Simulations for the SoLID Detector

A thesis submitted in partial fulfillment of the requirements for the Bachelor of Science degree in Physics from the College of William and Mary

by

Jack Anderson

Advisor: Wouter Deconinck

Senior Research Coordinator: Henry Krakauer

Date: April 28, 2014

Abstract

A simulation program named remoll_solid was created for the purpose of testing experiments of the Solenoidal Large Intensity Detector (SoLID) detector, which is to be used in Hall A of the Jefferson Lab. It makes use of a large solenoidal magnet which houses the target and the initial area from which particles pass from the target towards the signal-detecting calorimeters. The detector is unique in its design due to its use of baffles, six plates with curved spokes that can be arranged to limit the energy range of particles passing from the target into the particle detectors at the back end of the design. The large, donut shaped calorimeters on the far side of the baffles were the area of particular interest for the simulations. There are two such calorimeters, a pre-shower and a shower detector.

The simulations focused on the Deep Inelastic Scattering (DIS) experiment, using a target of deuterium and beam energies of 6.6 GeV and 11.0 GeV. Simulations ran to test a class of results specific to the calorimters, and we made alterations to the event generator and hit counters until the results in the calorimeter matched the expectations, and we had an even distribution of hits over the calorimeters. The successful completion of this step allows for alterations of the placement and configuration of the calorimeters by simply altering the Geometry Description Markup Language (GDML) files that are contained within the programs folder. Additionally, creation of a User Interface began, and is still under construction, to make the program more accessible for collaborators.

Introduction

The project I worked on, under the guidance of Dr. Deconinck, involved the simulation of experiments for one of the next big detectors that will be placed in Jefferson Lab in Newport News, the SoLID detector. SoLID, standing for Solenoidal Large Intensity Detector, currently has three experiments approved for run-time at Jefferson Lab's electron accelerator. The first of these plans to use Deep Inelastic Scattering (DIS) to make precise measurements of the parity-violating electroweak asymmetry of polarized electrons scattering off quarks. Within this experiment is time alloted for running with three different targets: deuterium, hydrogen, and Lead-208[1]. The SoLID project is still in the planning stages, thus the simulations are imperative not only for predicting radiative corrections and background information, but also for finding the ideal configurations of the different pieces of the detector. Such simulations will indicate how the final physical version of the detector should be arranged.

The specific point of interest for my part was the Shashlik calorimeters at the back end of the detector were the primary focus. A Shashlik calorimeter consists of layers of scintillating and absorbing material, in this case lead as the absorbing material and a scintillating plastic to generate the signal for the PMTs to detect. The two calorimeters are placed at the back of the design, toroidal in shape and over 2 meters in radius. Their shape can be seen in Figure 4.

Method

My work on the project began in March of 2013, when I first looked at SoLID GEMC, a Geant4 program that made use of a preliminary design of the detector, and the MonteCarlo Method for generating events. At the time of this paper, Geant4 is the leading toolkit for simulations of particles interacting with matter[4]. Created by CERN, Geant4 comes with the option of a number of rigorously tested "physics lists," and one of which can be implemented in a simulation. These lists define the physical interactions that will occur between two particles based on the our most current understanding of the Standard Model. It soon became clear that SoLID GEMC had a number of issues that made both running the simulations and sharing results difficult, and with little documentation, it seemed that taking the project in a slightly different direction would be a good course of action. Over the summer, working jointly between William & Mary and Jefferson Lab, I began work on a new program, remoll_solid, that would pair an existing simulator and the Geometry Description Markup Language (GDML) files that described the geometries in SoLID GEMC. "Remoll" is a reference to the name of the simulation program I was writing the geometry of SoLID into, MOLLER, another future experiment planned for Jefferson Lab.

Having been designed by different groups, the two programs (SoLID GEMC and Moller) had different methods of reading through the geometry files, so the task became altering the geometry files pulled from SoLID GEMC so that they could be read by the coding of the Moller simulator.

Though both programs used GDML, an XML-based language for defining geometries[2], to implement the detectors, their process of parsing through the geometry files differed, and thus the twenty or so files that defined different aspects of the very large detector needed to be altered. GDML allows for the creation of different pieces of material, of different shapes, to be placed in volumes defined by the writer of the code. These volumes can then be used, moving the entirety of the material with just the alteration of a few numbers in the mother file that brings all the aspects together.

