Undergrauate Research report

Gain Studies in Gaseous detectors

Ankit D. Mohapatra

Florida institute of Technology,
150 W University Blvd, Melbourne,
Florida, USA

E-mail: amohapatra2012@my.fit.edu

Contents

1 Introduction

In this report I have presented the work I did at Florida Institute of Technology under Dr. M.Hohlmann. I have given an outline of the simulation procedure and have also included the codes. This semester I worked on the simulation studies in Gaseous Electron multipliers (GEMs)(mostly related to zigzags) as well as Micro channel plates (MCPs). I have also included a brief description of the hardware work I did this semester.

2 Procedure for overall simulation

The gaseous detector simulation is done using a combination of different softwares. These include:

- We fist have to build a Computer Aided (CAD) model of the detector, and then apply suitable potential (or loads as they are called in Finite element Analysis). For our studies, we would be using Ansys. One of the reason why we choose Ansys is that it has curved solid elements (the ones which are used for meshing) and hence give a better estimate of the potential, and electric field. After the potential is solved, it is written to a list file (.lis files) in a format that can be read by Garfield/Garfield++. The problem that comes with the use of finite element softwares is that they give an error as high as 50 % near the electrodes. To avoid this error, there has been lot of interest to use Boundary element Method (BEM) softwares. However, the BEMs still don't have an interface with Garfield++, and use of lot of computing time. However, in the near future, I hope that these shortcomings would be removed.

- Independent of the CAD model, one can do a simulation of the gaseous properties of the gas mixture used in a detector. We would be using Magboltz, which was written by Dr. Steve Biaggi. Magboltz takes as input cross-sections of various gases, and computes the drift velocity, electron energy distribution function, diffusion, townsend coefficients and other important gaseous detector properties. It has interface to Garfield as Garfield++. These simulation studies were done in the previous semester, and hence the results are not included in this report.

- The next step in the simulation is to read the field map files in Garfield. Garfield is a drift chamber detector simulation program written by Dr. Rob Veenhof. A recent C++ version, Garfield++ has been developed and in our simulation we would be using primarily Garfield++. Garfield primarily tracks the electrons, and stores the electron properties, which can then be used to compute the gain in the detectors, as well as their position and energy resolutions.

- Apart from gain, Garfield can also compute the signals in case of detectors, The signals can then be written to data files, which can then be processed further in PSpice.

3 Gain simulation procedure

The first step in any detector simulation is to understand what results one wants to study from the simulation. It is to be understood that not one single model can be used to study all the properties.
In this report we would be concerned more about the gain simulation. First we read the field map files from Ansys and store them as bin files, as the bin files are faster to read. The unit cell made in Ansys is then made periodic in Garfield/Garfield++. As stated earlier (1), we would be using Garfield++ (C++) and not Garfield (Fortran). The reasons for the choice are:

- The magboltz version used in Garfield are not updated, unlike Garfield++ which has the interface to the latest version. This would mean that the gases whose cross-sections may have been modified after 2008, would give slightly incorrect results in Garfield.

- The second, and the most important reason to use Garfield++ is that Garfield++ has routines which can track electrons in magnetic field. This is important as our studies for both Micro-channel Plates and GEMs involve magnetic field.

- Garfield++ is written using ROOT (A data analysis program developed by CERN) classes (the exact executable is called garfroot), and hence has access to the better statistical tools offered by ROOT, in comparison to Garfield.

Now that the cell structure is defined in Garfield++, one needs to identify the drift medium. Garfield++, by default chooses the medium with the lowest dielectric constant as the drift medium, which in our case happens to be gas. Next, the interface to Magboltz is called,

which then is followed by setting the gas composition, temperature and pressure. Normally in gaseous mixtures like Argon and Carbondioxide, the max electron energy hardly reaches more than 200 eV. In accordance with that, we set the max electron energy at 200 eV.
One of the parameters we get from Magboltz is the townsend coefficient. One can get the value of the gain by exponentiating the integral of the townsend coefficent over the entire detector length.

$$G = e^{\int \alpha(E/p)dx} \tag{3.1}$$

It turns out that the simulated gain obtained from this procedure is different from the measured gain, in some cases by a factor of 10. The discrepancy in gain is explained by including the penning transfer mechanism. Penning transfer is a group of processes where a gas having a higher excitation energy than the ionization energy of the the other gas in the admixture ionizes the gas. As an example, lets consider Argon-CO_2. The ionization potential of CO_2 is $\approx$ 13 eV. But argon has excitation energies higher than that, hence an electron from the excited state on collision (interaction) with a CO_2 molecule can ionize the molecule. The modification in gain due to Penning transfer is controlled by the transfer ratio. This ratio depends on the percentages of the component gas mixtures. For example, Argon-CO_2 in the ratio 70-30 has a transfer ratio of 57, a small modification in the gas composition has a considerable change on the transfer ratio. Apart from penning transfer, the other process which enhances the gain is Photon feedback. It may so happen in gaseous detectors that photons are produced (When the gas molecules deexcite from higher excited states) Unfortunately, this correction has not been added to Garfield++. To get around this problem, one scales the simulated gain with the experimental gain and modifies the transfer ratio. In future, this process would be added to Garfield++.
Garfield++ has different electron tracking functions. In our case we would be using the AvalancheMicroscopic tracking class. In avalanche microscopic tracking, an electron with an initial random energy (as given by the user) is drifted in an isotropic direction. Due to the electric field, the electron gains energy, and collides with the electron. Based on the energy of the electron after the collision the electron is either lost to attachment or produces a secondary electron. The central point of the tracking algorithm is matching electron energy after collision(s) with respective cross-sections. Apart from AvalancheMicroscopic, Garfield++ also has AvalancheMC, which also happens to be the tracking routine for ions. Until recently this was the default tracking algorithm. The accuracy of this algorithm depends upon the knowledge of the mean free path of electrons in gases, and is also the weakness in the algorithm, as it is known that the distance between collisions is not a constant in the gas mixture.
In the AvalancheMicroscopic method, one gives the starting energy to the electron as well as the starting direction and starting position, normally the energy given is about 0.5 eV. In principle, one could use any starting energy as the avalanche process is essentially independent of the initial electron energy. Essentially the electrons are tracked till they reach the anode, but in case one is just interested to study the gain in gaseous detectors, the tracking of the electrons can be stopped about 50 microns below the Gaseous detector structure. The main advantage of the approach is that it greatly reduces the computing

time, and at the same time is as accurate as tracking electrons to anode. The reason for the accuracy of this method in gain studies is that the avalanche primarily occurs inside the gaseous detector structure (the places of high electric field), and the energy of the electron is not optimal for attachment loss. One can easily see that this approach may fail in case we want to study the x-y distribution of the electron endpoints, to get an estimate of the position resolution. After the electrons have been tracked, we collect the total number of electrons and ions produced in an avalanche and corresponding to them we store the electron endpoints. One of the biggest difficulties in gain simulations is the need for high computing power, which means that one cannot run the program in one system and accumulate enough statistics to predict the trend in properties in various detectors. To overcome this difficulty we would be running our simulations as batch jobs in lxplus cluster parallely. This would mean that we could subdivide the whole simulation into multiple batch jobs each having say less number of avalanches per job. In case of magnetic field, one has to set the magnetic field at an an appropriate angle to the electric field and also enable the microscopic tracking routine in magnetic fields.

4 Position resolution and signals

In case of GEM detectors, lot of interest has been shown towards the use of zigzag instead of the conventional pas strips for readout. However there have been concerns over the use of zigzag in magnetic field. In order to study this effect in detail, one has to store the electron x-y endpoints for those electrons which have a z-coordinate that equals that of the anode. Normally in gain simulation, one just builds the anode as a continuous sheet of metal, but in case of electron endpoint simulation, we have to build the anode in a way that is specific to the problem. The problem comes from the fact that while the pitch of the GEM detectors (or for that matter MCPs) is of the order of microns, the pitch of the strips extend to millimeters. At first sight, one may be tempted to build a big GEM structure which would then capture the periodicity of readout strips. This structure would not be a good model of the actual detector as the error in the metal would be higher than usual.To overcome this difficulty, I proposed 2 different approaches. These are listed below.

- **Cut based approach:** This approach was inspired by the fact that the zigzags resembled the triangle wave, which is of the form $y = Sin^{-1}(Sin(\pi x))$. This could be easily extended to the case of the strips with appropriately defined regions (say like a $<$ x,y $<$ b). Although this approach seems plausible and elegant, on a second thought one can find a subtle flaw in the argument. This point becomes clear once one thinks of the change in the electric field pattern of a parallel plate capacitor when instead of using continuous sheet of metal one uses a segmented anode. The electric field lines are straight in case of the general parallel plate capacitor, whereas they would bend in case of segmented anodes. The change in pattern would be more pronounced in case of zig-zags.

