Update on dE/dx

Tracking Meeting

Charles Hughes
chughes2@iastate.edu
March 29, 2023

Progress Update

- (Reminder) From Workfest
  - Using [coresoftware/simulation/g4simulation/g4eval/TrackEvaluation.cc]
- From last week's meeting
  - dE should come from cluster ADC
  - dx should come from layer thickness and track state angles (α & β)
- From Jin Huang: exercise (debug dE and dx)
  - Simulate cosmic ray muons
  - Simulate proton, He4, Li7, Be9, B11 (see if <dE> scales as Z^2)

Progress Update

- **dE should come from cluster ADC**
  - [Using the functionality in add_cluster_size in TrackEvaluation.cc]

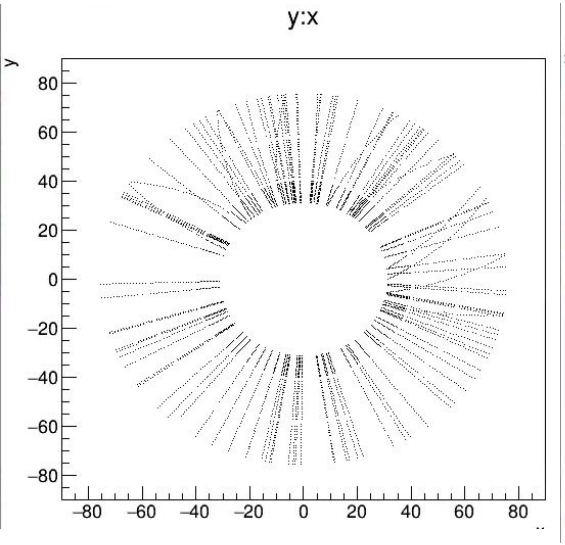
```
//! number of hits associated to cluster
void add_cluster_size( TrackEvaluationContainerv1::ClusterStruct& cluster, TrkrCluster* trk_clus)
{

  TrkrClusterv5 *trk_clusv5 = dynamic_cast<TrkrClusterv5*> (trk_clus);
  cluster.size = trk_clusv5->getSize();
  cluster.phi_size = trk_clusv5->getPhiSize();
  cluster.z_size = trk_clusv5->getZSize();
  cluster.ovlp = trk_clusv5->getOverlap();
  cluster.edge = trk_clusv5->getEdge();
  cluster.adc = trk_clusv5->getAdc();
  cluster.max_adc = trk_clusv5->getMaxAdc();

}
```

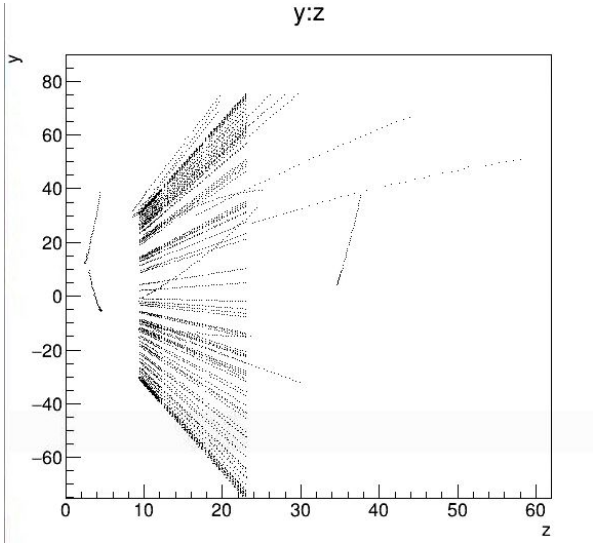
This in turn comes from [getADC() in trackbase/TrkrClusterv5.h] * (slide 16)

