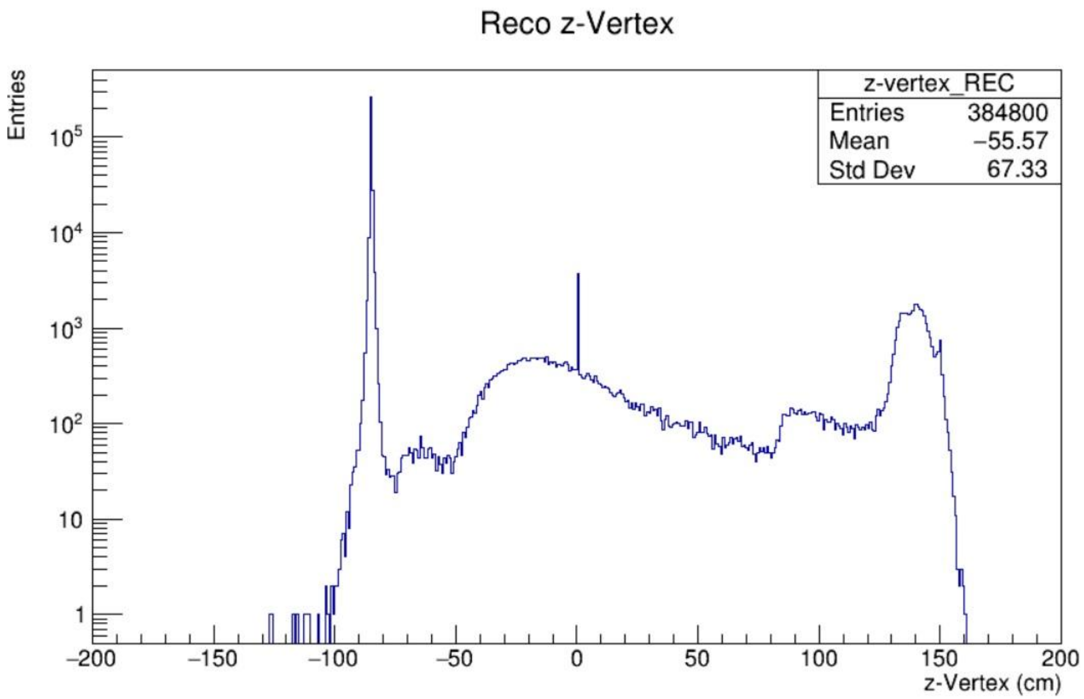
Fixed Target Analysis: PID and Track Efficiency for the MPD experiment

Studens:
Adrian Lara
Francisco Reyes

Supervisors:
Dr. Vadim Kolesnikov
Dra. Ivonne Alicia Maldonado Cervantes
Dr. Viktar Kireyeu
Natalia Kolomoyets

Primary Vertex Analysis

Primary Vertex

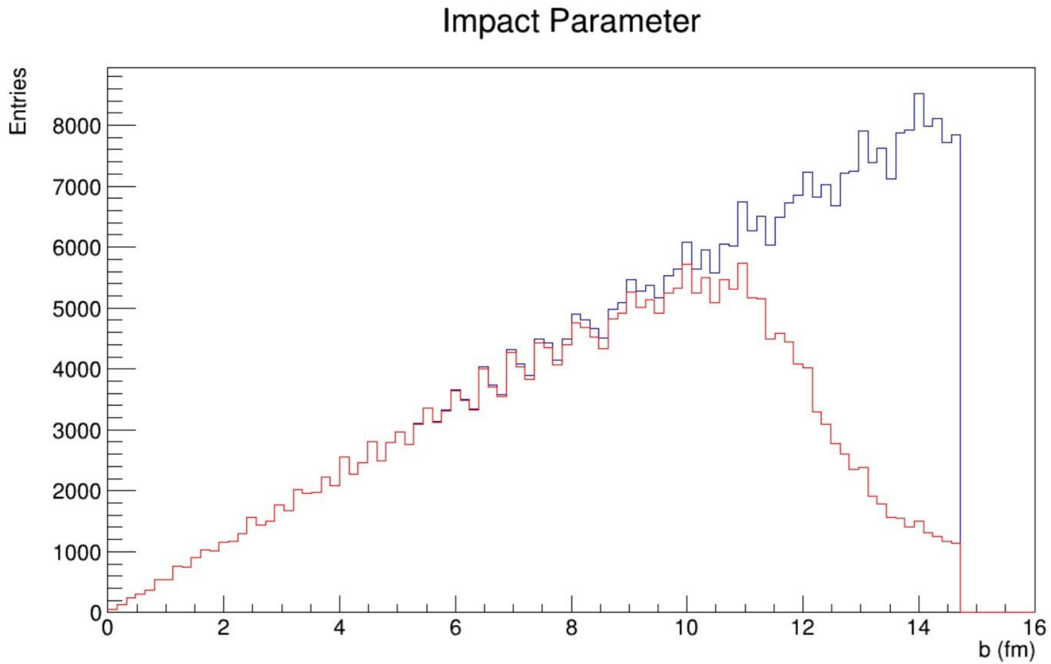


Distribution of z-Vertex with all generated events. From runMC.C file we establish primary vertex at -85cm. So having primary vertex at -50 to 150 its not clear.

A percentage of how many events are near the peak has been made.

Percentage of events
|z - 85| < 15:
80.164%

Primary Vertex

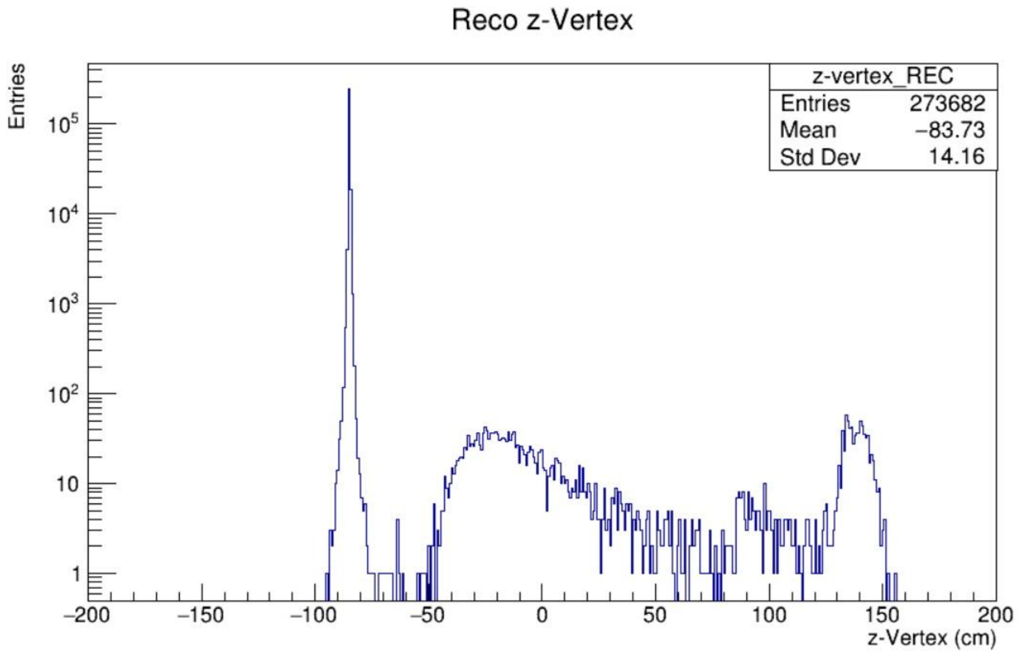


We analyzed the impact parameter, it was seen that to try to exclude the peripheral collisions we need to make a cut in MonteCarlo Tracks

Applied cuts :
MCTracks > 308
= 124 (Xe) + 184 (W)

We restrict MCTracks to have more particles than the initial ones, for 124 for Xe and 184 for W.

Primary Vertex



Again we calculate percentage of events and we obtain.

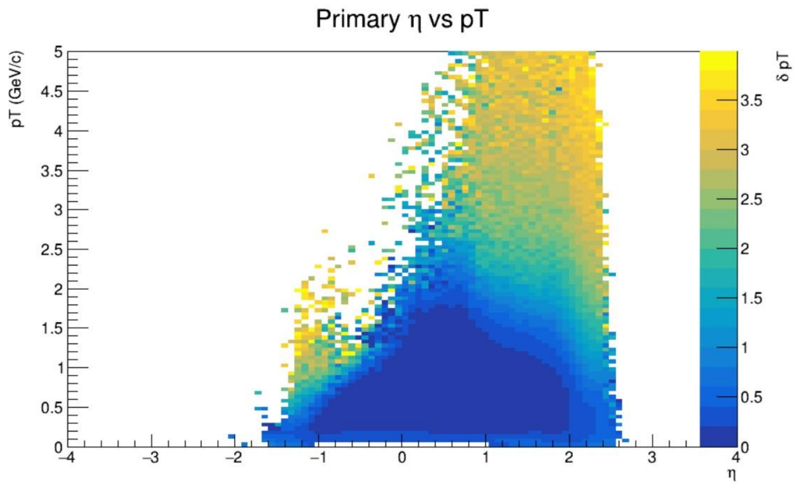
Percentage of Events
|z - 85| < 15:
98.95%

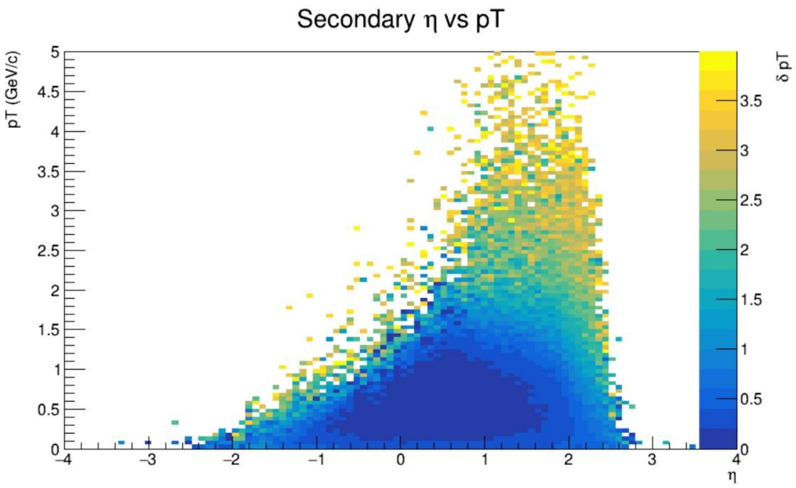
Cuts obtained:
Z between -100 cm and -70 cm

Track Selection

Track Selection

One of the tasks performed is to clean the distributions of the phase space, with them we can observe the effectiveness of the cuts applied for the reconstructed tracks. Here we show phase space of Primary and Secondary particles.





resolution of pT is on palette

Track Selection

Selection criteria

To select tracks we rely on transversal momentum resolution.

$$\delta p_T = \frac{\left|p_T^{Reco} - p_T^{MC}\right|}{p_T^{MC}}$$

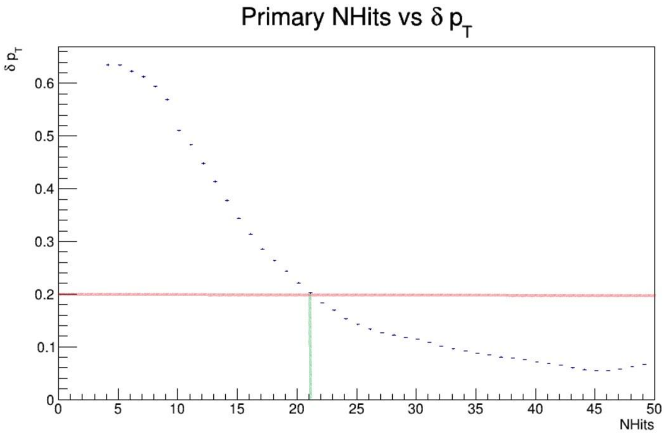
Looking for pT resolution less than 0.2.

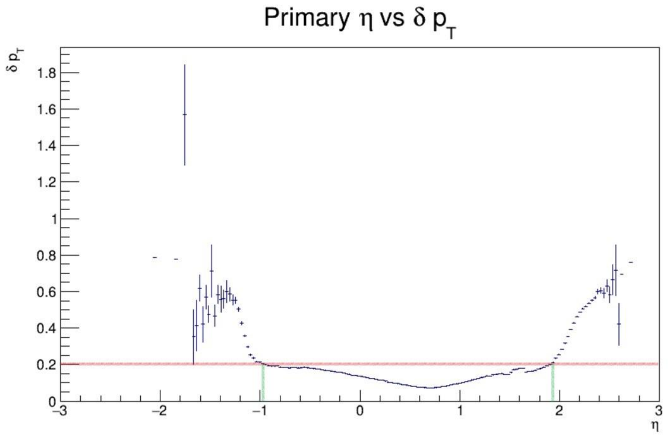
Procedure

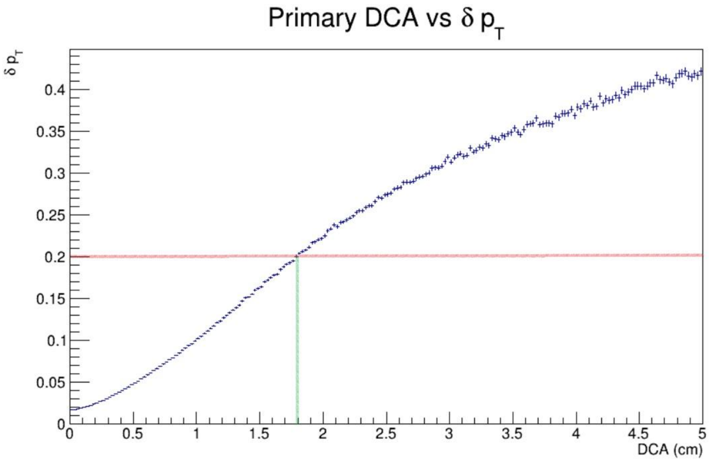
Primary and secondary particle TProfile were created with respect to the variables Nhits, pseudorapidity, DCA vs pT resolution.

The cuts obtained from the primary vertex were left unchanged.

