TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS

TODO: Thanks to everyone...

CHAPTER 1

INTRODUCTION

1.1 PREPARING FOR BONUS12

1.2 DEEP INELASTIC INCLUSIVE DATA ANALYSIS

CHAPTER 2

PHYSICS FORMALISM

2.1 NUCLEON STRUCTURE

2.2 ELECTRON-SCATTERING KINEMATICS

2.3 ELASTIC REGIME

2.4 RESONANCE REGION

2.5 DEEP INELASTIC SCATTERING

2.6 THE QUARK-PARTON MODEL

2.7 QUANTUM CHROMODYNAMICS

2.8 NUCLEON STRUCTURE-FUNCTION RATIO F_N^2/F_P^2

2.9 DIFFICULTIES IN EXTRACTING F_N^2/F_P^2 FROM DEUTERIUM

2.9.1 BOUND NUCLEON STRUCTURE

2.9.2 BACKGROUNDS

2.10 BARELY OFF-SHELL NUCLEON STRUCTURE

CHAPTER 3

THE BONUS12 EXPERIMENT

3.1 CONTINUOUS ELECTRON BEAM ACCELERATOR FACILITY

3.2 CEBAF LARGE ACCEPTANCE SPECTROMETER

3.2.1 CENTRAL TIME OF FLIGHT

3.2.2 SOLENOID MAGNET

3.2.3 HIGH-THRESHOLD CHERENKOV COUNTER

3.2.4 DRIFT CHAMBERS

3.2.5 TORUS MAGNET

3.2.6 LOW-THRESHOLD CHERENKOV COUNTER

3.2.7 FORWARD TIME OF FLIGHT

3.2.8 PRE-SHOWER CALORIMETER

3.2.9 ELECTROMAGNETIC CALORIMETER

3.3 BONUS12 RTPC

3.3.1 TARGET

3.3.2 CATHODE AND GROUND FOIL

3.3.3 SENSITIVE REGION

3.3.4 GASEOUS ELECTRON MULTIPLIER FOILS

3.3.5 READOUT PAD-BOARD

3.3.6 TRANSLATION BOARDS AND ELECTRONICS

3.3.7 CONSTRUCTION AND INTEGRATION

3.4 DRIFT-GAS MONITORING SYSTEM

3.4.1 PURPOSE

The Drift-gas Monitoring System (DMS) is essentially a drift chamber designed to measure the drift-velocity of electrons in the drift-gas mixture. It does this by detecting β electrons radiated by two ^{90}Sr sources in coincidence with ionization electrons that those β's create (see Fig.). Each β must travel straight up through the hole in the skeleton of the DMS into the drift gas, create an ionization electron in the sensitive region, and be detected at the opposite end by an scintillator-photomultiplier tube combination. The ionization electrons that are created in that sensitive region are forced to move toward the anode via the electric field created by the cathode-anode combination and the field shaping electrodes that ensure the field is uniform in between. By knowing the distance between the sources and the arrival time of ionization electrons at the anode, we can calculate the drift velocity of those electrons. That drift velocity along with readings from attached pressure and temperature sensors, will tell us a bit more about the gas mixture changes and its effect on the velocity.

The construction of the DMS consists of the skeleton, electronics, and sources. The skeleton is made of six detachable Delrin sides held together by screws and

maintains its air-tightness by way of rubber gaskets. The cathode, anode, and field-shaping electrodes are made of conducting metals. Those together with the photomultiplier tubes all rely on a high-voltage power supply. Finally the sources are Eckert & Ziegler ^{90}Sr sources that will be housed in a plastic and attached to the DMS in a light-tight manner.

3.4.2 ELECTRIC FIELD

3.4.3 DESIGN

3.4.4 CONSTRUCTION AND TESTING

3.4.5 INTEGRATION

CHAPTER 4

SIMULATION AND DEVELOPMENT

The simulation and development of the BONuS12 experiment has been ongoing essentially since the original BONuS6 experiment in the early 2000's. The goal with BONuS has always been to detect low momentum spectator protons in coincidence with scattered electrons at high Bjoken-x. This is the purpose of the simulation and development (*i.e.* to optimize a detector that will result in high statistics in the relevant kinematic range).

