Real-seeded tracking – comparison between ACTS v21.1 and v30

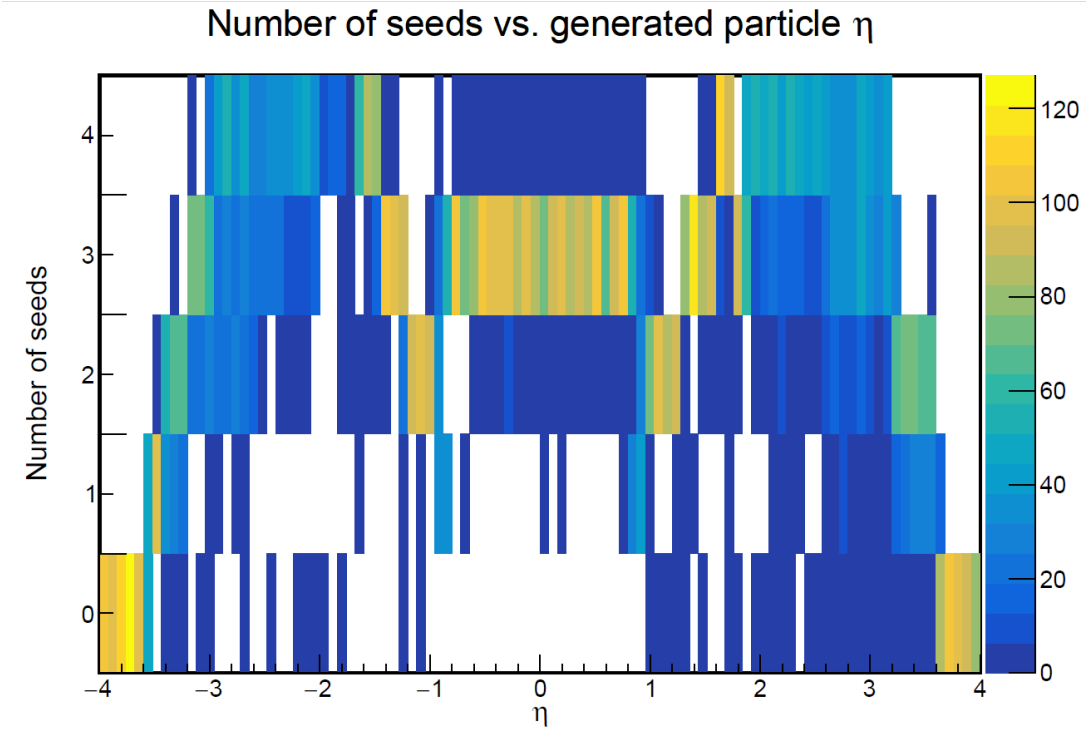
Barak Schmookler

Outline

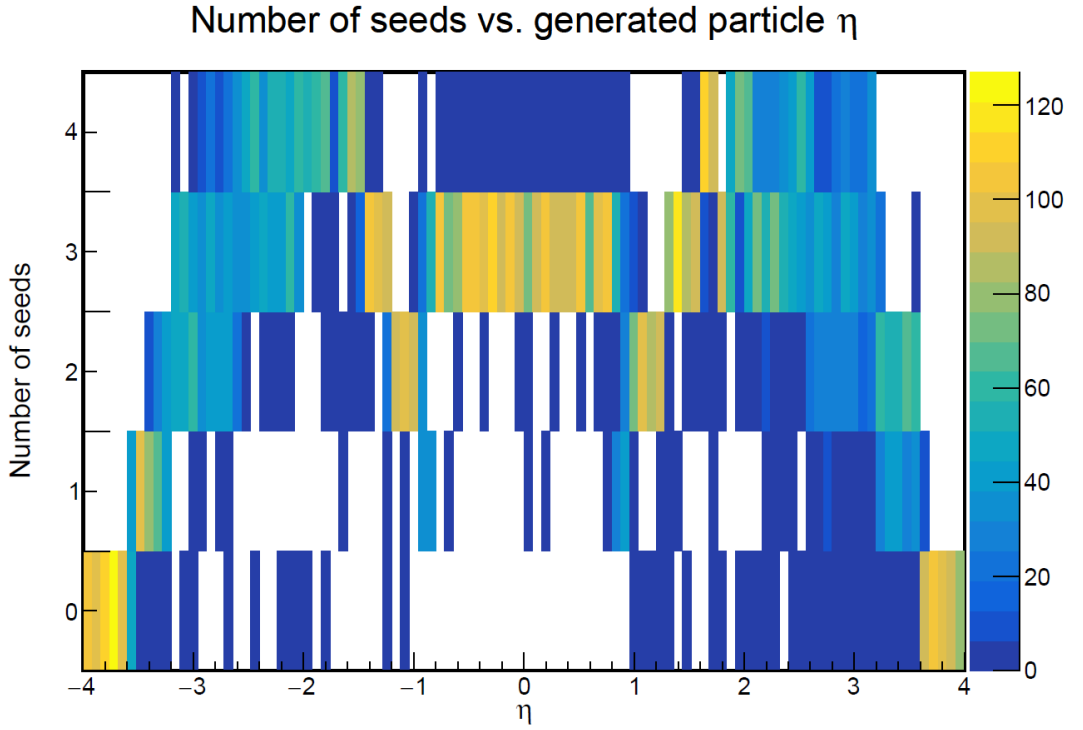
- EICRecon was recently updated to use ACTS v30. Previously, we were using ACTS v21.1. We want to study what affect, if any, this update has on the real-seeded tracking.
- Simulation summary:
  - Single negative muon generated
  - Uniform eta distribution from [-4,4]
  - Uniform momentum distribution from [0.5,20] GeV/c
  - Generation vertex: (0,0,0) mm
  - Same Geant4 (npsim) simulation file was used for both ACTS v30 and ACTS v21.1