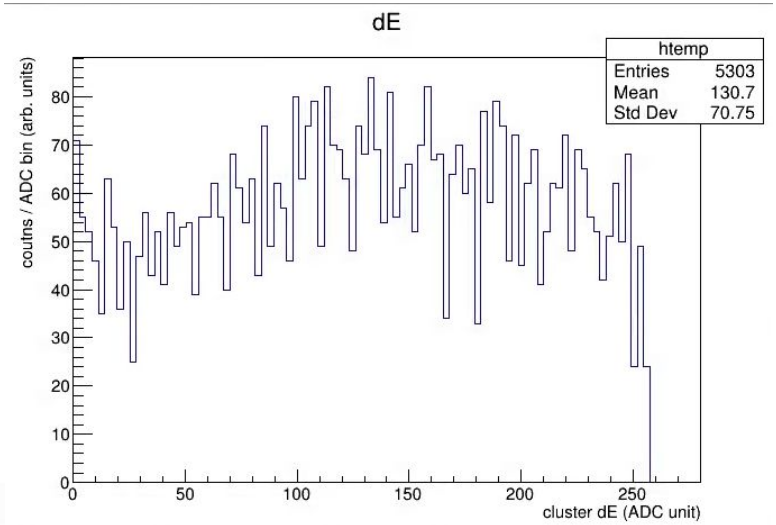
Progress Update

- **dE should come from cluster ADC**
  - [Using the functionality in add_cluster_size in TrackEvaluation.cc]

100 events , protons only , η = 0.3, variable φ, p = 100 GeV/c

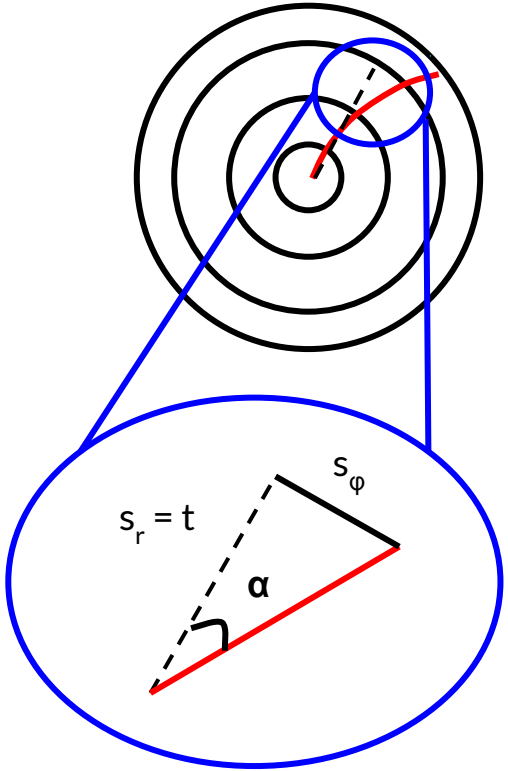


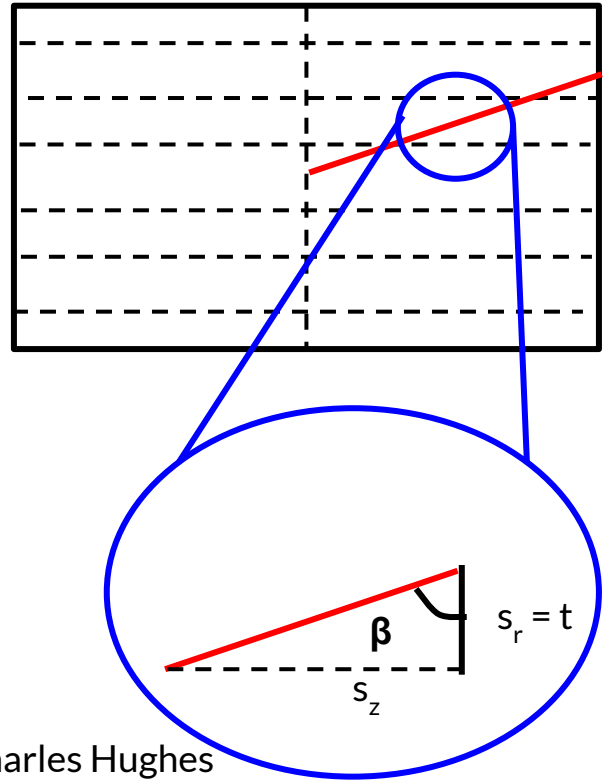




Progress Update

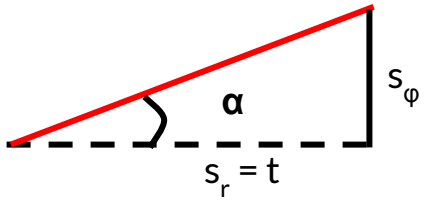
- dx should come from layer thickness and track state angles (α & β)