This chapter will focus on the methods used in that detector optimization in preparation for the BONuS12 experimental run. It will cover simulations done to improve geometry, determine electron drift time, and understand energy loss through detector components. The chapter will also go over the construction of the detector. Finally, the process of reconstruction will be covered, which is the way we ultimately determine the kinematic values necessary to recover the structure functions we're interested in.

4.1 GEANT4 MONTE CARLO (GEMC)

Much of the simulations done in preparation for and during the CLAS12 use the Geant4 Monte Carlo (GEMC) software developed by Maurizio Ungaro at Jefferson Lab. GEMC, as the name indicates, uses a toolkit called Geant4. Geant4 was developed by CERN. It was released as a successor in the GEANT software toolkit series, whose first released was in 1998. Since then Geant4 has become an international collaboration of contributers and maintainers with applications ranging from nuclear physics to medical physics.

The purpose of the Geant4 toolkit is to simulate the passage of particles through matter. This can mean anything from particles going through biological material (*e.g.* simulating the effects of radiation on human tissue) to simulating particles moving through detectors, which will clearly be of much interest to us here. In order to understand how the BONuS12 experiment simulations were conducted with GEMC, let's first become a little more familiar with Geant4.

Geant4 uses the object-oriented programming language C++ in various facilities to exploit its features. The first defining characteristic of this toolkit is its ability to define geometry, or physical layout of an experiment. This lets us consider how this geometry effects the particles moving through the materials in the experiment. The path that these particles takes as well as the interactions with the materials they pass through is another facility in Geant4 known as tracking.

The Geant4 Monte Carlo (GEMC) is a C++ framework that utilizes Geant4 and the Monte Carlo method of randomized sampling in order to obtain particle behavior through materials. At a very basic level, GEMC can define particle momenta and angles as well as detector geometry and material in order to understand the particle's behavior in through that material. One can define a variety of output variables of interest in these simulations like total energy deposited, position, or momentum. There is much more that can be done with this simulation platform, that we'll discuss as we move along through the following sections. Let's first talk about one more tool that was used for simulations called Garfield++.

4.2 GARFIELD++

While Geant4 and GEMC both deal with the simulation of particles through matter, the particles of interest in the BONuS12 Experiment will also go through gases and be under the influence of electric and magnetic fields. For the simulation of such environments, we use a toolkit called Garfield++, which was developed at CERN. This is extended version of the original Garfield platform that incorporates MagBoltz in the C++ language. MagBoltz solves the Boltzmann transport equations for electrons in gas mixtures under the influence of electric and magnetic fields. This allows Garfield++ to simulate electrons traveling in a gaseous medium under the influence of electric and magnetic fields. The other programs utilized to create a mesh of the RTPC and solve the electromagnetic equations inside the RTPC are GMSH and ElmerSolver, respectively. We'll talk more about each of those and their purpose in our discussion of drift electrons as well as gas-mixture optimization.

4.3 BONUS12 RTPC SIMULATIONS

Throughout the next section, we're going to focus on the simulations that shed light on particle behaviors in the detector, drove optimization efforts, and offered insight about expected results. We'll go over all the tools that were used for the

simulations and how each one was utilized and implemented. Computer simulations are immensely powerful and tend to be much less expensive than physical exploration and experimentation. We'll discuss how the packages already presented can come together to simulate the entire BONuS12 Experiment from the RTPC to its inclusion in the CLAS12 detector.