Seed multiplicity

ACTS v21.1

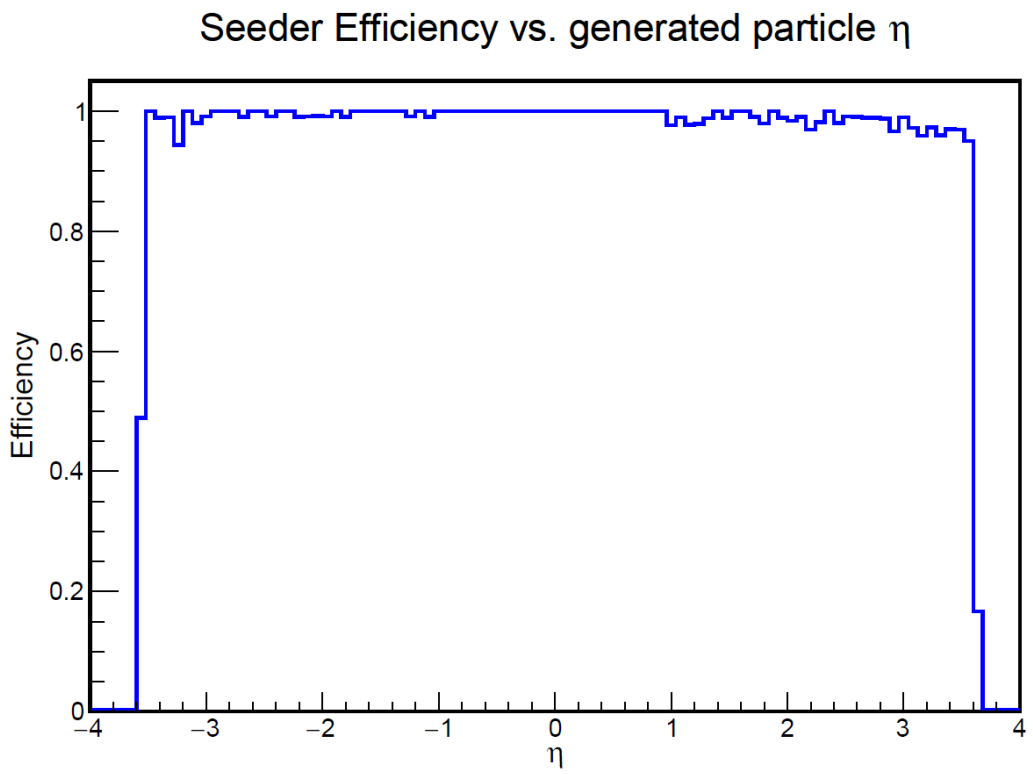


ACTS v30

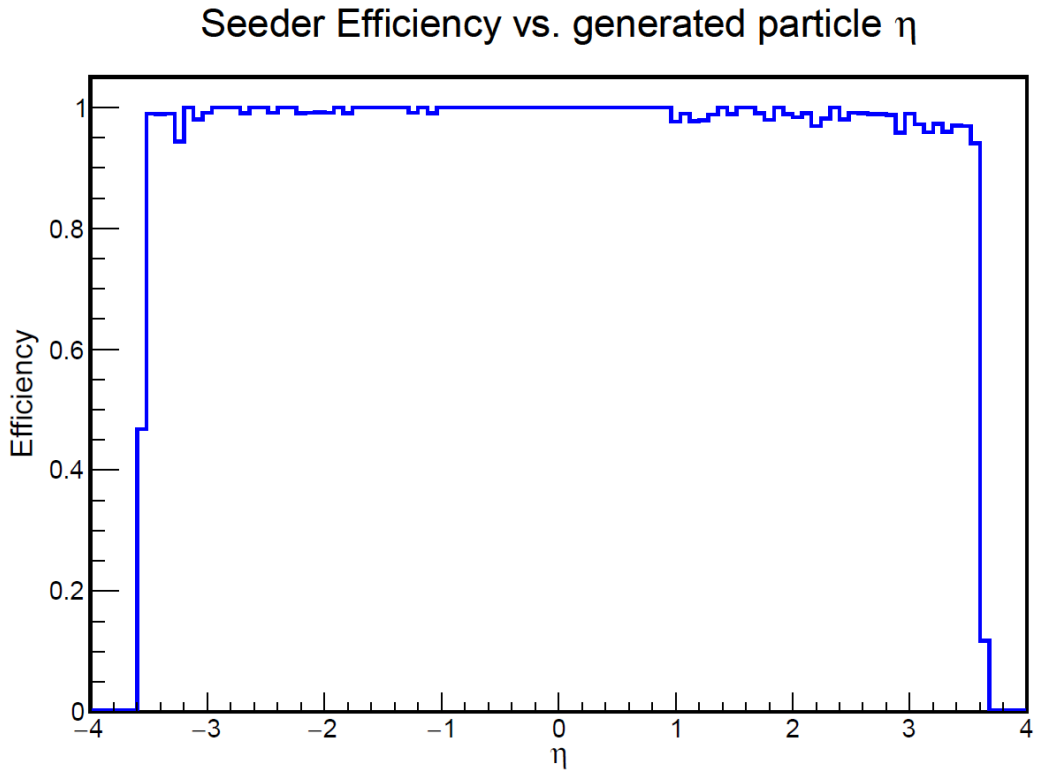


Seed efficiency

ACTS v21.1

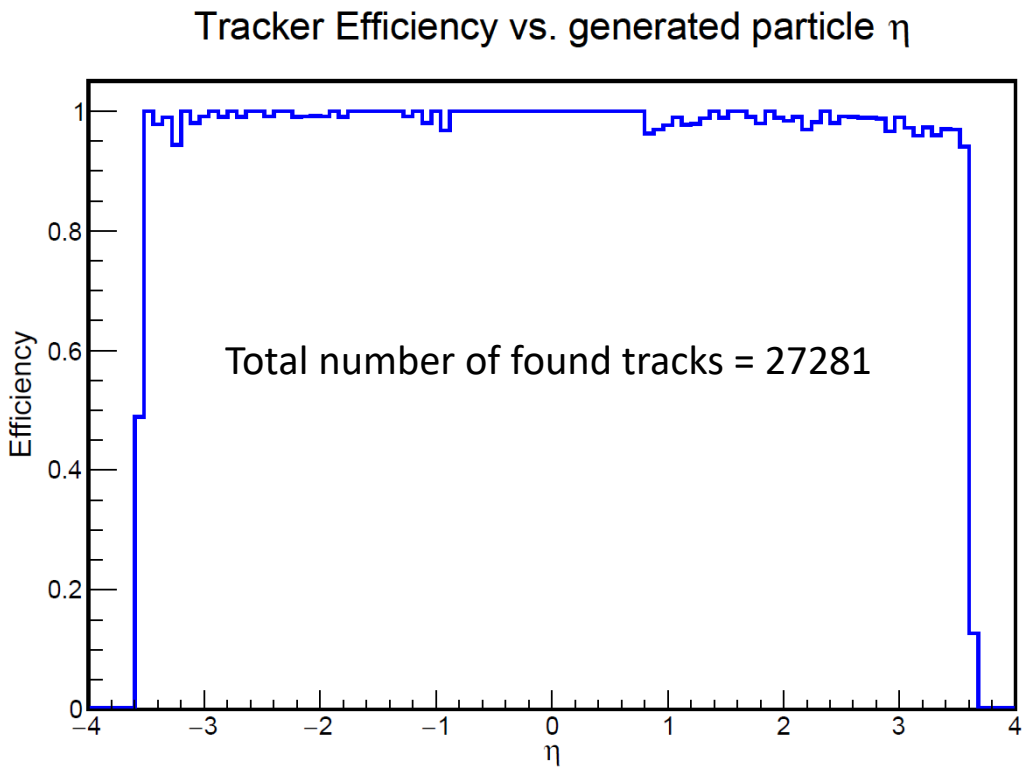


ACTS v30

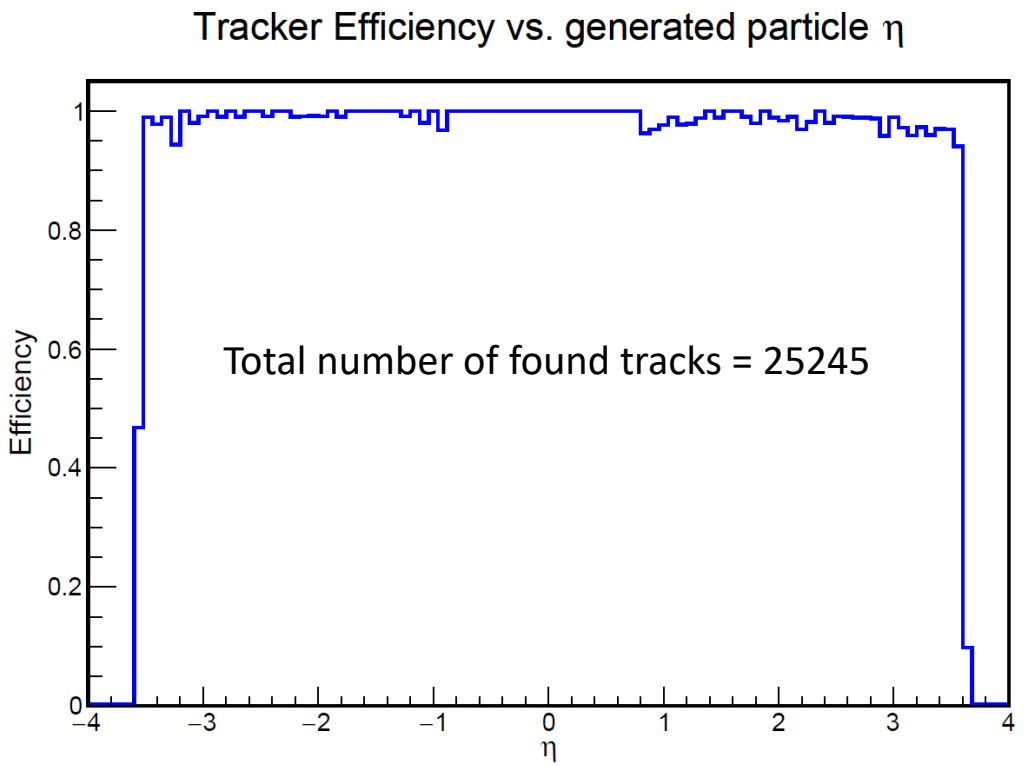


Track efficiency

ACTS v21.1



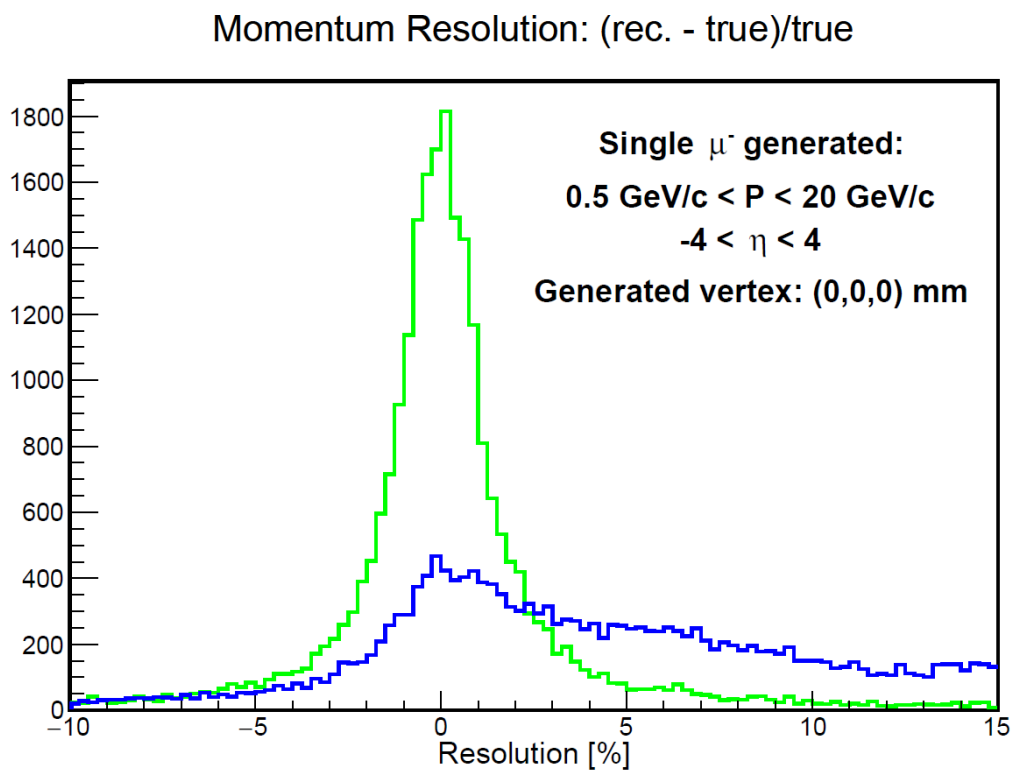
ACTS v30