Track Selection



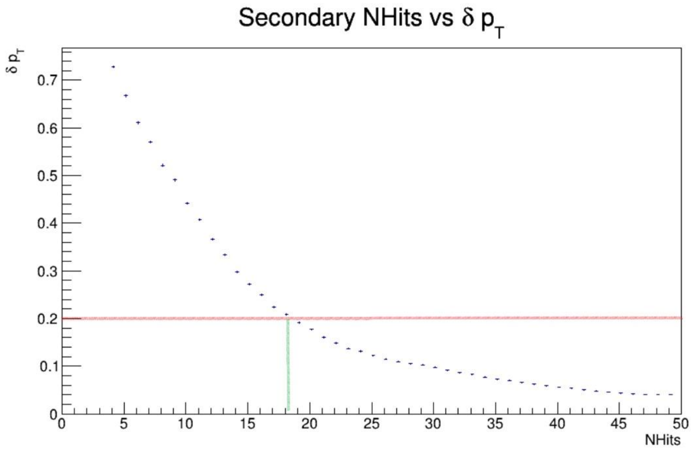


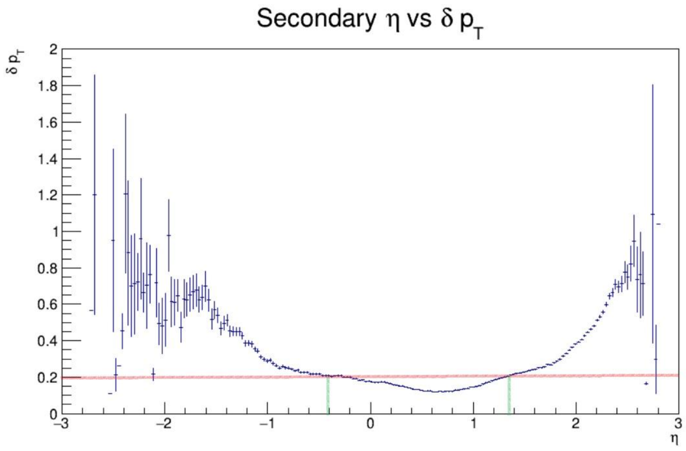


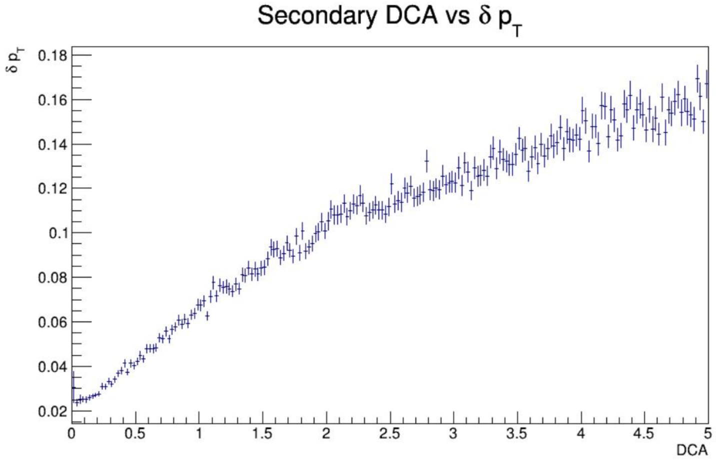
Cuts:

MCTracks > 304

Track Selection



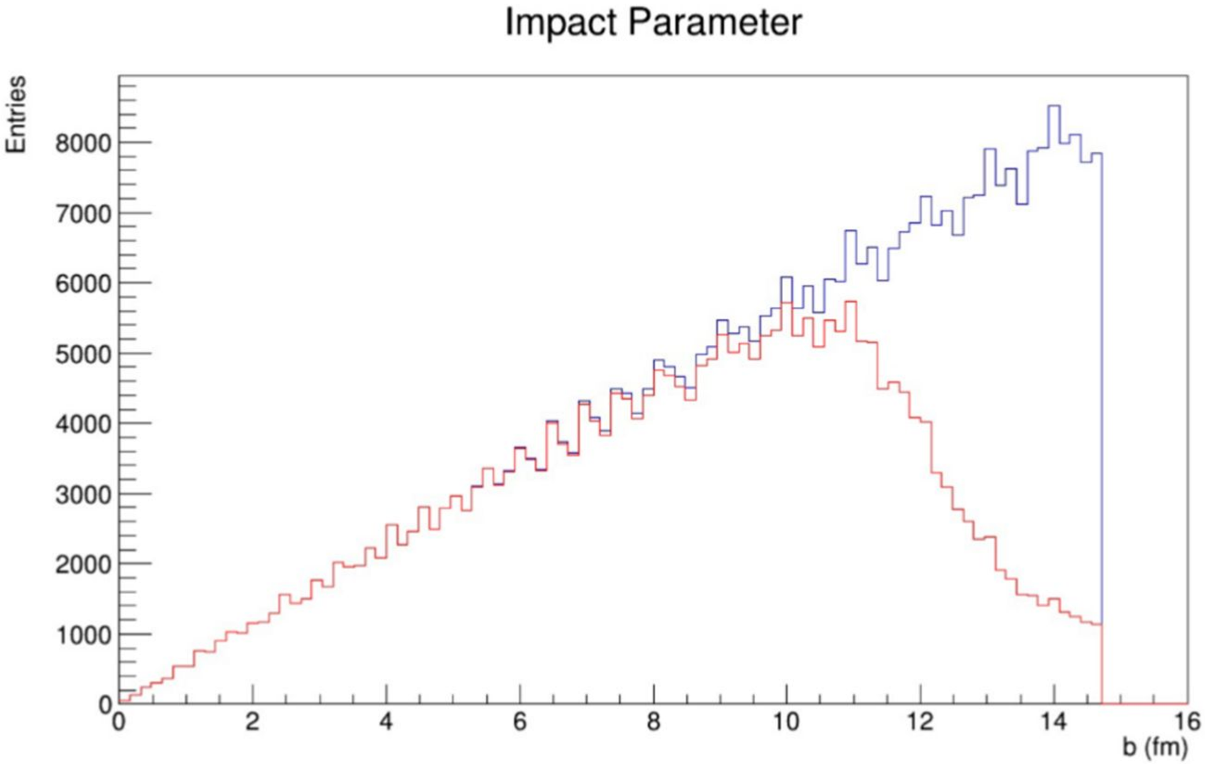




Cuts:

MCTracks > 304

Track Selection



To exclude even more peripheral events we made a cut at b < 13 fm

Cut Obtained:
b < 13 fm

Track Selection

Obtained cuts for Accepted Tracks

Primary Vertex

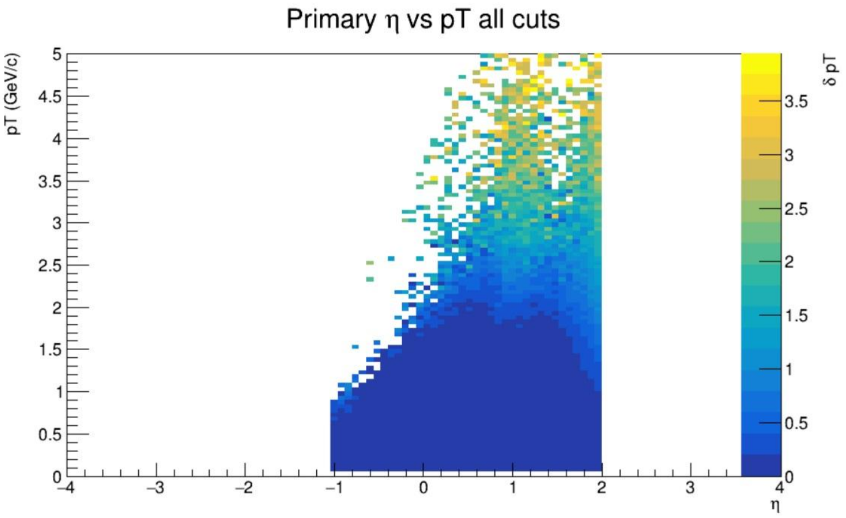
- MC Tracks > 308
- Z-Vertex (-100,-70)

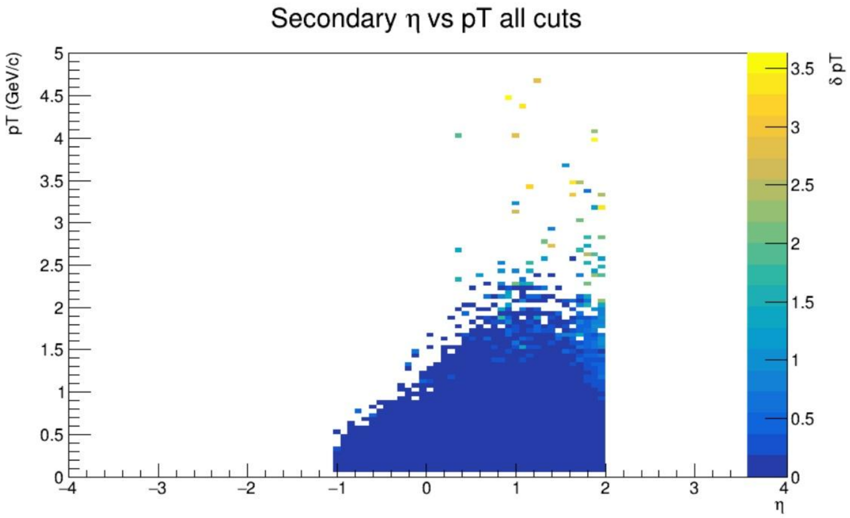
Cut on eta, DCA, Number of Hits, b

- Eta (-1 , 2)
- DCA <= 2
- Number of Hits => 20
- Impact parameter b < 13

Track Selection

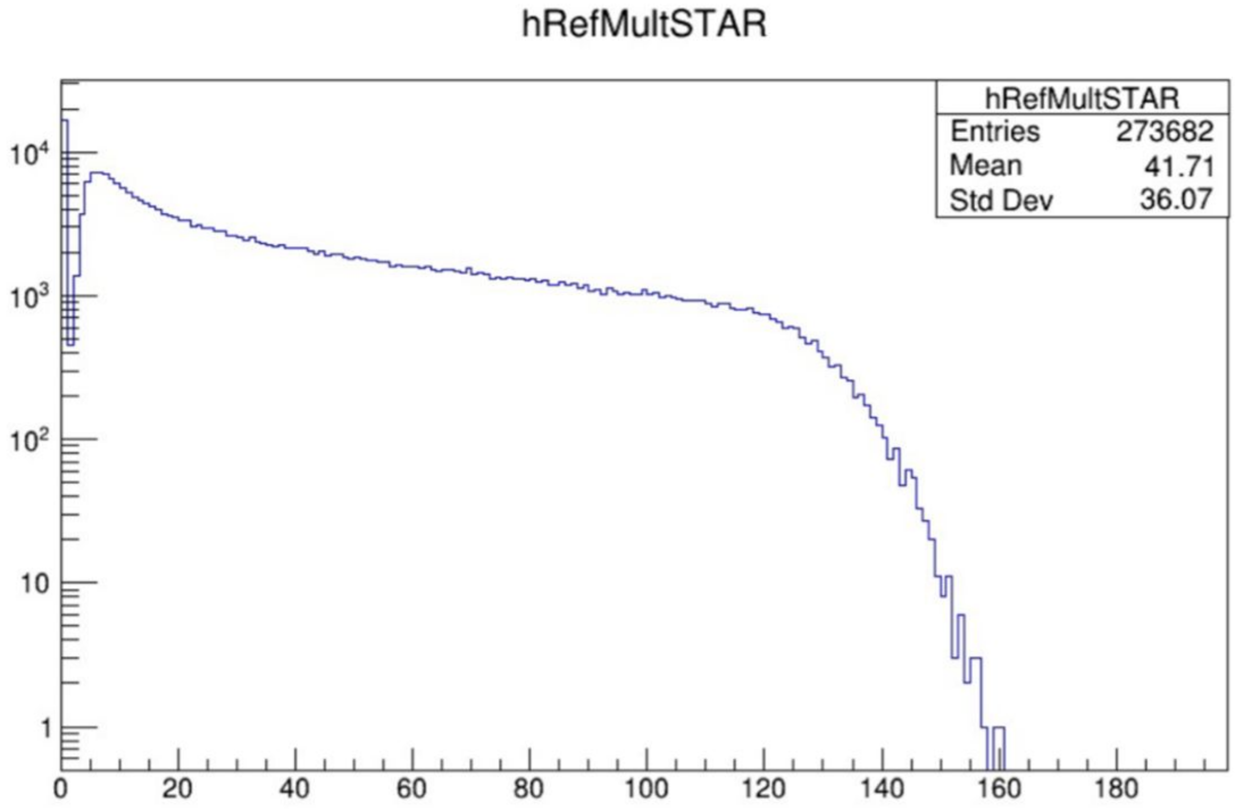
From the distributions below, we can observe how the cuts affect the phase space, also we can observe that resolution is too big for particles with pT > 2 GeV/c and that it is not symmetric with respect to pseudorapidity, below zero values, pT reach a maximum at 1 GeV/c.





resolution of pT is on palette

Track Selection



From Accepted Reconstructed Tracks we obtain multiplicity distribution, this distribution was made for Centrality determination

Applied cuts:

MC Tracks > 308

Eta (-1 , 2)

DCA <= 2

NHits => 20

b < 13

Centrality Determination: MC Glauber

MC Glauber

Glauber package

Necessary data

- Nuclei Radius:
  - Xe = 4.763 +- 0.01
  - W = 5.373 +- 0.01
- Skin Thickness:
  - Xe = 0.523
  - W = 0.523
- Atomic Number:
  - Xe = 124
  - W = 184
- Inelastic cross section:
  - 26.94035 mb
- Statistics 10 to 1 on total of Events:
  - 3,848,000 a 384,800

Centrality Framework

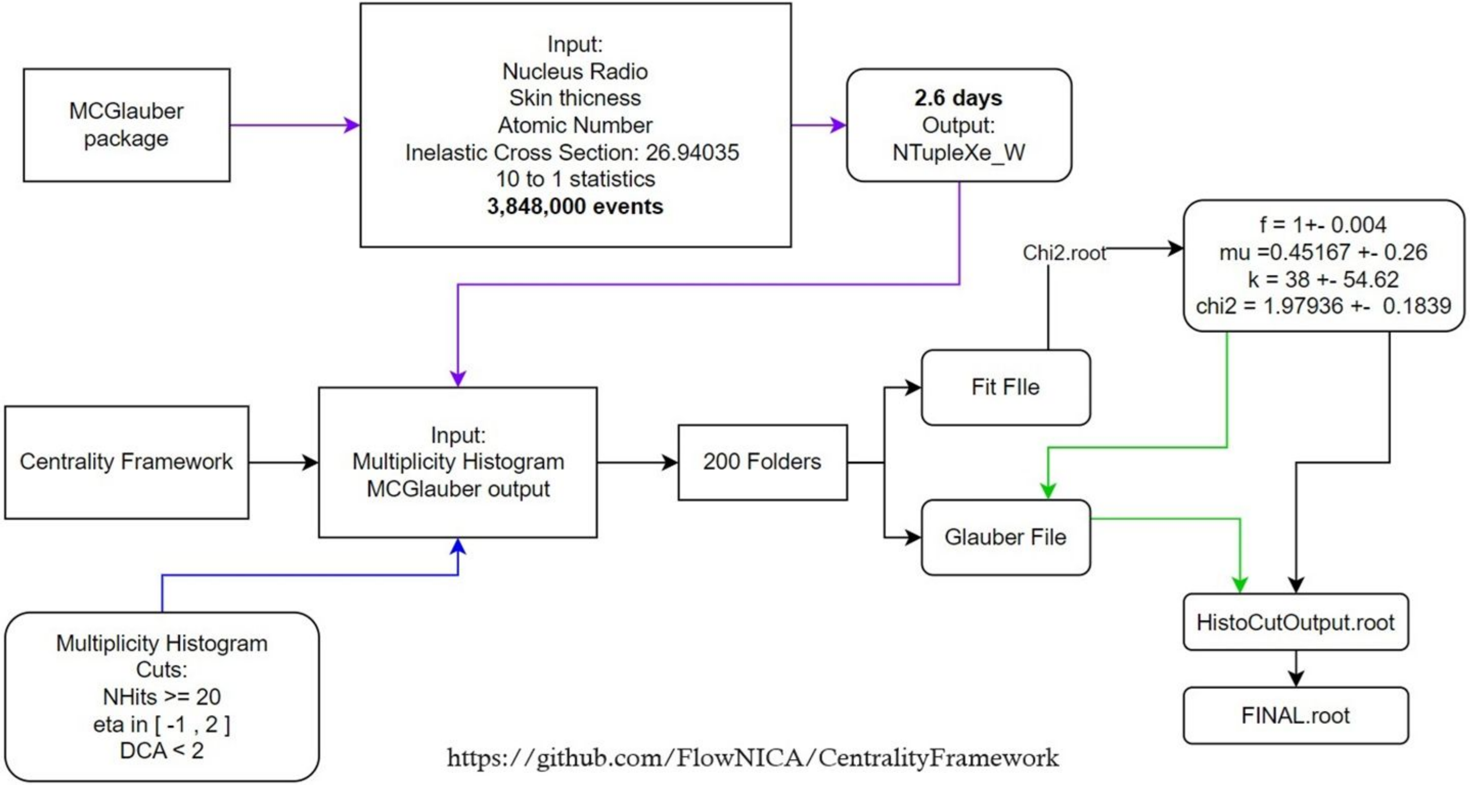
Necessary data

- Multiplicity Distribution
- Output file from Glauber package

To run MC Glauber method, the necessary data its presented.
All instruction on how to run MC Glauber its on Centrality Framework Github.
https://github.com/FlowNICA/CentralityFramework