- A second and an even more elegant way of doing the signal simulation would be to break the simulation into two phases-one avalanche phase (continuous anode plate)

and the other drift phase (pads/zigzags). In case of the avalanche phase, the simulation is same as the gain simulation, with the electrons tracked till they are 50 microns out of the lowest part of the gas detectors (the lower electrode). The beauty of the method lies in the drift part. Instead of giving the electron initial energy and the electron starting point as random numbers one could just give them the input as the endpoints (outputs) obtained from the first avalanche phase. The cell used for the second phase has no gaseous detector , and is nothing but a parallel plate capacitor with any imaginable shape of anodes. The distance between the plates is equal to the induction space of the gas-based detector minus the 50 microns the electrons have traveled and the potential difference between the plates is such that the induction field is reproduced. It is to be kept in mind that unlike the avalanche phase, no additional electrons are going to be produced in the drift phase, hence the tracking routine to be used is DriftElectron (which does not allow the electron to multiply) instead of AvalancheMicroscopic. As one can clearly see we have removed the problem with the pitches of the various components namely the detector itself and the readout (by making them completely different). The method draws inspiration from the solution technique of coupled differential equations to decoupled ones.

5 Results

5.1 GEMs

The gain in case of single GEM was reproduced in zero magnetic field(1). There was very good agreement with the data at higher potentials rather than with data at lower potential. The anomaly comes from the feedback effect, which could also account for the discrepancy between the transfer ratios.
Taking a step further, the gain simulation was done with different magnetic fields as well as different angles between E an B fields. One could see that with higher magnetic field as well as greater E-B angle, the gain decreased, with the effect becoming more pronounced at the extreme conditions (2 T and 90 ^{0})(2).

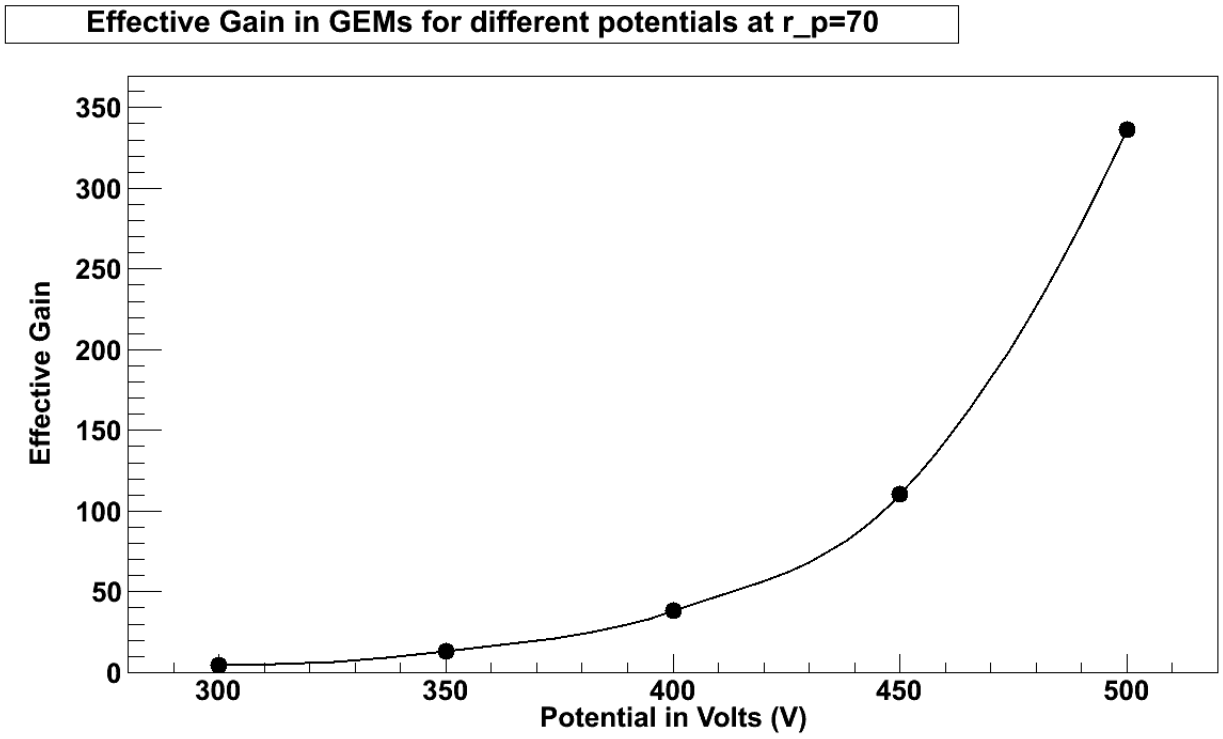

Figure 1. Variation of the Gain in Gems for different potentials

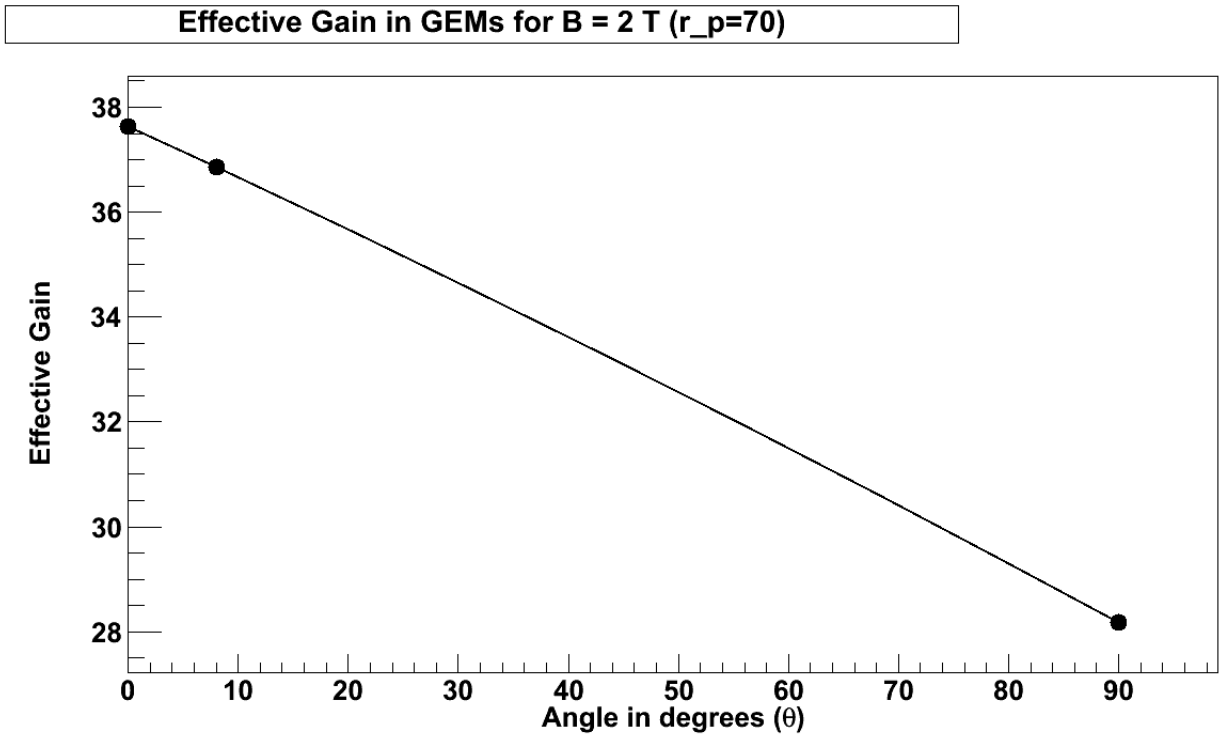

Figure 2. Variation of the Gain in GEMs for different E-B angles at B=2T

In order to study the x-y distribution, it is important to choose the correct potential. The main criteria for the choice being higher statistics and better agreement with the experimental data. The potentials used for gain study were from 300 V to 500 V in steps of 50 V. The 300 V logically have the highest statistics, are computationally less intensive, but do not agree with the data (The agreement comes at a transfer ratio of 100 % !). On the other end of the spectrum is 500 V which although is very computationally intensive, matches very well with the data. To strike a balance between both the factors, it seems that 400 V would be the most appropriate voltage. The results for the simulation are presented below both for magnetic as well as nonmagnetic fields(4). When the magnetic field is switched on, one can see clearly that the cluster moves in accordance with the Lorentz force. In gain simulation, one does just a single electron simulation, but in case of x-y distribution, one should track the electron cloud, and use heed. However if one magnifies the region of the starting position of the electron endpoint, then we can get a good enough approximation from single electron avalanche model only. 5