4.3.1 GEOMETRY & MATERIALS

The first thing to do when simulating the BONuS12 RTPC in GEMC is do define its geometry and materials. This is done via Perl file, where one can use predefined materials from Geant4 (*e.g.* G4_KAPTON for Kapton, G4_Cu for copper, etc.) or define your own materials. Geometries are defined both in Geant4 and GEMC by solid types like "tube", "box", "sphere", etc. Since the BONuS12 RTPC is is made of several different cylinders, most of the geometry definitions are of type "tube". Therefore, we specify the dimensions in terms of r, ϕ and z. For example, the drift volume is defined in the code by

```
$detector{"name"} = "sensitive_drift_volume";
$detector{"mother"}    = "rtpc";
$detector{"description"} = "Sensitive drift volume";
$detector{"color"}     = "ff88994";
$detector{"type"}      = "Tube";
$detector{"dimensions"} = "$rmin*mm $rmax*mm $z_half*mm $phistart*deg
    $pspan*deg";
$detector{"material"}  = $mate;
$detector{"style"}     = 1;
$detector{"sensitivity"} = "rtpc"; ## HitProcess definition
$detector{"hit_type"}   = "rtpc"; ## HitProcess definition
print_det(\%configuration, \%detector);
```

where the material (`$mate`) is made of 80% He_3 and 20% CO_2 in this case and `$rmin`= 30.0, `$rmax`= 70.0, `$z_half`= 192.0, `$phistart`= 0.0, `$pspan`= 360.0. One defines the units within the declaration of `$detector{"dimensions"}`, like `$rmin*mm` would be 30.0 mm for example. There are other variable names that you see in within the detector attributes above that are important to understand. The variable `style` describes whether the type is a solid (`style` = 1) or wire frame (`style` = 0). The

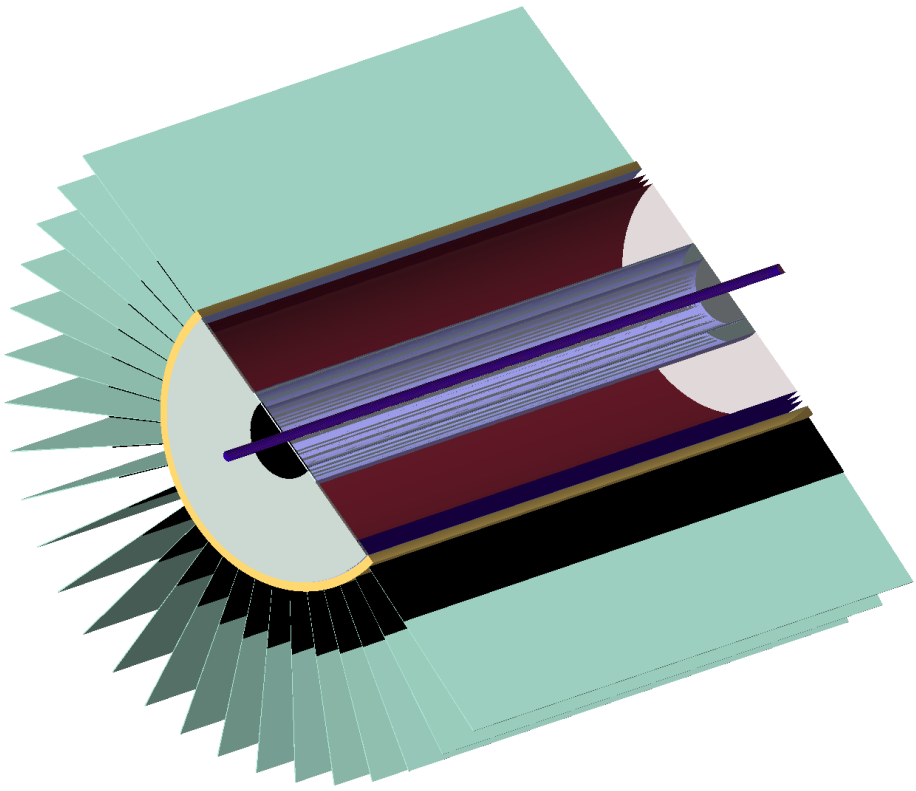

Fig. 1: BONuS12 RTPC in GEMC

`sensitivity` variable directs GEMC to add the output of this region to the correct bank in the output file. In order to define what particle-interaction output variables appear in the output file for the given volume, we use the `hit_type` variable. We'll talk more about the hit type in Section 4.3.3 when we discuss what to do when ionization occurs in the drift region.

