# ACTS Integration for B0 Trackers

Sakib Rahman
University of Manitoba
29 June 2023

30 src/algorithms/tracking/TrackerSourceLinker.cc

Viewed

```
@@ -80,21 +80,43 @@ eicrecon::TrackerSourceLinkerResult *eicrecon::TrackerSourceLinker::produce(std:
```

```
        Acts::Vector2 loc = Acts::Vector2::Zero();
        Acts::Vector2 pos;
```

```
+       auto hit_det = hit->getCellID()&0xFF;
+       auto onSurfaceTolerance = 0.0001;       // By default, ACTS uses 0.1 micron as the on surface tolerance
+       if (hit_det==150){
+        onSurfaceTolerance = 0.001;           // Ugly hack for testing B0. Should be a way to increase this tolerance in geometry.
+       }
+
```

```
        try {
            // transform global position into local coordinates
            // geometry context contains nothing here
            pos = surface->globalToLocal(
                    Acts::GeometryContext(),
                    {hit_pos.x, hit_pos.y, hit_pos.z},
-                   {0, 0, 0}).value();

+                   {0, 0, 0}, onSurfaceTolerance).value();

            loc[Acts::eBoundLoc0] = pos[0];
            loc[Acts::eBoundLoc1] = pos[1];
        }
        catch(std::exception &ex) {
-
-           m_log->warn("Can't convert globalToLocal for hit:
+           auto inverse = ((surface->transform(Acts::GeometryContext())).inverse()) ;
+           m_log->warn("Can't convert globalToLocal for hit:
```

Recap on [PR#405 on EICrecon]

1) B0Tracker detector factory was added

2) EICrecon performs a round trip check (global->local and local->global) on hits passed to ACTS. It was failing due to low on surface tolerance limit. Hacky hard-coded solution to get around it was to set the tolerance using detector id. Need a better way to do this.

10 src/global/tracking/tracking.cc

Viewed

```
@@ -33,14 +33,16 @@ void InitPlugin(JApplication *app) {
33
34      // Tracker hits collector
35      app->Add(new JChainFactoryGeneratorT<TrackerHitCollector_factory>(
36  -                   {
37  -                       "SiBarrelTrackerRecHits",       // Si tracker hits
38                          "SiBarrelVertexRecHits",
39                          "SiEndcapTrackerRecHits",
40                          "TOFBarrelRecHit",              // TOF hits
41                          "TOFEndcapRecHits",
42  -                       "MPGDBarrelRecHits",            // MPGD
43  -                       "MPGDDIRCRecHits"},
44
45                      "CentralTrackingRecHits"));     // Output collection name
46
```

```
@@ -33,14 +33,16 @@ void InitPlugin(JApplication *app) {
33
34      // Tracker hits collector
35      app->Add(new JChainFactoryGeneratorT<TrackerHitCollector_factory>(
36  +                   {
37  +                       "SiBarrelTrackerRecHits",       // Si tracker hits
38                          "SiBarrelVertexRecHits",
39                          "SiEndcapTrackerRecHits",
40                          "TOFBarrelRecHit",              // TOF hits
41                          "TOFEndcapRecHits",
42  +                       "MPGDBarrelRecHits",            // MPGD
43  +                       "MPGDDIRCRecHits",
44  +                       "B0TrackerRecHits"              // B0TRK
45  +                   },
46
47                      "CentralTrackingRecHits"));     // Output collection name
48
```

2) B0TrackerRecHits needed to be passed as input in tracking.cc

2 src/utilities/eicrecon/eicrecon.cc

Viewed

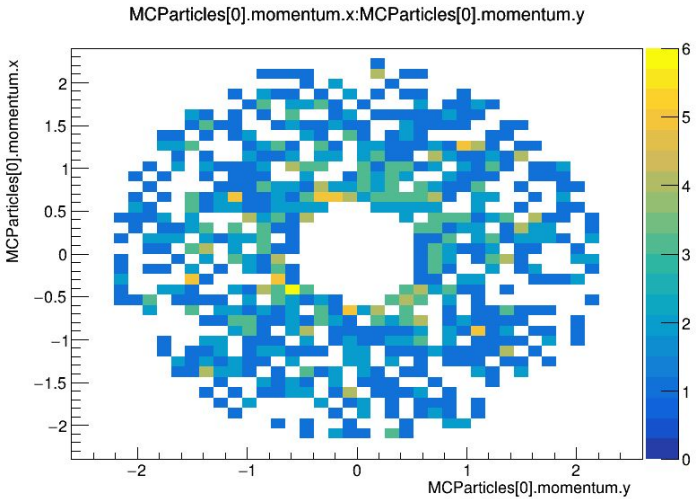
```
@@ -30,13 +30,13 @@ std::vector<std::string> EICRECON_DEFAULT_PLUGINS = {
30          "BHCAL",
31          "FHCAL",
32          "B0ECAL",
33  -       "B0TRK",
34          "ZDC",
```

```
@@ -30,13 +30,13 @@ std::vector<std::string> EICRECON_DEFAULT_PLUGINS = {
30          "BHCAL",
31          "FHCAL",
32          "B0ECAL",
33          "ZDC",
```