The components I originally focused on when first writing the program were the target, the spoked baffles, the calorimeters, and the nosecap of the

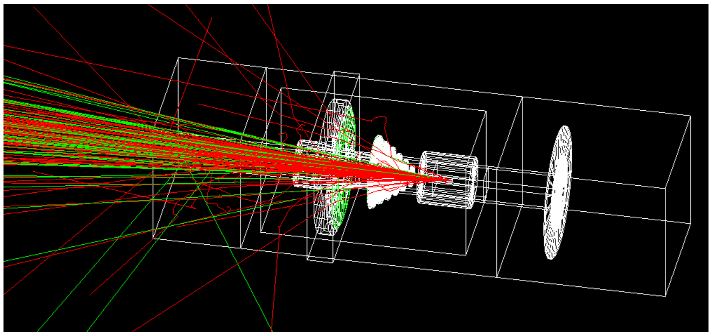


Figure 1: An early run of the program with just a few parts of the detector

beamline. I felt these were the components imperative to testing if this new program would be a viable replacement for SoLID GEMC. The different aforementioned pieces involved varying levels of difficulty (the six baffles required over 1,000,000 lines of code), but once complete, the nature allowed for easy rearrangement of the different volumes without having to open any file but the mother file. All three targets, (deuterium, liquid hydrogen and lead-208) were added in for ease of access by others interested in using the program.

Each step of the process required opening the program with visualization macros to ensure that the program was generating the proper geometries. Once this was complete, the next thing that needed to be done, in order to simulate the proposed experiments for SoLID, was to create a field map file, a file that defined a magnetic field at points throughout the detector. This was a bit tricky, with the large size and solenoidal shape of the detector. Dr. Seamus Riordan, the creator of the Moller simulator, was helpful in this regard, and I soon had a very rough, but working, simulator for the SoLID detector with GDML-based geometries.

It should be noted at this point that the field map file, due to its enormous size, cannot be added to the folder of the program and must copied from a separate link.

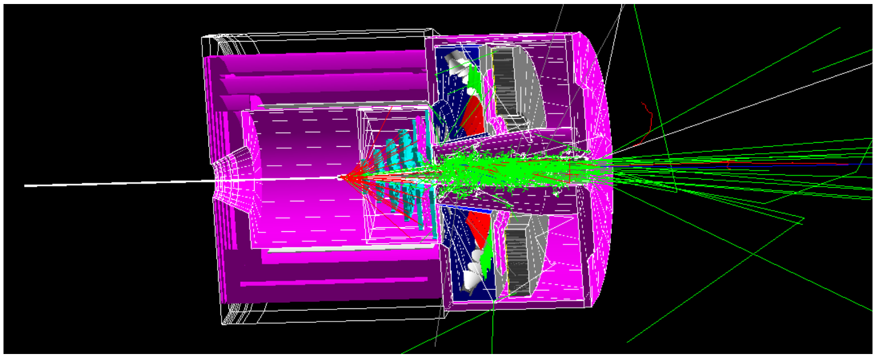


Figure 2: The current layout of the SoLID simulation

Though the design for the detector has not been finalized, the simulation is meant to reflect our best approximation as to what it will look like, and can easily be altered as we find necessary changes to the eventual physical version of the detector.

The area of particular interest to me was the Calorimeter, the largest particle-identifying piece of the detector, and the component of the detector that measures energy the particles. The simulation generates a ROOT file with each run, and in the ROOT files are leaves that record various kinematic properties of events from the run. ROOT is an object-oriented set of frameworks which allows for easy storage and organization of files with large amounts of data[3]. Each parameter of interest, such as the type or energy of a particle, is defined in the the coding for solid_remoll (which was taken from remoll) and a "leaf" is created for each of these parameters, containing all the pertinent information. Among these leaves is a class that gives properties of tracks just within the calorimeter, but at first the information contained in the class did not match our expectations. At first I thought it might be an error with the simulation itself; that the tracks were not making their way to calorimeter as we had hoped. But an analysis of the hits throughout the entire detector, but restricted to the coordinates of the calorimeter, proved that this was not the case. Rakitha Bemminiwattha, a post-doc based out of Jefferson Lab, discovered that the sum counter for the hits for each event was not resetting after each event, causing the sum to quickly exceed the maximum allowed by the file that created the ROOT tree. The coding called for tracks that exceeded set maximums to be thrown out. This prevented data from being stored in the tree. Remedying this error resulted in much more sensible findings in the simulations.