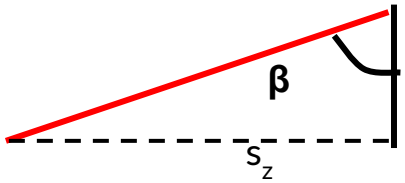
Progress Update

- dx should come from layer thickness and track state angles (α & β)



$\tan(\alpha) = s_\varphi / s_r$

$s_\varphi = s_r * \tan(\alpha)$



$\tan(\beta) = s_z / s_r$

$s_z = s_r * \tan(\alpha)$

$$dx = \sqrt{s_r^2 + s_\phi^2 + s_z^2}$$

$$dx = \sqrt{s_r^2 + (s_r \tan(\alpha))^2 + (s_r \tan(\beta))^2}$$

$$dx = s_r \sqrt{1 + \tan^2(\alpha) + \tan^2(\beta)}$$

$$dx = t \sqrt{1 + \tan^2(\alpha) + \tan^2(\beta)}$$

Code snippet from TrackEvaluation.cc - Line 745

std::atan2 (y,x) = atan(y/x) = opposite over adjacent

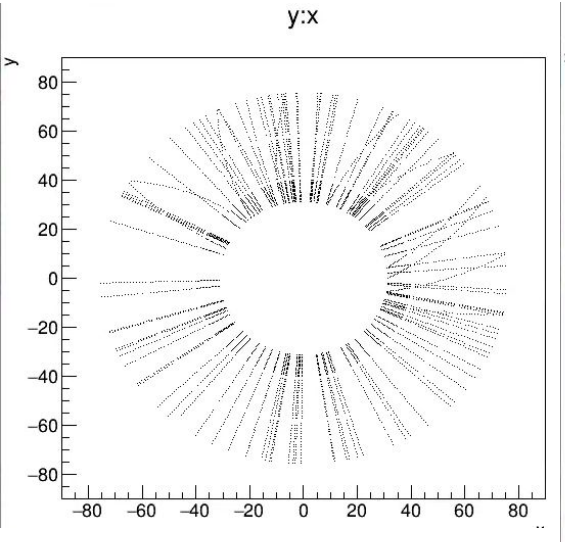
```
/*
store state angles in (r,phi) and (r,z) plans
they are needed to study space charge distortions
*/
const auto cosphi( std::cos( cluster.trk_phi ) );
const auto sinphi( std::sin( cluster.trk_phi ) );
const auto trk_pphi = -state->get_px()*sinphi + state->get_py()*cosphi;
const auto trk_pr = state->get_px()*cosphi + state->get_py()*sinphi;
const auto trk_pz = state->get_pz();
cluster.trk_alpha = std::atan2( trk_pphi, trk_pr );
cluster.trk_beta = std::atan2( trk_pz, trk_pr );
```