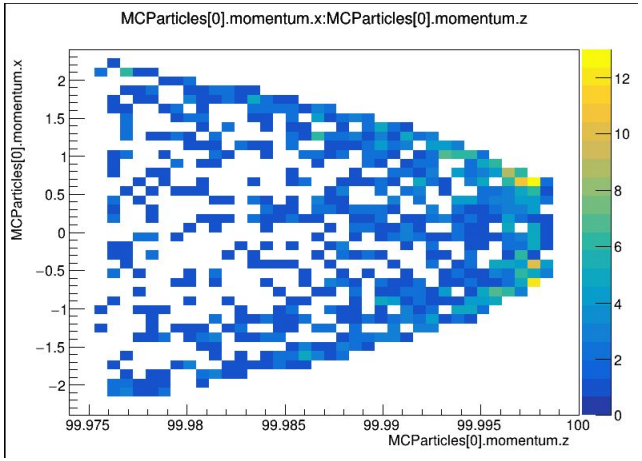
3) Make sure it's the list of default plugins or you can activate it during running

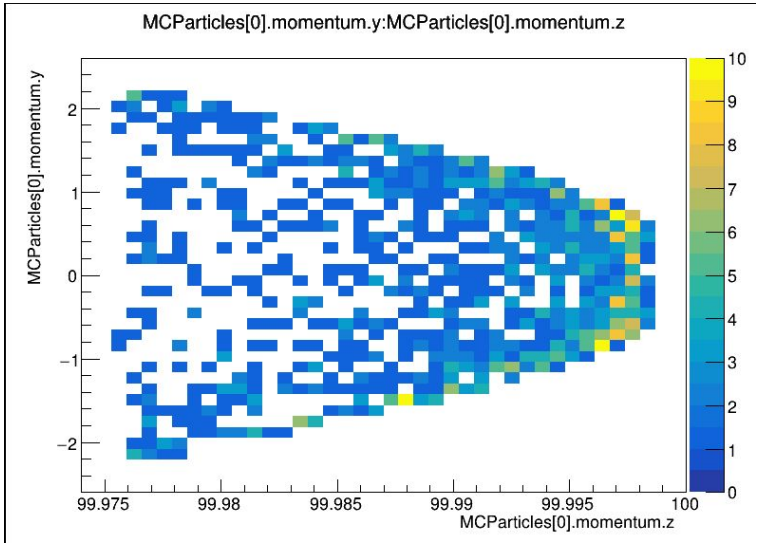
# Study with 100 GeV protons

```
from DDSim.DD4hepSimulation import DD4hepSimulation
from g4units import mm, GeV, MeV, mrad
SIM = DD4hepSimulation()

SIM.enableGun = True
SIM.gun.thetaMin = 6*mrad
SIM.gun.thetaMax = 22*mrad
SIM.gun.momentumMin = 100*GeV
SIM.gun.momentumMax = 100*GeV
SIM.gun.distribution = 'uniform'
SIM.gun.particle = 'proton'
SIM.outputFile = 'result.edm4hep.root'
```