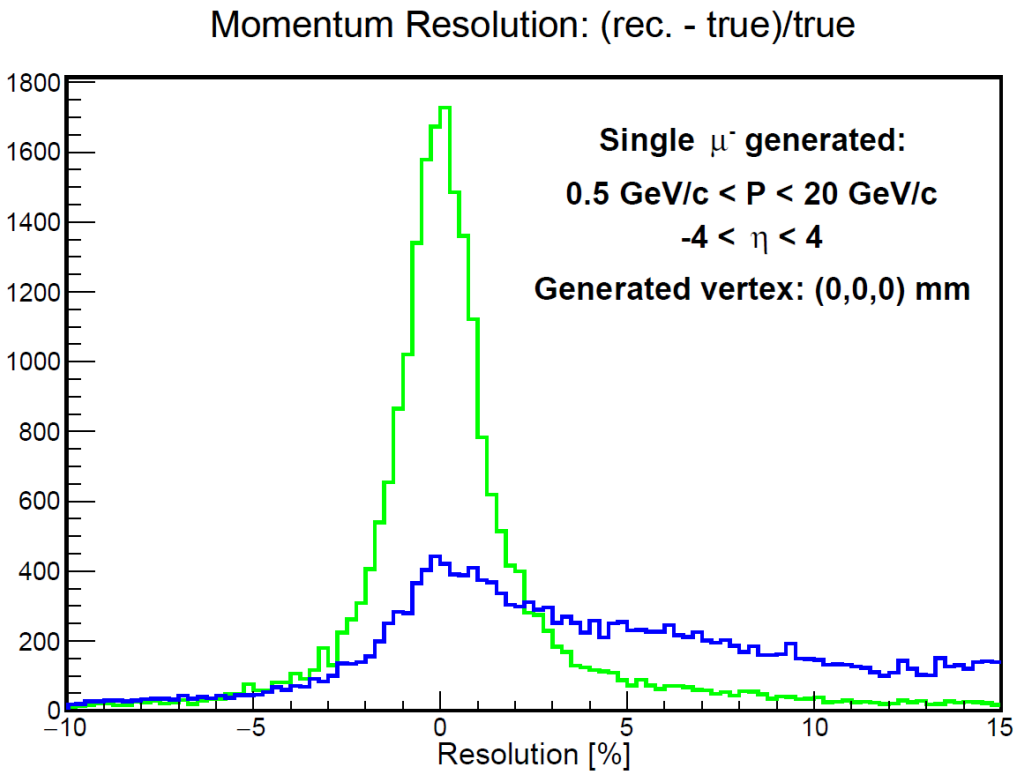
Seed level
Track level

Momentum resolution

ACTS v21.1



ACTS v30

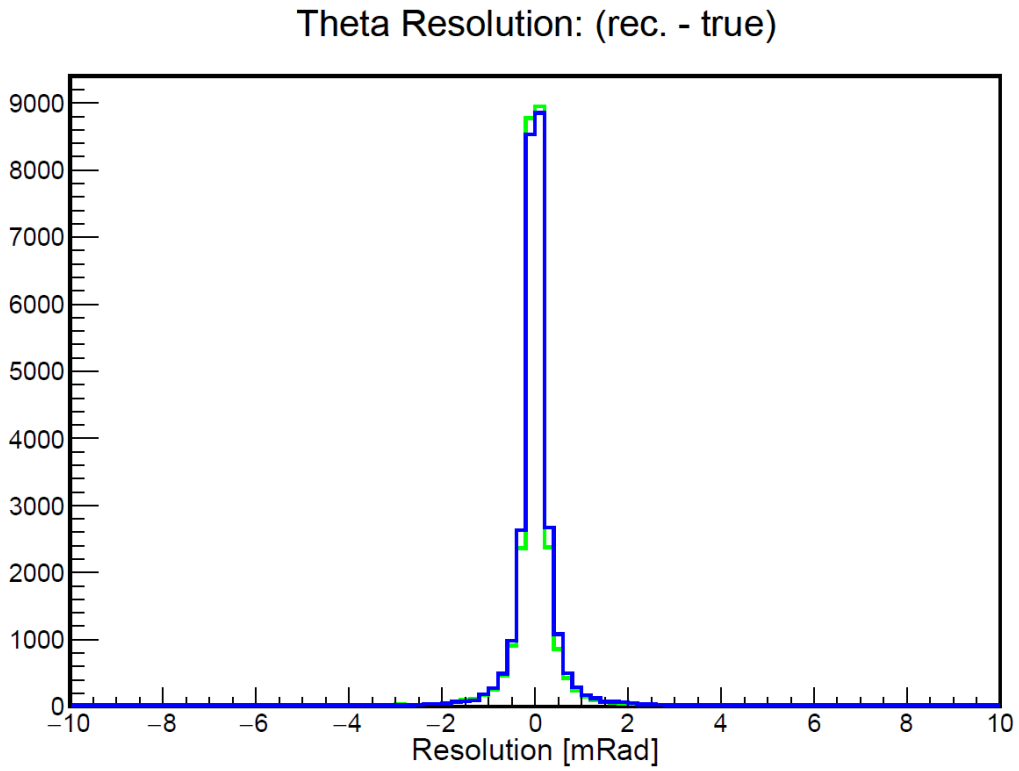


Seed level

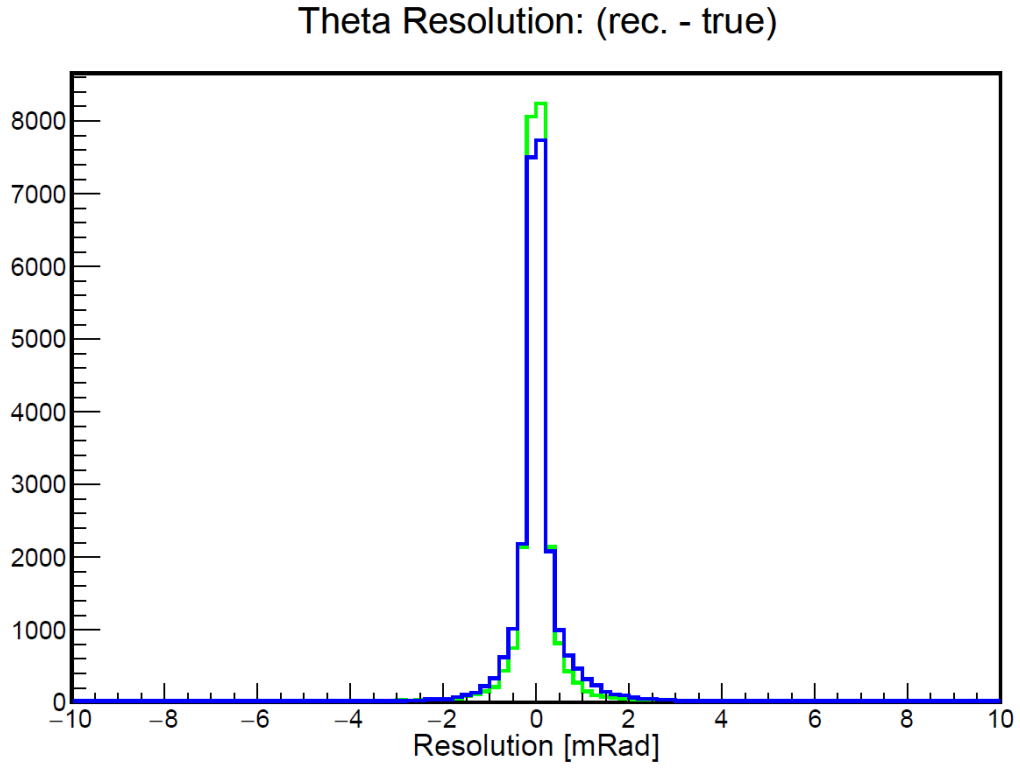
Track level

Theta resolution

ACTS v21.1



ACTS v30

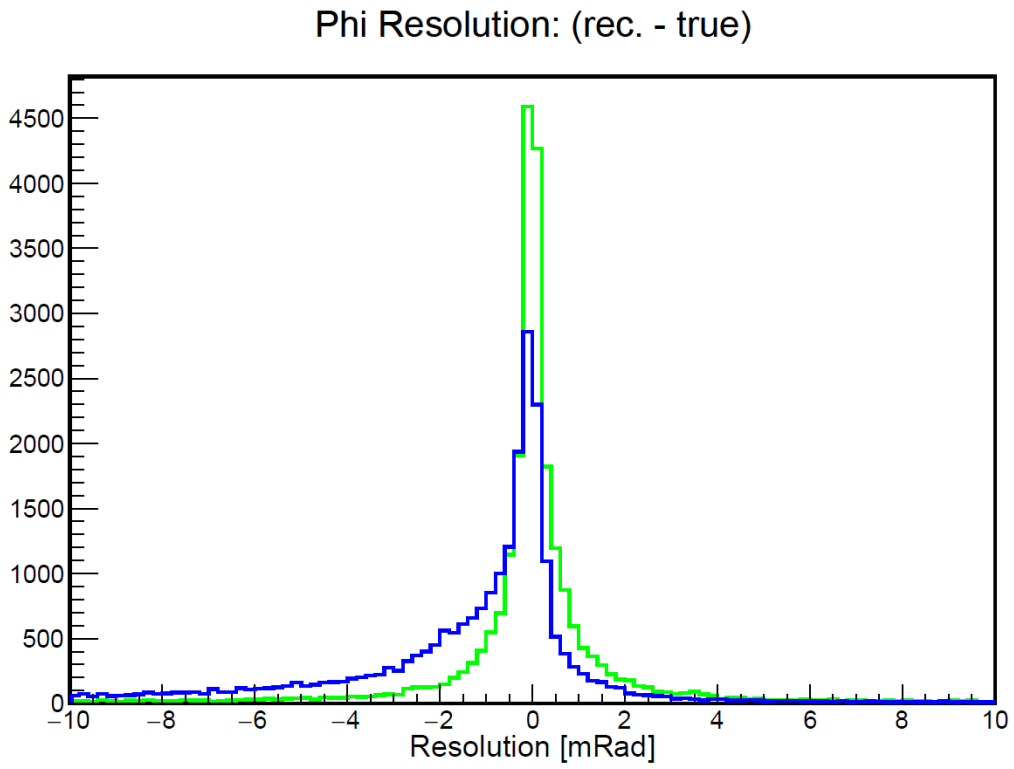


Seed level