Not all of the RTPC details can be implemented into GEMC, so we only include the important components in the GEMC simulation. That includes the main detector parts like the target, ground and cathode foils, GEM foils, and readout pad-board. Then there are the components that had to be included in order to understand their effect on the particles that may be traveling through them (*e.g.* down-stream end plate, electronics and translation boards, support ribs and spines, etc). Most of these secondary components had to be simplified in order to save time during

the simulation process. For example, a cylindrical volume of smeared density was included outside of the readout pad-board. This density included a proportional amount of the support ribs and spines, electronics, and air. The final geometry can be seen in Fig. 1, where a slice was taken in order to see the internal structure.

Once the geometry is set up for the RTPC, it must be inserted within the CLAS12 detectors in GEMC. Because the detector is to be in the center of the solenoid magnet, its center position for the RTPC is at $z = 0$ mm. That makes for positioning in GEMC easy, since the defaults for two more detector variables are zeroes. That is `pos`, which is for defining the position of the detector, is a vector of zeroes. The other, `rot` which defines the rotation matrix of the detector, is also filled with zeroes. Because these two are set to the default values, they don't have to be included in the detector variable definitions.

The file that brings all these detectors together in GEMC is an XML (*i.e.* extended markup language) file called a gcard. This file is where one defines not only which detectors to include in the simulation, but also what variables to include in the output file and what the incoming particle beam should be (*e.g.* momentum, angle, spread, etc.). For example, if one desired 10.6 GeV/c electrons to travel at 0° scattering angle θ and 0° around ϕ, with a spread of ±10 MeV/c in momentum, ±10° in θ and ±180° in ϕ, the code would be:

```
<option name="BEAM_P" value="e-, 10.6*GeV, 0.0*deg, 180.0*deg"/>
<option name="SPREAD_P" value="10*MeV, 10*deg, 180.0*deg"/>
<option name="BEAM_V" value="(0, 0, 0)cm"/>
<option name="SPREAD_V" value="(0.3, 20)cm"/>
```

Notice that in this code snippet, the vertex of the particle `BEAM_V` is set to zero and there is a spread on that vertex `SPREAD_V` of 0 cm $\leq r \leq$ 0.3 cm and -20.0 cm $\leq z \leq 20$ cm), which spans the diameter and length of the BONuS12 RTPC target.

This method of generating particles makes use of GEMC's internal event generator. The particles that can be generated makes use of the Geant4 particle bank. The trouble with this internal generator is that we don't have access to multiple particles that we may want to examine (*i.e.* secondary particles). For that we have to look toward another method of generating particles and how to import that file into GEMC.

4.3.2 EVENT GENERATOR

For the purpose of our GEMC simulations in BONuS12, we are primarily concerned with the reaction $eD \rightarrow e'pX$ and so we need a means of generating such particles. For that we use an external particle generator called Pythia. Pythia is a program for generating high-energy physics events, which is precisely what we need. It uses theory and models on collisions between particles like e^-, e^+, p and $\bar{p}$ (*i.e.* anti-proton) to generate output in a file format name Lund, after the University where the program was developed.

TABLE I: Lund file header

Column	Quantity
1	**Number of particles**
2	Mass number of the target
3	Atomic number oif the target
4	Target polarization
5	**Beam Polarization**
6	Beam type: electron=11, photon=22
7	Beam energy (GeV)
8	Interacted nucleon ID (2212 or 2112)
9	Process ID
10	Event weight

This Lund output file format has very specific variables that we can take advantage of in GEMC. The first line of this Lund file contains header information for the particles to follow from the collision simulation. This header contains 10 different columns, listed in Table I. The items in bold are used by GEMC. Given the number of particles listed under column 1, there will be a list below the header with particle details for each (see Table II). That is, if there is a 5 listed under the first column in the header, then below the header will be 5 lines for each of the particles. For a simulation with multiple events, subsequent events appear after the last particle of the previous beginning again with the header line.