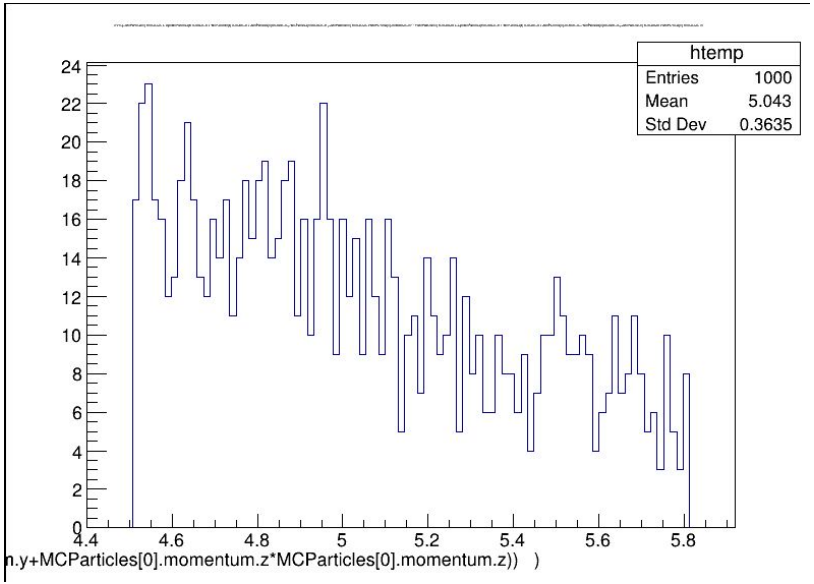



MCParticle[0] correspond to the generated proton

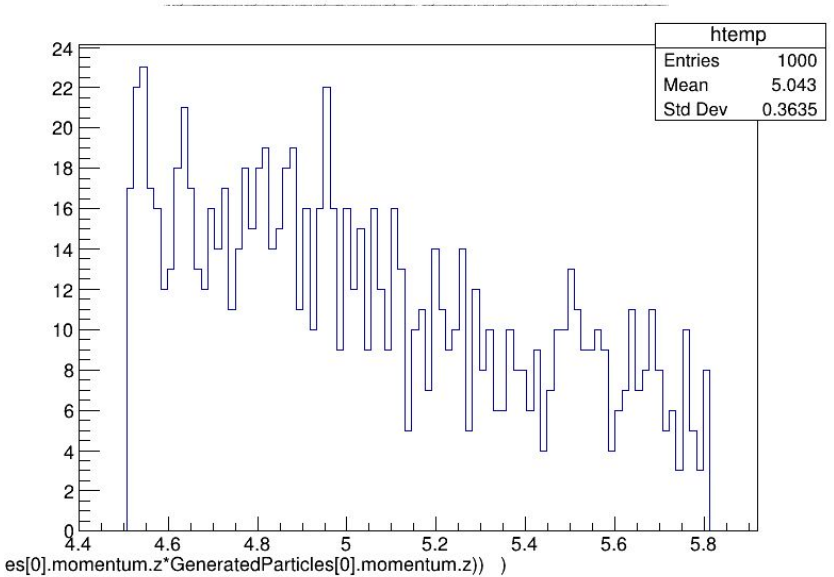


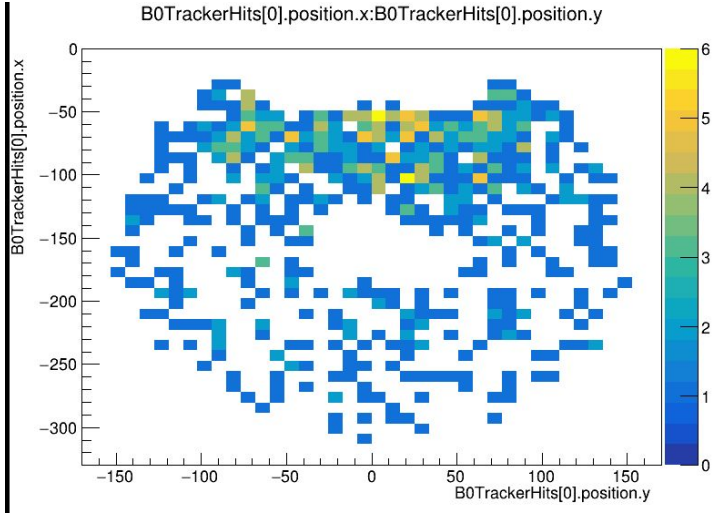
MCParticles Eta Distribution (Full sim root file)



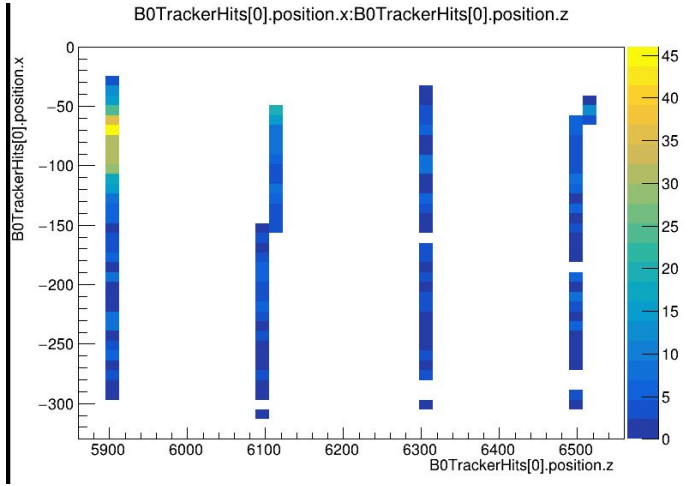

Generated Particles Eta Distribution (Reco root file)