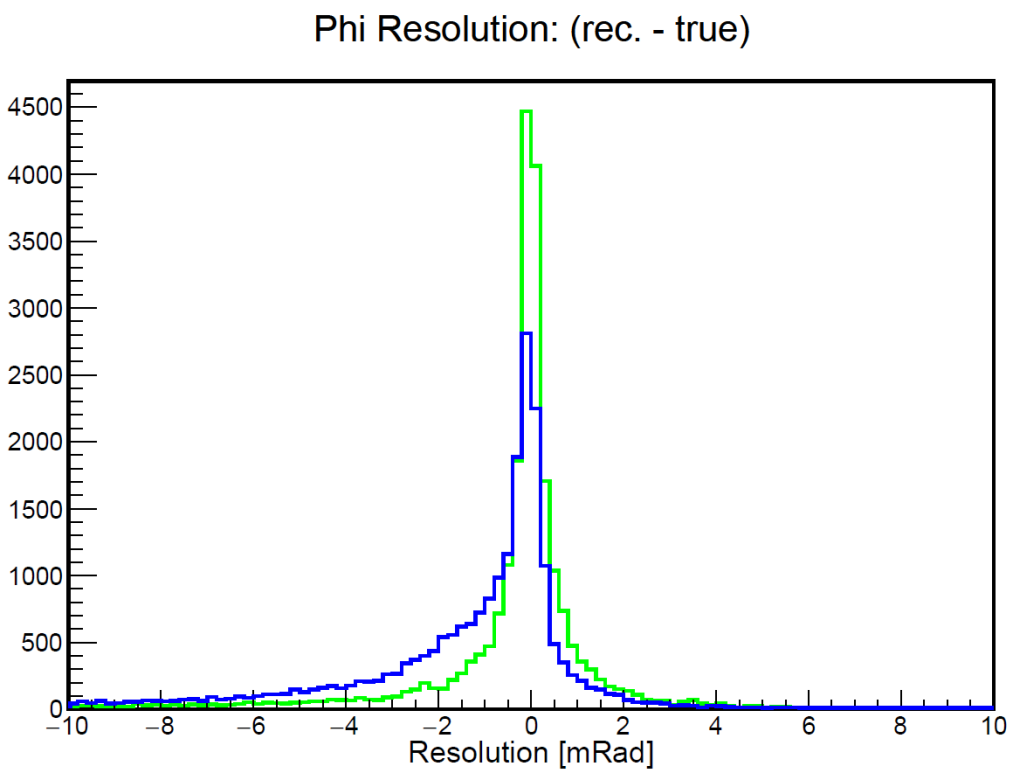
Track level

Phi resolution

ACTS v21.1



ACTS v30

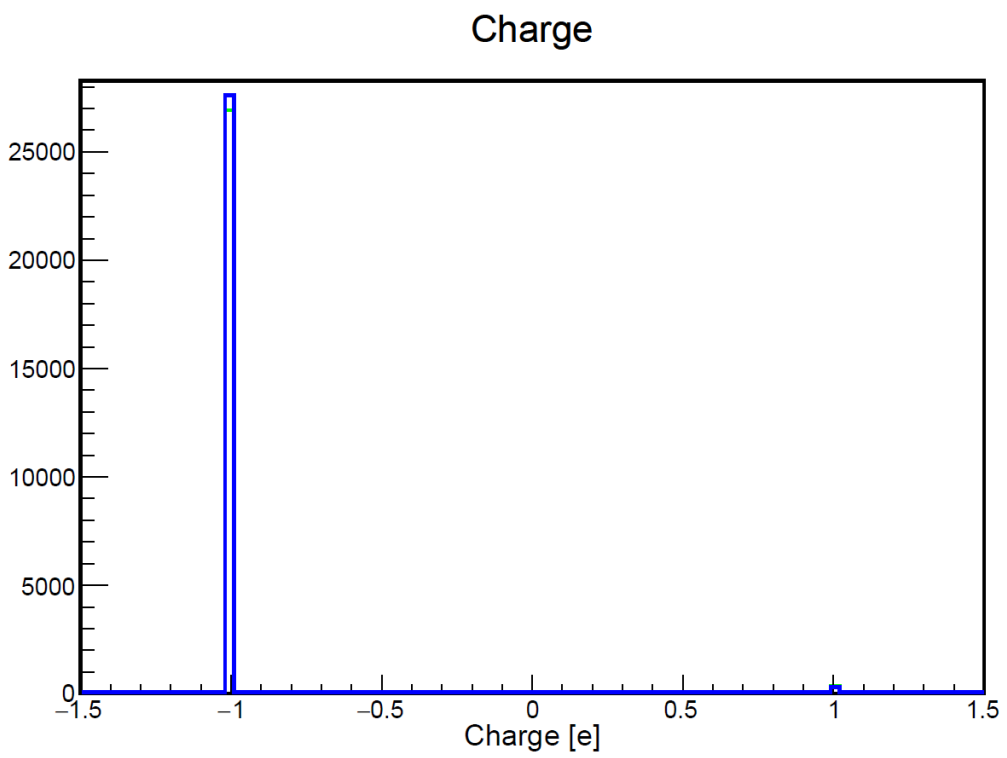


Seed level

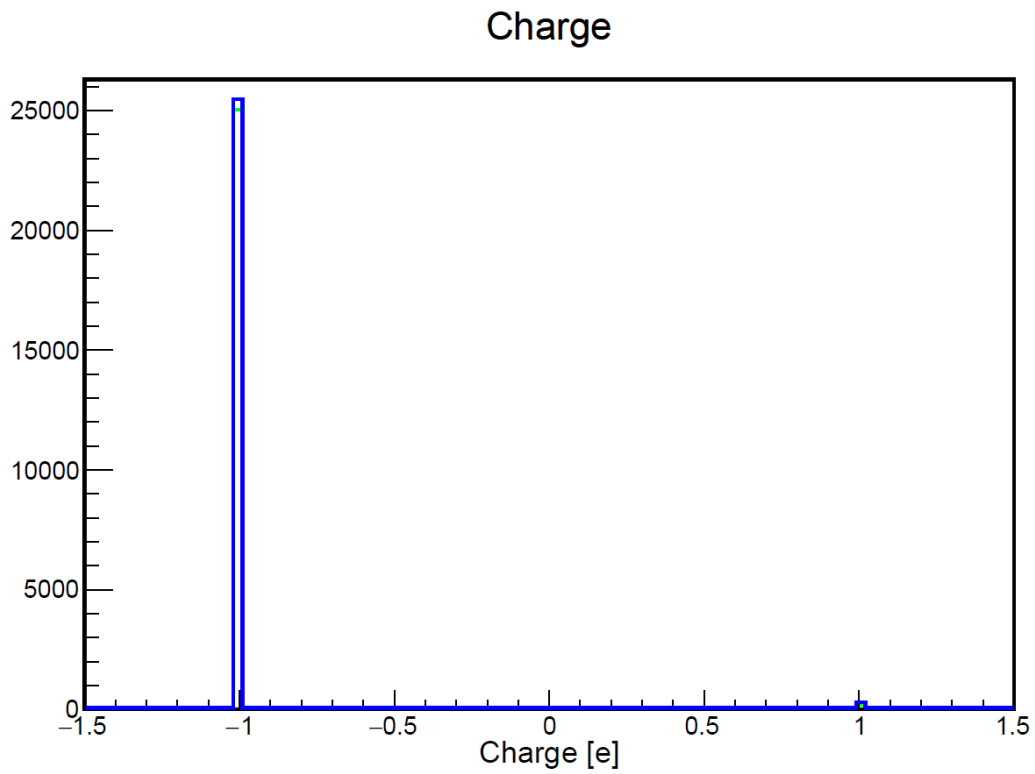
Track level

Charge reconstruction

ACTS v21.1



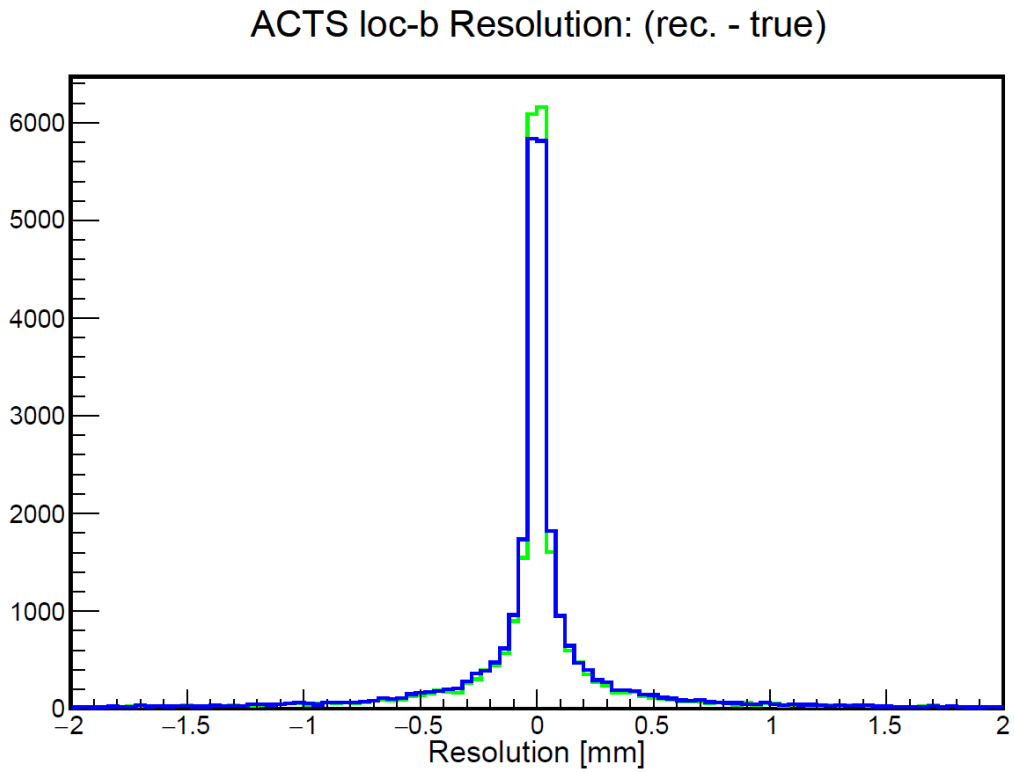
ACTS v30



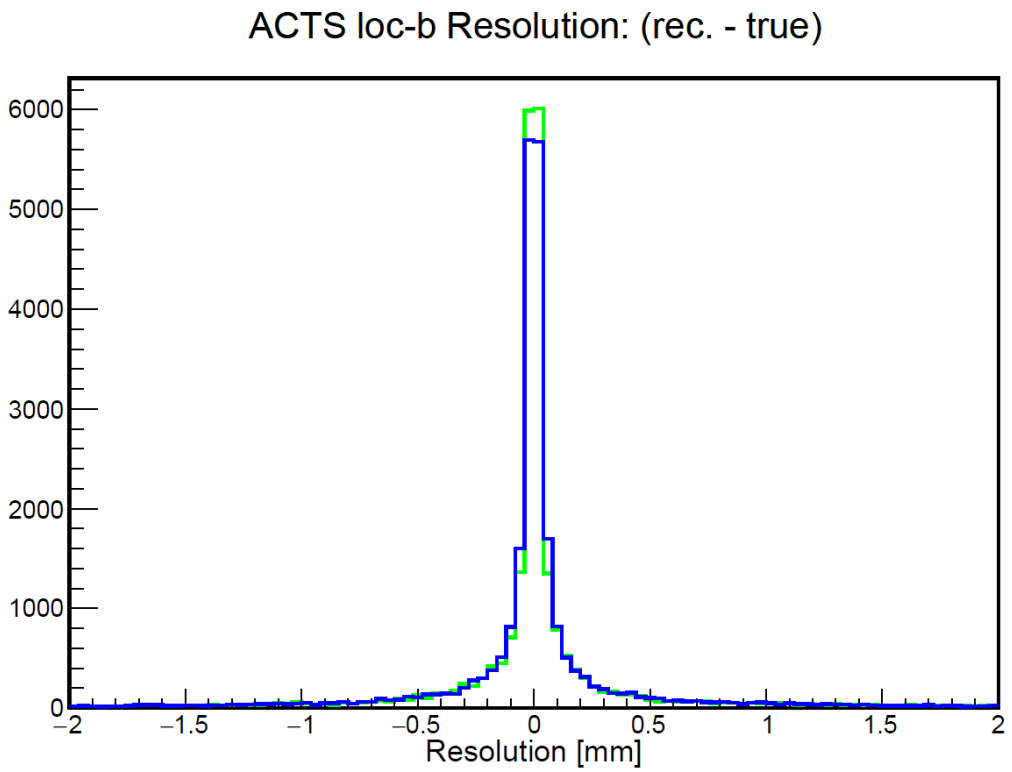
Seed level
