2 is implemented. This histogram is meant to display the data corresponding corresponding to the paddles triggered by the simulated physics in the experiment, which represents useful points of data for data analysis and understanding the impact of the simulation in what was recorded by the CDet. Therefore, the bins are not displayed when they are not filled with any data, as these correspond to paddles not triggered in the simulation, with the histogram from the previous section displaying the Ggometry to provide necessary context to this data.

As the bins are indexed, the values of the equivalent paddle in the ROOT data file are simply copied to the appropriate bin. This has the additional benefit of also containing the number of photons recorded by each triggered paddle, and is displayed with brighter colours that correspond to a larger number of photons. The ability to fill this histogram with only data from the simulation is possible because the simulation's output storage ROOT file stores these hits separately from the hits also summed with noise.

If the "Real Hits" option is selected, this histogram is overlaid over the histogram from Section 5.1.2. As this histogram is transparent when it is displayed, only the triggered paddles are displayed, and these appear in an orange colour, so that they can be distinguished from the blue colour of the paddles triggered by Noise Hits. As they are displayed over the opaque histogram containing the geometry, sufficient context is provided for the location of each triggered paddle.

5.1.4 Noise Hits

Another histogram very similar to the one in Section 5.1.3 is implemented. However, it is are not filled with hits from real particles, but rather with generatred noise hits. Similar to Triggered Paddles, unfilled bins are transparent and only filled bins are visible, again corresponding to paddles triggered and not triggered by noise.

However, unlike the hits triggered by simulated physics, noise hits are not stored separately, but rather summed together with the simulated physics hits in an array meant to represent a noisy output. Thus, to fill a `TH2Poly` with only noise hits, the values from the summed array are only copied to the corresponding paddles in the noise hit histogram only if the values are either 0 or not the same as the equivalent value in the array containing only simulated physics hits.

Like simulated physics hits, this histogram is visible when the ”noise hits” checkbox is selected; this histogram is given a light blue to avoid confusion with the paddles triggered by particles, and can be displayed with all the other data.

5.1.5 Reconstructed In-Plane Positions

Two sets of data are calculated at the start of every event being processed in the display. First, the calculated in in-plane positions are filled for each real and noise hit, stored in separate arrays. Then, these arrays are stored as TH2PolyMarker Points that can be overlaid over all the histograms used to display the geometry and paddle data.

When the ”Time-to-Distance Calibration” checkbox is selected, event display is refreshed, with all of the other selected items shown. As the TH2PolyMarker Points have a draw function, the appropriate points are shown on the event display, overlaying everything else that is displayed.

Due to the simulated hits and noise being selectable, Time-to-Distance calculated data will not show when the other relevant option is not selected. As such, the calculated values for the real hits are only shown when the real hits are selected. Likewise, the calculated noise hits are only shown when noise hits are selected. It is then possible for the "Timing-Distance Calibration" checkbox to be selected, but not have any information shown, if the other two checkboxes are not selected.

In Fig. 5.2, we show the event display again, now for an event with both real and noise hits selected.