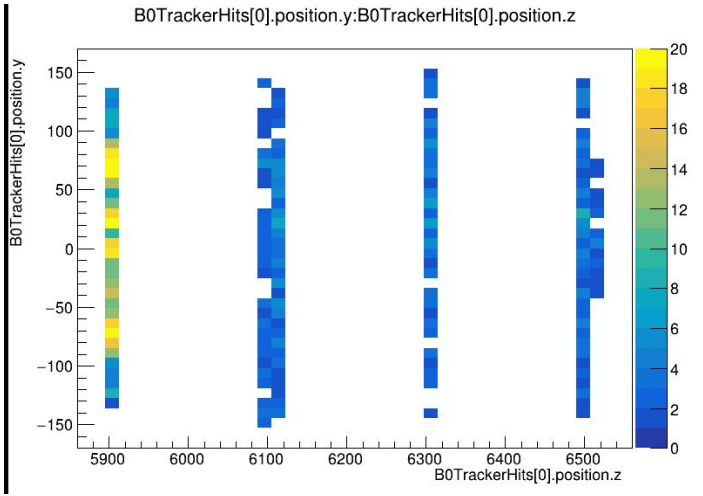


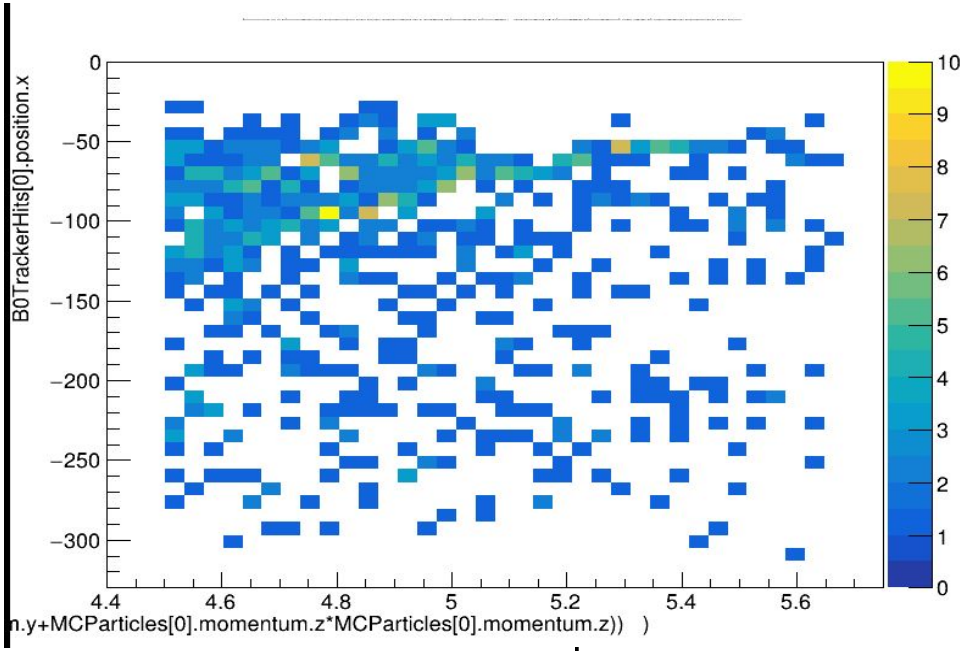
# Hits on B0

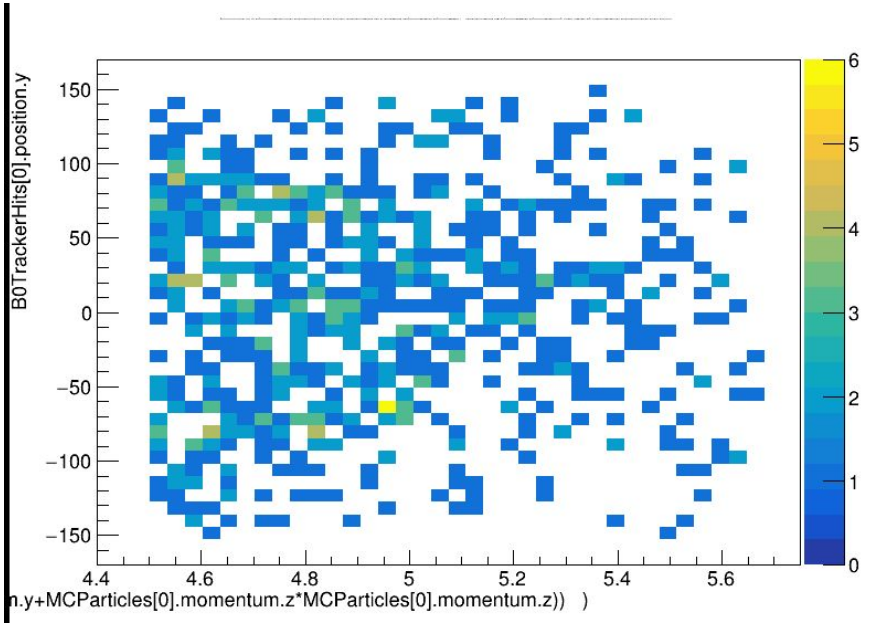



B0TrackerHits[0] is the first hit on any B0 plane for a event (not necessarily the generated proton)