Track level

Loc-b resolution

ACTS v21.1



ACTS v30

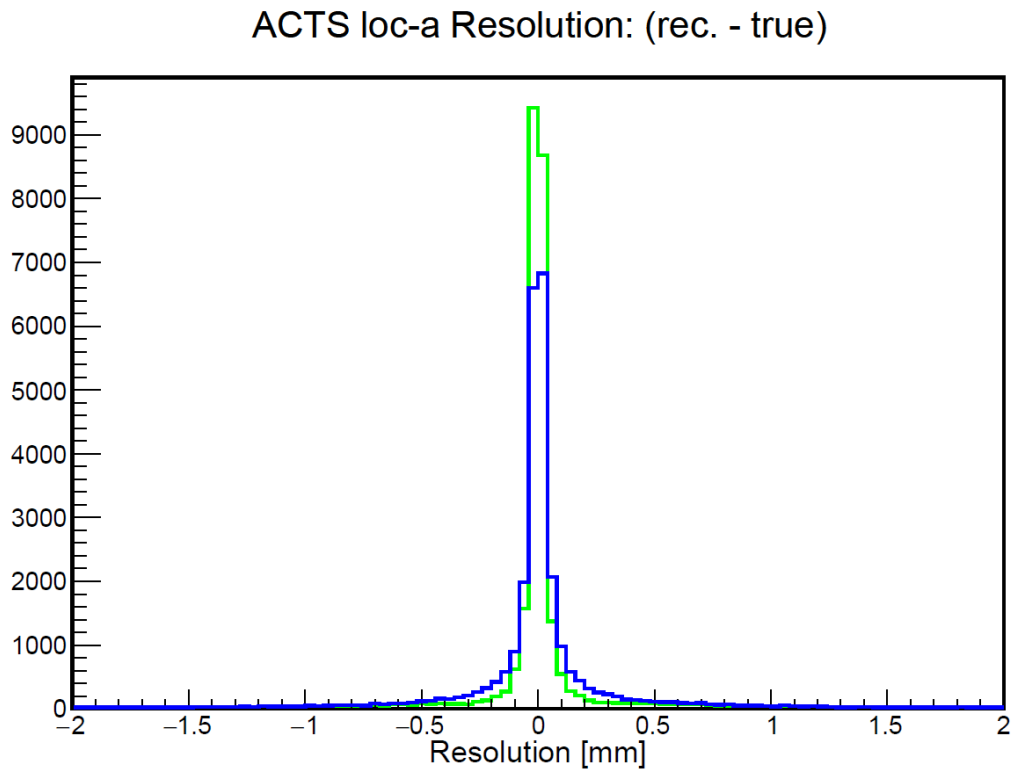


Loc-b: longitudinal impact parameter with respect to (0,0,0)

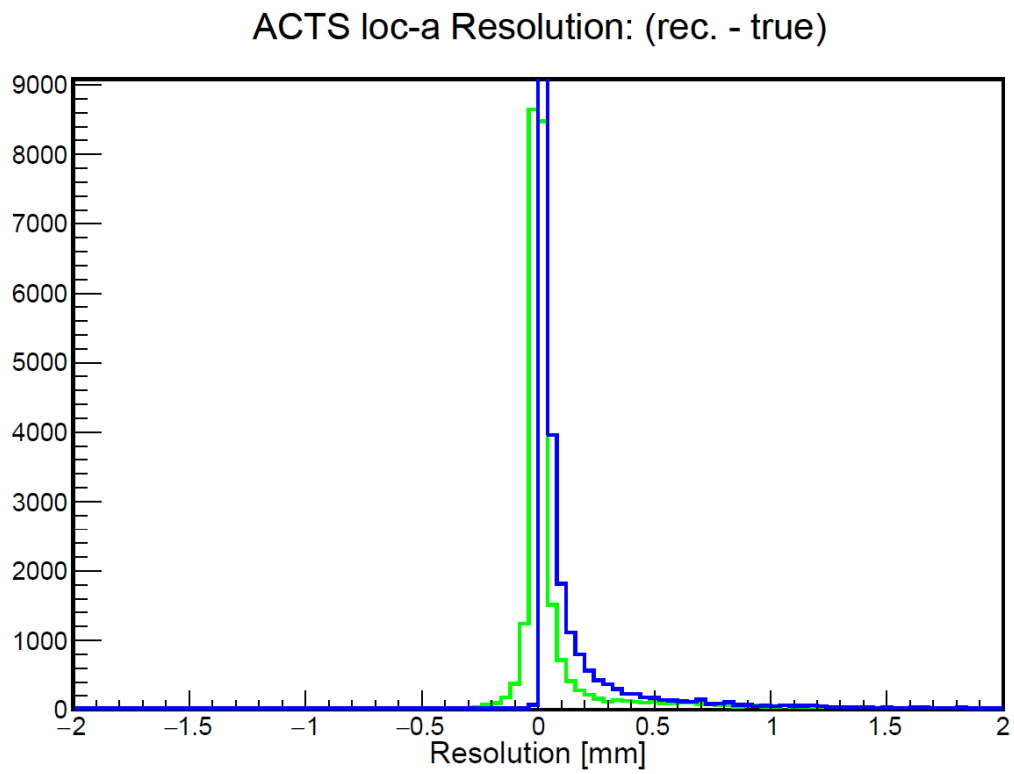
Seed level
Track level

Loc-a resolution

ACTS v21.1



ACTS v30



Loc-a: signed transverse impact parameter with respect to (0,0,0)