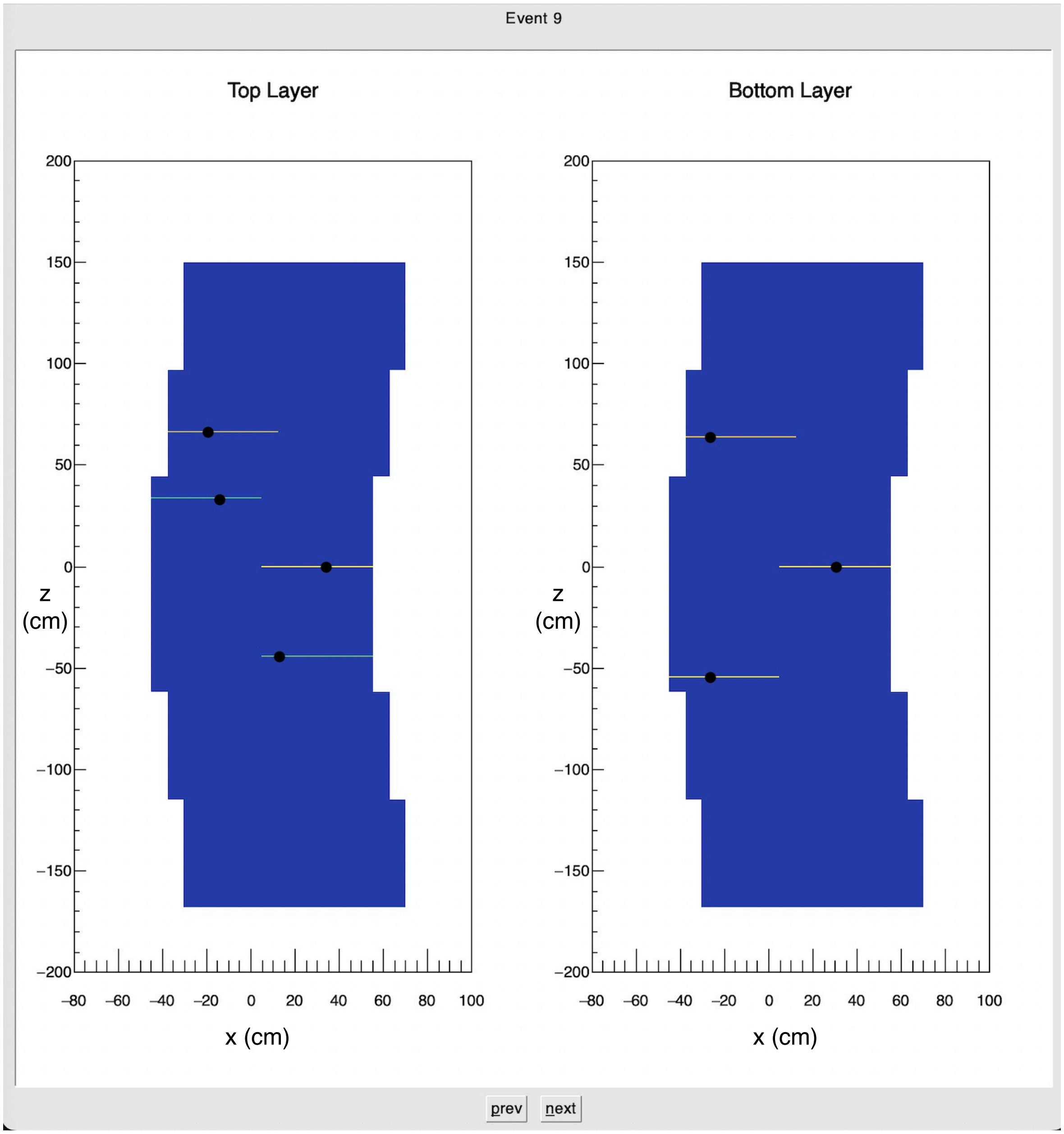

Figure 5.2: The Event Display, with all the options turned on, for one specific Event. The different options are clear in the figure, with the orange lines representing the paddles triggered by simulated particles, and the blue lines triggered by simulated noise. The dots represent the reconstructed hit positions of these Events.

Chapter 6

Conclusions and Future Developments

This thesis provides a detailed summary of a series of projects that were undertaken to improve the CDetMC simulation, and to develop additional analysis algorithms and other useful tools to aid in future software development and data analysis.

We have implemented, within the GEANT4-based CDet framework, a full implementation of the entire CDet detector, together with a system that was developed to optimize and automate the simulation of a large number of events. Significant improvements have been made to the CDetMC event generation scheme, and now include the possibility to generate realistic incident particle directories with position, momentum, and energy depositions that model those expected in future SBS experiments. While we have focused primarily on the Gep-V experiment, which will measure elastic $e - p$ scattering, because of the generalized approach that has been taken, it will be possible (and straightforward) to simulate distributions from other experiments with SBS as well. A new algorithm has been implemented to make a determination of the in-plane hit position of incident particles. While the in-plane momentum component calculated from these in-plane positions is poorly defined, due to the small distance between the layers of CDet, the new knowledge of the in-plane positions themselves is expected to prove useful, when combined with information from other detector subsystems. A new event display tool has been developed to aid in the simulation and analysis of noise hits in CDet, and has already been valuable in the development of initial CDet data analysis software.

6.1 Future Development Possibilities

6.1.1 Integrating `addroot` into `Automate.sh`

As explained in Section 3.4, the function `addroot` is not integrated into the function `Automate.sh`, which limits the ability to generate large amounts of varied data without personnel monitoring of the batch queue cluster, due to the fact that one must wait for all of the simulation work to be completed in `Automate.sh` before calling `addroot`. This is because the simulation is performed independently on multiple cores, and calling the `addroot` function too early can result in errors, which is likely to happen if it is only called after only one arbitrary core has completed simulation.