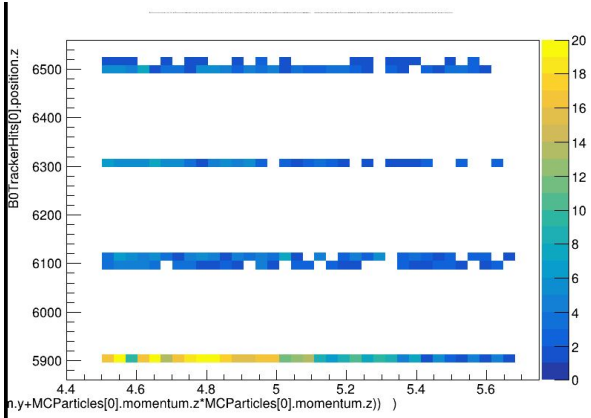


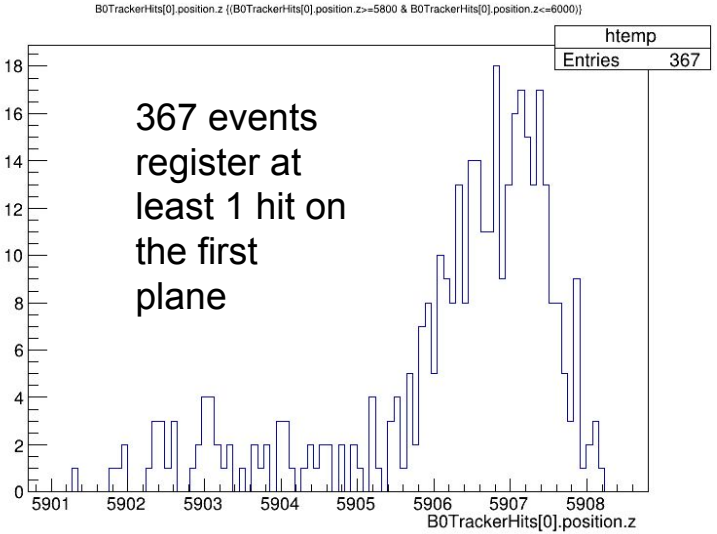





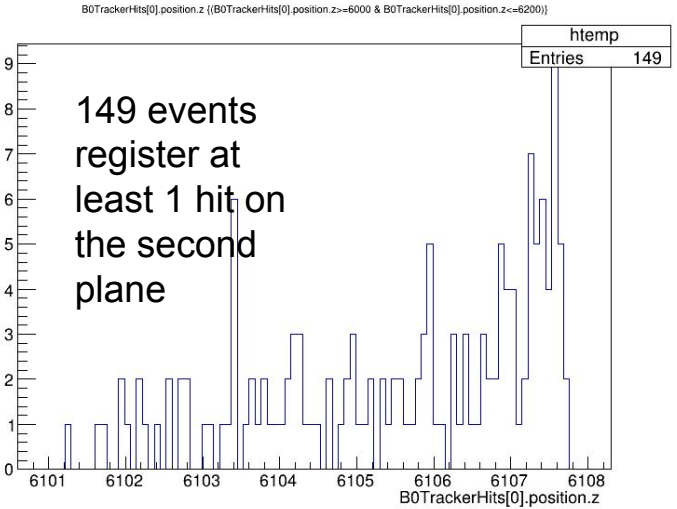




B0HitPosition vs
Eta distribution

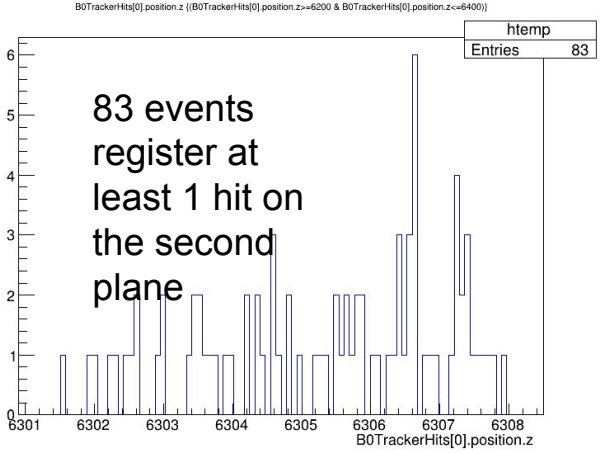




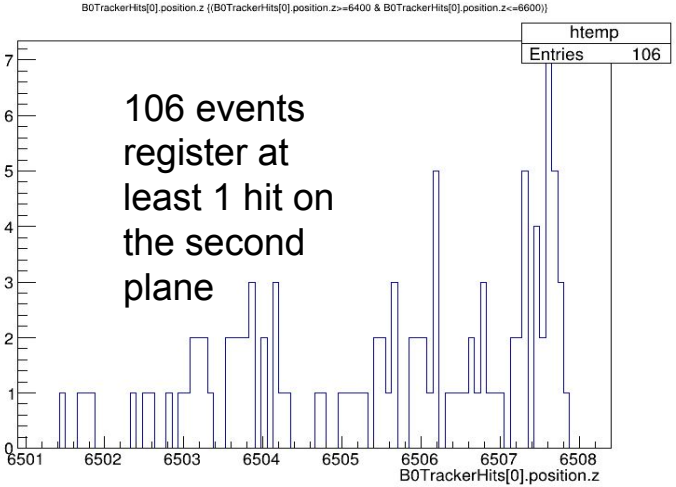

367 events register at least 1 hit on the first plane




149 events register at least 1 hit on the second plane