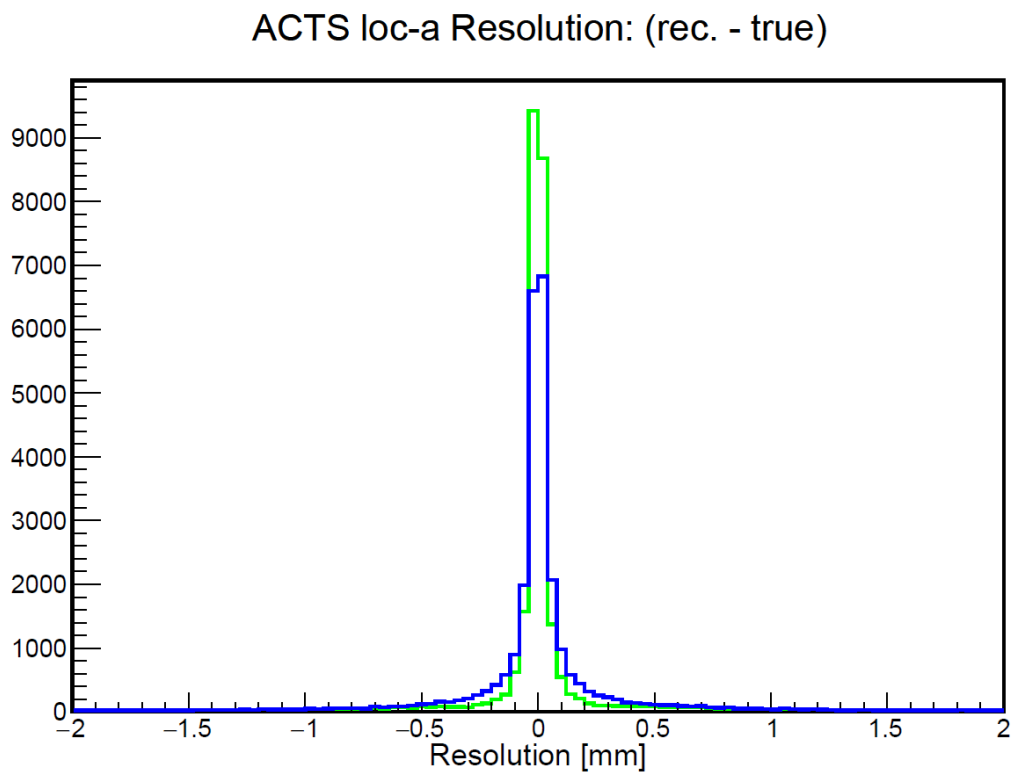
Seed level

Track level

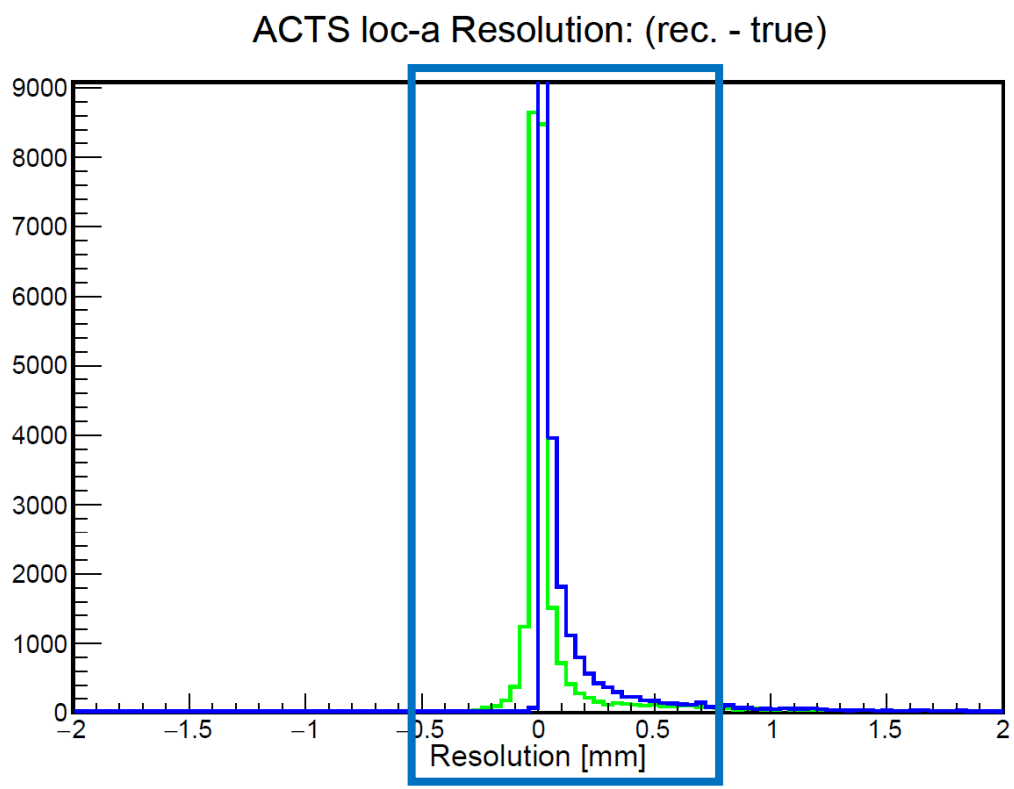
Loc-a resolution

At the seed level, ACTS loc-a is always being set positive in our v30 implementation.

ACTS v21.1



ACTS v30



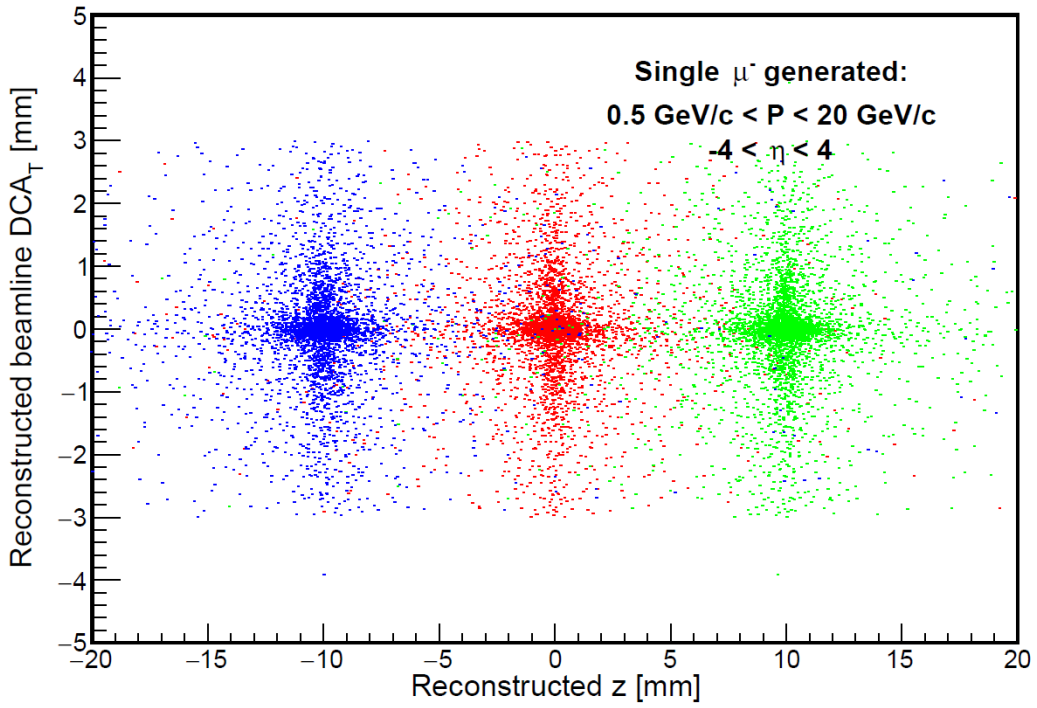
Loc-a: signed transverse impact parameter with respect to (0,0,0)

