Simulating the Response of a New β-Delayed Proton Detector

M. Roosa

April 2, 2018

1 Introduction

Two common thermonuclear astrophysical explosions, classical novae and type I x-ray bursts, depend heavily on the reaction rates of proton rich nuclides, notably ^{30}P(p,γ)^{31}S and ^{15}O(α,γ)^{19}Ne [1,2].

In order to place the necessary experimental constraints on these rates, a new micro pattern gas amplification detector (MPGAD) has been designed and built for use at the National Superconducting Cyclotron Laboratory on MSU's campus (Fig. 1). The Proton Detector will indirectly measure those reactions through the beta delayed charged particles released in the β-decays of ^{31}Cl and ^{20}Mg [3]. It has separate drift and proportional amplification (amp, for ease) regions provided by a flex-board field cage and a Micromegas structure [4]. These regions are characterized by the magnitude of their respective electric fields. The ionized charge, from decays inside the detector, is carried to the measurement pad (pad plane) by these electric fields. The drift region has a corresponding electric field large enough to transport charge but not cause it to multiply. This moves the charge from the ionization site to the beginning of the amp region (mesh) while conserving the number of ionized particles. The amp region, which has a much larger field, then multiplies the charge so it can induce a measurable signal in the pad.

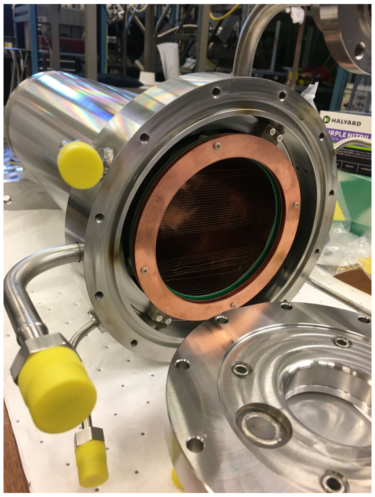

Figure 1: The Proton Detector, opened, with field cage visible.

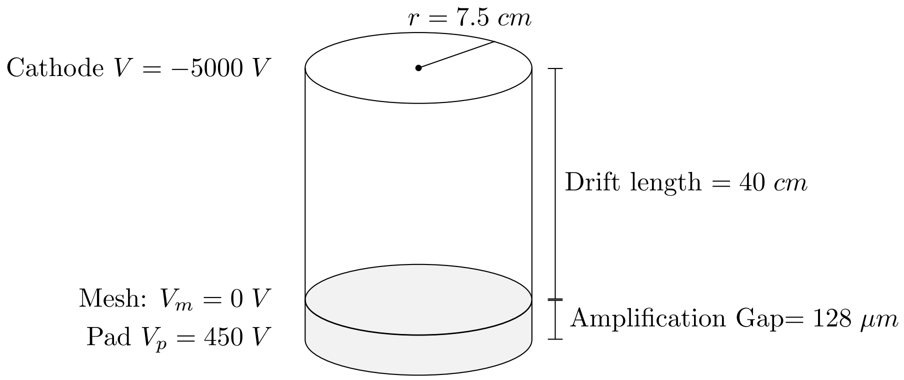

Figure 2: Pictured is a rough schematic of The Proton Detector's dimensions with typical operating voltages. Note: this gives electric fields for the drift region $\vec{E}_{drift} = 125\ V/cm$ and amp region $\vec{E}_{amp} = 35,156\ V/cm$ both pointing up the page. These are the same as assumed for the results section.

1.1 Project Goals

The focus of this project is to develop a simulation to model the electric response of The Proton Detector using Garfield++ and its attached libraries. Explicitly, we want to characterize how electrons ionized by nuclear decays within the detector, will move through the gas filled volume of the detector (drift) and how they will multiply in its regions of large electric field (amplification). Further, we want to know how the drift and amplification are effected by the choice of:

- Operating voltages
- Gas composition
- Detector dimensions, specifically amp gap.

A reliable simulation of electron transport through the drift and amp regions will provide a basis on which to make technical decisions regarding the operation of the detector with regard to the above.

2 Simulation Details

2.1 Garfield++ and Transport

Garfield++ uses a microscopic Monte Carlo method to simulate electron transport and gas properties. In application it calculates the paths of individual particles and the results need to be analyzed over a large number of simulated events. For an overview of Monte Carlo methods see [7] chapter 11.

