
# Off-Axis Tracking Issue Fix

## PCA Seeded Tracking

15th February, 2024

Harsimran Singh
Lokesh Kumar
Department of Physics, Panjab University
Chandigarh, India

# Outline

- Issue with current truth-seeded tracking
- Proposed Fix: PCA Seeded Tracking
  - Definition, approach & calculation
- Testing & Analysis Results
- Summary & Discussion

# Issue with current truth-seeded tracking

We set the initial track parameters based on the generated particle's momentum vector, charge, and creation point. This information is then fed into the CKF in addition to a line surface (perigee surface) along the z axis through (0,0,0).

```
// Insert into edm4eic::TrackParameters, which uses numerical values in its specified units
auto track_parameter = track_parameters->create();
track_parameter.setType(-1); // type --> seed(-1)
track_parameter.setLoc({static_cast<float>(std::hypot(v.x, v.y)), static_cast<float>(v.z)}); // 2d location on surface [mm]
track_parameter.setLocError({1.0, 1.0}); // sqrt(variance) of location [mm]
track_parameter.setTheta(theta); //theta [rad]
track_parameter.setPhi(phi); // phi [rad]
track_parameter.setQOverP(charge / pinit); // Q/p [e/GeV]
track_parameter.setMomentumError({0.01, 0.05, 0.1}); // sqrt(variance) on theta, phi, q/p [rad, rad, e/GeV]
track_parameter.setTime(mcparticle.getTime()); // time [ns]
track_parameter.setTimeError(10e9); // error on time [ns]
track_parameter.setCharge(charge); // charge
```

```
// Construct a perigee surface as the target surface
auto pSurface = Acts::Surface::makeShared<const Acts::PerigeeSurface>(Acts::Vector3(0,0,0));

// Create parameters
acts_init_trk_params.emplace_back(pSurface, params, charge, cov);
```

Link to Barak's Talk: https://indico.bnl.gov/event/21238/contributions/83543/attachments/51067/87297/tracking_111623.pdf

# Issue with current truth-seeded tracking

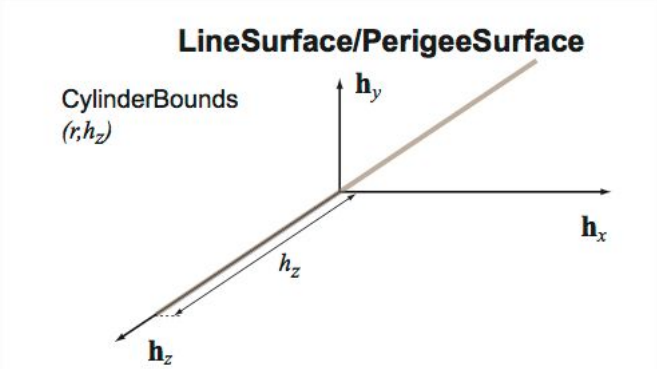
The created particle's momentum vector should be tangential to the cylinder surrounding the line surface which may not be the case always.