For the BONuS12 experiment, the event generator created a Lund file with various electron-proton deep-inelastic collisions that we must run through GEMC. To do this, instead of utilizing the GEMC internal event generator, we include the code

```
<option name="INPUT_GEN_FILE" value="LUND, even_gen.lund"/>
```

TABLE II: Lund particles

Column	Quantity
1	**Index**
2	Lifetime [nanoseconds]
3	**Type (1 is active)**
4	**particle ID**
5	Index of the parent
6	Index of the first daughter
7	**momentum x [GeV]**
8	**momentum y [GeV]**
9	**momentum z [GeV]**
10	Energy of the particle [GeV]
11	Mass of the particle [GeV]
12	**vertex x [cm]**
13	**vertex y [cm]**
14	**vertex z [cm]**

in the gcard that we use to give direction to GEMC. This file will serve to instruct GEMC how many particles are in each event and the type of particle, its momentum and its vertex. For our purposes in BONuS12, we have Lund files with primary electron-proton event with a number of additional protons that serve as background. The number of these background protons can vary, but the intent is always to best represent what we would expect to see. Once we run this file through GEMC with all the other variables defined that have been previously discussed, we need to take a look at what happens in the simulation when these protons travel through the RTPC.

4.3.3 DRIFT ELECTRONS

By how much charged particles bend when moving through a magnetic field depends on the magnitude of that field throughout its path, so it's very important to get an accurate magnetic field map. The BONuS12 RTPC sits inside the solenoid magnet in CLAS12, so knowing the map of that field (*ie*. the magnitude and direction

of the field at small steps in space around the magnet) is crucial to both simulation of data and reconstruction of realistic data.

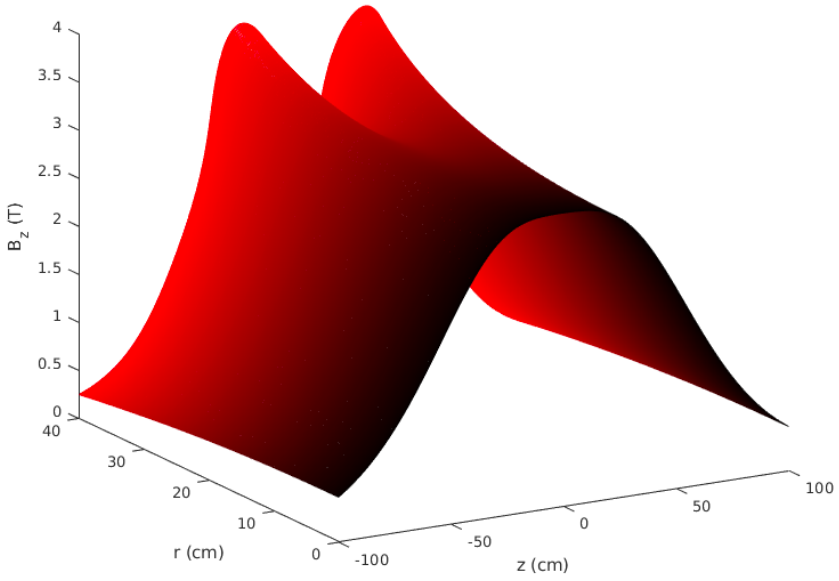

Fig. 2: CLAS12 Solenoid Field Map (V.Lagerquist)

This field map for the solenoid magnet in CLAS12 is mapped in steps of z and r (symmetric in ϕ) was developed by Victoria Lagerquist at Old Dominion University (see Fig. 2). The map itself takes as input course measurements of the B-field inside the solenoid and finds a more finely-structured field to use for other applications. One of those relevant applications is the simulation of protons traveling through the BONuS12 RTPC. GEMC does an acceptable job simulating those tracks. What GEMC does not do on its own is simulate the ionization electrons that are created by the protons traveling through the sensitive region of the RTPC.

For the simulation of those ionization electrons (also called drift electrons), we use the Garfield++ program. By starting electrons at different values of r throughout the sensitive region (*i.e.* 3 cm $\leq r \leq$ 7 cm) and using known values of the electric and magnetic fields, Garfield++ give us the time it takes that electron to reach the outer edge of the RTPC (*i.e.* 8 cm) as well as the change of angle that it makes. This drift time and drift angle, as those are known respectively, are of crucial importance to the BONuS12 experiment.