2.2 Detector Specifications

The geometry in Garfield++ was chosen to match the physical Proton Detector. The simulated volume is composed of two stacked cylinders, one for the amp region and one for the drift region (see Fig. 2). The program assumes uniform electric fields for the amp and drift regions and takes theses field strengths as inputs. By choosing the mesh to be at ground it can automatically set the voltages at the cathode and pad.
Because of the change in length scale from the drift to the amp region (3 orders of magnitude) the simulation considers their transport separately. The final positions, energy, and time are recorded at the end of the drift region and are then used to initialize new avalanches at the beginning of the amp region. This prevents problems with the transport algorithm's adaptive step sizing and provides any easy way to record the positions and velocity of electrons at the mesh.

2.3 Initializing Events

Most tests, with the exception of early gain calculations, were executed with electrons at 6 keV. This was primarily motivated by the availability of the ^{55}Fe X-ray source, which has a 5.9 keV photopeak, used to test the real detector. These initial electrons were assigned an isotropic, random, starting direction with only negative z components (pointing from the cathode to the pad plane, See Fig. 2).

3 Results

The results and figures listed here are generated with the following assumed operating conditions. These are typical of current use and provide a basis for confirming the validity of simulated results.

Initial energy	$\mathcal{E}_0 = 6\ keV$
Amplification Field	$\vec{E}_{amp} = 35.156\ kV/cm$
Drift Field	$\vec{E}_{drift} = 125\ V/cm$
Events	N = 1000
Amplification Gap	128 μm

To describe the spread of some distributions we will use resolution, defined as the ratio of the full width at half maximum (FWHM) of a distribution over its mean. Which for a Gaussian is:

$$\mathcal{R} = \frac{FWHM}{Mean} = \frac{2\sqrt{\ln(4)} \times \sigma}{Mean}$$

Where σ is the RMS (variance) of the Gaussian.

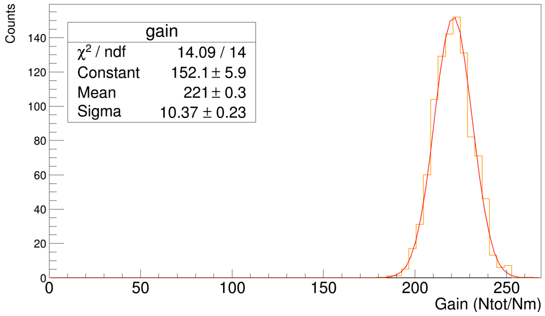

Figure 3: Simulated gain spectrum with fit of 6 keV electrons at typical operating conditions in P10.

3.1 Gain

Predicting the gain across the amplification gap is valuable because lets us infer the energy of the incident particles from the signal they induce on the pad. This, in effect, makes the detector a calorimeter. We can do this by working backwards. By knowing the number of electrons at the pad N_{total} and the gain we can know the number of particles at the mesh. The counts at the mesh is directly related to the number of ionizations produced by any secondary particles from nuclear decays in the gas. By knowing the ionization energy of the gas we can know the energies of those nuclear decays. This works specifically because the drift region is operated at a voltage too low to accelerate electrons fast enough to multiply.

$$Gain = \frac{N_{total}}{N_{mesh}} \qquad E_{event} \approx \frac{N_{total}}{Gain} \times E_{ionization}$$

After simulating the transport of 1000 events under typical conditions with the 128 μm amplification gap we found a gain of 221 with a resolution of 11.02%. The resolution is consistent with simulation and experiments of the ^{55}Fe 5.9 keV photopeak, for example [3, 5] respectively. Because of possible investment in a detector with amplification gap 64 μm, similar simulations were run for the new amplification gap and the comparison of results can be found in Table 1 with a summary of results from other tests.
In order to inform decisions about the choice of gas to use with the detector we also tested the gain through several gas mixtures (Fig. 4) with specific interest in the P10 and 5% Isobutane + 95% Ar mixtures.