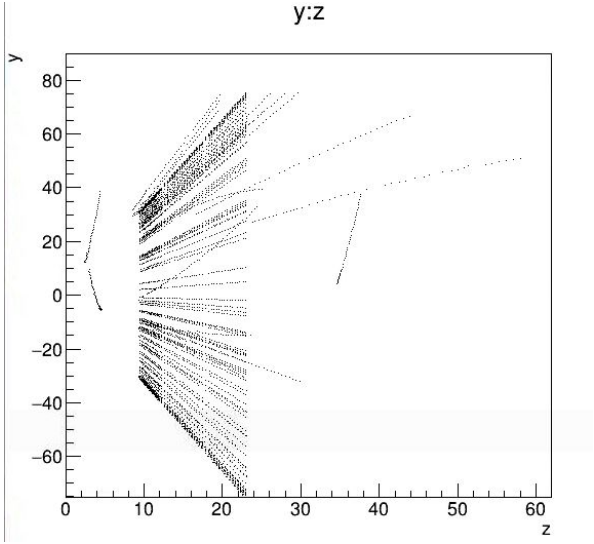
Progress Update

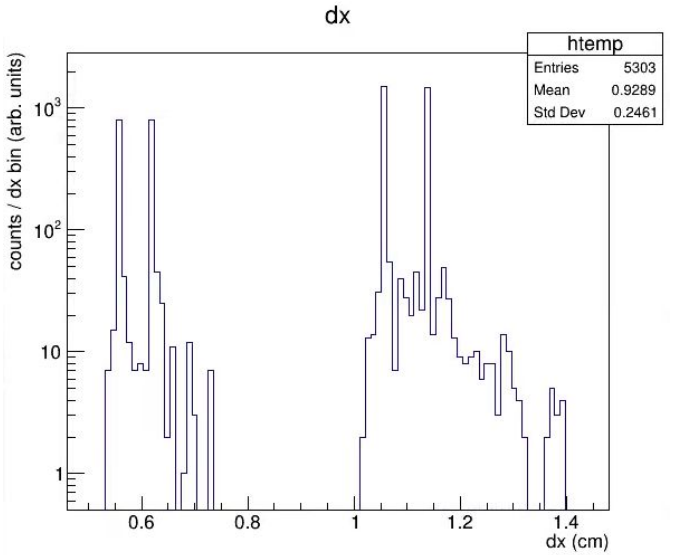
- **dx should come from layer thickness and track state angles (α & β)**

100 events , protons only , η = 0.3, variable φ, p = 100 GeV/c

$$dx = t\sqrt{1 + \tan^2(\alpha) + \tan^2(\beta)}$$



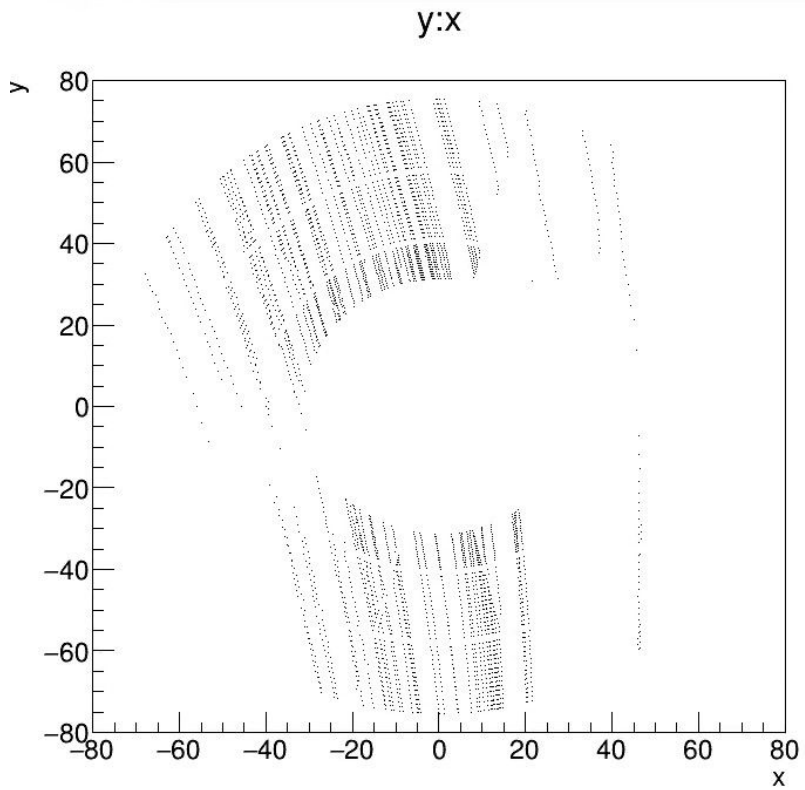




Progress Update

- **(from Jin Huang) Simulate cosmic ray muons (thanks to Michael Peters for helping !!!)**