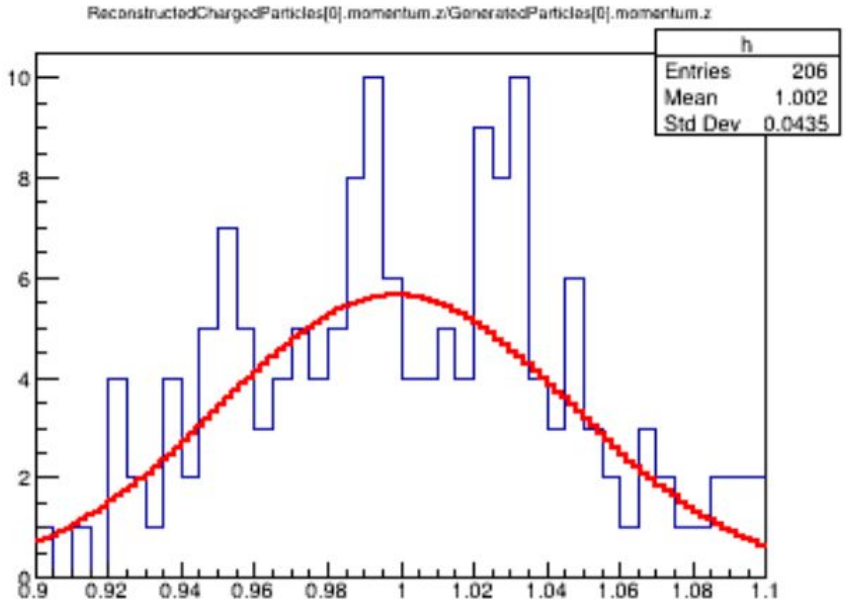



83 events register at least 1 hit on the second plane

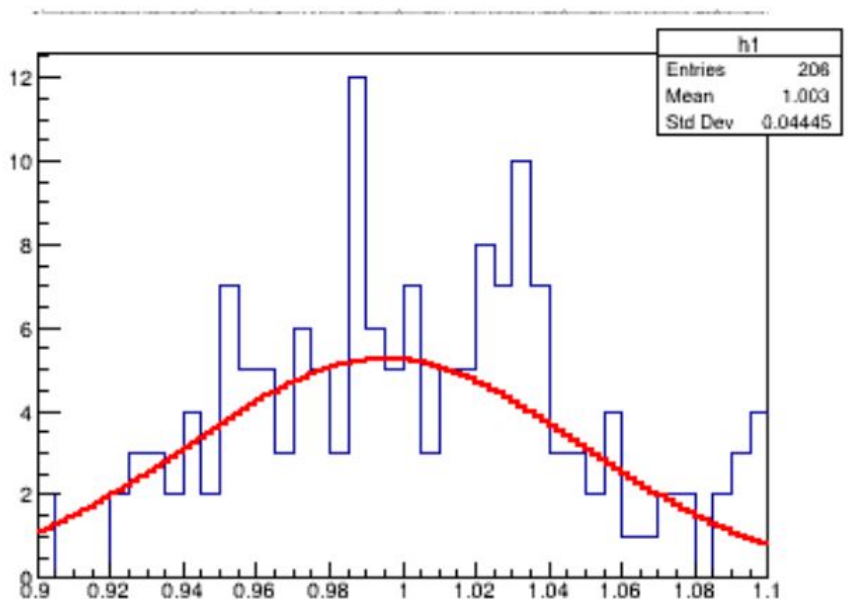



106 events register at least 1 hit on the second plane

Longitudinal

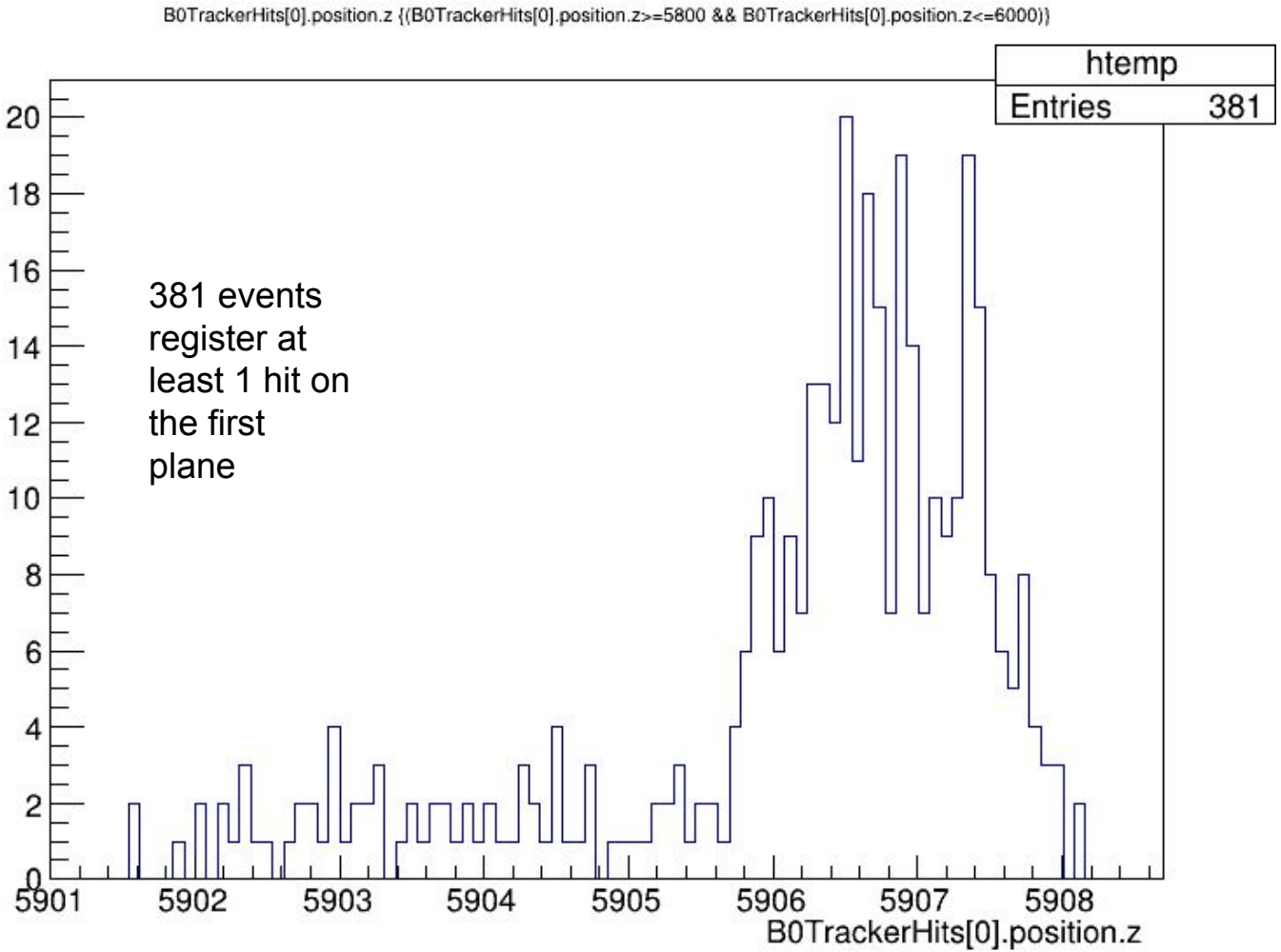



Transverse




206 events out of 1000 get reconstructed momentum