MC Glauber

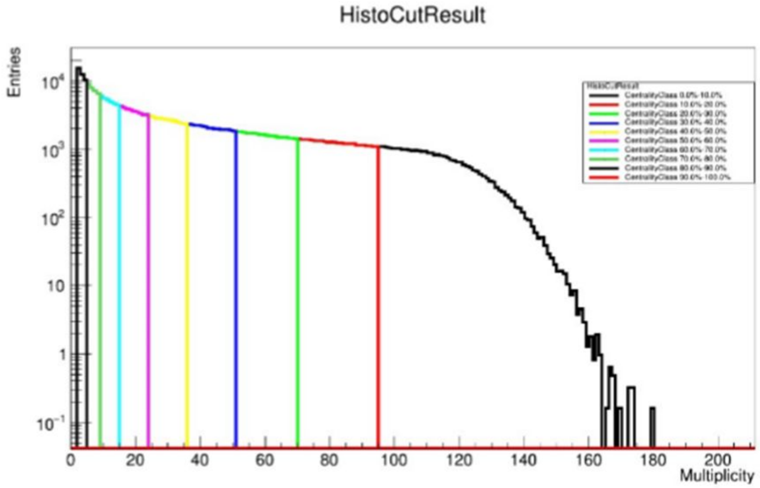
Visual Representation of MC Glauber Procedure



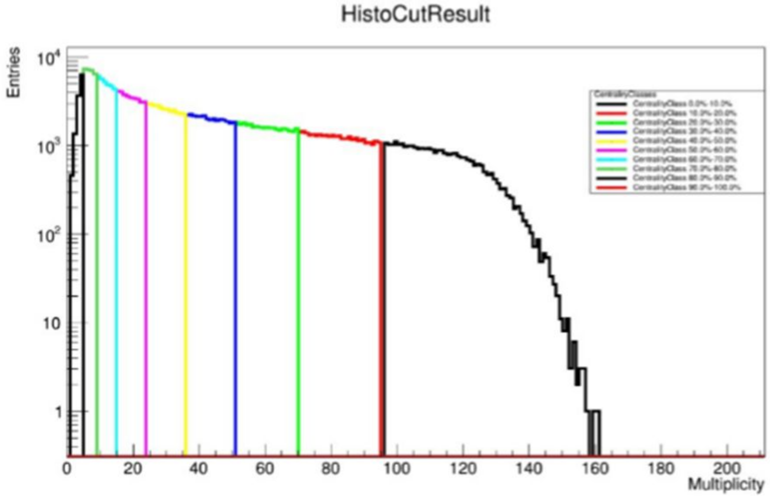
https://github.com/FlowNICA/CentralityFramework

MC Glauber

Result obtain on MC Glauber are distribution divided by centrality classes, here are represented with different colors.



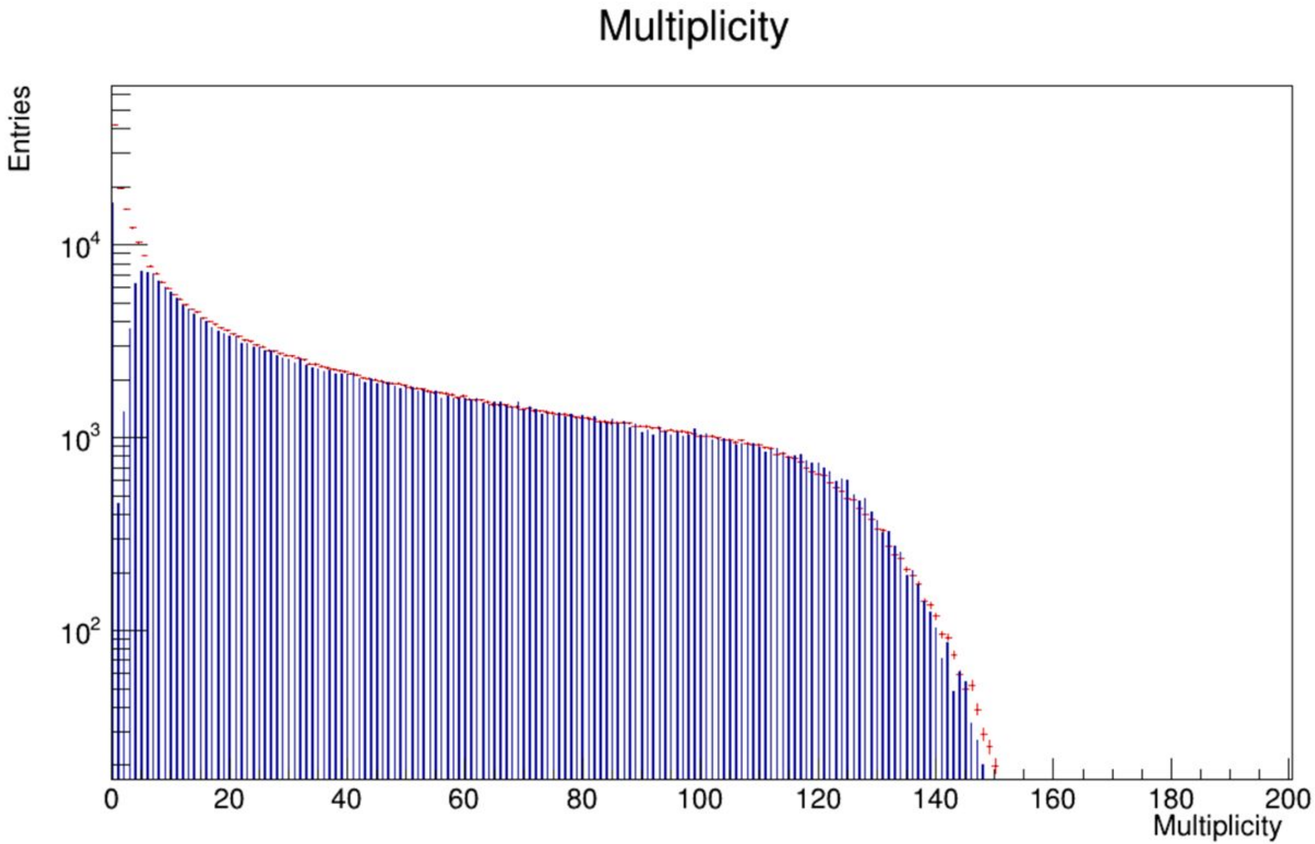
(a) Multiplicity fitted function



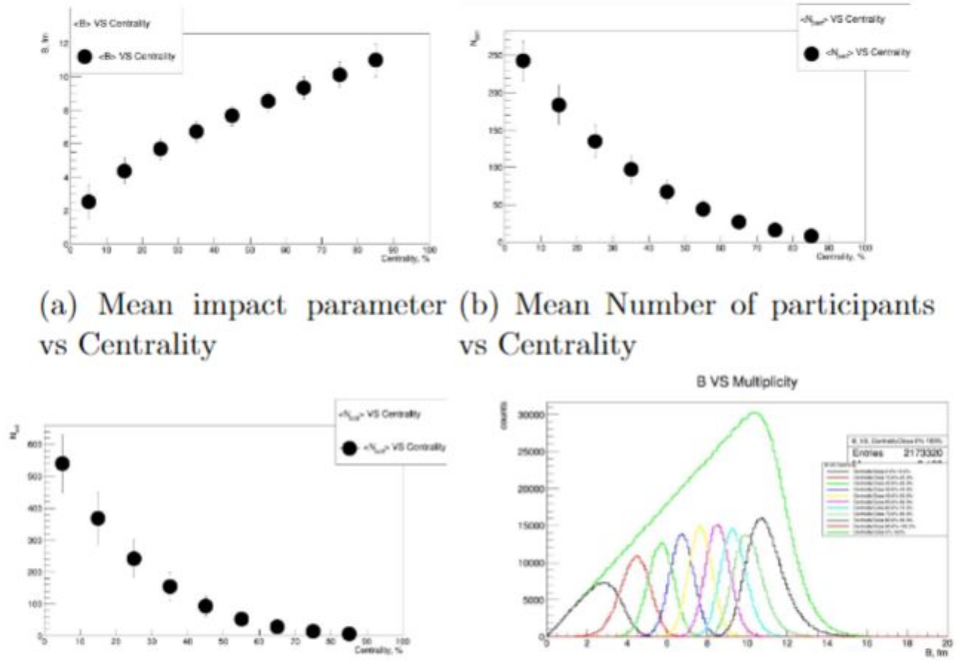
(b) Multiplicity input ditribution

MC Glauber

Multiplicity input distribution and fitted distribution on red.



MC Glauber



(a) Mean impact parameter vs Centrality

(b) Mean Number of participants vs Centrality

(c) Mean Number of collision vs Centrality

(d) Impact parameter divided by centrality Classes

Figure 16: Histograms divided by Centrality Classes

Histograms divided on centrality classes.

MC Glauber

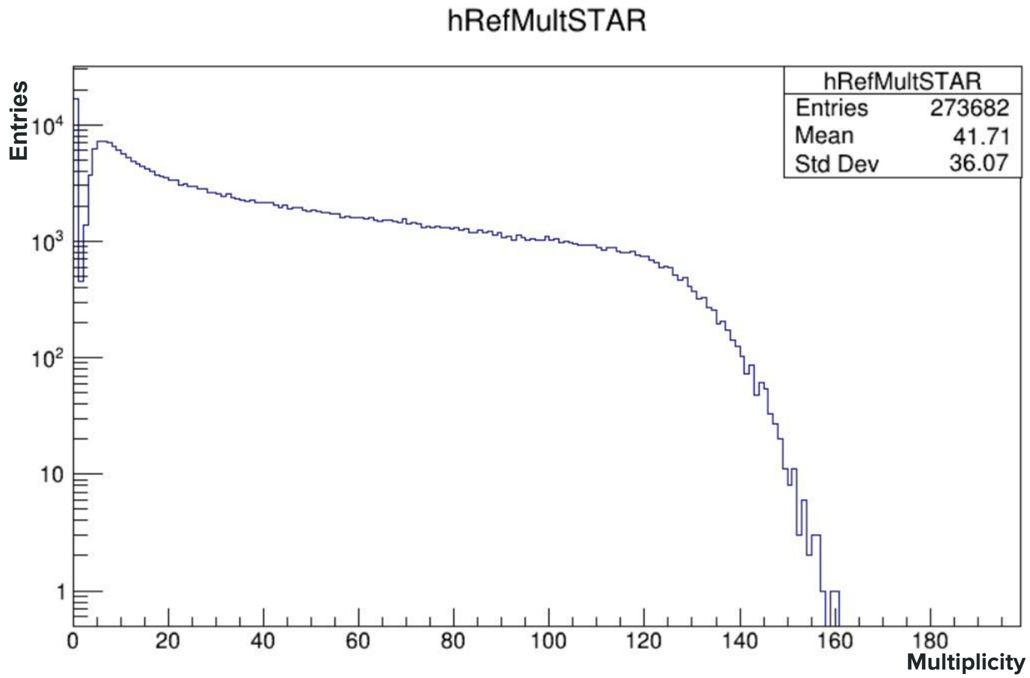
Centrality, %	N_{ch}^{min}	N_{ch}^{max}	$\langle b \rangle$, fm	RMS	$\langle N_{part} \rangle$	RMS	$\langle N_{coll} \rangle$	RMS
0 - 10	95	164	2.54	1.00	242.75	26.44	539.73	88.76
10 - 20	70	95	4.37	0.76	183.50	26.19	367.84	79.38
20 - 30	51	70	5.69	0.63	134.96	21.54	241.37	59.12
30 - 40	36	51	6.74	0.59	97.33	17.96	154.42	43.75
40 - 50	24	36	7.68	0.59	67.38	14.84	93.66	31.25
50 - 60	15	24	8.55	0.61	43.96	11.82	52.81	20.86
60 - 70	9	15	9.35	0.66	27.27	9.01	28.33	13.12
70 - 80	5	9	10.11	0.78	16.04	6.73	14.59	8.12
80 - 90	2	5	10.99	0.99	8.23	4.74	6.60	4.77

Table of Multiplicity, Impact parameter, Number of participants and Number of colliders divided of Centrality classes

Centrality Determination: Gamma Fit

GammaFit and Multiplicity

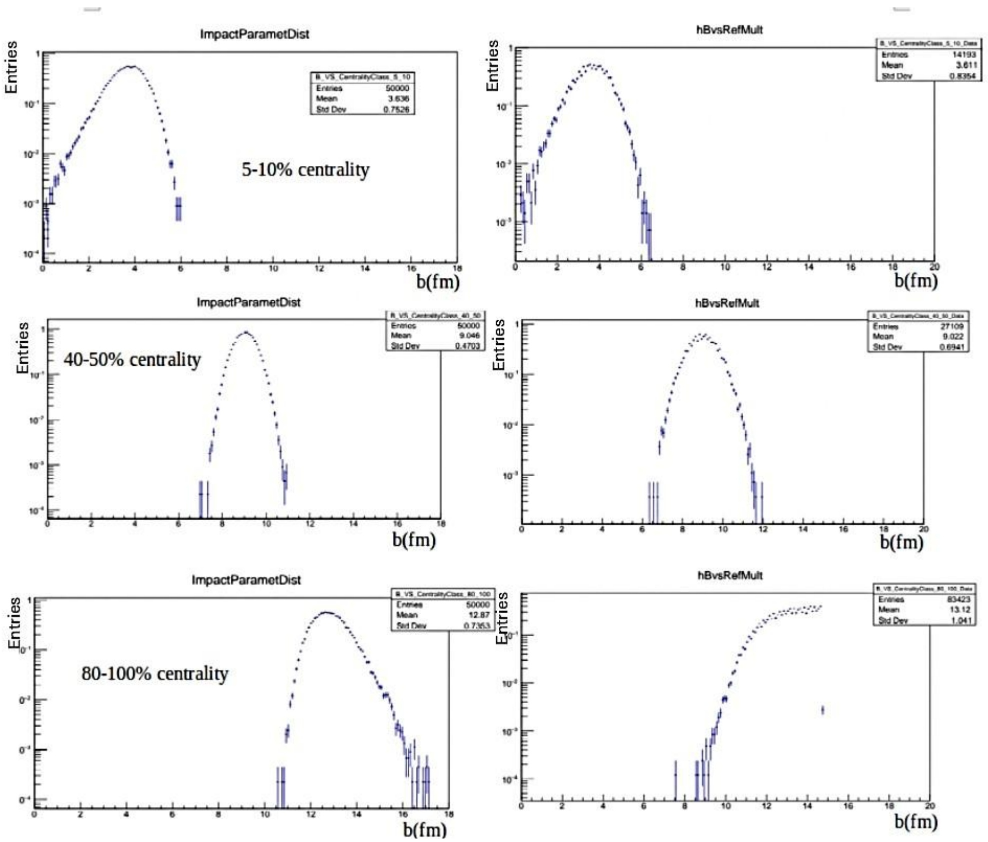
First the GammaFit method to perform the adjustment takes a multiplicity distribution from a file ". root", this multiplicity distribution is opposes from the reconstructed data and applied to it the cuts obtained earlier.