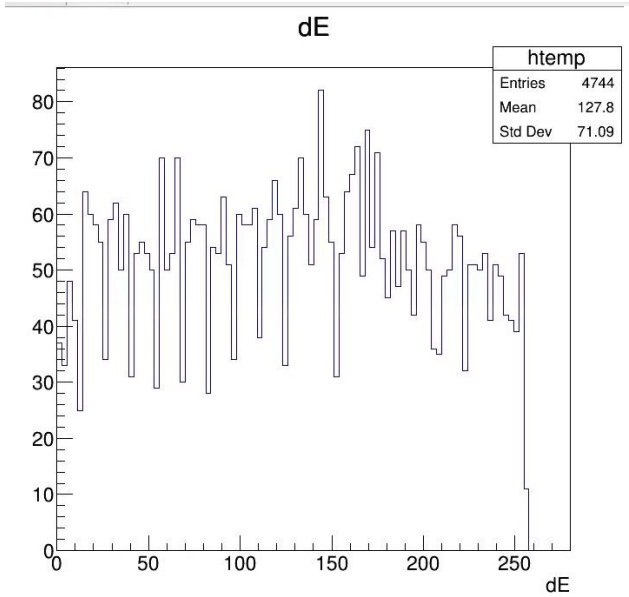
  - Vertex at (0,-400,0)
  - All muons with p = 3 GeV
  - η = 0
  - φ: {π/2 - π/20 , π/2 + π/20}

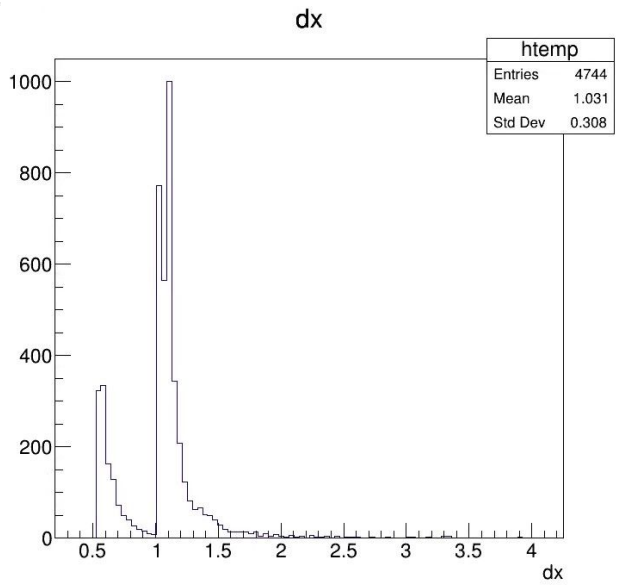


Progress Update

- **(from Jin Huang) Simulate cosmic ray muons (thanks to Michael Peters for helping !!!)**

  - Vertex at (0,-400,0)
  - All muons with p = 3 GeV
  - η = 0
  - φ: {π/2 - π/20 , π/2 + π/20}