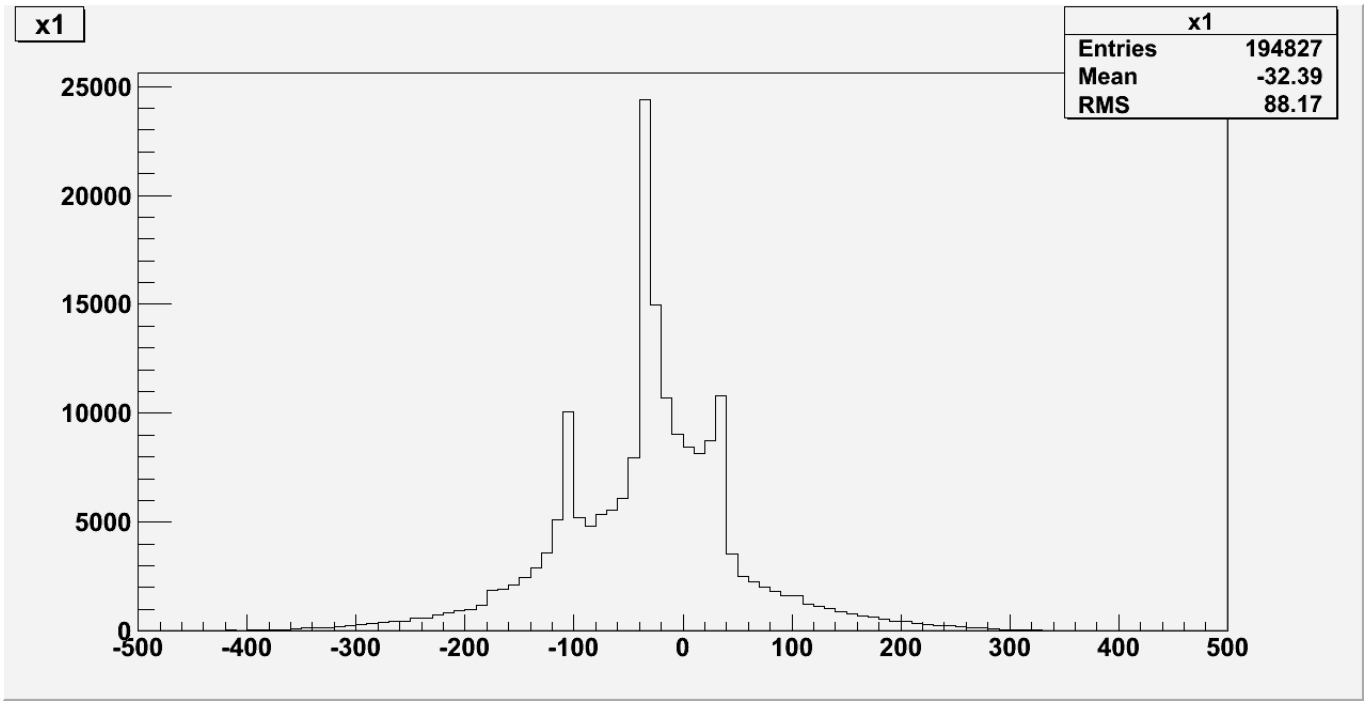

Figure 3. Plot of all the x coordinates of the electron endpoints from starting point of electron to anode

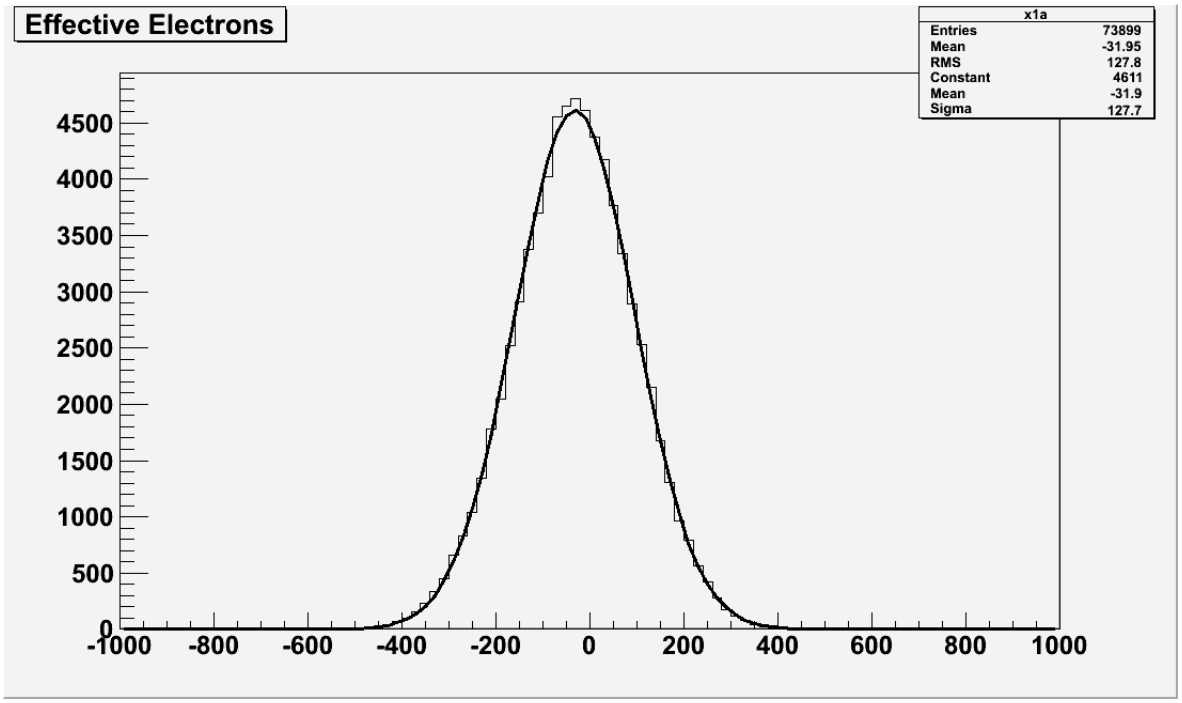

Figure 4. Plot of x coordinates of the electron end points for electrons which hit the anode

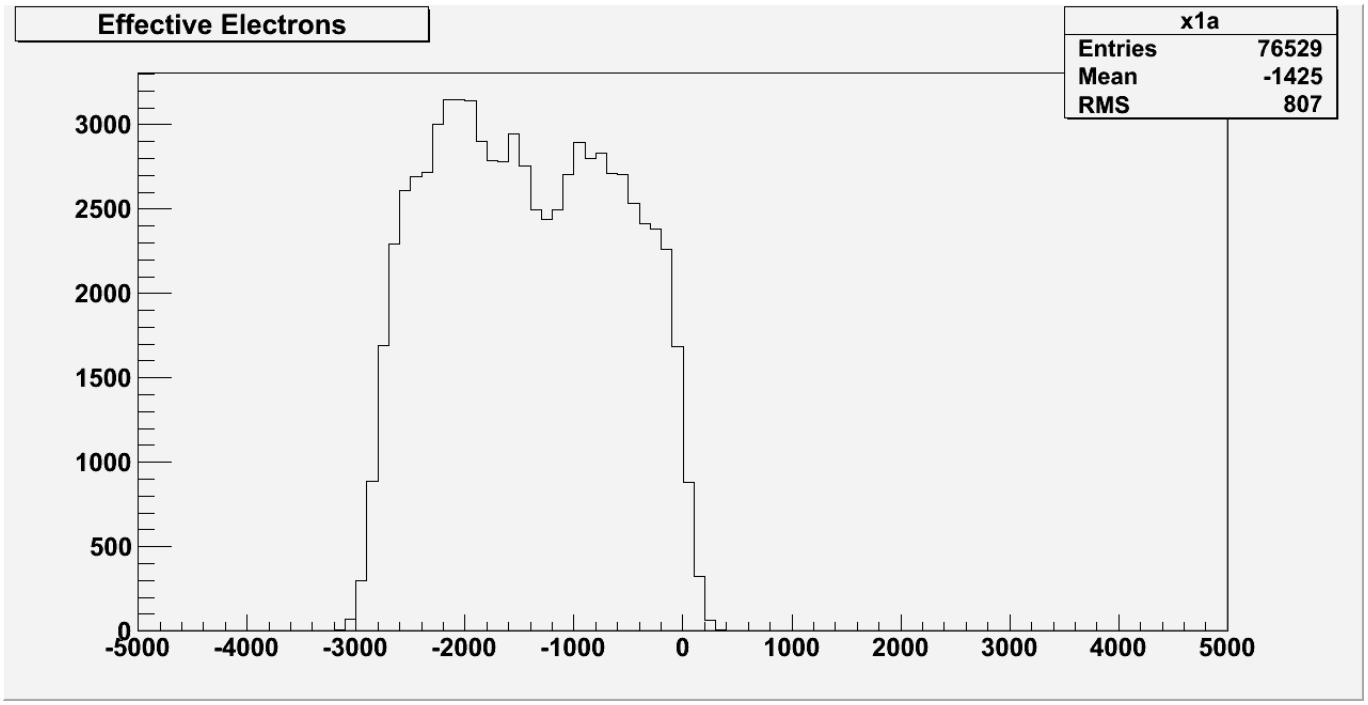

Figure 5. A look at the x distribution when the starting point is more spread out