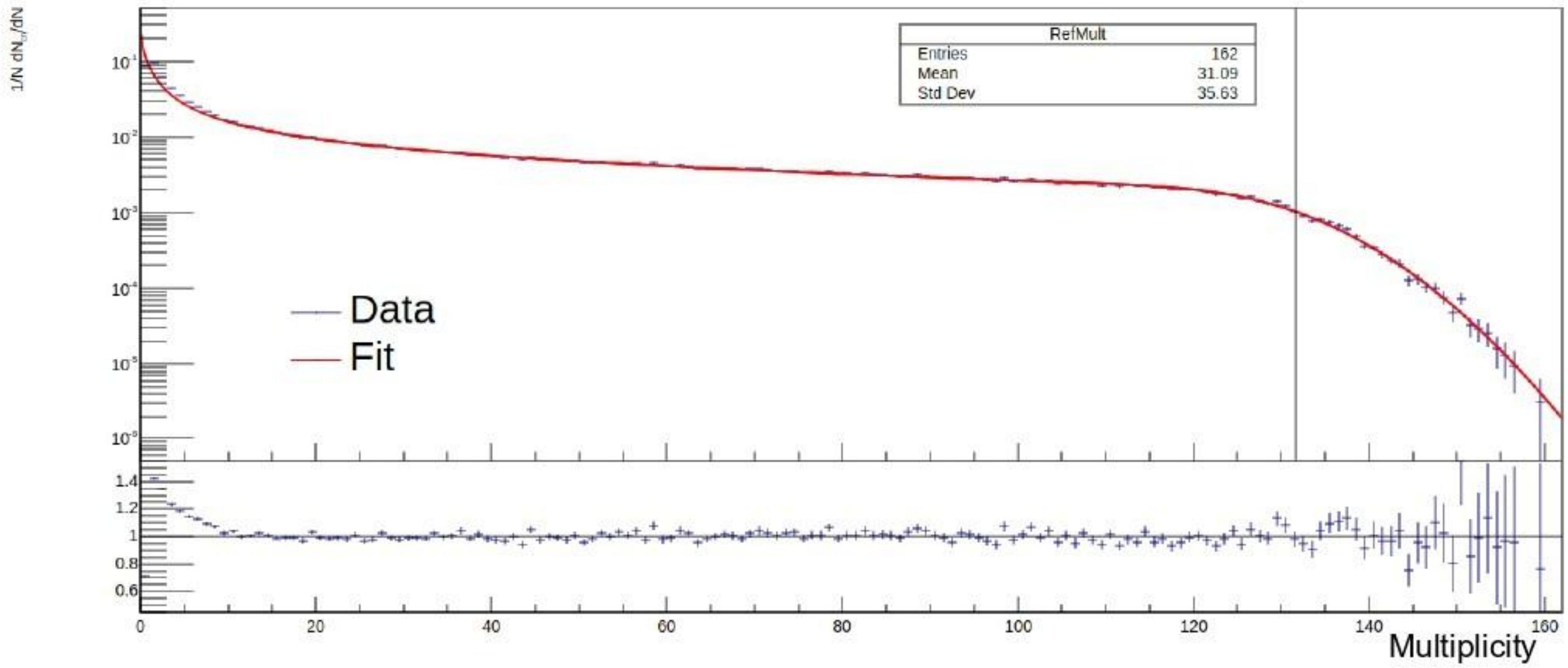
Cuts:
MCTracks > 308
Eta (-1 , 2)
DCA <= 2
NHits => 20
b < 13

Extrapolation of impact parameter by centrality

The adjustment is made by extrapolating and reconstructing the impact parameter for different classes of centrality, comparing with the data that it extracts directly from the simulation, in this case from the generator UrQMD



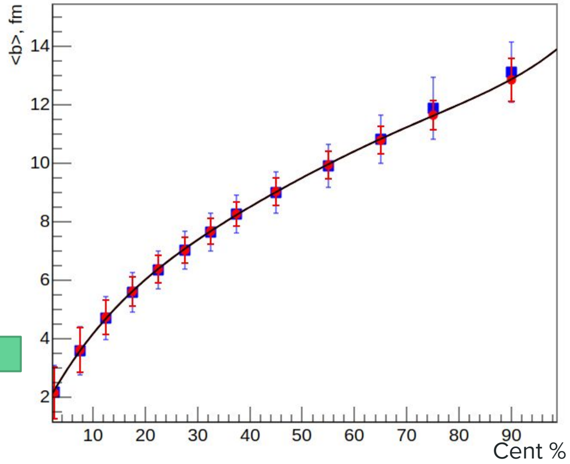
Adjustment of distribution

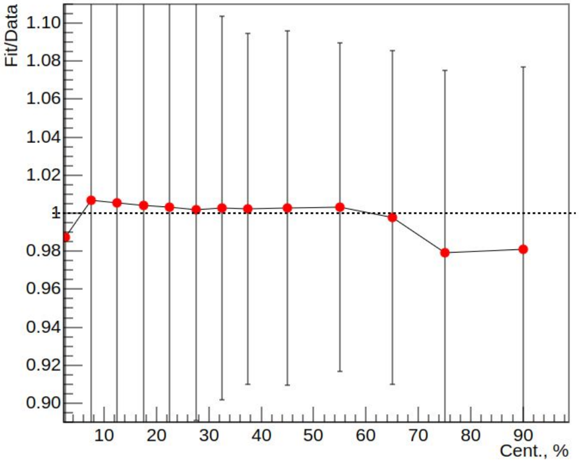


Then compare the adjustment from the reconstructed impact parameter with the input data as shown in the figure.

Comparison of impact parameter

Finally, the average of the impact parameter from the different classes of centrality is compared with the impact parameter extracted from the UrQMD





Parameter	Value
NDF	143
a1	-3.73
a2	0.164
a3	-2.84
chi^2	163.5
nknee	131.7
sigma	679.9
teta	0.647
chi^2/NDF	1.143111

In our case the Framework was configured so that all the adjustment is automatic, so at the end also the parameters used for the adjustment are extracted.

Multiplicity and impact parameter by centrality class

The Framework also gives us a table with the results of multiplicity and impact parameter by class of centrality

Centrality %	Multiplicity	Impact parameter
0-10	162-96	2.94-5.18
10-20	96-70	5.18-6.75
20-30	70-50	6.75-7.39
30-40	50-34	7.39-7.97
40-50	34-23	7.97-9.51
50-60	23-15	9.51-10.42
60-70	15-9	10.42-11.24
70-80	9-5	11.24-12.01
80-100	5-1	12.01-14.10

Comparison of MCGlauber and GammaFit methods

Comparison of multiplicity in centrality classes

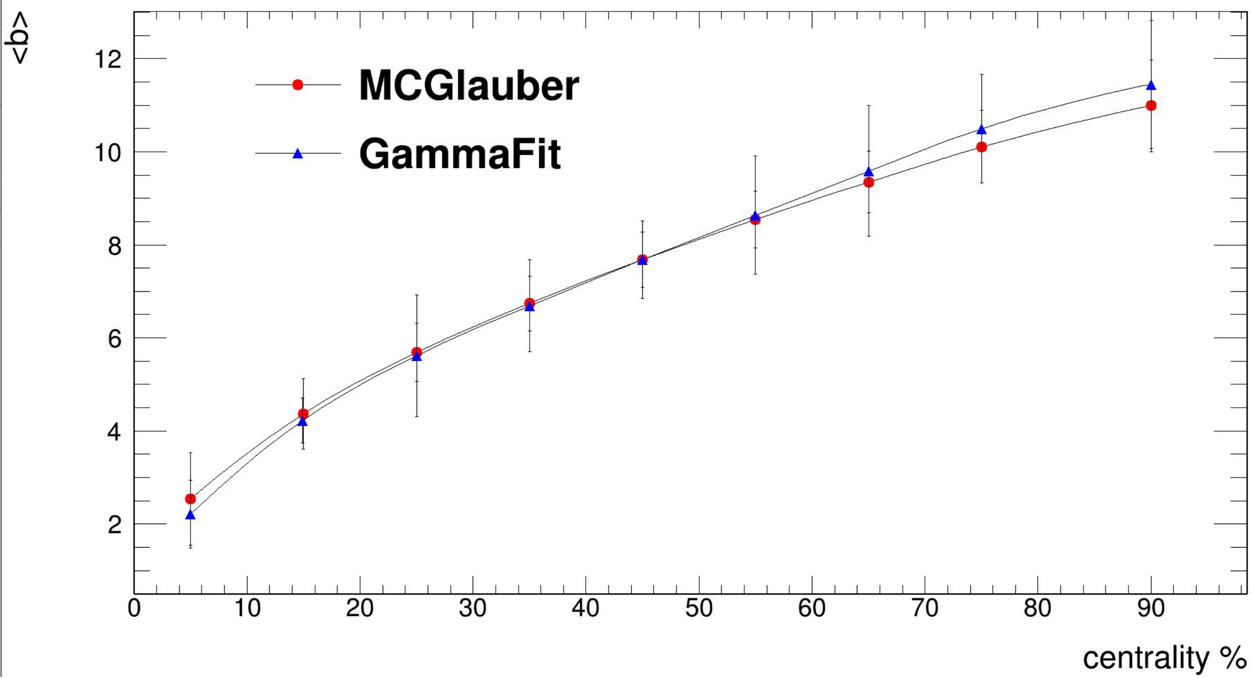
Centrality, %	$N^{min}_{chMCGlauber}$	$N^{max}_{chMCGlauber}$	$N^{min}_{chGammaFit}$	$N^{max}_{chGammaFit}$	ΔN^{min}_{ch}	ΔN^{max}_{ch}
[0 - 10]	95	164	96	162	1	2
[10 - 20]	70	95	70	96	0	1
[20 - 30]	51	70	50	70	1	0
[30 - 40]	36	51	34	50	2	1
[40 - 50]	24	36	33	34	9	2
[50 - 60]	15	24	15	23	0	1
[60 - 70]	9	15	9	15	0	0
[70 - 80]	5	9	5	9	0	0
[80 - 90]	2	5	1	5	1	0

In the table, we compare the data of multiplicity and the difference that exists between the two different methods, noting that in multiplicity the two methods are proportional.

Primary Kaons pT (Reco/MC)

Centrality %	$\langle b_{MCGlauber} \rangle$	$\langle b_{GammaFit} \rangle$
0-10	2.54	2.21
10-20	4.37	4.7
20-30	5.69	5.62
30-40	6.74	6.4
40-50	7.68	7.68
50-60	8.55	8.24
60-70	9.35	9.02
70-80	10.11	10.84
80-100	10.99	11.63

The same comparison was made for the average of the impact parameter, it can be seen that there are hardly any differences between the two methods.