A simple graphical user interface was also created for operating the simulation. It uses G4UIxm, Geant4's class for creating interactive interfaces. Unfortunately, currently the code only allows for the addition of menus and buttons linked to specific commands, so the user's flexibility with settings is contingent upon the depth to which the programmer goes to create a variety of buttons.

Analysis

The current status of the program is that it works for DIS, π^+ and π^- generators, and properly tracks hits in the calorimeter, for both the pre-shower and shower layers of the calorimeter. The majority of the particles detected in these layers, with the DIS generator, are electrons and photons, with a significant amount of neutrons and trace amounts of quarks and protons. We would have liked to see evidence of pions from the DIS generator as well but unfortunately there were no such tracks in the calorimeters. We can however do a pion/electron separation analysis by comparing files from the DIS and π^+ generators.

The user interface currently works to initialize the program, set the beam generator and alter the initial event parameters (the range of angles through

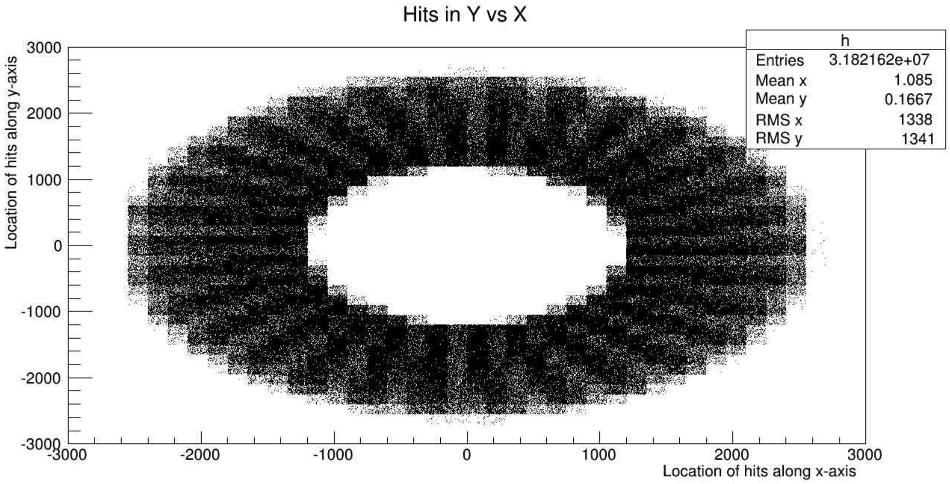


Figure 3: Y vs X distribution of hits in the shower layer of the calorimeter

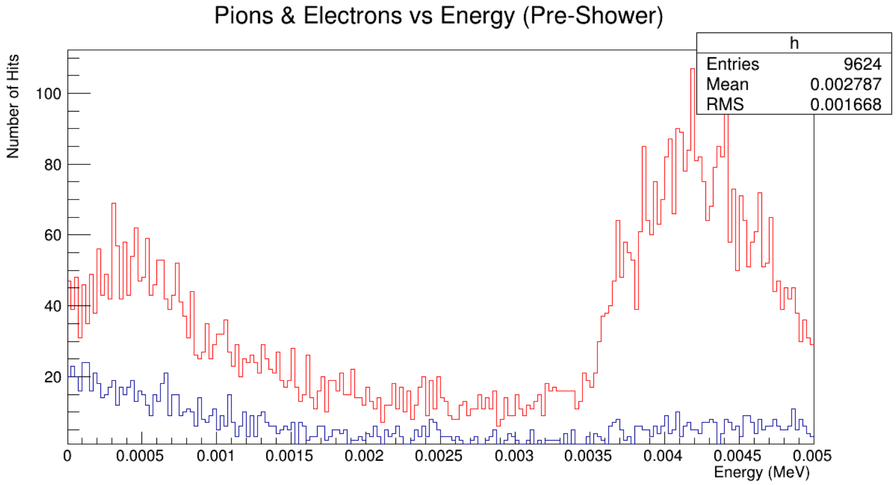
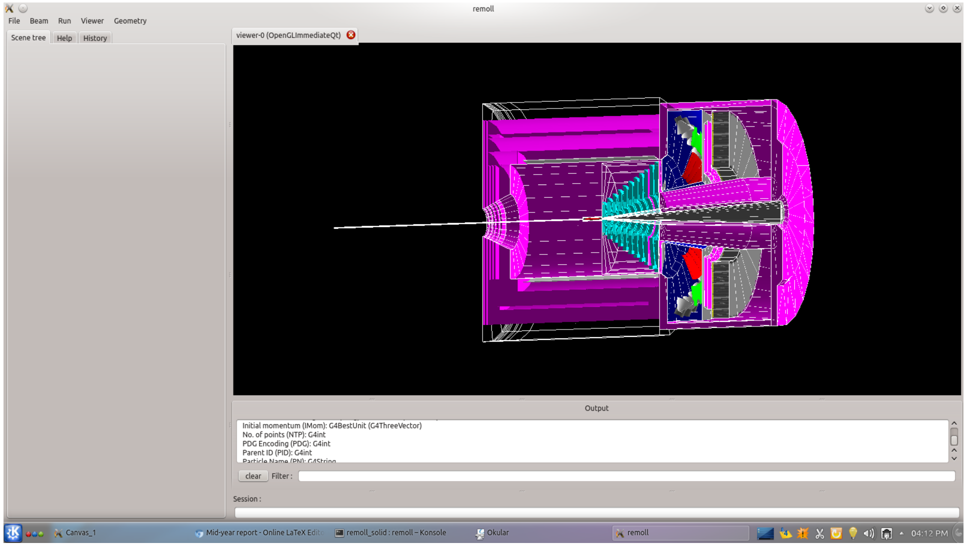


Figure 4: Pion/Electron separation for the Pre-shower layer. Electrons in red.



```
Initial momentum (IMom): G4BestUnit (G4ThreeVector)
No. of points (NTP): G4int
PDG Encoding (PDG): G4int
Parent ID (PID): G4int
Particle Name (PN): G4String
```

Figure 5: The user interface for remoll_solid. The dropdown menus include buttons to set initial conditions for runs of the program