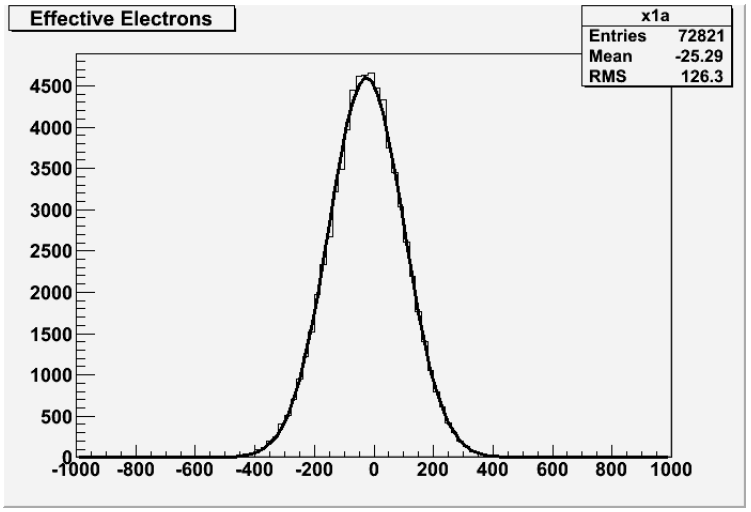

Figure 6. Plot of x distribution for B=0.5T and 0 ^{0}angle

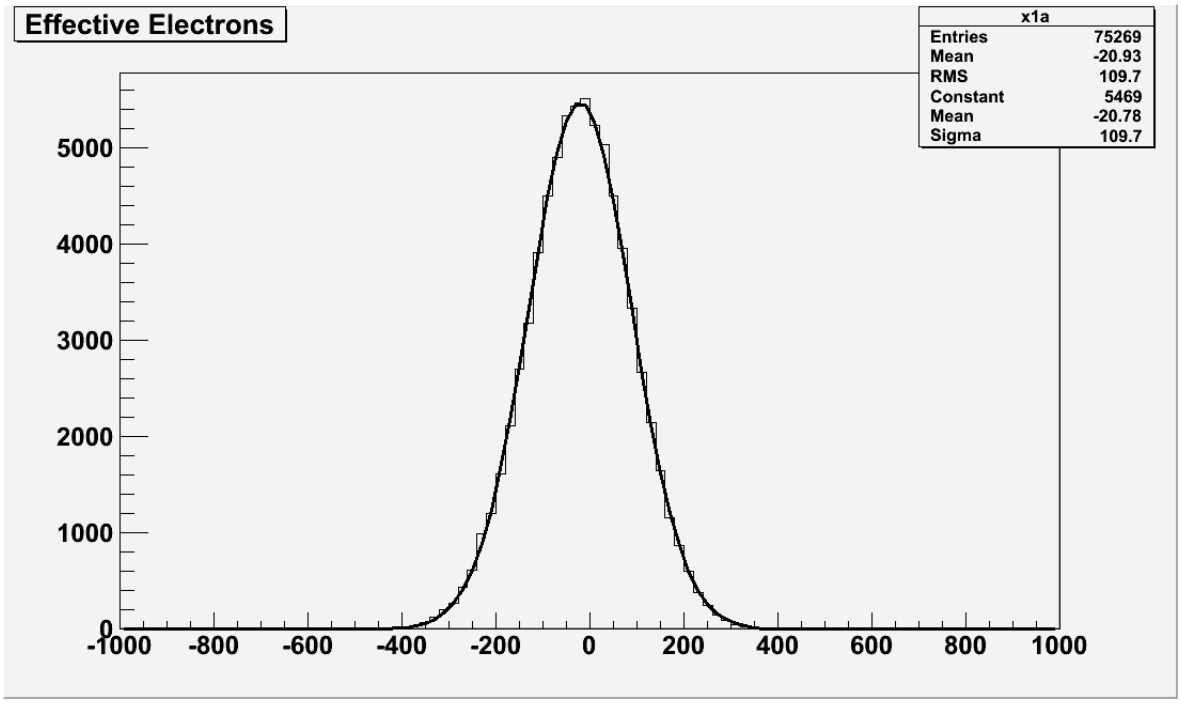

Figure 7. Plot of x distribution for B=2T and 0 ^{0}angle

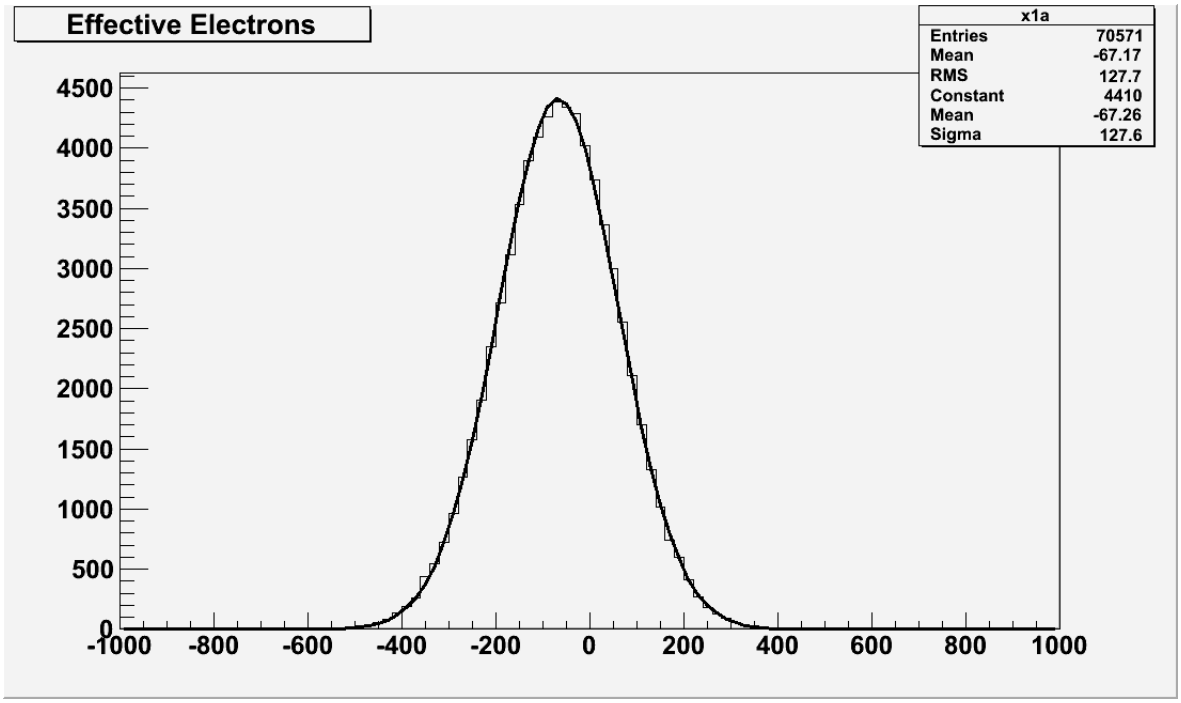

Figure 8. Plot of x distribution for B=0.5T and 8 ^{0}angle

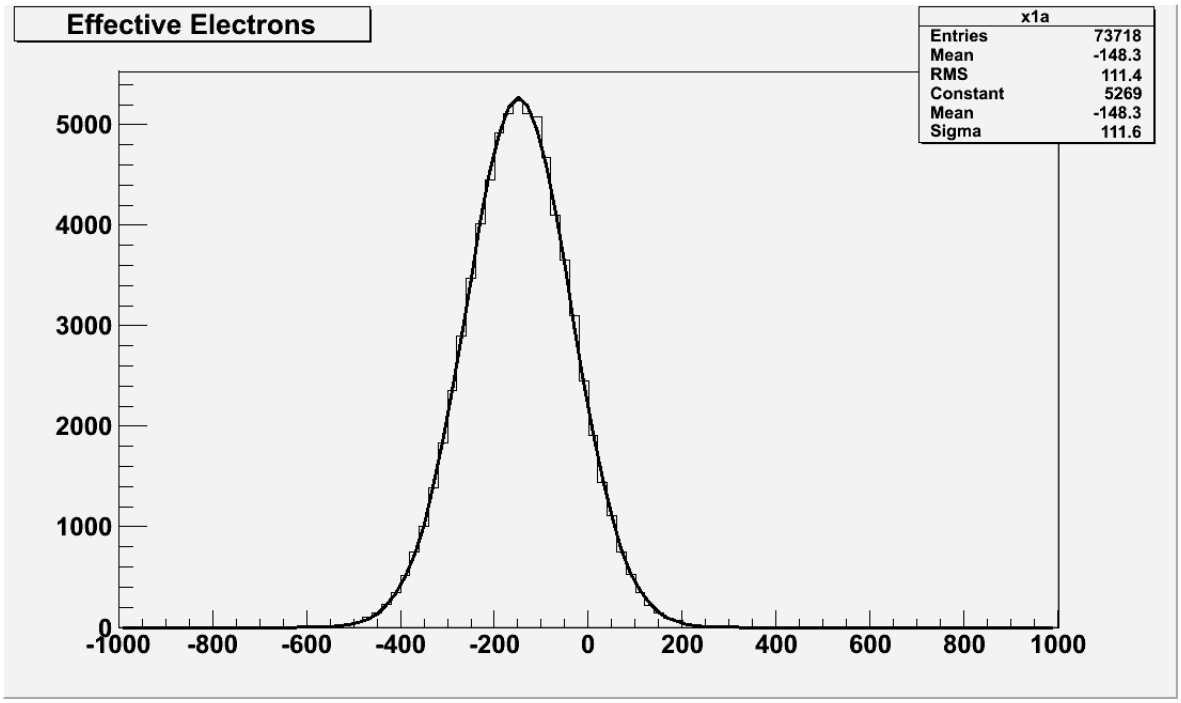

Figure 9. Plot of x distribution for B=2T and 8 ^{0}angle

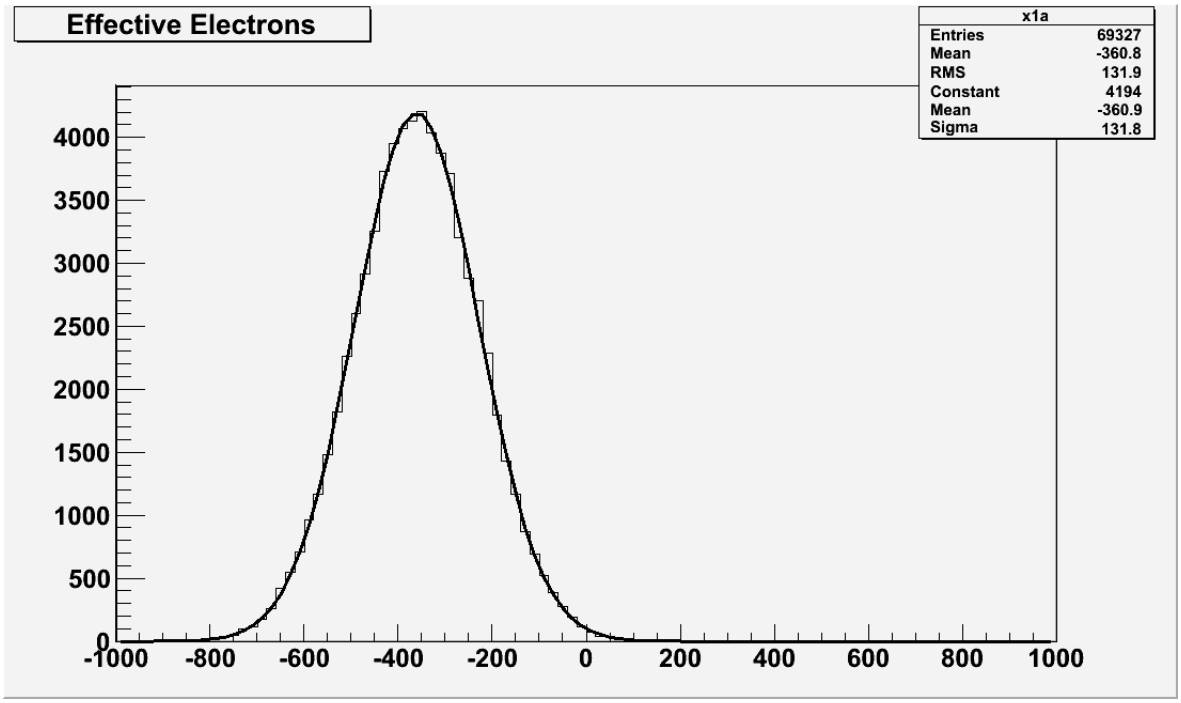

Figure 10. Plot of x distribution for B=0.5T and 90 ^{0}angle

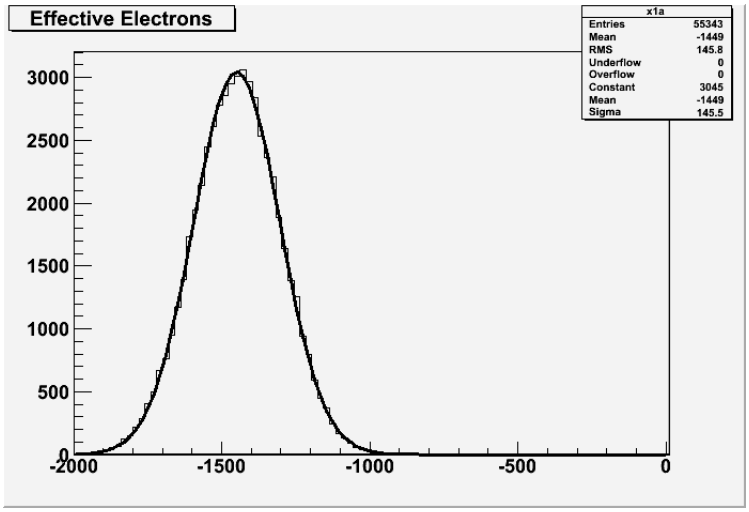

Figure 11. Plot of x distribution for B=2T and 90 ^{0}angle

5.2 Micro Channel Plates (MCPs)

There were some advances in MCP simulation as well. The most important was that of building of the MCP unit cell and the effective gain. The effective gain is the gain one would observe if there were no induction gap, and we were just collecting the electrons from the bottom surface of the MCP (similar to Peskov)

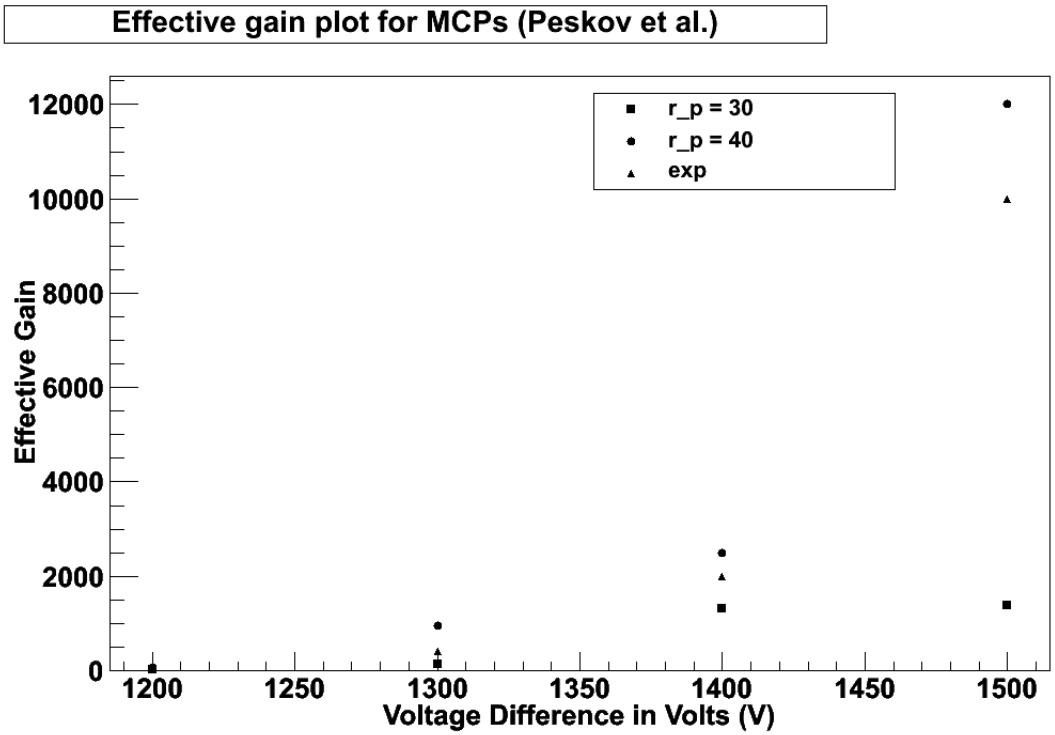