Track Efficiency

Track Efficiency

Efficiency: $$Eff = \frac{p_T^{Reco}}{p_T^{MC}}$$

We divide Reconstructed pT Distributions over MC pT Distribution.
For Track Efficiency we are using MonteCarlo association

Accepted Reconstructed Tracks

From Track Selection:

MC Tracks > 308

Z-Vertex (-100,-70)

Eta (-1 , 2)

DCA <= 2

Number of Hits => 20

Impact Parameter b < 13

Montecarlo Tracks

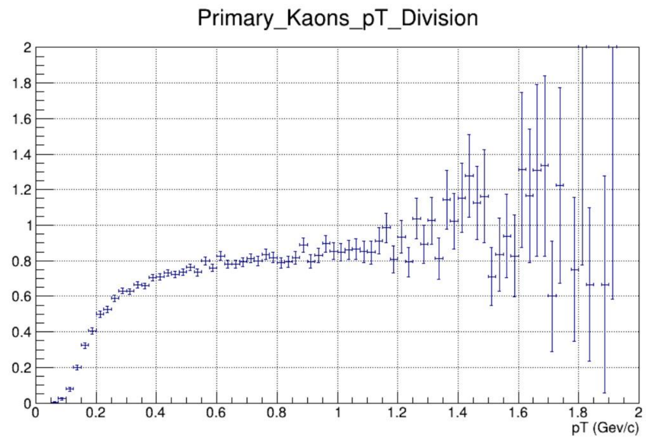
Matching Reco Tracks selection:

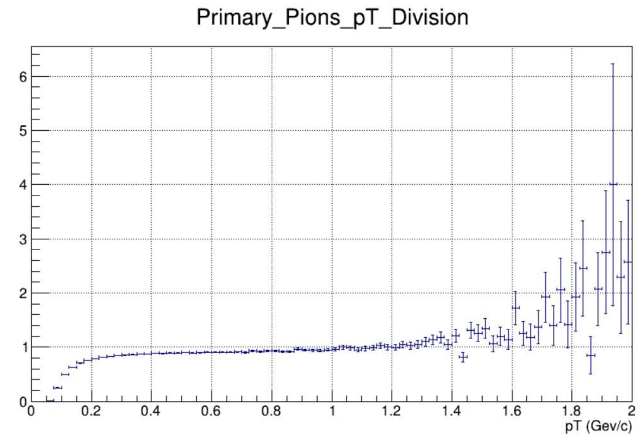
Z-Vertex (-100,-70)

Eta (-1 , 2)

Impact Parameter b < 13

Track Efficiency: Primary Particles

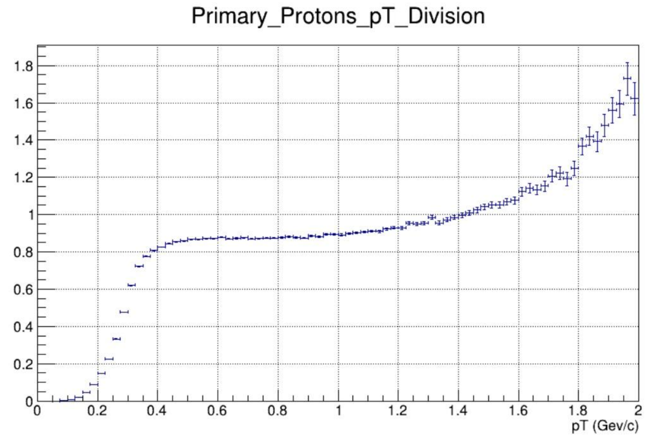


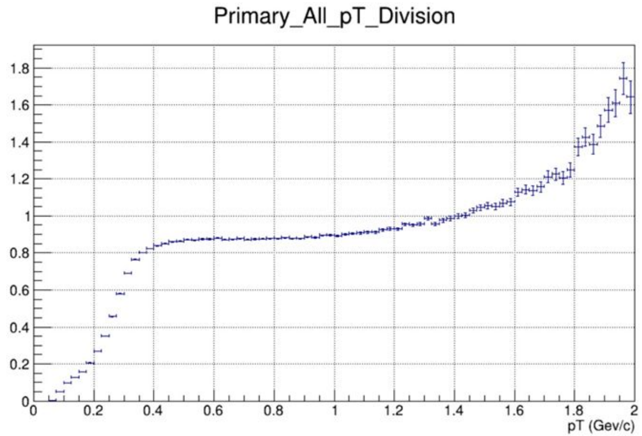


Efficiency stay below 1 till pT = 1 GeV/c. For greater values of pT reconstruction is not correct. This pattern is on protons as well.

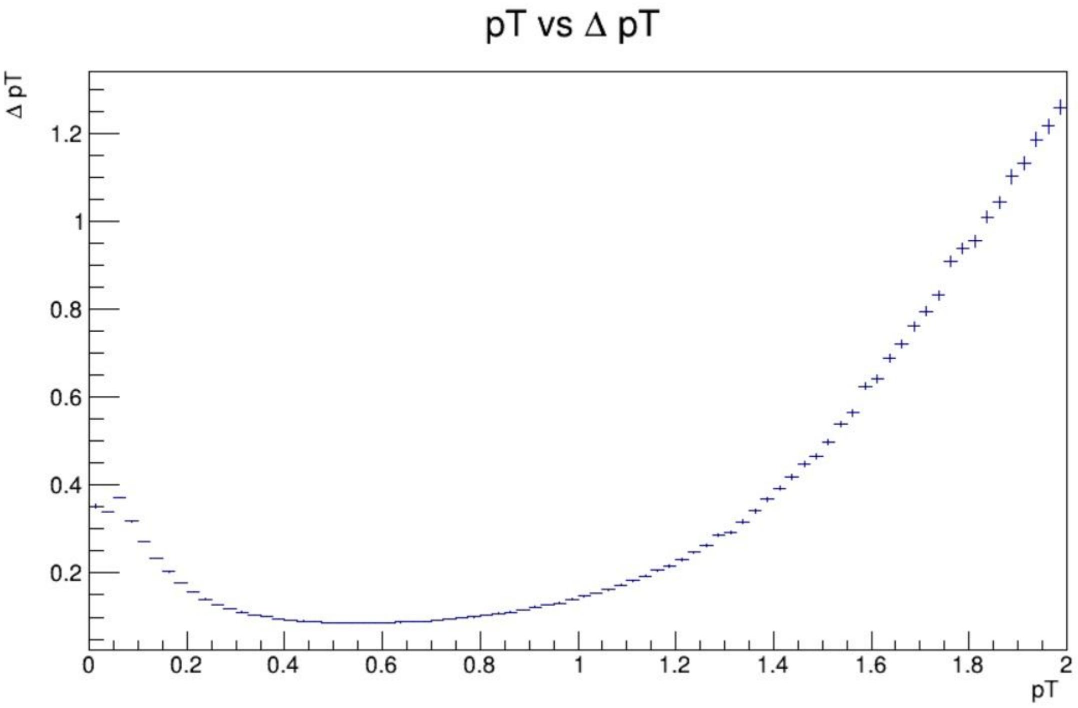
For Kaons and Pions stays below 1 til pT = 1.2 GeV/c.

Best Tracking Reconstruction is on pT < 1.2 GeV/c



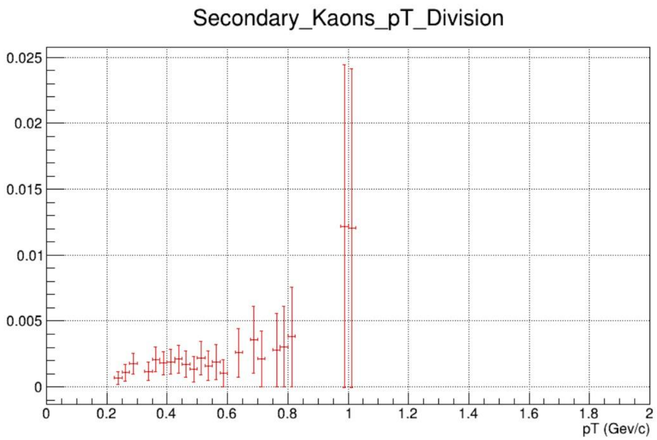


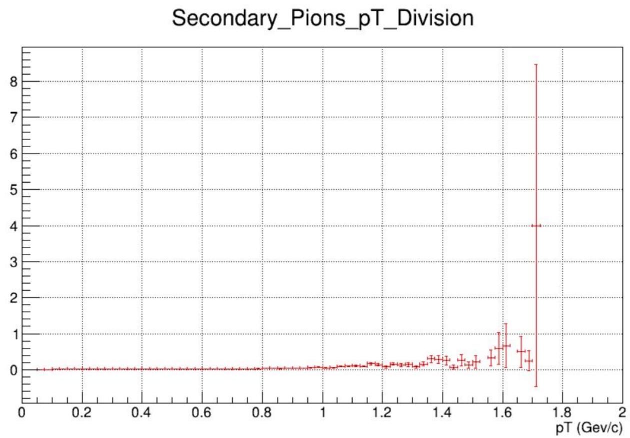
Track Efficiency

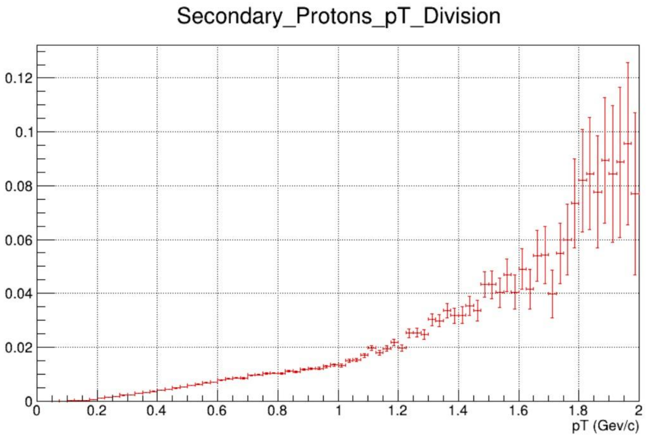


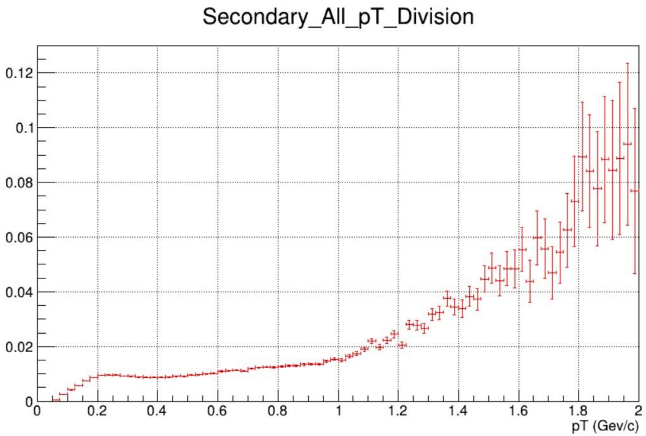
This pT value can be explain with transversal momentum distribution, here we can observe that pT has a lower resolution than 20% when pT values are below 1.2 GeV/c.

Track Efficiency: Secondary Particles







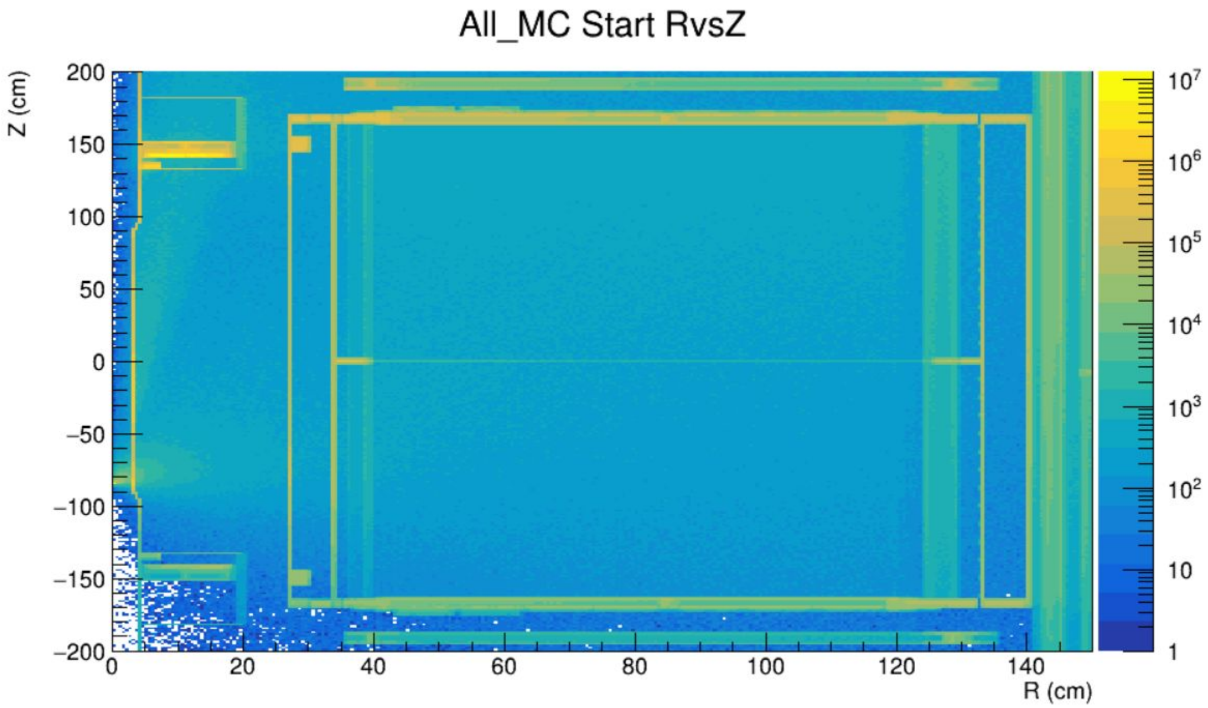


Efficiency is not done correctly, it should have a similar behavior as primary particles.

To explain this we take a look for starting point of secondary particles.

Track Efficiency

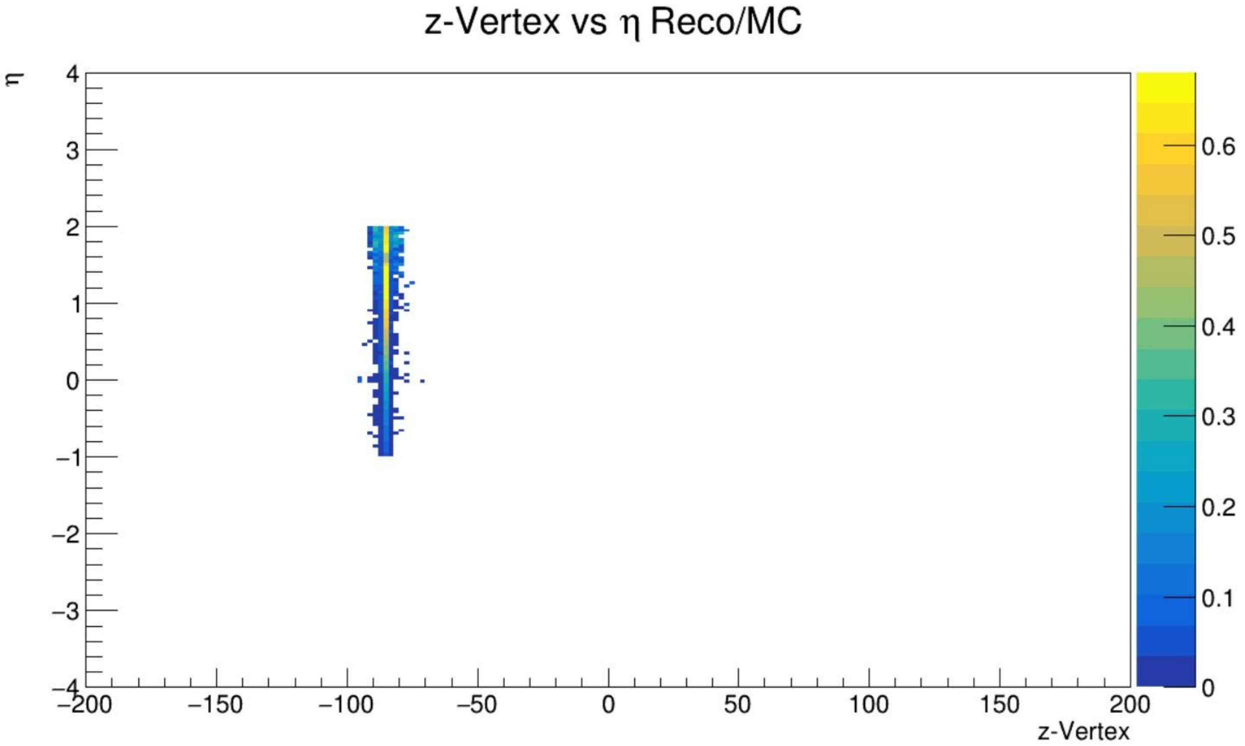
We can observed that most particles are being created on the edges of the detector, so reconstruction its not well done or particles won't collide with the detector and therefore we do not obtain enough reconstructed tracks



All Track Efficiency distribution are just preliminar work, improving them will be continue later on

Track Efficiency

This distribution gives Track Efficiency with respect to z-Vertex and pseudo rapidity, with this distribution we can perform a identification of centrality per event, using Centrality wagon of MpdRoot.