To solve this, it will be necessary for all the processes to communicate to a single program in some way. It is useful to note that all cores in the batch queue cluster share access to the same disk area. The `runlist_test` file contains a list of all the runs for the set of runs being generated, before it is overwritten by the next call to `Automate.sh`. Thus, a new text file can be generated by `Automate.sh`, in which a modified `GenerateParticles.C` can input its run number to after completion of simulation. As all runs will eventually be recorded in this file, the `GenerateParticles.C` can be programmed to only place a run number between higher and lower run numbers, with a line break to separate them. After each run number is recorded, a program can be called to compare this file to `runlist_test`. If it is identical, which will be the case when all the runs are done, it can then call the `addroot` script, using the first and last values of `runlist_test` as parameters.

6.1.2 Batch Mode

Due to the `addroot` script not being integrated into `Automate.sh`, it is currently not possible to send batches of `Automate.sh` with different parameters whilst also using the multiple core feature of the cluster. This heavily limits the experimenters to generate different datasets at one time, which can be compared to extract meaningful information about the functioning of the simulation.

However, this can be done when there is a single run number, and thus using a single core, as was done successfully when running the simulation on a stand-alone CPU. This works because `addroot` is not needed, as it is used only to sum up data from multiple run numbers. To do this, one must run a shell script that contains multiple calls to Automate.sh, all with the same run number. However, due to using only a single core, this process is significantly slower than using multiple cores.

With the modifications shown in Section 6.1.1, it is expected that a single call to `Automate.sh` will result in a single, summed root file, that does not need to wait for the simulation to be complete to have to use `addroot`. With modifications to `Automate.sh` to ensure that the program will not close until addroot is called, it will then be possible to send batches of `Automate.sh` commands with different parameters, using multiple cores at much greater speeds than sending batches using a single core.

Bibliography

[1] URL: https://geant4.web.cern.ch/docs/.

[2] J. Alcorn et al. "Basic instrumentation for Hall A at Jefferson Lab". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 522.3 (2004), pp. 294–346. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2003.11.415. URL: http://www.sciencedirect.com/science/article/pii/S0168900203033977.

[3] Kerry L. Mellott Anna Pla-Dalmau Alan D. Bross. "Low-cost extruded plastic scintillator, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 466 (3 2001), pp. 482–491. DOI: https://doi.org/10.1016/S0168-9002(01)00177-2.

[4] R. G. Arnold et al. "Measurement of Elastic Electron Scattering from the Proton at High Momentum Transfer". In: *Phys. Rev. Lett.* 57 (2 July 1986), pp. 174–177. DOI: 10.1103/PhysRevLett.57.174. URL: https://link.aps.org/doi/10.1103/PhysRevLett.57.174.

[5] John Arrington et al. "Nucleon Form Factors – A Jefferson Lab Perspective". In: *Journal of Physics: Conference Series* 299 (2011). URL: https://iopscience.iop.org/article/10.1088/1742-6596/299/1/012002/pdf.

[6] *BC-400/BC-404/BC-408/BC-412/BC-416 Premium Plastic Scintillators*. BICRON. 1998. URL: https://www.phys.ufl.edu/courses/phy4803L/group_I/muon/bicron_bc400-416.pdf.

[7] F Benmokhtar et al. "Precision Measurement of the Neutron Magnetic Form Factor up to $Q^2 = 18.0$ (GeV/c)^2 by the Ratio Method". In: *Jefferson Lab Program Advisory Committee* (2008).

[8] JB Birks. "The Theory and Practice of Scintillation Counting". In: *The Theory and Practice of Scintillation Counting*. Ed. by JB Birks. International Series of Monographs in Electronics and Instrumentation. Pergamon, 1964, pp. 39–67. ISBN: 978-0-08-010472-0. DOI: https://doi.org/10.1016/B978-0-08-010472-0.50008-2. URL: http://www.sciencedirect.com/science/article/pii/B9780080104720500082.

[9] G. D. Cates et al. "Flavor Decomposition of the Elastic Nucleon Electromagnetic Form Factors". In: *Physical Review Letters* 106.25 (June 2011). ISSN: 1079-7114. DOI: 10.1103/physrevlett.106.252003. URL: http://dx.doi.org/10.1103/PhysRevLett.106.252003.

[10] E Cisbani et al. “Update on E12-07-109 to PAC 47: Large Acceptance Proton Form Factor Ratio Measurements up to 14.5 GeV^2 Using the Recoil-Polarization Method”. In: *Jefferson Lab Program Advisory Committee* (2019).

[11] E. Cisbani et al. “Update on E12-07-109 to PAC 47: Large Acceptance Proton Form Factor Ratio Measurements up to 14.5 GeV2 Using the Recoil-Polarization Method”. In: *PAC 47, Jefferson Lab* (2019).