By knowing the drift time and drift angles of electrons starting at various values of r and z, we can plot the points and fit the points to an equation. We can then use these equations in GEMC to find the drift time and drift angle of a drift electron created at any point along the path of the proton in the sensitive region of the RTPC. In order to speed up simulation efforts, simulation electrons were created at $r = 3$ cm to $r = 7$ cm at 0.5 cm increments and $z = -19$ cm to $z = 19$ cm at 5 cm increments. This give us 81 data points to work with.

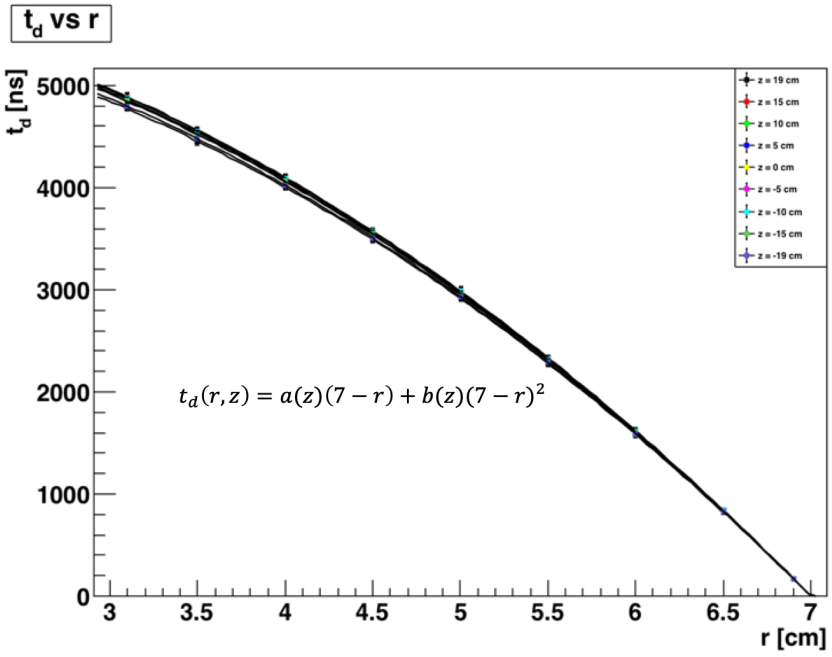

Fig. 3: Drift time (t_d) vs. r for drift electrons

4.3.4 EXPECTED RESOLUTIONS

4.3.5 DRIFT CHAMBER OCCUPANCIES

4.4 DMS SIMULATIONS

4.4.1 GEOMETRY

4.4.2 ELECTRIC FIELD

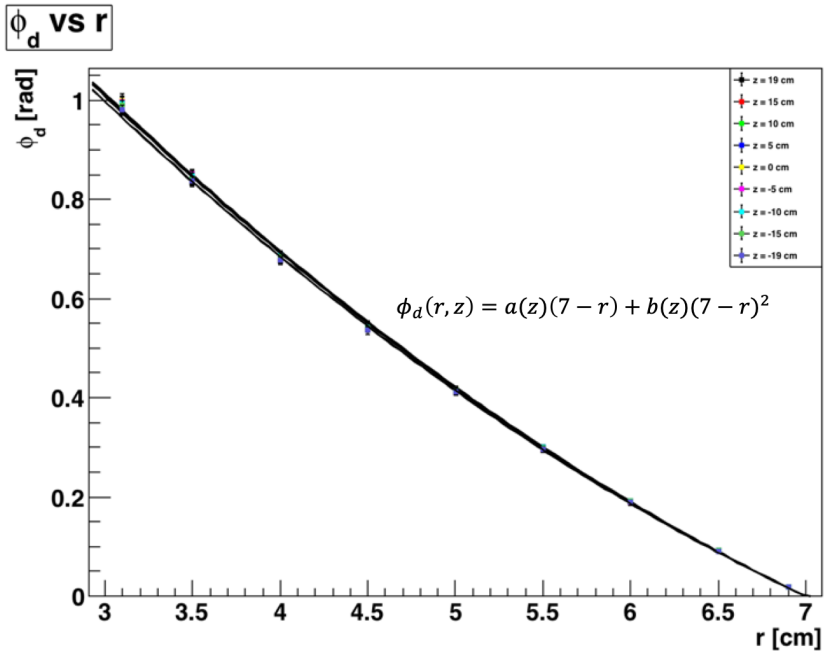

Fig. 4: Drift angle (ϕ_d) vs. r for drift electrons

4.4.3 DRIFT VELOCITY

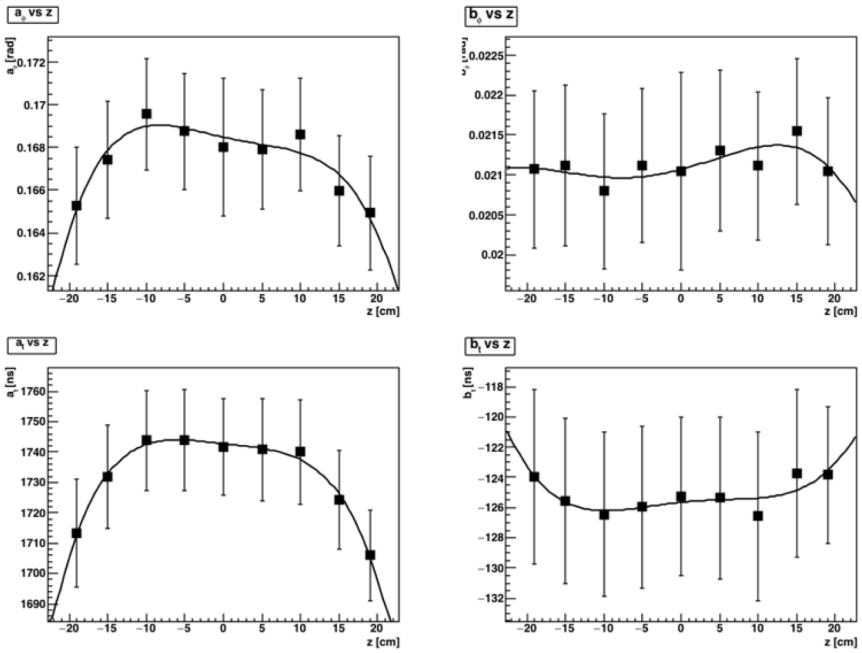

Fig. 5: Parameters a and b for drift electron equations

CHAPTER 5

DATA ANALYSIS

5.1 ELECTRON RECONSTRUCTION

5.2 PROTON RECONSTRUCTION