Figure 12. Plot of effective gain in MCP

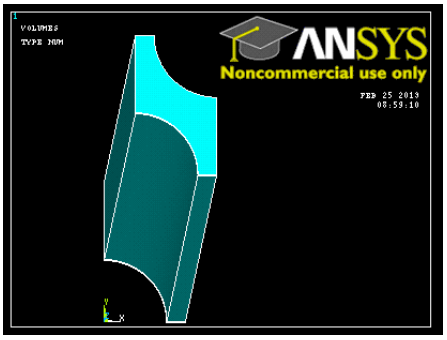

Figure 13. CAD model of the MCP

5.3 MCP Hardware

In the hardware side, the main problem arose from the choice of the cathode. The peek of the MCP because of it's height would tightly hold the MCP together, and hence there were concerns about the passage of gas inside the detector. As an alternative, it was suggested by Dr. Hohlmann that instead of using drift planes (metal plates) as cathodes, one could use GEMs. The main problem with the use of GEM was the presence of dielectric, which could lead to loss of electrons. The second cause of concern was how to provide potential to the two sides of the GEM. It was suggested that one should make both the surfaces

equipotential. Also a metal mesh was found in the lab, which would be suitable to use for our purpose, however the condition of the mesh is not very promising. Below I present the pictures of the MCPs when they were soldered, so that HV could be applied to them.