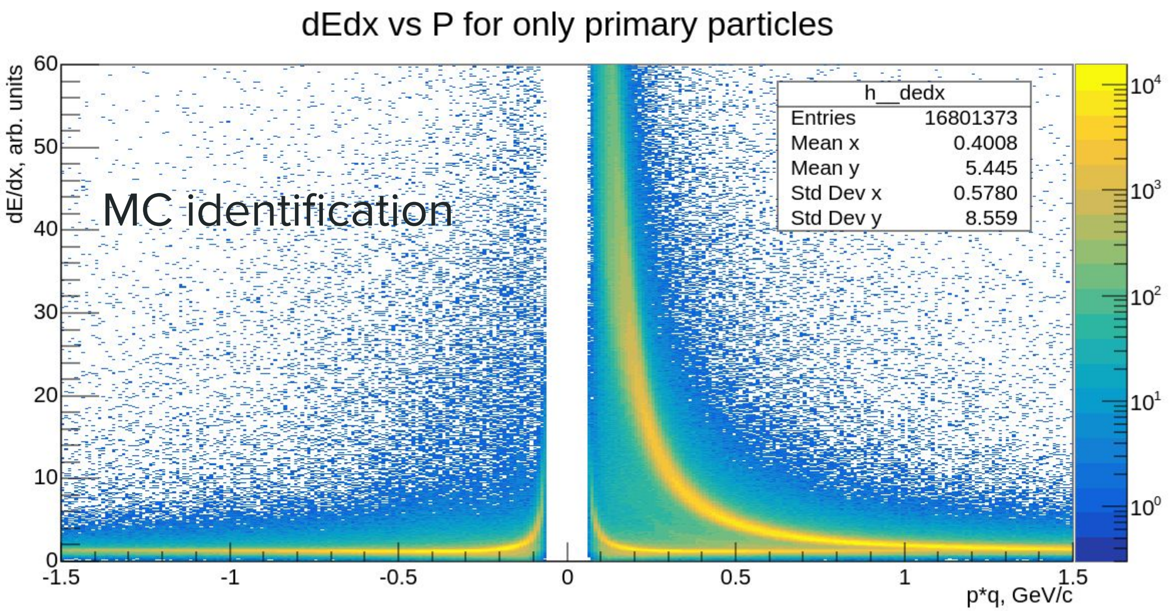
Particle Identification: Primary Particles.

Cuts on MC Tracks:
Eta (-1, 2)
Impact parameter b < 13

Cuts on Reco Tracks:
Eta (-1 , 2)
DCA <= 2
NHits => 20
Impact parameter b < 13

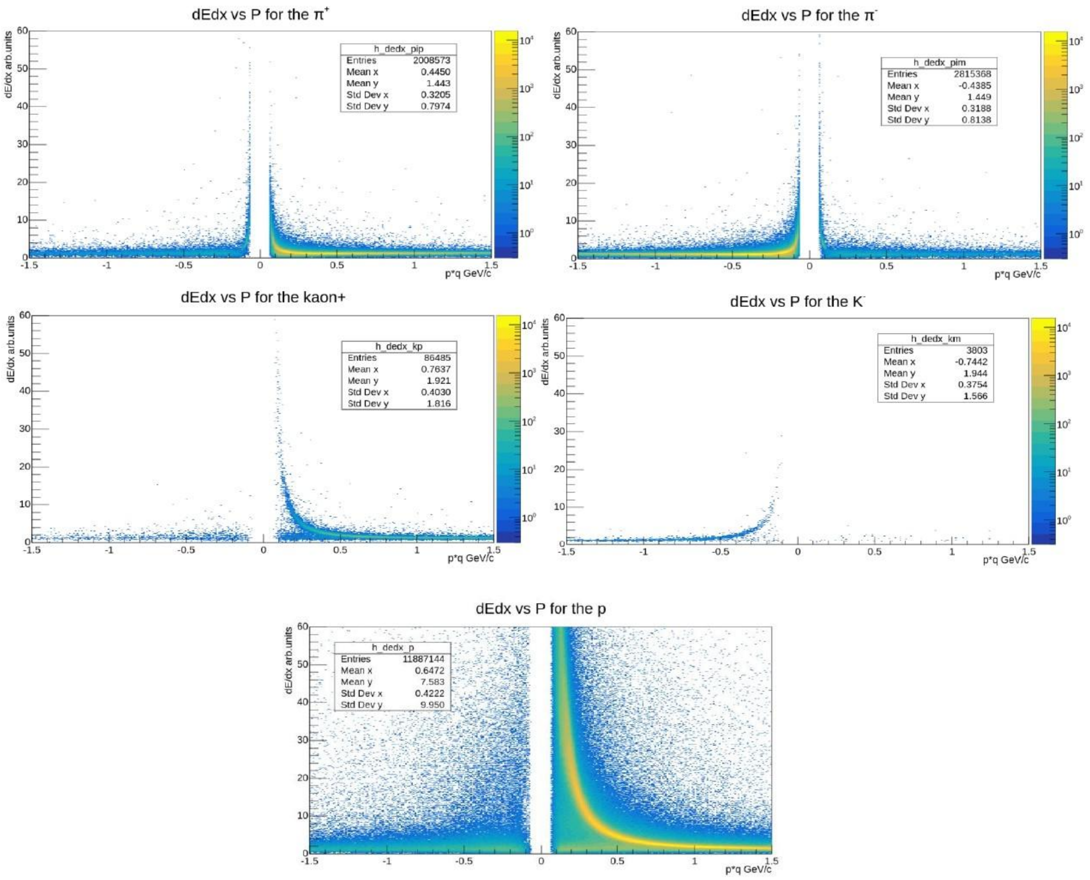
Loss energy

For particles identification, reconstructed data were used for energy loss histograms as function of the moment, using the cuts to have more limp distributions



Energy losses for different species of particles

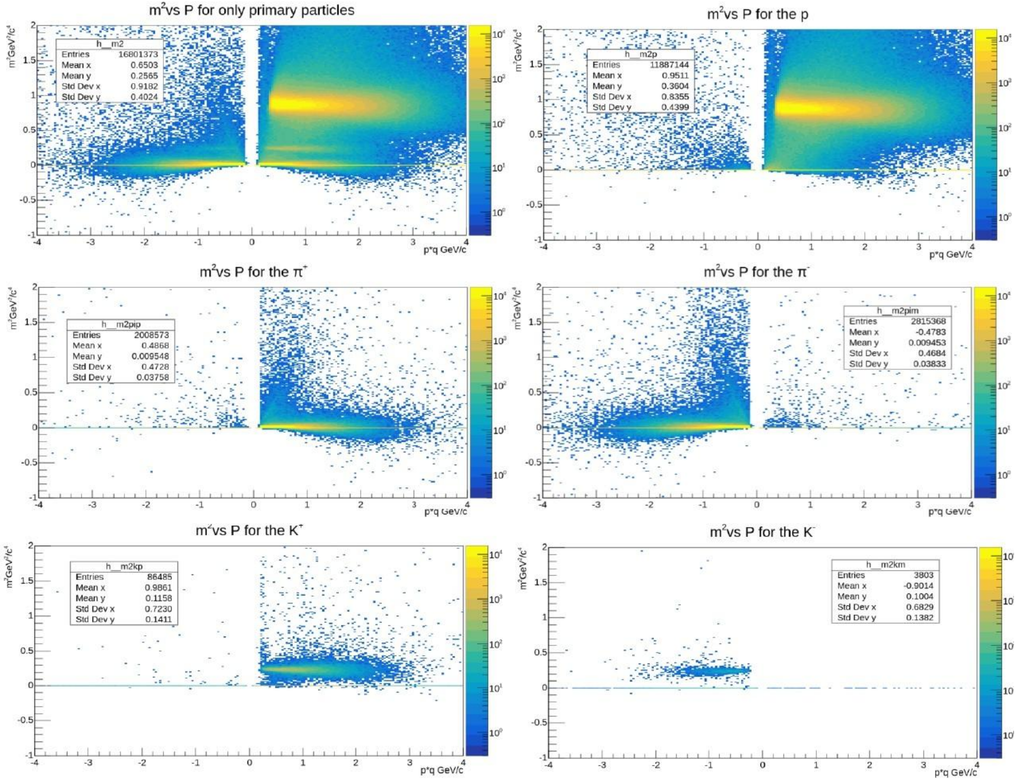
MC identification



Monte carlo identification is used to have energy loss distributions for different species of particles

Cuts on Reco Tracks:
Eta (-1 , 2)
DCA <= 2
NHits => 20
Impact parameter b < 13

Mass^2



MC identification

The same way to the energy loss histograms were obtained mass^2 histograms

Cuts on Reco Tracks:
Eta (-1 , 2)
DCA <= 2
NHits => 20
Impact parameter b < 13

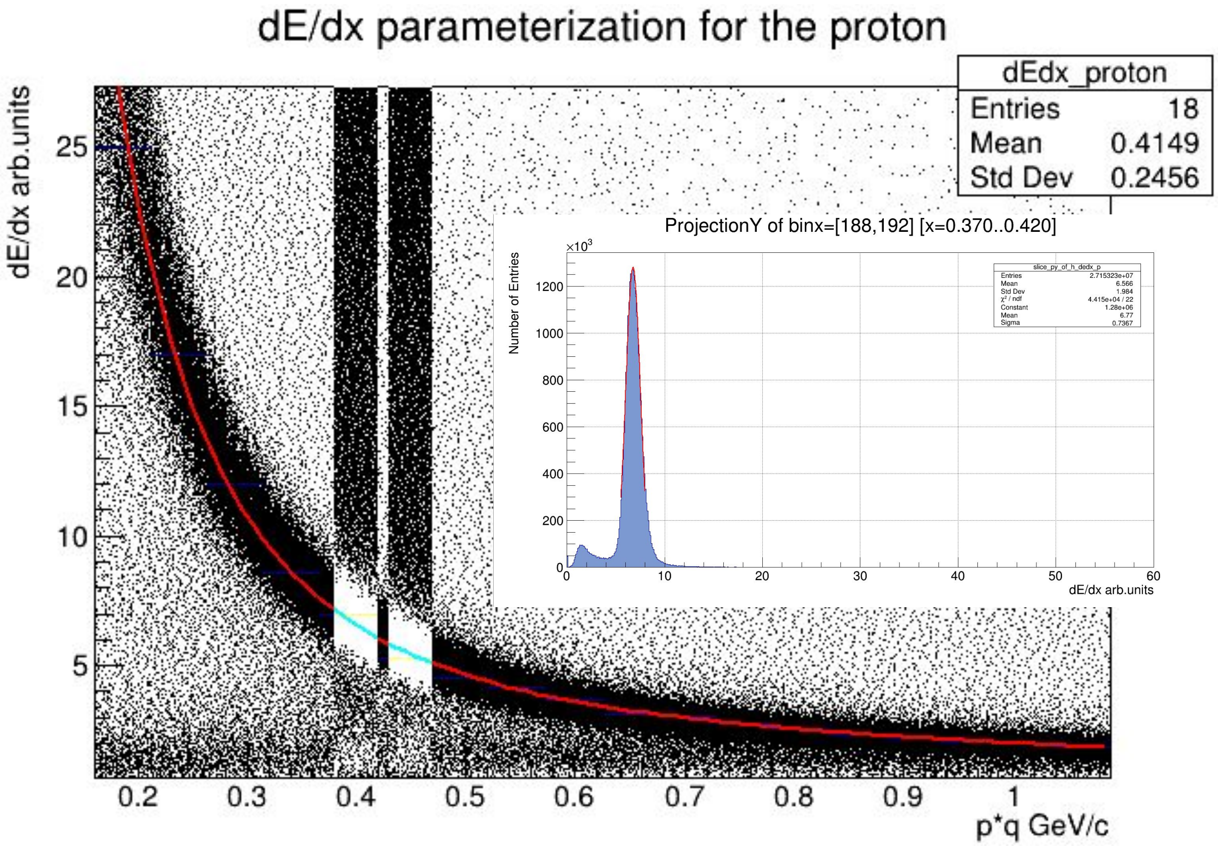
Bethe Bloch equation

$$\frac{dE}{dx} = \frac{a_0}{(\frac{p}{E})^{a_3}}(a_1 - (\frac{p}{E}^{a_3}) - \ln(a_2 + (\frac{m}{p})^{a_4})$$

An adjustment was made using the Bethe Bloch equation, in order to describe the energy loss that different species of particles have when passing through the detector

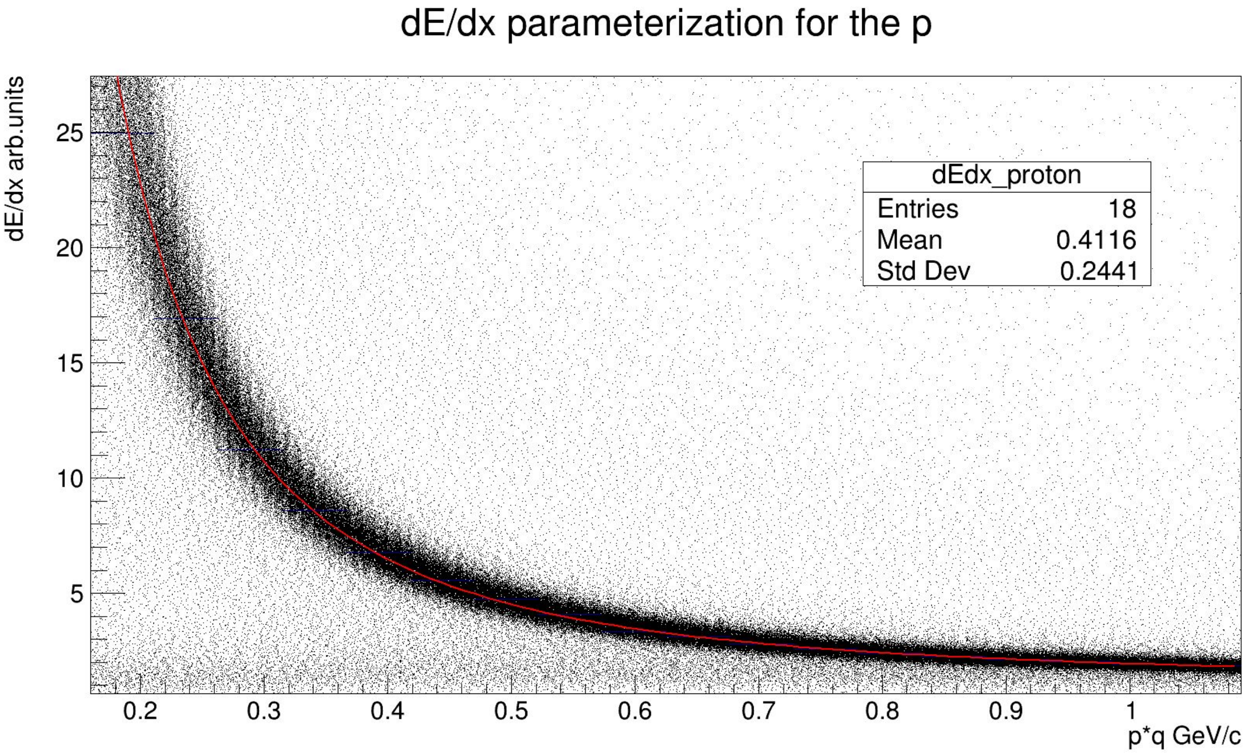
Adjustment for power loss

Projections on the Y-axis were used to create slices of 5 bins, to have 18 points where they adjusted Gaussians to find the mean value and their respective sigma