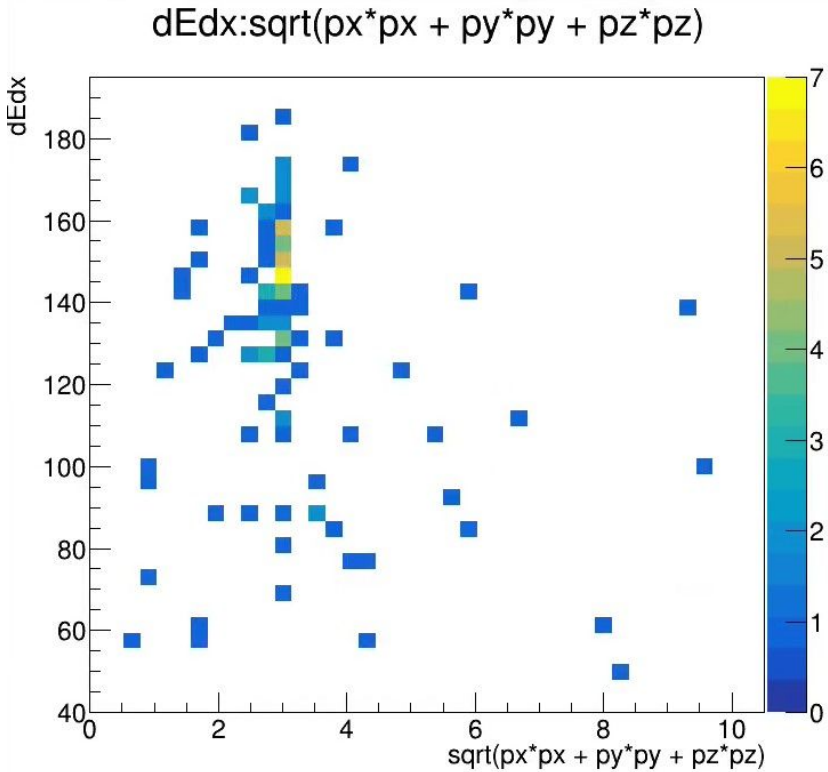




Progress Update

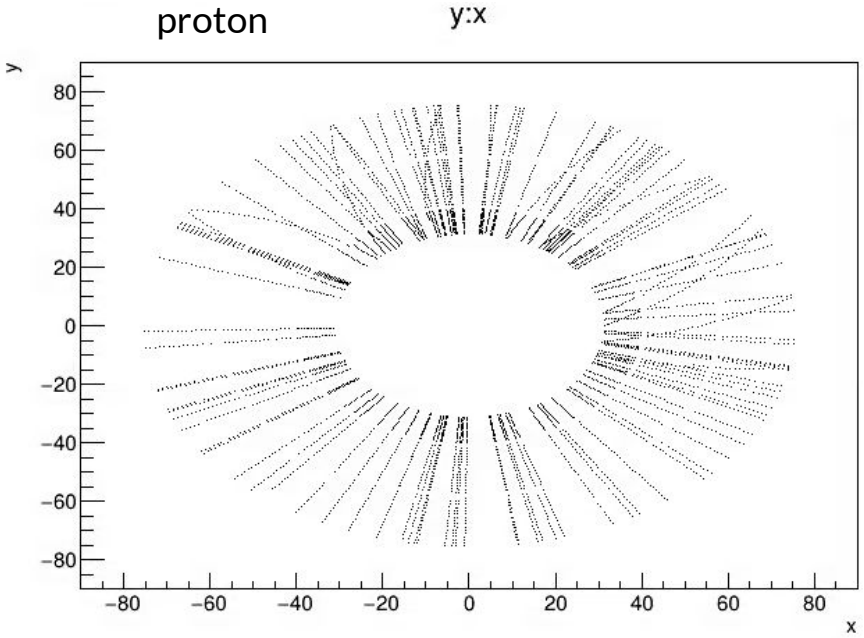
- **(from Jin Huang) Simulate cosmic ray muons (thanks to Michael Peters for helping !!!)**

  - Vertex at (0,-400,0)
  - All muons with p = 3 GeV
  - η = 0
  - φ: {π/2 - π/20 , π/2 + π/20}
  - dE/dx (20 % truncated mean)



Progress Update

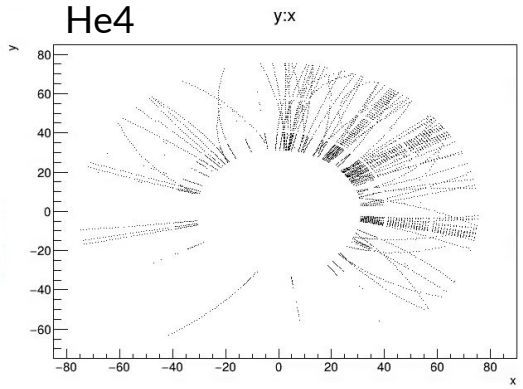
- **(from Jin Huang) Simulate light ions in TPC - looking for <ADC> increases as Z^2**
  - Vertex at (0,0,0)
  - All ions with p = 100 GeV
  - η = 0.3
  - φ: {- π, π}
  - Protons, He4, Li7, Be9, B11