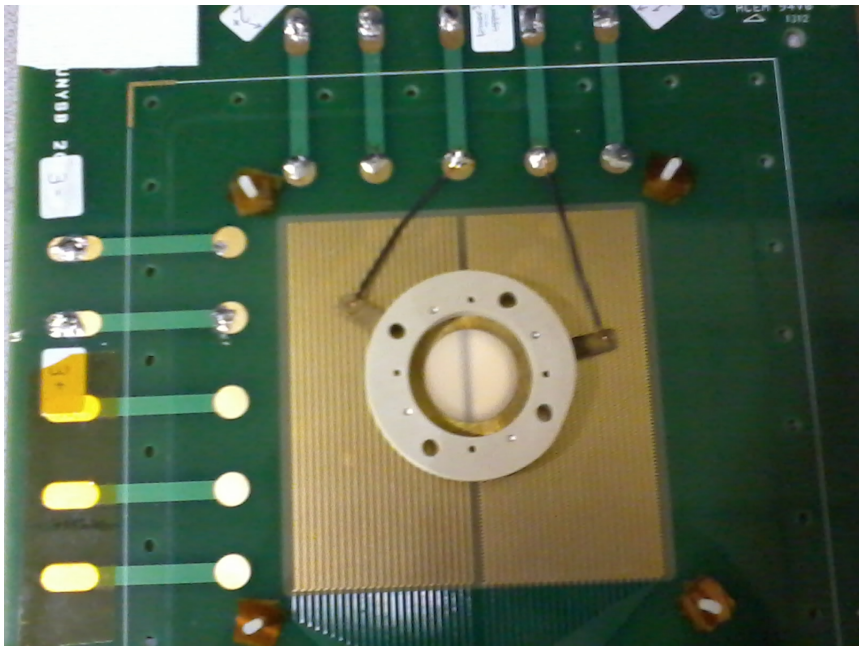

Figure 14. Microchannel plates

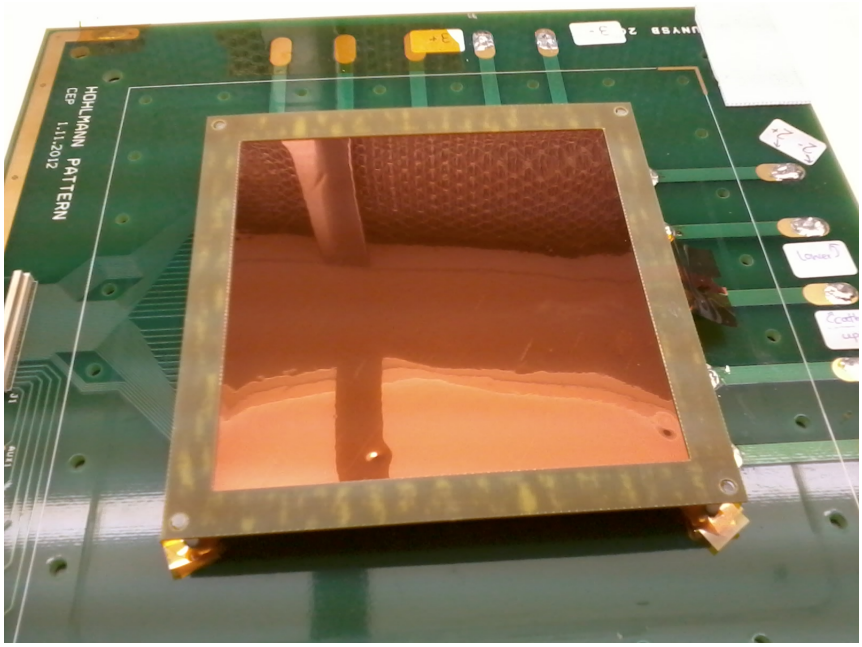

Figure 15. Microchannel plates with the cathode as drift plane

6 Code

First the Ansys codes for GEMs

```
FINISH
/CLEAR,START
/PREP7
! No polynomial elements
/PMETH,OFF,1

! Set electric preferences
KEYW,PR_ELMAG,1
KEYW,MAGELC,1

! Select element
ET,1,SOLID123

! Material properties
MP,PERX,1,1e10  ! Metal
MP,RSVX,1,0.0   !
MP,PERX,2,1.0   ! Gas
MP,PERX,3,4.0   ! Permittivity of FR4


!Electric field in V/cm
d_field=2000
t_field=3000
v_gem=400

! Distance in microns
drift_space=1000
induction_space=1000
thick=50
dzcopper=5
dzcathode=5
dzanode=5
z1cathode=thick+dzcopper+drift_space
z2cathode=z1cathode+dzcathode
z1anode=-(dzcopper+induction_space)
z2anode=z1anode-dzanode



! Potentials on surfaces
```

```
anode=0
gem_lower=anode-(t_field*(induction_space/10000))
gem_upper=gem_lower-v_gem
cathode=gem_upper-(d_field*(drift_space/10000))

!BUILDING THE GEM
pitch=140 !in microns
X=pitch/2
Y=SQRT(3)*pitch/2
d_outer=70
d_inner=50

BLOCK,0,X,0,Y,0,thick
BLOCK,0,X,0,Y,thick,thick+dzcopper
BLOCK,0,X,0,Y,0,-dzcopper

CONE,d_outer/2,d_inner/2,0,thick/2,0,360
CONE,d_inner/2,d_outer/2,thick/2,thick,0,360
CYLIND,d_outer/2,,0,-dzcopper,0,360
CYLIND,d_outer/2,,thick,thick+dzcopper,0,360

WPOFFS,X,Y,0

CONE,d_outer/2,d_inner/2,0,thick/2,0,360
CONE,d_inner/2,d_outer/2,thick/2,thick,0,360
CYLIND,d_outer/2,,0,-dzcopper,0,360
CYLIND,d_outer/2,,thick,thick+dzcopper,0,360


WPOFFS,-X,-Y,0

VSBV,1,4,,DELETE,DELETE ! 12
VSBV,12,5,,DELETE,DELETE ! 1
VSBV,1,8,,DELETE,DELETE ! 4
VSBV,4,9,,DELETE,DELETE ! 1

VSBV,2,7,,DELETE,DELETE ! 4
VSBV,4,11,,DELETE,DELETE ! 2

VSBV,3,6,,DELETE,DELETE ! 4
VSBV,4,10,,DELETE,DELETE ! 3

! ELECTRODES
```

```
! CATHODE

BLOCK,0,X,0,Y,z1cathode,z2cathode

! ANODE

BLOCK,0,X,0,Y,z1anode,z2anode

! GAS
BLOCK,0,X,0,Y,z1cathode,z1anode

VSBV,6,1,,,KEEP
VSBV,7,2,,,KEEP
VSBV,6,3,,,KEEP

VGLUE,ALL

!/COLOR,VOLU,12,1
!/COLOR,VOLU,4,4
!/COLOR,VOLU,0,5
!/COLOR,VOLU,15,6
!/COLOR,VOLU,10,8

!1-> Dielectric
!4-> cathode
!5-> Anode
!6-> lower electrode
!8-> upper electrode
!9-> Gas

! DIELECTRIC
VSEL,S,VOLU,,1
VATT,2,,1

! Metals
VSEL,S,VOLU,,4,6
VATT,1,,1
VSEL,S,VOLU,,8
VATT,1,,1

! Gases
VSEL,S,VOLU,,9
VATT,3,,1
```

```
! Lets apply loads(voltages)

VSEL,S,VOLU,,4
ASLV,S
DA,ALL,VOLT,cathode

VSEL,S,VOLU,,8
ASLV,S
DA,ALL,VOLT,gem_upper

VSEL,S,VOLU,,6
ASLV,S
DA,ALL,VOLT,gem_lower

VSEL,S,VOLU,,5
ASLV,S
DA,ALL,VOLT,anode


! boundary conditions for gas and dielectric (optional)
VSEL, S,,,1
ASLV, S
ASEL, R, LOC, X, 0
DA, ALL, SYMM
VSEL, S,,,1
ASLV, S
ASEL, R, LOC, X, X
DA, ALL, SYMM
VSEL, S,,,1
ASLV, S
ASEL, R, LOC, Y, 0
DA, ALL, SYMM
VSEL, S,,,1
ASLV, S
ASEL, R, LOC, Y, Y
DA, ALL, SYMM

VSEL, S,,,9
ASLV, S
ASEL, R, LOC, X, 0
DA, ALL, SYMM
VSEL, S,,,9
```

```
ASLV, S
ASEL, R, LOC, X, X
DA, ALL, SYMM
VSEL, S,,,9
ASLV, S
ASEL, R, LOC, Y, 0
DA, ALL, SYMM
VSEL, S,,,9
ASLV, S
ASEL, R, LOC, Y, Y
DA, ALL, SYMM



! IT NOW BOILS DOWN TO THIS

! MESHING
MSHKEY,0
SMRTSIZE,1
VSEL,S,VOLU,,1,9,8
ASLV,S
VMESH,ALL

! Solve the field
/SOLU
SOLVE
FINISH

! Display the solution
/POST1
/EFACET,1
PLNSOL, VOLT,, 0

! Write the solution to files
/OUTPUT, PRNSOL, lis
PRNSOL
/OUTPUT

/OUTPUT, NLIST, lis
NLIST,,,,COORD
/OUTPUT
```

```
/OUTPUT, ELIST, lis
ELIST
/OUTPUT

/OUTPUT, MPLIST, lis
MPLIST
/OUTPUT
```