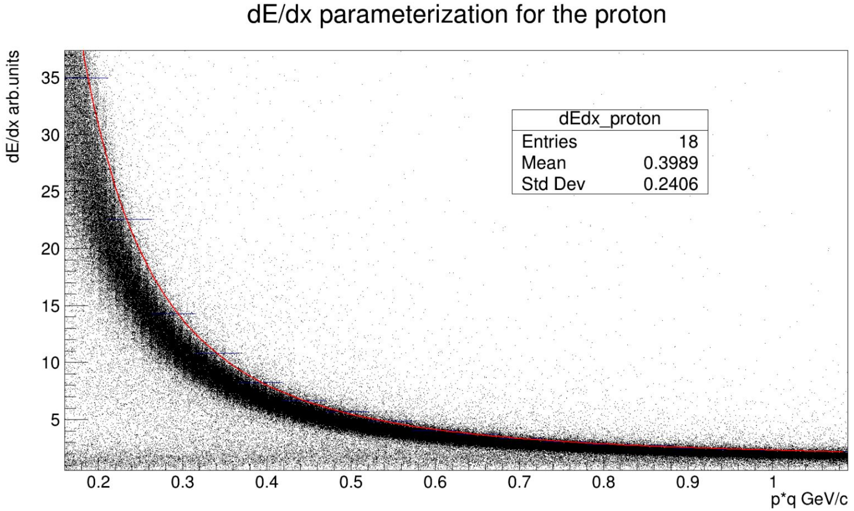
Adjustment for different species of particles

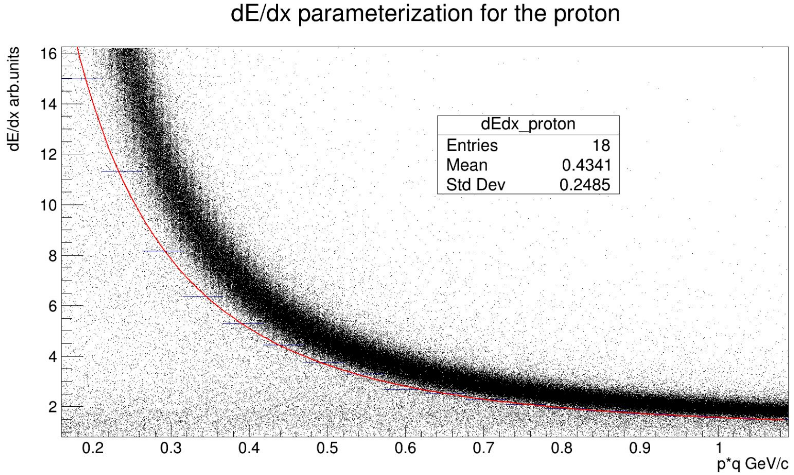
Adjustments were then made in the same way for protons, pions and kaons.



Mean and Sigmas value

With the mean value, first the value of its sigmas was added twice and the adjustment projected, then two sigmas were subtracted.





Parameters for the Bethe Bloch equation

By adjusting the mean value more than 2 times and less than 2 times the sigma, values are obtained for the Bethe Bloch equation for different species of particles.

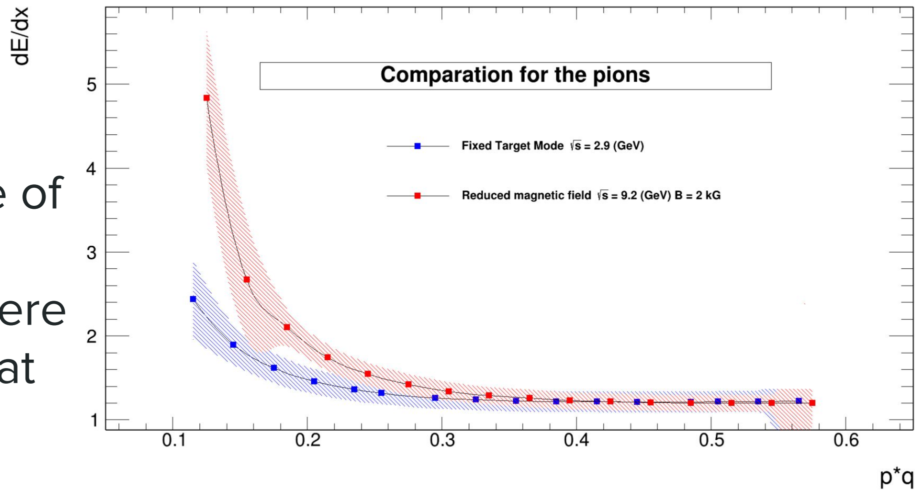
Parameter	p max	p min
p0	-1.3055	-1.338
p1	0.3812	0.8953
p2	1.4613	1.955
p3	1.521	0.7643
p4	1.502	2.4008

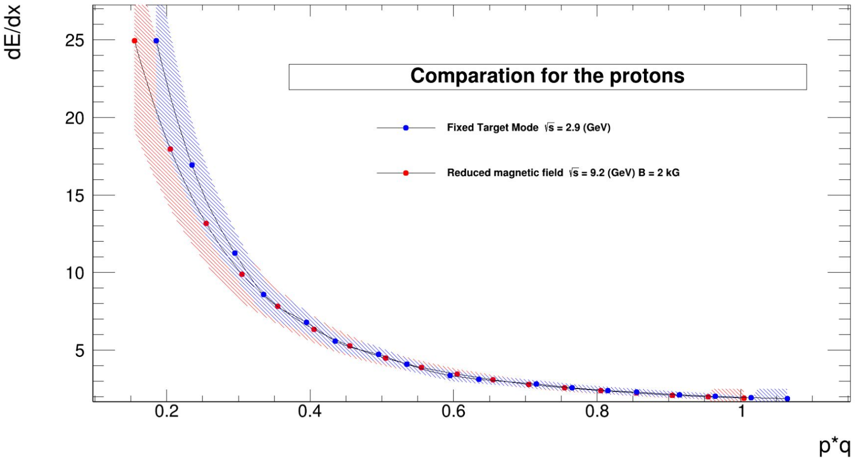
Parameter	π^+ max	π^+ min
p0	-0.694978	-0.4814
p1	-0.5444038	-0.6593
p2	-0.221145	-0.3791
p3	6.0991	6.8815
p4	-0.380694	-0.3162

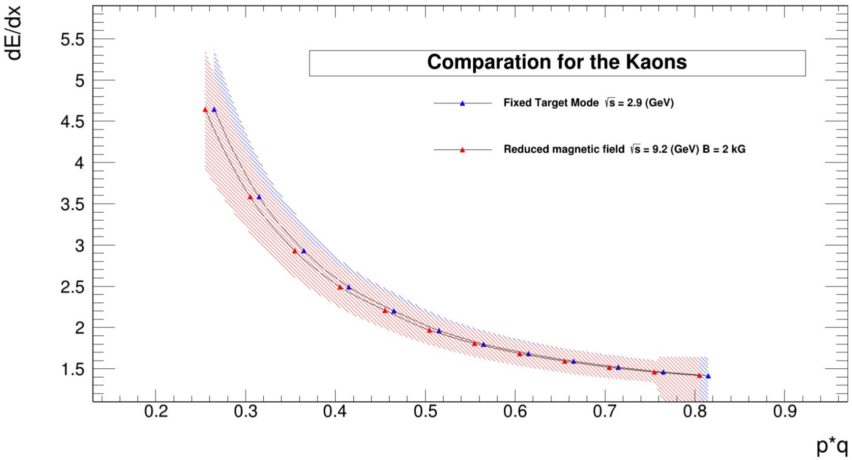
Parameter	K^+ max	K^+ min
p0	-0.954	-1.1444
p1	-0.0273	0.4036
p2	0.8683	1.2918
p3	2.395	1.2918
p4	0.60565	2.9028

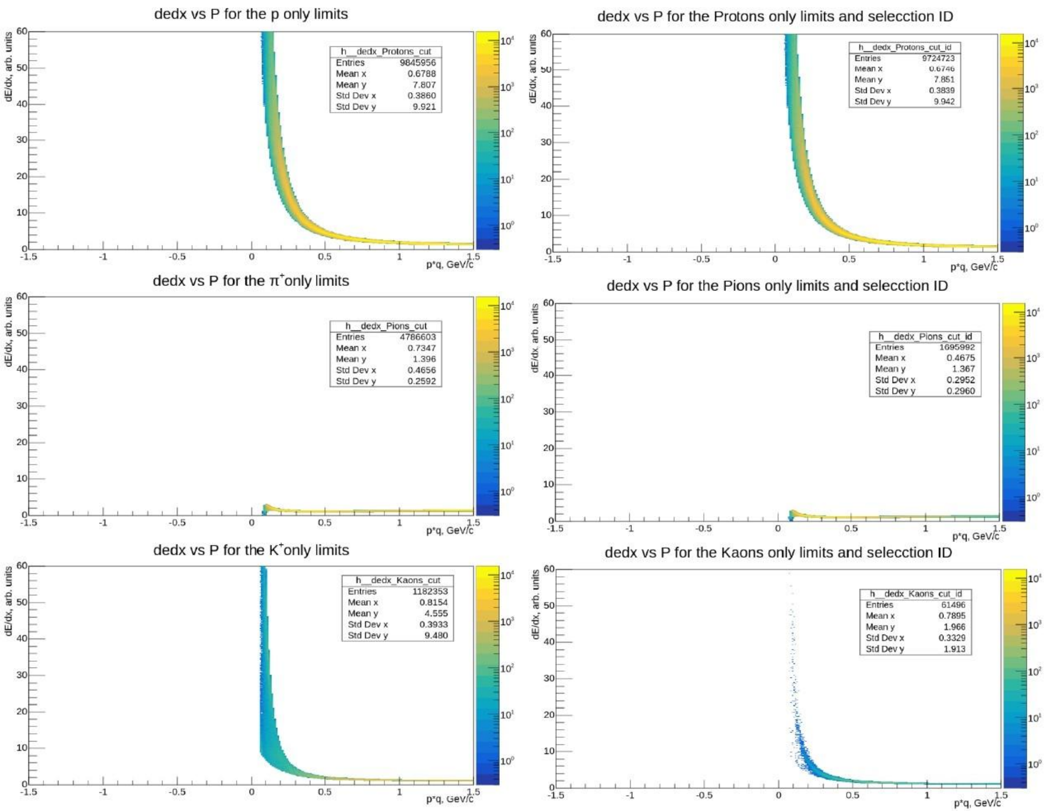
Comparison with reduced magnetic field

The adjustments were compared with those of the reduced magnetic field of my colleague Alejandro San Juan, noting that at low pT there are more notable differences for pions, but at higher pT have to follow the same energy losses







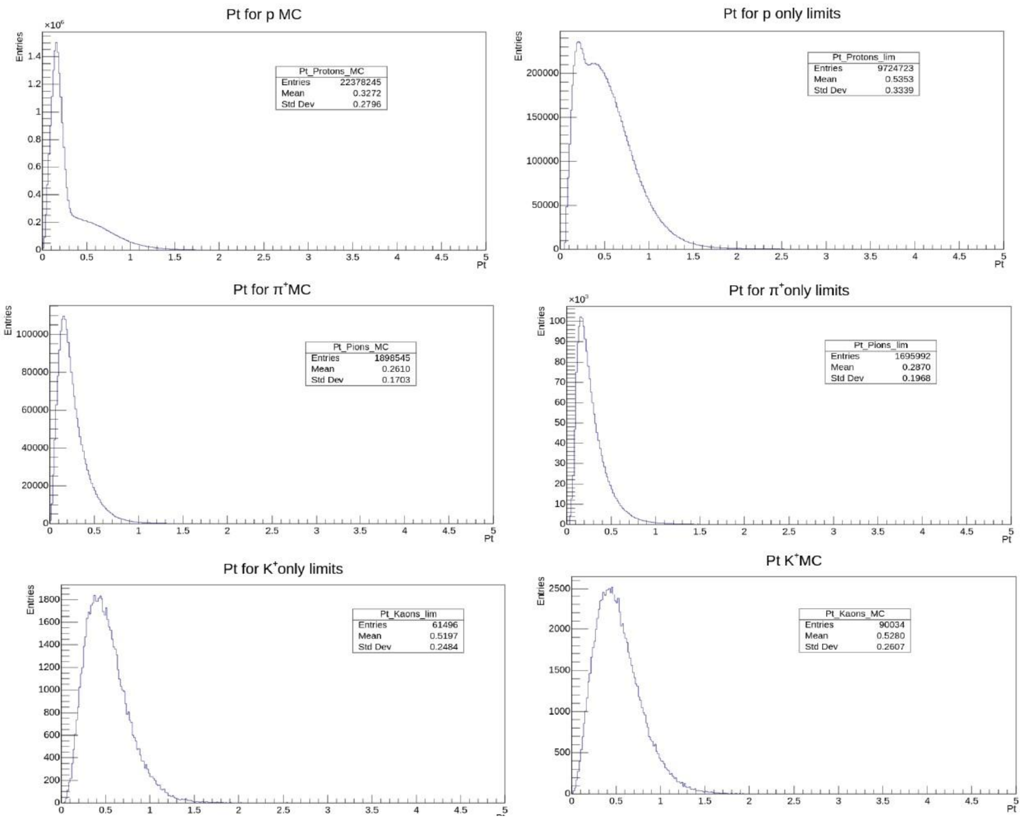


Comparison with limits and with limits and MC identification

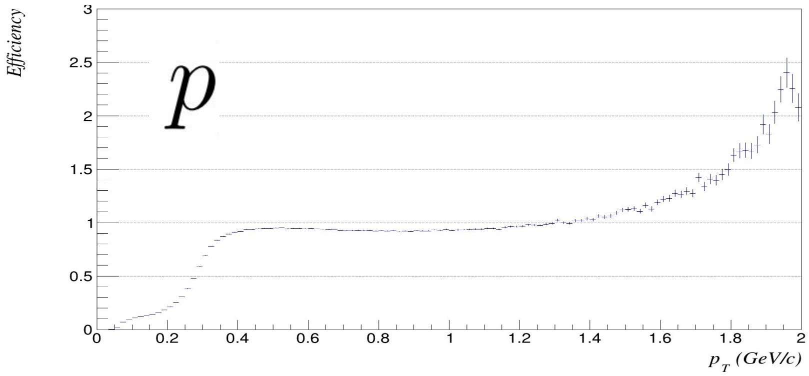
With the adjustments of the mean value and sigma were used to restrain the entry of particles and compared with histograms with the cuts and with MC identification.