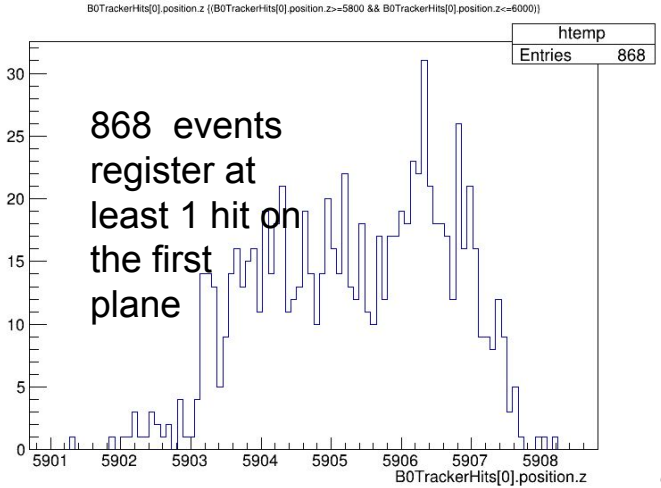
Study with 100 GeV protons
(Main magnetic field turned off)

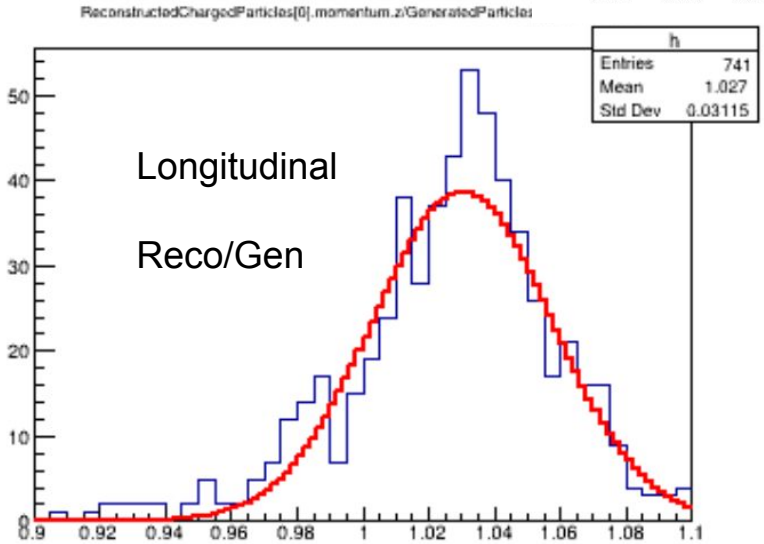


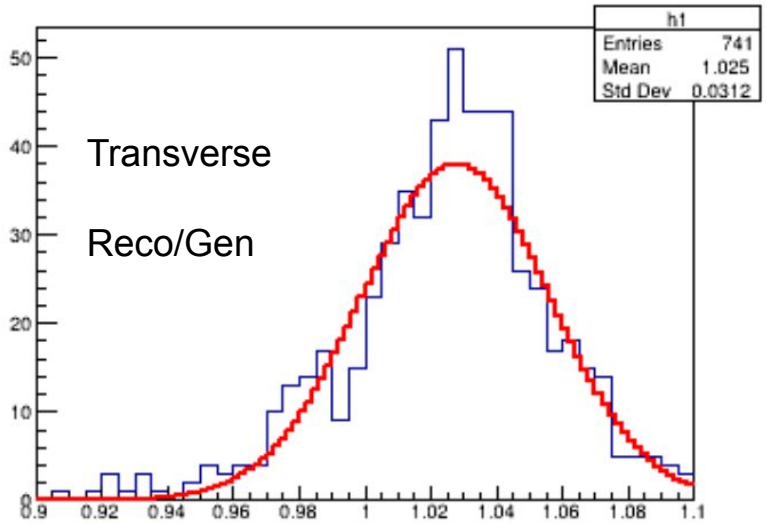

No better than the case with the main field on

Study with 100 GeV protons
(central magnetic field turned off+apply 25 mrad crossing angle boost)