[12] I C Cloet et al. “Survey of nucleon electromagnetic form factors.” In: *Few-Body Syst.* 46 (Mar. 2009). DOI: 10.1007/s00601-009-0015-x.

[13] Jefferson Lab Collaboration. *Jefferson Lab*. URL: https://www.jlab.org/.

[14] *Documentation of g4sbs*. Jefferson Lab Hall A. URL: https://hallaweb.jlab.org/wiki/index.php/Documentation_of_g4sbs.

[15] Loyal Durand. “Inelastic Electron-Deuteron Scattering Cross Sections at High Energies”. In: *Phys. Rev.* 115 (4 Aug. 1959), pp. 1020–1038. DOI: 10.1103/PhysRev.115.1020. URL: https://link.aps.org/doi/10.1103/PhysRev.115.1020.

[16] R. Fernow. *Intoduction to Experimental Physics*. Cambridge University Press, 1986.

[17] E.M. Henley and A. Garcia. *Subatomic Physics*. World Scientific Publishing Co. Pte. Ltd., 2007.

[18] Charles Earl Hyde-Wright and Kees de Jager. “Electromagnetic Form Factors Of The Nucleon And Compton Scattering”. In: *Annual Review of Nuclear and Particle Science* 54.1 (2004), pp. 217–267. DOI: 10.1146/annurev.nucl.53.041002.110443. eprint: https://doi.org/10.1146/annurev.nucl.53.041002.110443. URL: https://doi.org/10.1146/annurev.nucl.53.041002.110443.

[19] K.-H. Kaiser et al. “The 1.5GeV harmonic double-sided microtron at Mainz University”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 593.3 (2008), pp. 159–170. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2008.05.018. URL: https://www.sciencedirect.com/science/article/pii/S0168900208007341.

[20] G. Peter Lepage and Stanley J. Brodsky. “Exclusive Processes in Quantum Chromodynamics: The Form Factors of Baryons at Large Momentum Transfer”. In: *Phys. Rev. Lett.* 43 (8 Aug. 1979), pp. 545–549. DOI: 10.1103/PhysRevLett.43.545. URL: https://link.aps.org/doi/10.1103/PhysRevLett.43.545.

[21] Dietrich Müller. “Electron Showers of High Primary Energy in Lead”. In: *Phys. Rev. D* 5 (11 June 1972), pp. 2677–2683. DOI: 10.1103/PhysRevD.5.2677. URL: https://link.aps.org/doi/10.1103/PhysRevD.5.2677.

[22] *Photomultiplier Tubes*. URL: https://www.sense-pro.org/lll-sensors/pmt.

[23] Fons Rademakers Rene Brun. “ROOT - An Object Oriented Data Analysis Framework”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*. Vol. 389. AIHENP’96 Workshop, Lausanne. Sept. 1997, pp. 81–86. URL: http://root.cern.ch/.

[24] M. N. Rosenbluth. "High Energy Elastic Scattering of Electrons on Protons". In: *Phys. Rev.* 79 (4 Aug. 1950), pp. 615–619. DOI: 10.1103/PhysRev.79.615. URL: https://link.aps.org/doi/10.1103/PhysRev.79.615.

[25] *Scintillating Products: Scintillating Optical Fibers*. Saint-Gobain Crystals. URL: https://ethz.ch/content/dam/ethz/special-interest/phys/particle-physics/precisionphysicsatlowenergy-dam/TeachingContent/ASL/bicronfiber.pdf.

[26] Elton S. Smith. "The 12 GeV JLab Upgrade Project". In: *Nuclear Physics A* 827.1-4 (Aug. 2009), pp. 599c–604c. ISSN: 0375-9474. DOI: 10.1016/j.nuclphysa.2009.05.132. URL: http://dx.doi.org/10.1016/j.nuclphysa.2009.05.132.

[27] *TH2Poly Class Reference*. ROOT By CERN. URL: https://root.cern.ch/doc/master/classTH2Poly.html.

[28] B Wojtsekhowski et al. "Large Acceptance Proton Form Factor Ratio Measurements at 13 and 15 $(\mathrm{GeV/c})^2$ Using Recoil Polarization Method". In: *PAC32* (2012). URL: https://www.jlab.org/exp_prog/proposals/07/PR12-07-109.pdf.

[29] R. L. Workman et al. "Review of Particle Physics". In: *PTEP* 2022 (2022), p. 083C01. DOI: 10.1093/ptep/ptac097.