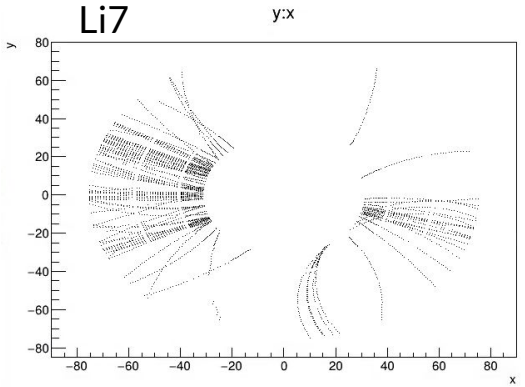


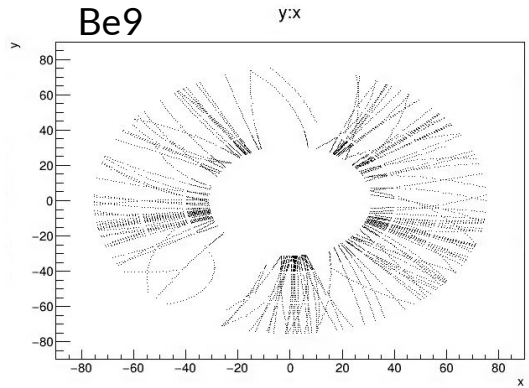
Progress Update

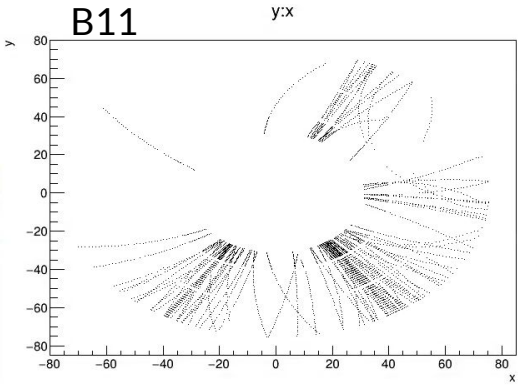
- **(from Jin Huang) Simulate light ions in TPC - looking for <ADC> increases as Z^2**

  - Vertex at (0,0,0)
  - All ions with p = 100 GeV
  - η = 0.3
  - φ: {- π, π}
  - Protons, He4, Li7, Be9, B11







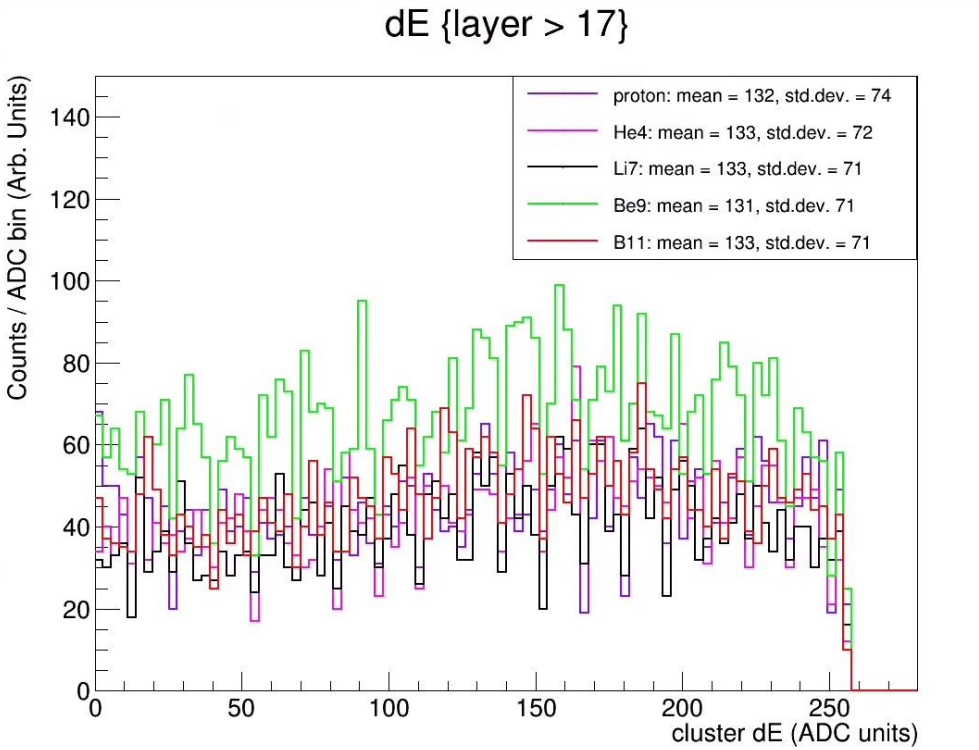


Progress Update

- **(from Jin Huang) Simulate light ions in TPC - looking for <ADC> increases as Z^2**

  - Vertex at (0,0,0)
  - All ions with p = 100 GeV
  - η = 0.3
  - φ: {- π, π}
  - Protons, He4, Li7, Be9, B11

NO SCALING AT ALL - WHAT'S GOING ON?



Progress Update

- NO SCALING AT ALL - WHAT'S GOING ON?