Next is given the code for parallel plate capacitors with anode described by one sheet of metal

```
FINISH
/CLEAR,START
/CWD,U:\ansys\zigzag
/OUTPUT,output,txt
/PREP7
! No polynomial elements
/PMETH,OFF,1

! Set electric preferences
KEYW,PR_ELMAG,1
KEYW,MAGELC,1

! Select element
ET,1,SOLID123

! Material properties
MP,PERX,1,1e10  ! Metal
MP,RSVX,1,0.0   !
MP,PERX,2,1.0   ! Gas
MP,PERX,3,4.0   ! Permittivity of FR4


!Electric field in V/cm

! Distance in microns
drift_space=0
induction_space=1000
dzcathode=5
dzanode=5
z1cathode=drift_space
z2cathode=z1cathode+dzcathode
z1anode=-(induction_space)
```

```
z2anode=z1anode-dzanode

! Potentials on surfaces
t_field=3000 ! In kV/cm
anode=0
cathode=(induction_space/10000)*t_field


! CATHODE
pitch=140 !in microns
X=pitch/2
Y=SQRT(3)*pitch/2

BLOCK,0,X,0,Y,z1cathode,z2cathode

! ANODE

BLOCK,0,X,0,Y,z1anode,z2anode

! GAS
BLOCK,0,X,0,Y,z1cathode,z1anode


VGLUE,ALL
VLIST,ALL
!VPLOT,2
!1-> Cathode
!2-> Anode
!4-> Gas

! Metals
VSEL,S,VOLU,,1,2
VATT,1,,1

! Gases
VSEL,S,VOLU,,4
VATT,2,,1

! Lets apply loads(voltages)

VSEL,S,VOLU,,1
ASLV,S
DA,ALL,VOLT,cathode
```

```
VSEL,S,VOLU,,2
ASLV,S
DA,ALL,VOLT,anode

! IT NOW BOILS DOWN TO THIS

! MESHING
MSHKEY,0
SMRTSIZE,1
VSEL,S,VOLU,,4
ASLV,S
VMESH,ALL

! Solve the field
/SOLU
SOLVE
FINISH

! Display the solution
/POST1
/EFACET,1
PLNSOL, VOLT,, 0

! Write the solution to files
/OUTPUT, PRNSOL, lis
PRNSOL
/OUTPUT

/OUTPUT, NLIST, lis
NLIST,,,,COORD
/OUTPUT

/OUTPUT, ELIST, lis
ELIST
/OUTPUT

/OUTPUT, MPLIST, lis
MPLIST
/OUTPUT
```

Anodes with segmented pads

```
FINISH
/CLEAR,START
/CWD,U:\ansys\zigzag
/OUTPUT,output,txt
/PREP7
! No polynomial elements
/PMETH,OFF,1

! Set electric preferences
KEYW,PR_ELMAG,1
KEYW,MAGELC,1

! Select element
ET,1,SOLID123

! Material properties
MP,PERX,1,1e10  ! Metal
MP,RSVX,1,0.0   !
MP,PERX,2,1.0   ! Gas
MP,PERX,3,4.0   ! Permittivity of FR4


!Electric field in V/cm

! Distance in microns
drift_space=0
induction_space=1000
dzcathode=5
dzanode=5
z1cathode=drift_space
z2cathode=z1cathode+dzcathode
z1anode=-(induction_space)
z2anode=z1anode-dzanode

! Potentials on surfaces
t_field=3000 ! In kV/cm
anode=0
cathode=(induction_space/10000)*t_field


! CATHODE
pitch=300 !in microns
X_cat=pitch*10
```

```
Y_cat=X_cat

BLOCK,0,X_cat,0,Y_cat,z1cathode,z2cathode

! ANODE
X_an=500
Y_an=Y_cat
pitch=2000
*DO,i,0,1,1
BLOCK,(pitch+X_an)*i,((pitch+X_an)*i)+X_an,0,Y_an,z1anode,z2anode
*ENDDO

! GAS
BLOCK,0,X_cat,0,Y_cat,z1cathode,z2anode

VSBV,4,2,,,KEEP
VSBV,5,3,,,KEEP
VLIST,ALL

VGLUE,ALL
VLIST,ALL
VPLOT,5
!1->Cathode
!2,3->Anode
!5->Gas
! Metals
VSEL,S,VOLU,,1,3
VATT,1,,1

! Gases
VSEL,S,VOLU,,5
VATT,2,,1

! Lets apply loads(voltages)

VSEL,S,VOLU,,1
ASLV,S
DA,ALL,VOLT,cathode

VSEL,S,VOLU,,2,3
ASLV,S
DA,ALL,VOLT,anode
```

```
! IT NOW BOILS DOWN TO THIS

! MESHING
MSHKEY,0
SMRTSIZE,1
VSEL,S,VOLU,,5
ASLV,S
VMESH,ALL

! Solve the field
/SOLU
SOLVE
FINISH

! Display the solution
/POST1
/EFACET,1
PLNSOL, VOLT,, 0

! Write the solution to files
/OUTPUT, PRNSOL, lis
PRNSOL
/OUTPUT

/OUTPUT, NLIST, lis
NLIST,,,,COORD
/OUTPUT

/OUTPUT, ELIST, lis
ELIST
/OUTPUT

/OUTPUT, MPLIST, lis
MPLIST
/OUTPUT
```

Ansys code for zigzags

```
FINISH
/CLEAR,START
/CWD,U:\ansys\zigzag
/OUTPUT,output,txt
/PREP7
! No polynomial elements
```

```
/PMETH,OFF,1

! Set electric preferences
KEYW,PR_ELMAG,1
KEYW,MAGELC,1

! Select element
ET,1,SOLID123

! Material properties
MP,PERX,1,1e10  ! Metal
MP,RSVX,1,0.0   !
MP,PERX,2,1.0   ! Gas
MP,PERX,3,4.0   ! Permittivity of FR4


!Electric field in V/cm

! Distance in microns
drift_space=0
induction_space=1000
dzcathode=5
dzanode=5
z1cathode=drift_space
z2cathode=z1cathode+dzcathode
z1anode=-(induction_space)
z2anode=z1anode-dzanode

! Potentials on surfaces
t_field=3000 ! In kV/cm
anode=0
cathode=(induction_space/10000)*t_field


! CATHODE
dis=300
!pitch=300 !in microns
X_cat=1000
Y_cat=6000+dis+10

BLOCK,0,X_cat,0,Y_cat,z1cathode,z2cathode

! ANODE
```

```
X=500
Y=2000
WPOFFS,0,0,z1anode
CSYS, 4
K,,0,0,0
K,,X,Y,0
K,,X,2*Y,0
K,,0,Y,0
WPOFFS,0,0,-dzanode
CSYS, 4
K,,0,0,0
K,,X,Y,0
K,,X,2*Y,0
K,,0,Y,0

V,9,10,11,12,13,14,15,16
WPOFFS,0,0,dzanode
K,,2*X,0,0
K,,2*X,Y,0
WPOFFS,0,0,-dzanode
CSYS, 4
K,,2*X,0,0
K,,2*X,Y,0


V,10,11,18,17,14,15,20,19


WPOFFS,0,0,dzanode
WPOFFS,0,Y+dis,0
K,,0,0,0
K,,X,Y,0
K,,X,2*Y,0
K,,0,Y,0
WPOFFS,0,0,-dzanode
CSYS, 4
K,,0,0,0
K,,X,Y,0
K,,X,2*Y,0
K,,0,Y,0

V,21,22,23,24,25,26,27,28
WPOFFS,0,0,dzanode
```

```
K,,2*X,0,0
K,,2*X,Y,0
WPOFFS,0,0,-dzanode
CSYS, 4
K,,2*X,0,0
K,,2*X,Y,0


V,22,23,30,29,26,27,32,31

VLIST,ALL

VADD,2,3
VADD,4,5

!Gas
WPOFFS,0,-(Y+dis),dzanode-z1anode
CSYS,0
BLOCK,0,X_cat,0,Y_cat,z1cathode,z2anode

VLIST,ALL

VSBV,3,2,,,KEEP
VSBV,4,6,,,KEEP

VPLOT,3
!VDGL,2,3
VGLUE,ALL
VLIST,ALL

VPLOT,2,4,2
!1->CATHODE
!2->ANODE
!4->ANODE
!5->GAS

! Metals
VSEL,S,VOLU,,1,2
VATT,1,,1

VSEL,S,VOLU,,4
VATT,1,,1
```

```
! Gases
VSEL,S,VOLU,,5
VATT,2,,1

! Lets apply loads(voltages)

VSEL,S,VOLU,,1
ASLV,S
DA,ALL,VOLT,cathode

VSEL,S,VOLU,,2,4,2
ASLV,S
DA,ALL,VOLT,anode

! IT NOW BOILS DOWN TO THIS

! MESHING
MSHKEY,0
SMRTSIZE,1
VSEL,S,VOLU,,5
ASLV,S
VMESH,ALL

! Solve the field
/SOLU
SOLVE
FINISH

! Display the solution
/POST1
/EFACET,1
PLNSOL, VOLT,, 0

! Write the solution to files
/OUTPUT, PRNSOL, lis
PRNSOL
/OUTPUT

/OUTPUT, NLIST, lis
NLIST,,,,COORD
/OUTPUT
```

```
/OUTPUT, ELIST, lis
ELIST
/OUTPUT

/OUTPUT, MPLIST, lis
MPLIST
/OUTPUT
```