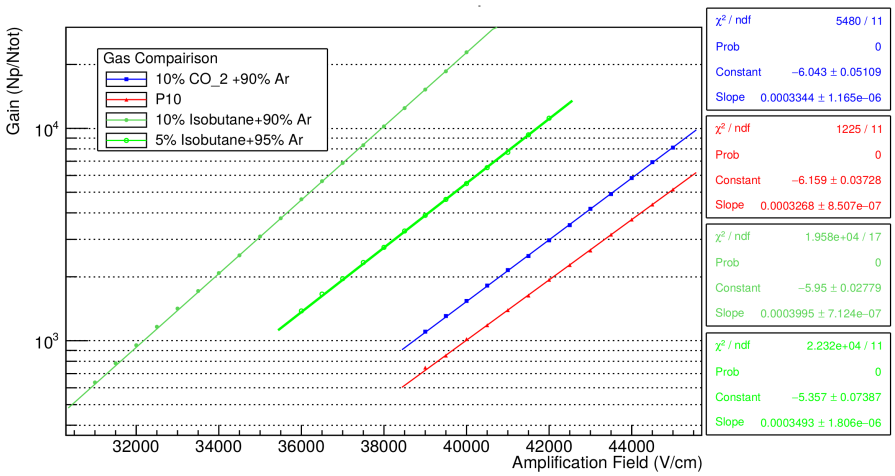

Figure 4: Gain plotted against amplification field for several gas mixtures. This simulation was specialized with different dimensions to cut computation time. Fits are of the form $f(x) = e^{p0+p1*x}$ where $p0$ is the "constant" and $p1$ is the "slope" . The fit details for each gas appear from top to bottom in the same order as the legend.

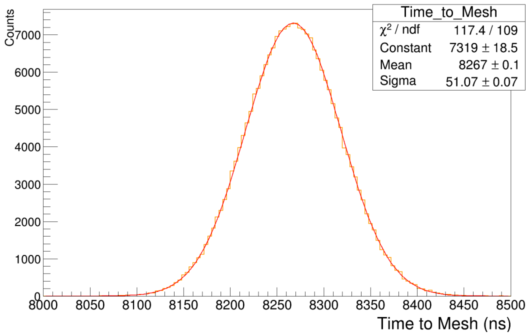

Figure 5: Simulated distribution of arrival times at the mesh with fit. Taken after 40 cm of drift through P10.

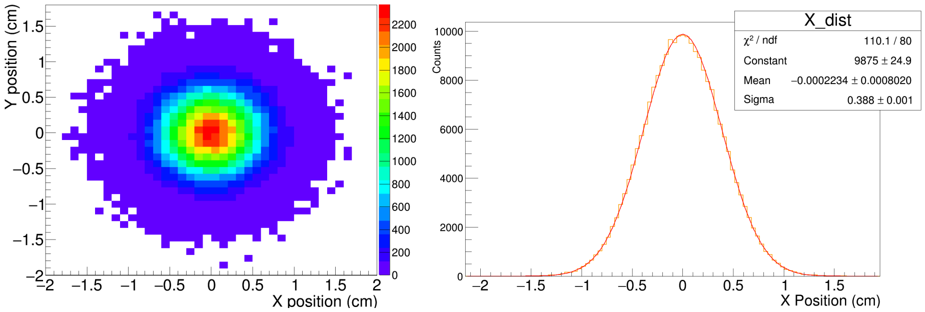

Figure 6: Simulated spatial distribution of particles in X and Y visualized together. Taken after 40 cm of drift through P10.

3.2 Space and Time distributions

Although The Proton Detector in phase one is focused on calorimetry, it is important to have the capacity to predict distribution of times of flight (Fig. 5) and the spatial distribution (Fig. 6) of electrons within the detector. In phase one this will give us information about the inter-pad charge deposition. This will become more important as we move into phase two, which will involve upgrading The Proton Detector with a higher granularity pad plane making it a Time Projection Chamber (TPC).

Value	Mean	RMS.	Resolution
X Position (cm)	−.0002 ± .0008	0.338 ± .001	-
Y Position (cm)	.0003 ± .0008	.3879 ± .0006	-
Time of Arrival (ns)	8267.0 ± 0.1	51.07 ± 0.07	1.45%
Gain	221.0 ± 0.3	10.3 ± 0.2	11.02%
Gain (64 μm amp gap)	25500. ± 100.	890. ± 80.	8.14%

Table 1: Tested values in P10. Assume Amp. Gap = 128 μm unless otherwise stated.

4 Conclusion and Prospects

Garfield++ offers the framework to pick up where other simulations like Geant4 leave off by calculating electron transport and gas diffusion. With it we have developed a simulation that, with the help of the NSCL batch servers, can provide high statistics electric response predictions for The Proton Detector. We are currently working to establish the validity of simulated results with success thus far. Once confirmed the simulation will be used to help choose operating conditions for The Proton Detector by further exploring its parameter space. Additionally, the simulation can be expanded to include more realistic fields and incorporate initial particles other than electrons.