which the events are permitted to escape the target), but UI still has trouble calling the field map, and thus simulations still need to be run by calling macros.

Conclusion

Some kinks still need working out, as we hope to find a suitable cut-off energy based on pion/electron separation. We hope for a pion rejection rate of 200:1. The pre-shower analysis such a separation is satisfactory, but in the shower layer of the calorimeter there is no discrepancy between energy levels of the different particles.

Simulations will be run to mirror those already recorded for SoLID GEMC, and once we have (hopefully) verified that remoll_solid can be used to generate the same data as SoLID GEMC, members of the SoLID collaboration can move over to remoll_solid and will no longer have to use SoLID GEMC. Currently, remoll_solid is located on a GIT repository, making easy access for it to anyone, as opposed to some programs which require authorization for access.

A readme will be necessary for users to understand how to alter the GDML files to fit their specific needs, as well as how to operate the user interface (or customize it).

Literature Cited

1. Jefferson Lab Facility, SoLID Hall A Collaboration. Submission at PAC 34
http://hallaweb.jlab.org/collab/PAC/PAC34/PR-09-012-pvdis.pdf

2. Geometry Description Markup Language for Physics Simulation and Analysis Applications R. Chytracek, J. McCormick, W. Pokorski, G. Santin IEEE Trans. Nucl. Sci., Vol. 53, Issue: 5, Part 2, 2892-2896

3. ROOT Team, The. "ROOT." . CERN, 1 Jan. 1995. Web. 3 Apr. 2014. ¡http://root.cern.ch/drupal/content/about¿.

4. "Geant 4." . CERN, 21 Mar. 2014. Web. 3 Apr. 2014. ¡http://geant4.cern.ch/¿.