pT Rec and pT MC

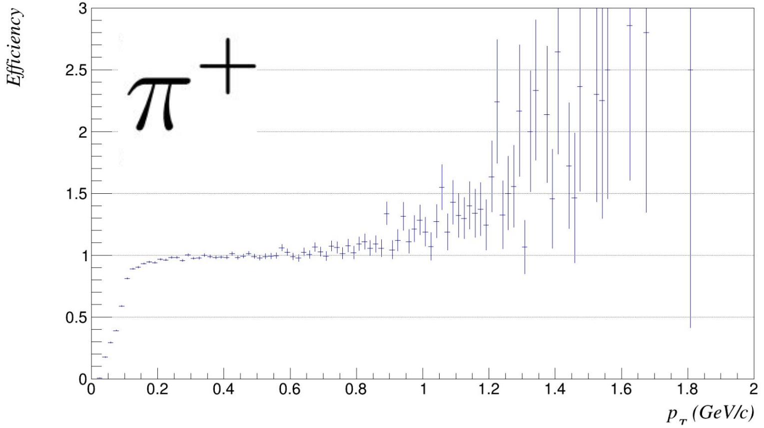
With the same section that was made for energy loss selecting limits of mean value and two sigmas pT histograms were obtained for reconstructed, also pT histograms were obtained with MC data

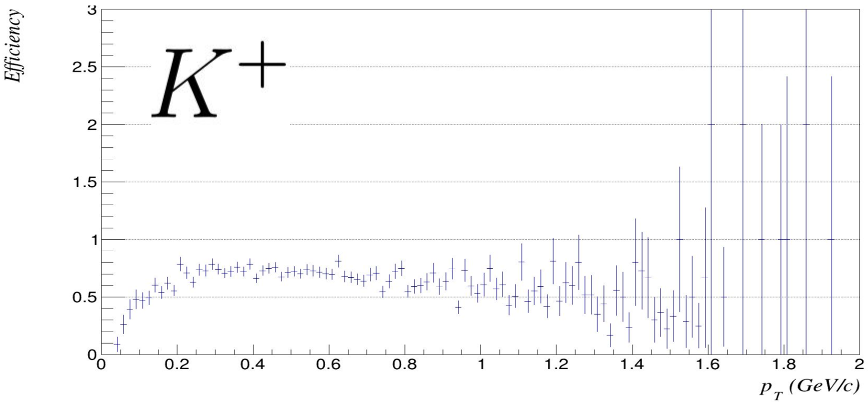


pT Rec and pT MC



In order to verify that the selection of particles was as good, pT efficiency histograms were obtained for pions, protons and kaons





Future work

In the future it is planned to continue working with the collaboration:

- **Making adjustments for energy losses for more particles (antipiones and antikaon)**
- **Identification for the masses**
- **Adding studies for centrality by species of particles using PID**

Thank you for your attention

Спасибо, до встречи

Appendix: Code

```
  //MonteCarlo Loop
for (Int_t i = 0; i < NumTracksMC; i++){
  MpdMCTrack *MCTrack = (MpdMCTrack*)fMCTracks->UncheckedAt(i);

  Int_t abspdgcode;
  Double_t pTMC;
  Double_t P;
  Double_t Pz;
  Double_t Eta;

  abspdgcode = TMath::Abs(MCTrack->GetPdgCode());

  pTMC = MCTrack->GetPt();
  P = MCTrack->GetP();
  Pz = MCTrack->GetPz();
  Eta = 0.5*TMath::Log((P + Pz)/(P - Pz));

  if (abspdgcode == 321 || abspdgcode == 2212 || abspdgcode == 211 || abspdgcode == 11)// kaons, protons, pions, electrons
  {
    if(pTMC <= 2.5 && MCTrack->GetMotherId() == -1 && Impactb < 13 && Eta > -1 && Eta < 2)
    {//Matching cuts of RecoTracks
      MC_pT->Fill(pTMC);
      HisMCPT->Fill(pTMC);
      ZVertex_Eta_MC->Fill(ZReco,Eta);
    }
  }

  if(abspdgcode == 321 || abspdgcode == 2212 || abspdgcode == 211){// kaons, protons, pions
              CounterMC++; //Counting the number of kaones, protons and pions
            }
}
```

```
  if (Impactb < 13 && Eta_Reco > -1 && Eta_Reco < 2 && DCA <= 2 && NumHits >= 20 )
  { //Impact Parameter Cut
    Prof3D[p][3]->Fill(Eta_Reco,Pt_Reco,Diff_Pt);
    Prof3D[2][3]->Fill(Eta_Reco,Pt_Reco,Diff_Pt);
    Multiplicity++;

    HistMCPSKpPi1[0][p][1][0]->Fill(Pt_MC);
    HistMCPSKpPi1[1][p][1][0]->Fill(Pt_Reco);
    HistMCPSKpPi1[0][p][3][0]->Fill(Pt_MC);
    HistMCPSKpPi1[1][p][3][0]->Fill(Pt_Reco);

    Prof_2D_Ntracks_zVertex_cuts->Fill(ZReco,NumTracksVertex,DzVertex);
    ZVertex_Eta_R->Fill(ZReco,Eta_Reco);
    pT_DiffpT_Reco_cuts->Fill(Pt_Reco,Diff_Pt);

    if (Pt_Reco <= 2.5)
    {
      HisRecoPT->Fill(Pt_Reco);
      Reco_pT->Fill(Pt_Reco);
    }
  }
}
//Fill Multiplicity cuts on b, eta, DCA, NHits
hRefMult->Fill(Multiplicity);
hBvsRefMult->Fill(Multiplicity,Impactb);
```

https://github.com/iamaldonado/START_Summer24/blob/main/AdrianLara/Classes_created/principal/Fixed_Analysis.cxx

Appendix: Code

```
  if (ZReco > -100 && ZReco < -70)//Cut on Primary Vertex
  {
    if (Impactb < 13 && Eta_Reco > -1 && Eta_Reco < 2 && DCA <= 2 && NumHits >= 20 )
    { //Impact Parameter Cut
      Multiplicity++;

      if(Pt_MC == 0) continue;
      HistMCPSKpPi1[1][p][1][0]->Fill(Pt_Reco);
      HistMCPSKpPi1[1][p][3][0]->Fill(Pt_Reco);

      Dpt_pt_Multi_Profile_cuts->Fill(Pt_Reco,Diff_Pt);
    }
  }
 }
 DZVertex_zVertex->Fill(ZReco,DzVertex);
 DZVertex_Multi->Fill(Multiplicity,DzVertex);
 DZVertex_Multi_Profile->Fill(Multiplicity,DzVertex);
 zVertex_b->Fill(ZReco,Impactb);
}
```

```
Division[0][j][1][k] = new TH1F(hist_name , title_name , HistMCPSKpPi1_bins[k].X , HistMCPSKpPi1_inter[k].Xlow , HistMCPSKpPi1_inter[k].Xup );
Division[0][j][1][k]->Divide(HistMCPSKpPi1[1][j][1][k],HistMCPSKpPi1[0][j][1][k]);
Division[0][j][1][k]->SetOption("E1");
```

https://github.com/iamaldonado/START_Summer24/blob/main/AdrianLara/Classes_created/Fixed_TrackEff/Fixed_Track.cxx

Appendix: Code

```
    int ntrmc = mMCTracks -> GetEntries();
    for (long int i = 0; i < ntrmc; i++) {

      MpdMCTrack* mctrack = (MpdMCTrack*) mMCTracks -> At(i);                          // Monte Carlo track is open for reading

      int   pdg                  = mctrack -> GetPdgCode();                               // Track PDG code
      int   prodId               = mctrack -> GetMotherId();                              // Track primacy: -1 = primary, any other = secondary
      float rapidity_mc          = mctrack -> GetRapidity();                              // Particle rapidity (CAN BE WRONG!!!)
      float pt_mc                = mctrack -> GetPt();                                    // Particle transverse momentum
      float p_mc                 = mctrack -> GetP();                                     // Particle full momentum
      float pz_mc                = mctrack -> GetPz();                                    // Particle momentum z-component


    Double_t Eta_MC = 0.5 * TMath::Log((p_mc + pz_mc) / (p_mc - pz_mc + 1e-16));


    //if   (Impactb < 13 && Z > -100 && Z < -75 && Eta_MC < 2 && Eta_MC > -1 ){
    if   (Impactb < 13 && Eta_MC < 2 && Eta_MC > -1 ){

    if( mctrack -> GetMotherId() != -1)continue;


    if (pdg == 2212)
    {
            //Proton (p)
            pt_protones_mc-> Fill (pt_mc);

    }else if (pdg == 321)
    {
            //Kaonplus (kp)
            pt_kaones_mc-> Fill (pt_mc);

    }else if (pdg == 211)
    {
            //pionplus (pip)
            pt_piones_mc-> Fill (pt_mc);
    }

}

}//mc
```

https://github.com/iamaldonado/START_Summer24/tree/main/FrankReyes

Appendix: Code

```
int ntr   = mKalmanTracks -> GetEntries();
 for (long int i = 0; i < ntr; i++) {
 MpdTrack*mpdtrack = (MpdTrack*) mMpdGlobalTracks->UncheckedAt(i);       // Global track
 MpdTpcKalmanTrack *kftrack  = (MpdTpcKalmanTrack*) mKalmanTracks->UncheckedAt(i); // The corresponding TPC Kalman track is also used
 int    kfcharge = kftrack -> Charge();                                              // for the charge
 double p        =  kftrack -> Momentum3().Mag() * kfcharge;                         // full momentum and
 double dedx     =  kftrack -> GetDedx();                                            // dE/dx information
 float pt_recon = mpdtrack -> GetPt();

 int mcId = kftrack -> GetTrackID();                                                 // MpdTpcKalmanTrack::GetTrackID() gives the ID of the corresponding Monte Carlo track
 MpdMCTrack* mctrack = (MpdMCTrack*) mMCTracks -> At(mcId);                          // Monte Carlo track is open for reading
 int    pdg                 = mctrack -> GetPdgCode();                               // Track PDG code
 int    prodId              = mctrack -> GetMotherId();                              // Track primacy: -1 = primary, any other = secondary

 if(mctrack -> GetMotherId() != -1) continue; //primary tracks
 //if(RECOmID == 2 && mctrack -> GetMotherId() != -1) continue; //secondary tracks
```

Appendix: Code

```
   Double_t proton_min =  ((-1.33806) / TMath::Power(p / TMath::Sqrt(p * p + 0.88), (0.76431))) * ( ( (0.895334) - TMath::Power(p / TMath::Sqrt(p * p + 0.88), (0.76431))) - (TMath::Log((1.95555) + T
   Double_t proton_max =   ((-1.30554) / TMath::Power(p / TMath::Sqrt(p * p + 0.88), (1.52176))) * ( ( (0.381237) - TMath::Power(p / TMath::Sqrt(p * p + 0.88), (1.52176))) - (TMath::Log((1.46136) +


Double_t pi_max = ((-0.694978) / TMath::Power(p / TMath::Sqrt(p * p + 0.01949), 6.0991)) * ( ( (-0.544038) - TMath::Power(p / TMath::Sqrt(p * p + 0.01949), 6.0991)) - (TMath::Log((-0.221145) + TMath
Double_t pi_min = ((-0.481401) / TMath::Power(p / TMath::Sqrt(p * p + 0.01949), (6.88153))) * ( ( (-0.65936) - TMath::Power(p / TMath::Sqrt(p * p + 0.01949), (6.88153))) - (TMath::Log((-0.37918) + T

Double_t kaon_max = ( (-0.954003) / TMath::Power(p / TMath::Sqrt(p * p + 0.2437), (2.39503)))  *  ( ( (-0.0273105) - TMath::Power(p / TMath::Sqrt(p * p + 0.2437), (2.39503))) - (TMath::Log((0.868322
Double_t kaon_min = ( (-1.14442) / TMath::Power(p / TMath::Sqrt(p * p + 0.2437), (0.119658)))  *  ( ( (0.403622) - TMath::Power(p / TMath::Sqrt(p * p + 0.2437), (0.119658))) - (TMath::Log((1.29185)
```

https://github.com/iamaldonado/START_Summer24/tree/main/FrankReyes

Appendix: Code

```
if (Impactb < 13 && Eta_Reco > -1 && Eta_Reco < 2 && DCA <= 2 && NumHits >= 20 ){  //Cuts

   if (dedx >=proton_min && dedx<=proton_max  ){
   dedx_protonescut -> Fill(p, dedx);
   ptcut_protones -> Fill (pt_recon);
    }

    if (dedx >=pi_min && dedx<=pi_max  ){
   dedx_pionescut -> Fill(p, dedx);
   ptcut_piones -> Fill (pt_recon);
    }

   if (dedx >=kaon_min && dedx<=kaon_max  ){
   dedx_kaonescut -> Fill(p, dedx);
   ptcut_kaones -> Fill (pt_recon);
   }


   h__dedx -> Fill(p, dedx);
   h__m2 -> Fill(p, mpdtrack -> GetTofMass2());
```

https://github.com/iamaldonado/START_Summer24/tree/main/FrankReyes

```
        if (pdg == 2212)
    {
       //Proton (p)
       h__dedxp->Fill(p, dedx);
       h__m2p -> Fill(p, mpdtrack -> GetTofMass2());
       if (dedx >=proton_min && dedx<=proton_max  ){
       iddedx_protonescut -> Fill(p, dedx);
       idptcut_protones -> Fill (pt_recon);
        }

    }else if (pdg == 321)
    {
       //Kaonplus (kp)
       h__dedxkp->Fill(p, dedx);
       h__m2kp -> Fill(p, mpdtrack -> GetTofMass2());
       if (dedx >=kaon_min && dedx<=kaon_max  ){
       iddedx_kaonescut -> Fill(p, dedx);
       idptcut_kaones -> Fill (pt_recon);
       }
    }else if (pdg == -321)
    {
       //Kaonminus (km)
       h__dedxkm->Fill(p, dedx);
       h__m2km -> Fill(p, mpdtrack -> GetTofMass2());
}else if (pdg == 211)
    {
       //pionplus (pip)
       h__dedxpip->Fill(p, dedx);
       h__m2pip -> Fill(p, mpdtrack -> GetTofMass2());
       if (dedx >=pi_min && dedx<=pi_max  ){
       iddedx_pionescut -> Fill(p, dedx);
       idptcut_piones -> Fill (pt_recon);
        }

    }else if (pdg == -211)
    {
       //pionminus (pip)
       h__dedxpim->Fill(p, dedx);
       h__m2pim -> Fill(p, mpdtrack -> GetTofMass2());
   }
        //}//kalman

}//cuts recos
```

Appendix: Code

https://github.com/iamaldonado/START_Summer24/tree/main/FrankReyes