This leads to a bias in the track-level reconstruction

(vx,vy,vz) = (0,0,0) mm
(vx,vy,vz) = (0,0,+10) mm
(vx,vy,vz) = (0,0,-10) mm

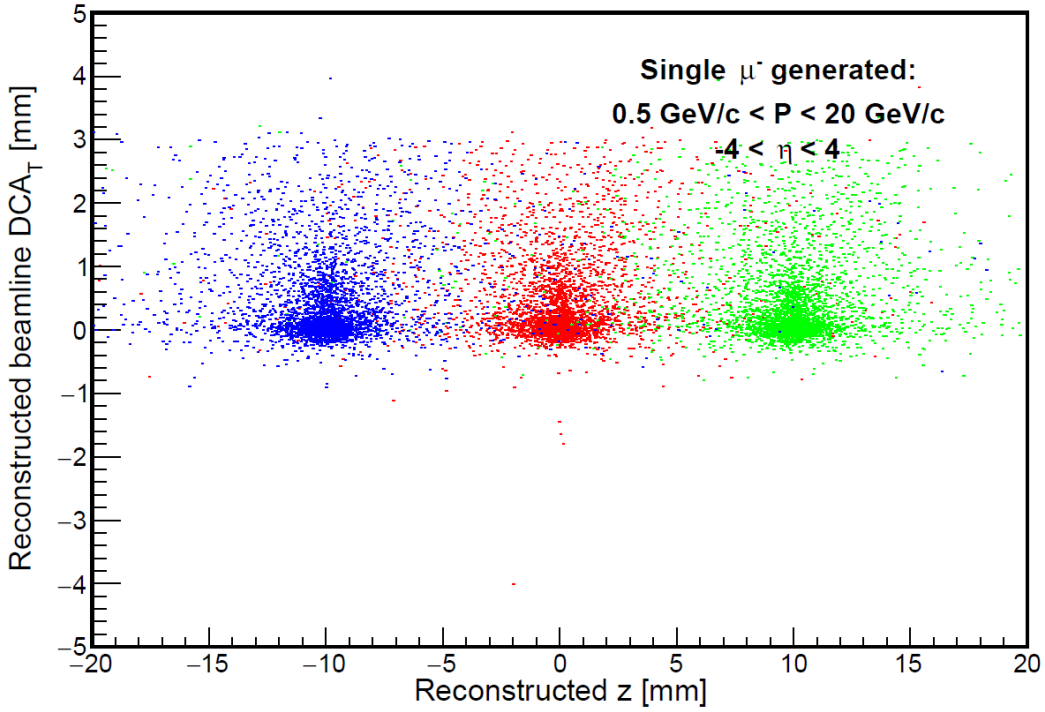
ACTS v21.1

Real-seeded tracking



ACTS v30

Real-seeded tracking



Reconstructed z: longitudinal impact parameter with respect to (0,0,0)
Reconstructed transverse DCA: signed transverse impact parameter with respect to (0,0,0)

Cause of this issue

- In our EICRecon seeding code, we call the *globalToLocal* function. If this is function is successful, then it is used to set the ACTS loc-a and loc-b values.
- If it fails, however, ACTS loc-a and loc-b are set by hand. In this case, ACTS loc-a gets (incorrectly) always set as a positive number.
- In ACTS v21.1, the call to the *globalToLocal* function would usually work successfully; but in ACTS v30, it is almost always failing.

```
207           const float z0 = seed.z();
208           auto perigee = Acts::Surface::makeShared<Acts::PerigeeSurface>(Acts::Vector3(0,0,0));
209           Acts::Vector3 global(xypos.first, xypos.second, z0);
210
211           auto local = perigee->globalToLocal(m_geoSvc->getActsGeometryContext(),
212                                               global, Acts::Vector3(1,1,1));
213
214           Acts::Vector2 localpos(sqrt(square(xypos.first) + square(xypos.second)), z0);
215           if(local.ok())
216             {
217               localpos = local.value();
218             }
```

https://github.com/eic/EICrecon/blob/main/src/algorithms/tracking/TrackSeeding.cc

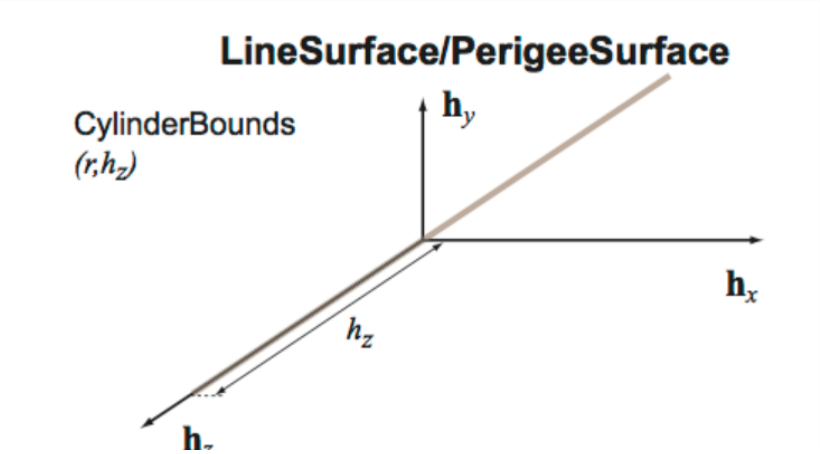
Cause of this issue

Line surface

`Acts::LineSurface` is a special kind of surface that depends on a reference direction, typically the unit momentum direction $\vec{d}$ of a particle. A point in space is considered *on surface* if and only if it coincides with the point of closest approach between the direction vector $\vec{d}$ and the line direction vector $\vec{z}$. As such, the function `Acts::LineSurface::globalToLocal()` can fail, if the argument position and direction do not fulfill this criterion. It is pure-virtual, meaning that it can not be instantiated on its own.

class **LineSurface** : *public* Acts::Surface

Base class for a linear surfaces in the TrackingGeometry to describe dirft tube, straw like detectors or the Perigee It inherits from Surface.



https://acts.readthedocs.io/en/latest/core/geometry/surfaces.html#line-surface

Passing in a constant vector is clearly wrong – we should pass in the seed momentum vector that we reconstructed. My guess is that in ACTS v21.1, for a particle generated at (x,y) = (0,0), the tolerance was set large enough that this was not an issue. But it has become one in ACTS v30. It was always an issue for off-beamline particles, so it should be fixed in any case.

```
207           const float z0 = seed.z();
208           auto perigee = Acts::Surface::makeShared<Acts::PerigeeSurface>(Acts::Vector3(0,0,0));
209           Acts::Vector3 global(xypos.first, xypos.second, z0);
210
211           auto local = perigee->globalToLocal(m_geoSvc->getActsGeometryContext(),
212                                               global, Acts::Vector3(1,1,1));
213
214           Acts::Vector2 localpos(sqrt(square(xypos.first) + square(xypos.second)), z0);
215           if(local.ok())
216             {
217               localpos = local.value();
218             }
```

https://github.com/eic/EICrecon/blob/main/src/algorithms/tracking/TrackSeeding.cc

Version comparison summary

- Other than for ACTS Loc-a, the seeding results and the resultant tracking with these seeds is very stable.
- The issue with the reconstruction of ACTS Loc-a seems to be more an issue with how we are calling the *globaltoLocal* function in our EICRecon track seeding code. Jeetendra Gupta and I are working to implement and test the fix.