Finally I present the complete Garfield++ code used for the simulation. One can see that the tree structures are widely used in the script.

```
#include <iostream>
#include <fstream>
#include <cmath>
#include <vector>
#include <stdlib.h>
#include <stdbool.h>
#include <stdio.h>

#include <TCanvas.h>
#include <TROOT.h>
#include <TApplication.h>
#include <TH1F.h>
#include <TH2F.h>
#include <TBuffer3D.h>
#include <TBuffer3DTypes.h>
#include <TVirtualViewer3D.h>
#include <TFile.h>

#include "MediumMagboltz.hh"
#include "ComponentAnsys123.hh"
#include "Sensor.hh"
#include "AvalancheMicroscopic.hh"
#include "ViewGeometry.hh"
#include "Plotting.hh"
#include "ViewField.hh"
#include "Random.hh"

#include <TChain.h>
#include "TObjArray.h"

#include <time.h>

using namespace Garfield;
```

```
using namespace std;

/* Tree to store variables produced by Garfield++

*/

int main(int argc, char * argv[]) {
  // Define variables.
  Double_t xi,yi,zi,ei,ti;
  Int_t nele,nelend;

  string dirname = argv[1]; // Directory Name

  TString fileName = argv[2]; // Filename

  double rp = 0.0; // Penning Parameter
  if (argv[3]) {
    rp = (atof(argv[3]))/100.0;
    if (rp>1.0 || rp<0.0) rp = 0.5;
  }

  double bfield = 0.0;  // Magnetic field
  if (argv[4]) {
    bfield = (atof(argv[4]))/10.0;
    if (bfield < 0.0) bfield = 0.0;
  }

  double ang = 0.0; // Angle between E&B fields
  if (argv[5]) {
    ang = atof(argv[5]);
    if (ang<0.0 || ang>180.0) ang = 0.0;
  }

  int nevents = 200; // Number of events.
  if (argv[6]) {
    nevents = atoi(argv[6]);
    if (nevents<1) nevents = 200;
  }
  if (nevents<1) nevents=200;

  TString treeName;

  Double_t x1[20000],y1[20000],z1[20000],e1[20000],t1[20000];
```

```
  Double_t x0[20000],y0[20000],z0[20000],e0[20000],t0[20000];
  Int_t stat[20000];

  fileName = "/afs/cern.ch/work/a/amohapat/zigzag/"+dirname+"/sgem_"+fileName+".root";
  cout << "Tree will be written in file " << fileName << endl;
  TFile *treefile = new TFile(fileName,"RECREATE","mytree");
  treefile->cd();
  treeName = "gemtree";
  TTree *tree = new TTree(treeName,"variables");

  cout << "Will write to File " << fileName << endl;

  tree->Branch("z1",z1,"z1[20000]/D");
  tree->Branch("status",stat,"status[20000]/I");
  tree->Branch("nele",&nele,"nele/I");
  tree->Branch("nelend",&nelend,"nelend/I");
  tree->Branch("x1",x1,"x1[20000]/D");
  tree->Branch("y1",y1,"y1[20000]/D");
  tree->Branch("e1",e1,"e1[20000]/D");
  tree->Branch("t1",t1,"t1[20000]/D");
  tree->Branch("x0",x1,"x1[20000]/D");
  tree->Branch("y0",y1,"y1[20000]/D");
  tree->Branch("e0",e1,"e1[20000]/D");
  tree->Branch("t0",t1,"t1[20000]/D");

  // Loading the Ansys files (weighting afterwards)
  ComponentAnsys123* gem = new ComponentAnsys123();
  gem->Initialise(dirname+"/ELIST.lis",dirname+"/NLIST.lis",dirname+"/MPLIST.lis",
  dirname+"/PRNSOL.lis","micron");
  gem->EnableMirrorPeriodicityX();
  gem->EnableMirrorPeriodicityY();
  cout<<bfield<<endl;
  double by=(bfield)*sin(ang*(22.0/7.0)/180.0);
  double bz=(bfield)*cos(ang*(22.0/7.0)/180.0);
  cout<<by<<" "<<bz<<endl;
  gem->SetMagneticField(0,by,bz);

  MediumMagboltz* gas=new MediumMagboltz();
  gas->SetComposition("ar",70.,"co2",30.);
//  gas->SetComposition("ar",45.,"co2",15.,"cf4",40.0);
//  gas->SetComposition("ar",80.,"cf4",20.);
//  gas->SetComposition("cf4",100.);
//  gas->SetComposition("ar",95.,"ch4",5.);
```

```
gas->SetTemperature(300.0);
gas->SetPressure(760.0);
gas->SetMaxElectronEnergy(200.);
gas->EnablePenningTransfer(rp, 0.0, "Ar");
gas->Initialise(true);
int numofmat=gem->GetNumberOfMaterials();

cout << "Number of materials: " << numofmat << endl;
for(int i=0;i<numofmat;i++){
  double perman = gem->GetPermittivity(i);
  cout<<"The permittivity of material "<<i<<" is "<<perman<<endl;
  if(fabs(perman-1)<1.e-13){gem->SetMedium(i,gas);}
}

Sensor* sensor = new Sensor();
sensor->AddComponent(gem);
const double pitch=0.0140;

AvalancheMicroscopic* aval = new AvalancheMicroscopic();
aval->SetSensor(sensor);
//  aval->EnableDebugging();

aval->EnableMagneticField();
// aval->EnableAvalancheSizeLimit(1000);

// Loop over primary electrons
//const double smear=pitch/2;
for(int i=0; i<nevents; i++) {
  const double smear = pitch/2;
  xi = -smear + RndmUniform() * smear;
  yi = -smear + RndmUniform() * smear;
  zi = 0.03;  ei = 0.1;  ti = 0.0;
  double dx0 = 0.0; double dy0 = 0.0; double dz0 = 0.0;
  float timeused = (float)clock()/CLOCKS_PER_SEC;
  if (timeused > 25000.0) {
    cout << "Time limit reached, " << timeused << " sec" << endl;
    break;
  }
  aval->AvalancheElectron(xi,yi,zi,ti,ei,dx0,dy0,dz0);

  int ne,ni;
  aval->GetAvalancheSize(ne,ni); nele = ne;
```

```
    double xa0, ya0, za0, ea0, ta0;
    double xa1, ya1, za1, ea1, ta1;
    int istat;

    nelend = aval->GetNumberOfElectronEndpoints();

    for (int j=0; j<nelend; j++) {
      aval->GetElectronEndpoint(j,xa0,ya0,za0,ta0,ea0,xa1,ya1,za1,ta1,ea1,istat);
      x0[j] = xa0; y0[j] = ya0;  z0[j] = za0; t0[j] = ta0; e0[j] = ea0;
      x1[j] = xa1; y1[j] = ya1;  z1[j] = za1; t1[j] = ta1; e1[j] = ea1;
      stat[j] = istat;

    }

    // Fill Tree
    tree->Fill();
  }

  // Write out the Tree File
  treefile->cd();
  tree->Write();
  treefile->Close();

  return 0;
}
```

7 Conclusions

Extensive simulations were done to study the x-y distributions of the electron endpoints in magnetic as well as non-magnetic fields. However one still has to implement the second approach of the signal simulation discussed earlier. Also in hardware, I would like to complete the Gain studies in MCPs, and comment on their feasibility.

Acknowledgments

I would like to thank Dr. M. Hohlmann for giving me an oppurtunity to work in his lab and also for his valuable inputs during the Monday meeting. I would also like to thank Vallary and Jessie for helping me out with the MCP hardware.

References

[1] Author, *Title*, *J. Abbrev.* **vol** (year) pg.

[2] Author, *Title*, arxiv:1234.5678.

[3] Author, *Title*, Publisher (year).