5.3 HELIX FITTER AND KALMAN FILTER

5.4 CALIBRATION

5.5 CUTS AND CORRECTIONS

5.6 KINEMATIC COVERAGE AND DATA BINNING

5.7 ACCEPTANCE CORRECTION

5.8 ELECTRON DETECTION EFFICIENCY

5.9 BACKGROUND SUBTRACTION

5.10 CROSS SECTION CALCULATION

5.11 RADIATIVE CORRECTIONS

5.12 SYSTEMATIC ERROR EVALUATION

CHAPTER 6

RESULTS

REFERENCES

[1] H. Van de Sompel, M. L. Nelson, and R. Sanderson, "HTTP Framework for Time-Based Access to Resource States – Memento, Internet RFC 7089." `https://tools.ietf.org/html/rfc7089`, 2013.

[2] R. T. Fielding, J. Gettys, J. C. Mogul, H. F. Nielsen, L. Masinter, P. J. Leach, and T. Berners-Lee, "Hypertex Transfer Protocol – HTTP/1.1, Internet RFC-2616." `https://tools.ietf.org/html/rfc2616`, 1999.

[3] H. Van de Sompel, M. L. Nelson, R. Sanderson, L. L. Balakireva, S. Ainsworth, and H. Shankar, "Memento: Time Travel for the Web," Tech. Rep. arXiv:0911.1112, 2009.

APPENDIX A

SPECIES

TODO: To be updated later!

VITA

Nathan M Dzbenski
Department of Physics
Old Dominion University
Norfolk, VA 23529

TODO: To be updated later!

Typeset using LaTeX.