- Some things I observed during the simulations:

1) To simulate anything > Z = 1, you have to pass this code to to the simple generator or the particle gun: 100ZZZAAAI - where ZZZ = charge and AAA = mass number and I = excitation. This means for e.g. Lithium 7 one would pass 1000030070

2) Simple generator can only correctly instantiate up to Z = 2 (alpha/helium3). For anything bigger than Z = 2, one needs to use the particle gun. If one does not use the particle gun and uses the simple event generator then one gets "geantinos" that are only transported and don't interact with the detector. I don't fully understand this but this webpage explains it a bit

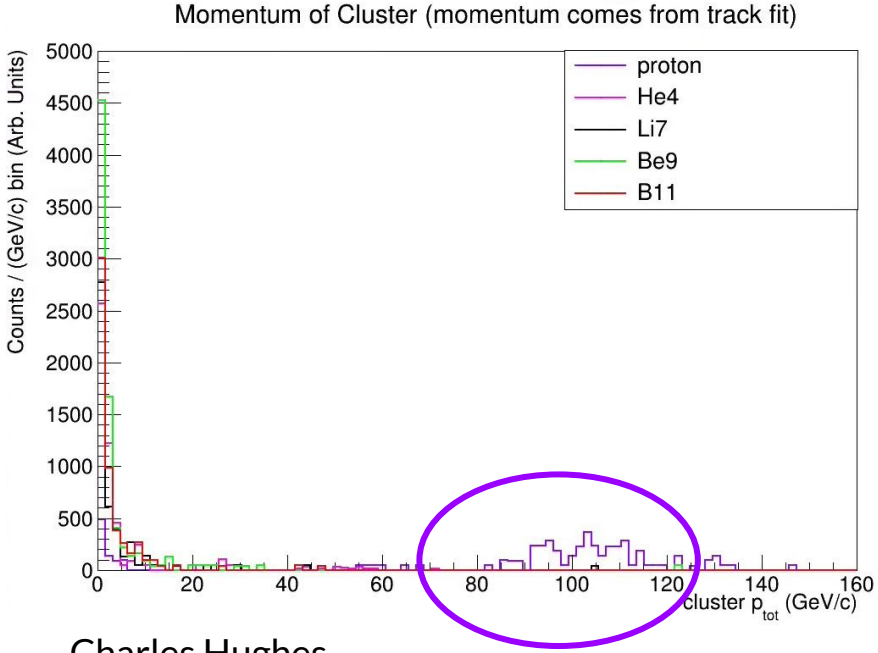
3) Something VERY strange is going on with momenta for Z >= 2. The simulation knows I am requesting particles with p = 100 GeV, but the particles that get inserted into the detector have very low and variable momentum. Not sure why this seems to be only true for Z >= 2 (I have the pipe and pipe absorber turned off)

Progress Update

- NO SCALING AT ALL - WHAT'S GOING ON?

3) Something VERY strange is going on with momenta for Z >= 2. The simulation knows I am requesting particles with p = 100 GeV, but the particles that get inserted into the detector have very low and variable momentum. Not sure why this seems to be only true for Z >= 2 (I have the pipe and pipe absorber turned off)

- Vertex at (0,0,0)
- All ions with p = 100 GeV
- η = 0.3
- φ: {- π, π}
- Protons, He4, Li7, Be9, B11
- **Only protons have p ~ 100 GeV**



Progress Update

- **NO SCALING AT ALL - WHAT'S GOING ON?**

4) Noticed that ADC counts always peak at 256 - this is because of [getADC() in trackbase/TrkrClusterv5.h]

```
66    // cluster info
67    //
68    unsigned int getAdc() const override {
69      uint8_t tmp = m_adc;
70      return tmp ;
71
72    }
```

In principle, can have single channel ADC (hits) much higher than this

Not sure if this needs to change for analyses that need cluster ADC

Summary

Agenda for Next Time:

- Need to understand what needs to be fixed in simulation/reconstruction
  - Simulation: What is going on with momentum?
  - Reconstruction: What is going on with ADC?
  - Reconstruction: Have I calculated dx correctly?

Agenda Further Ahead:

- Agree on how to get most probable dE/dx (truncated mean, template fit, … ?)
- Create utility for end user to get dE/dx for reconstructed track
- More realistic gains/looking toward data reconstruction