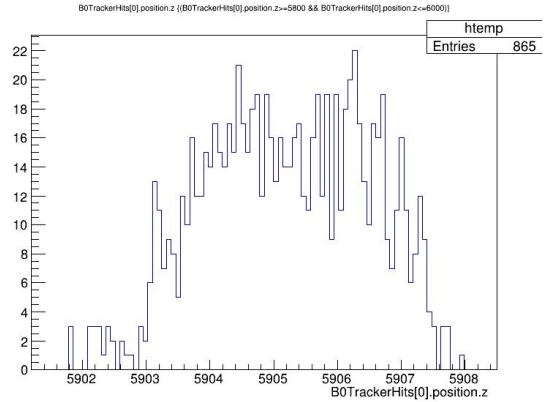
```
from DDSim.DD4hepSimulation import
DD4hepSimulation
from g4units import mm, GeV, MeV, mrad
SIM = DD4hepSimulation()

SIM.enableGun = True
SIM.gun.thetaMin = 6*mrad
SIM.gun.thetaMax = 22*mrad
SIM.gun.momentumMin = 100*GeV
SIM.gun.momentumMax = 100*GeV
SIM.gun.distribution = 'uniform'
SIM.gun.particle = 'proton'
SIM.outputFile = 'result.edm4hep.root'
SIM.crossingAngleBoost = -25*mrad
```

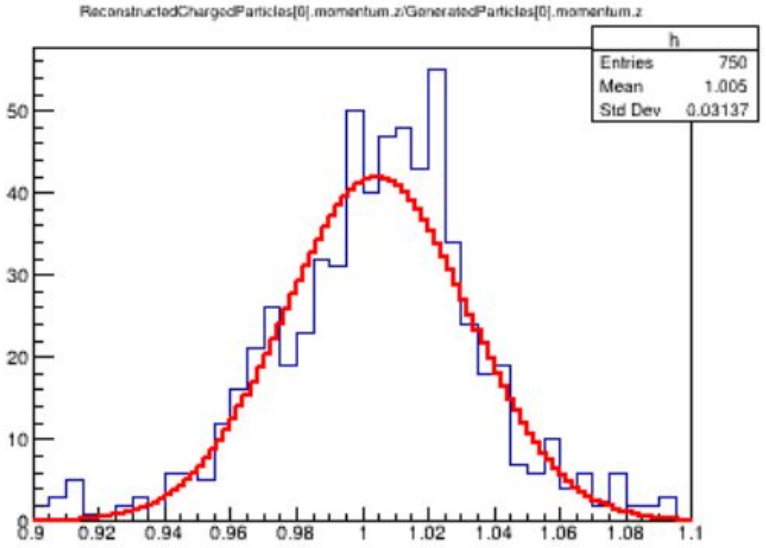


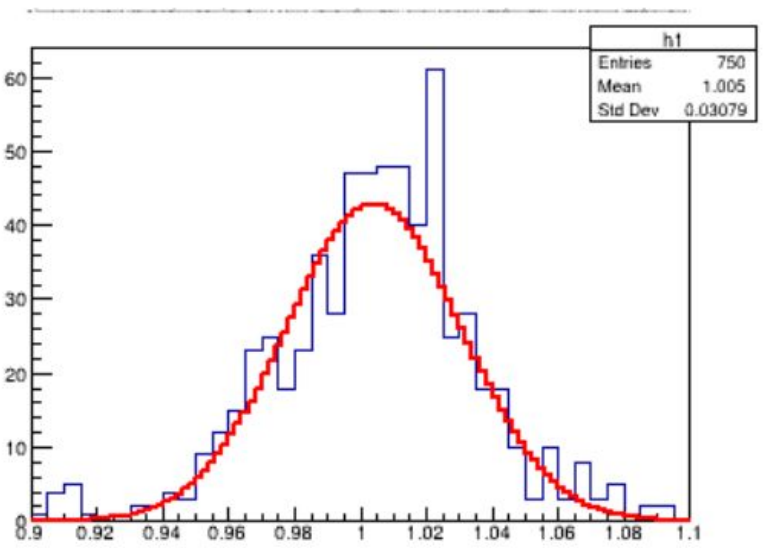



Study with 100 GeV protons
(central tracker and magnetic field on+apply 25 mrad crossing angle boost)


B0TrackerHits[0].position.z {(B0TrackerHits[0].position.z>=5800 && B0TrackerHits[0].position.z<=6000)}
htemp
Entries 865
B0TrackerHits[0].position.z


ReconstructedChargedParticles[0].momentum.z/GeneratedParticles[0].momentum.z
h
Entries 750
Mean 1.005
Std Dev 0.03137


h1
Entries 750
Mean 1.005
Std Dev 0.03079

# To Do

1) Understand the skew in the reconstructed/generated momentum distribution plots. Skews left when central magnetic field on and right when central magnetic field off. Could be a DD4hep related issue (https://github.com/AIDASoft/DD4hep/pull/1080) but needs more investigation.
2) Test with full geometry and crossing angle boost applied.
3) The axis range in reco/gen momentum plots is restricted. Check how badly the outliers fail.
4) Test with realistic seeder
5) Check B0Tracker has reasonable default thresholds in eicrecon.
6) Understand the effect of track quality cuts
7) Understand what changed that's making acts navigation to B0 work now compared to a few months ago.