Another (not version related) issue

If we generate a negative muon from

(x,y,z) = (10,0,0) mm

with a momentum direction of

(px,py,pz) = {+cos(10^o),+sin(10^o), 0},

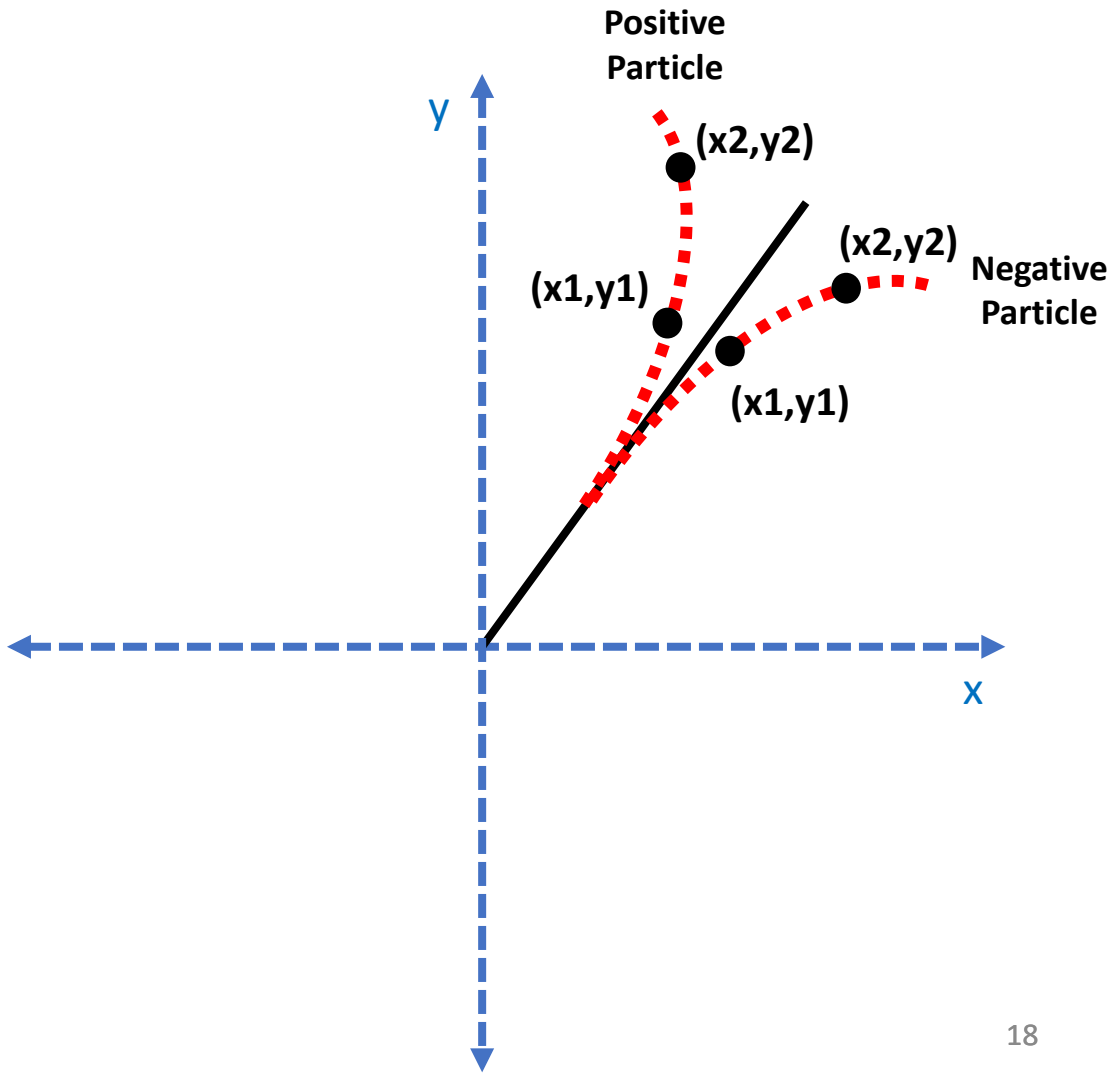
we see the seed charge reconstructed as positive. This causes the seed phi to be wrong and the track reconstruction to fail.

```
CentralTrackSeedingResults = (vector<edm4eic::TrackParametersData>*)0x59186a0
 CentralTrackSeedingResults.type = -1, -1
 CentralTrackSeedingResults.loc.a = 1.748047, 1.722656
 CentralTrackSeedingResults.loc.b = -0.000000, -0.013325
 CentralTrackSeedingResults.locError.xx = 0.100000, 0.100000
 CentralTrackSeedingResults.locError.yy = 0.100000, 0.100000
 CentralTrackSeedingResults.locError.xy = 0.000000, 0.000000
 CentralTrackSeedingResults.theta = 1.570544, 1.570720
 CentralTrackSeedingResults.phi = -2.966284, -2.967019
 CentralTrackSeedingResults.qOverP = 0.064824, 0.047646
```

```
CentralTrackSeedingResults.time = 10.000000, 10.000000
CentralTrackSeedingResults.timeError = 0.100000, 0.100000
CentralTrackSeedingResults.charge = 1.000000, 1.000000
```

Another (not version related) issue

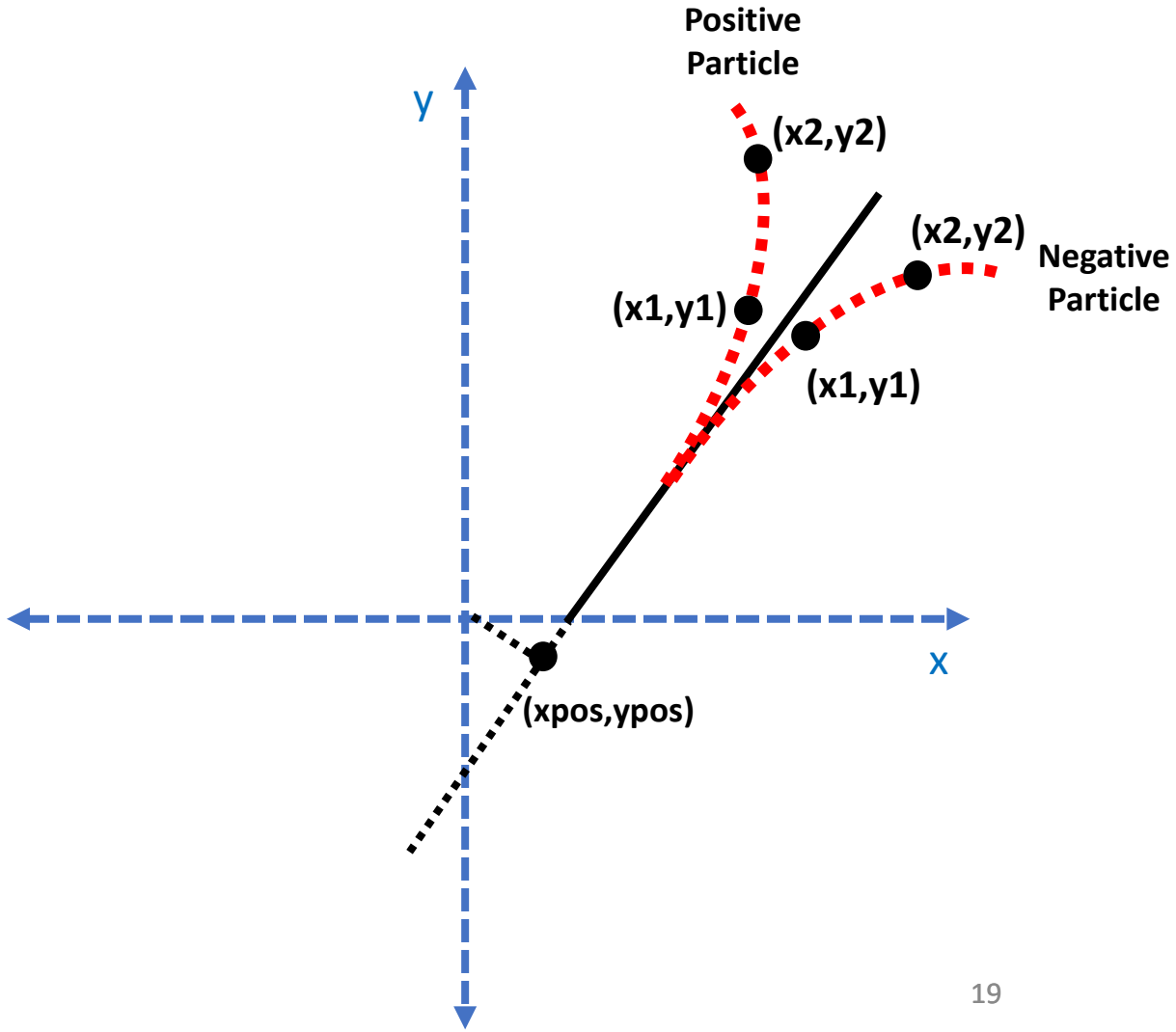
```
250    int eicrecon::TrackSeeding::determineCharge(std::vector<std::pair<float,float>>& positions) const
251    {
252      // determine the charge by the bend angle of the first two hits
253      int charge = 1;
254      const auto& firstpos = positions.at(0);
255      const auto& secondpos = positions.at(1);
256
257      const auto firstphi = atan2(firstpos.second, firstpos.first);
258      const auto secondphi = atan2(secondpos.second, secondpos.first);
259      auto dphi = secondphi - firstphi;
260      if(dphi > M_PI) dphi = 2.*M_PI - dphi;
261      if(dphi < -M_PI) dphi = 2*M_PI + dphi;
262      if(dphi < 0) charge = -1;
263
264      return charge;
265    }
```



https://github.com/eic/EICrecon/blob/main/src/algorithms/tracking/TrackSeeding.cc

Another (not version related) issue

```
250    int eicrecon::TrackSeeding::determineCharge(std::vector<std::pair<float,float>>& positions) const
251    {
252      // determine the charge by the bend angle of the first two hits
253      int charge = 1;
254      const auto& firstpos = positions.at(0);
255      const auto& secondpos = positions.at(1);
256                                  atan2(firstpos.second – ypos, firstpos.first – xpos)
257      const auto firstphi = atan2(firstpos.second, firstpos.first);
258      const auto secondphi = atan2(secondpos.second, secondpos.first);
259      auto dphi = secondphi - firstphi;  atan2(secondpos.second – ypos, secondpos.first – xpos)
260      if(dphi > M_PI) dphi = 2.*M_PI - dphi;
261      if(dphi < -M_PI) dphi = 2*M_PI + dphi;
262      if(dphi < 0) charge = -1;
263
264      return charge;
265    }
```



https://github.com/eic/EICrecon/blob/main/src/algorithms/tracking/TrackSeeding.cc

Another (not version related) issue

May be better to use matrix methods to determine charge:

Check sign of $\begin{vmatrix} x1 & y1 & 1 \\ x2 & y2 & 1 \\ x3 & y3 & 1 \end{vmatrix}$

```
250    int eicrecon::TrackSeeding::determineCharge(std::vector<std::pair<float,float>>& positions) const
251    {
252      // determine the charge by the bend angle of the first two hits
253      int charge = 1;
254      const auto& firstpos = positions.at(0);
255      const auto& secondpos = positions.at(1);
256                                  atan2(firstpos.second – ypos, firstpos.first – xpos)
257      const auto firstphi = atan2(firstpos.second, firstpos.first);
258      const auto secondphi = atan2(secondpos.second, secondpos.first);
259      auto dphi = secondphi - firstphi;  atan2(secondpos.second – ypos, secondpos.first – xpos)
260      if(dphi > M_PI) dphi = 2.*M_PI - dphi;
261      if(dphi < -M_PI) dphi = 2*M_PI + dphi;
262      if(dphi < 0) charge = -1;
263
264      return charge;
265    }
```

https://github.com/eic/EICrecon/blob/main/src/algorithms/tracking/TrackSeeding.cc