References

[1] C. Iliadis, A. Champagne, J. José, S. Starrfield, and P. Tupper, Astrophys. J. Suppl. Ser. 142, 105 (2002)

[2] A. Parikh, J. José, F. Moreno, C. Iliadis, Astrophys. J. Suppl. Ser. 178, 136 (2008)

[3] E. Pollacco, L. Trache, E. Simmons, A. Spiridon, M. McCleskey, B.T. Roeder, A. Saastamoinen, R.E. Tribble, G. Pascovici, M. Kebbiri, J.P. Mols, M. Raillot, Nucl. Instrum. Methods Phys. Res., Sect. A 723, 102 (2013)

[4] S. Giomataris, Ph. Rebourgeard, J.P. Robert, and G. Charpakb, Nucl. Instrum. Methods Phys. Res. A 376, 29 (1996)

[5] T. Dafni, E. Ferrer-Ribas, I. Giomataris, Ph. Gorodetzky, F. Iguaz, I.G. Irastorza , P. Salin, A. Tomás, Nucl. Instrum. Methods Phys. Res. A 608, 259 (2009)

[6] `https://garfieldpp.web.cern.ch/garfieldpp/`

[7] Hjorth-Jensen M., *Computational Physics*, University of Oslo (2013) `https://github.com/CompPhysics/ComputationalPhysicsMSU/blob/master/doc/Lectures/lectures2015.pdf`

[8] `https://portal.frib.msu.edu/bussupport/businessit/ITSystemsSupport/Shared%20Documents/Manuals/SLURM%20Batch\%20System\%20Getting\%20Started.pdf`

A Operating Procedures

A.1 Prerequisites

This simulation requires that ROOT(v6.02 or compatible) and Garfield++ be installed. To check if these two are working appropriately, find the Gem example in $GARFIELD_HOME/Examples/Gem and try making and running it. If it works in $GARFIELD_HOME but can not in your working directory you should ensure the environmental variables for Garfield++ and ROOT are added to the path.

A.2 Interactive Mode

Once Garfield++ and ROOT are working you can simulate events. If you are looking to produces graphics of the drift lines it is easiest to run the program interactively. In your working directory:

1. Open pdet.C and set your desired parameters. This can include changing:
   - What plots are turned on
   - Starting point of the initial particle
   - Gas properties (e.g. composition, pressure)
   - Any of the defaults listed below, etc.
2. Type ”make” into the terminal (this will run the makefile and compile the executable pdet)
3. Run pdet with the following command:

   ```
   ./pdet {int N} {double E} {string out_file} {double Amp_Field}
   ```

All of these variables have default values that can be changed for convenience. By default:

N = 1 Events
E = 6000 eV
Save the data to ”test.root”
Amp_field = 40 kV/cm

As an example, if I wanted to simulate:

10 Initial Photo-electrons
With 5.7keV starting energy
Save the data to ”test.root”
With an amplification field of 32 kV/cm

I would type the following in terminal and run:

```
./pdet 10 5700 test.root 32000
```

You can expect 6 keV electrons drifting 40 *cm* of P10 to take roughly 50 minutes of computation time.

A.3 Batch Mode

Alternatively, if you are trying to produce a large number of events, as is usually the case, running in batch mode is preferred. I have made a seperate script for batch mode operation (currently: pdet_nog.C). This is essentially the same as before except it has had all of the graphics implementation removed and only stores data that is of experimental interest(this is changed easily). This makes it quicker and take up less memory.
Because the TORQUE server at NSCL (*seaside*) does not have the latest implementation of ROOT, the simulation only works on the SLURM server (*fireside*). NSCL documentation for SLURM can be found in [8].
Edit all scripts in *fishtank* before logging into *fireside*, which is accessed from *turqoise* with:

```
>ssh ember.compute
```

From here access your working directory as usual. I have found it easiest to submit jobs using a pair of bash scripts. One with the commands to submit one job to the queue and another to run the first in a loop for the desired number of iterations. These are both short enough to include explicitly.

First I create a bash script to submit one job with the desired parameters listed in the previous section:

```
#!/bin/bash

#email update
#SBATCH --mail-type=FAIL
#time
#SBATCH -t 1:10:00

./pdet_nog 1 6000 out$1.root
```

The preamble here tells the server to email you if the job fails and asks for 1:10:00 (hours:mins:sec) of server time to complete the job. Smaller times will queue faster but you risk killing your job before completion.
Then I create a another script that loops the submit command:

```
#!/bin/bash

for i in {1..1000.1}
do
        sbatch sub.sh $i
done
```

Assuming that the previous script was saved in ”sub.sh”. The ”$i” here and ”$1” in the previous script lets you use the index of the loop (”i”) as an input to the ”./pdet” command. This is currently used to create unique, indexed names but has applications for incrementing other input parameters (e.g. testing a pressure range) with small modifications to the source code.