## 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.

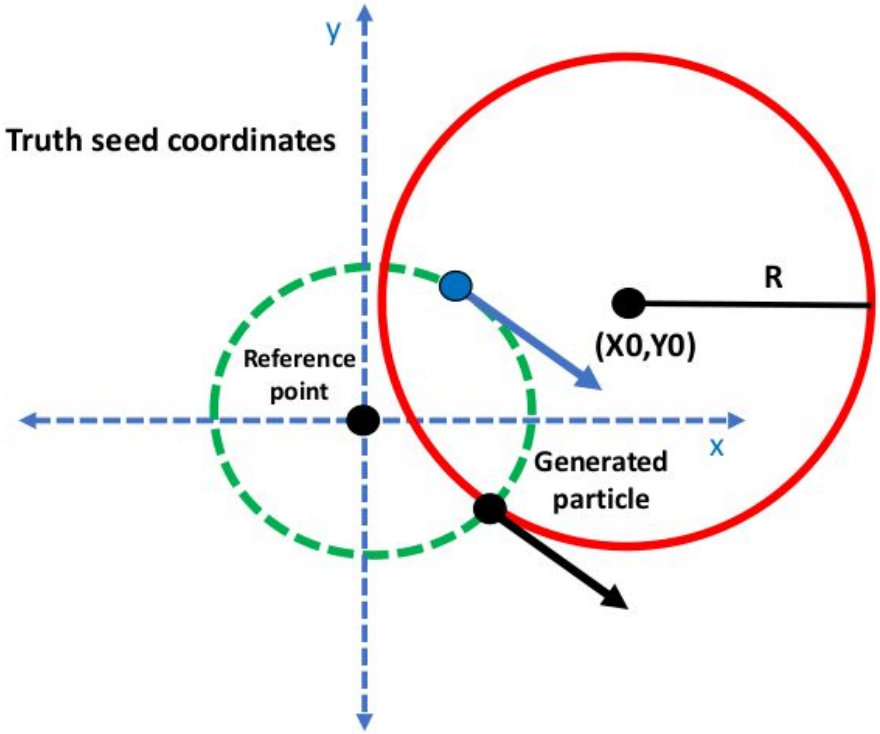
*class* **LineSurface** : *public* Acts::Surface

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




Link to Barak's Talk: https://indico.bnl.gov/event/21238/contributions/83543/attachments/51067/87297/tracking_111623.pdf

# Issue with current truth-seeded tracking




**Black arrow**: Generated particle at its creation point

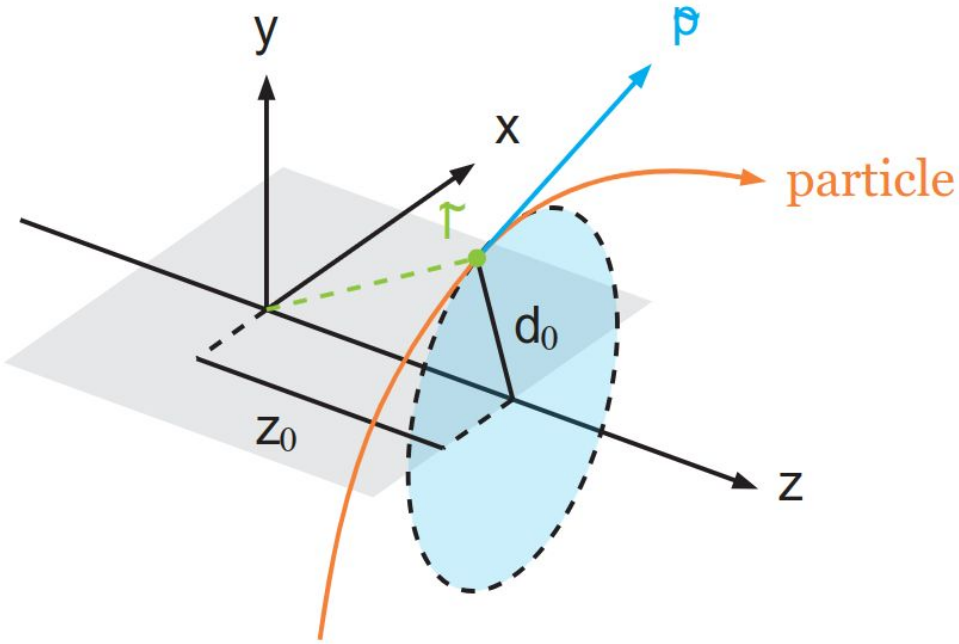
**Blue arrow**: Where the CKF will think the particle comes from in the current truth seeding implementation.

In Barak's talk, he gave an example where he generated particle at (10,0,0) mm in non-tangential direction to the line surface. CKF thought that particle was created at (-1.74,9.85,0) mm in tangential direction.

Link to Barak's Talk: https://indico.bnl.gov/event/21238/contributions/83543/attachments/51067/87297/tracking_111623.pdf

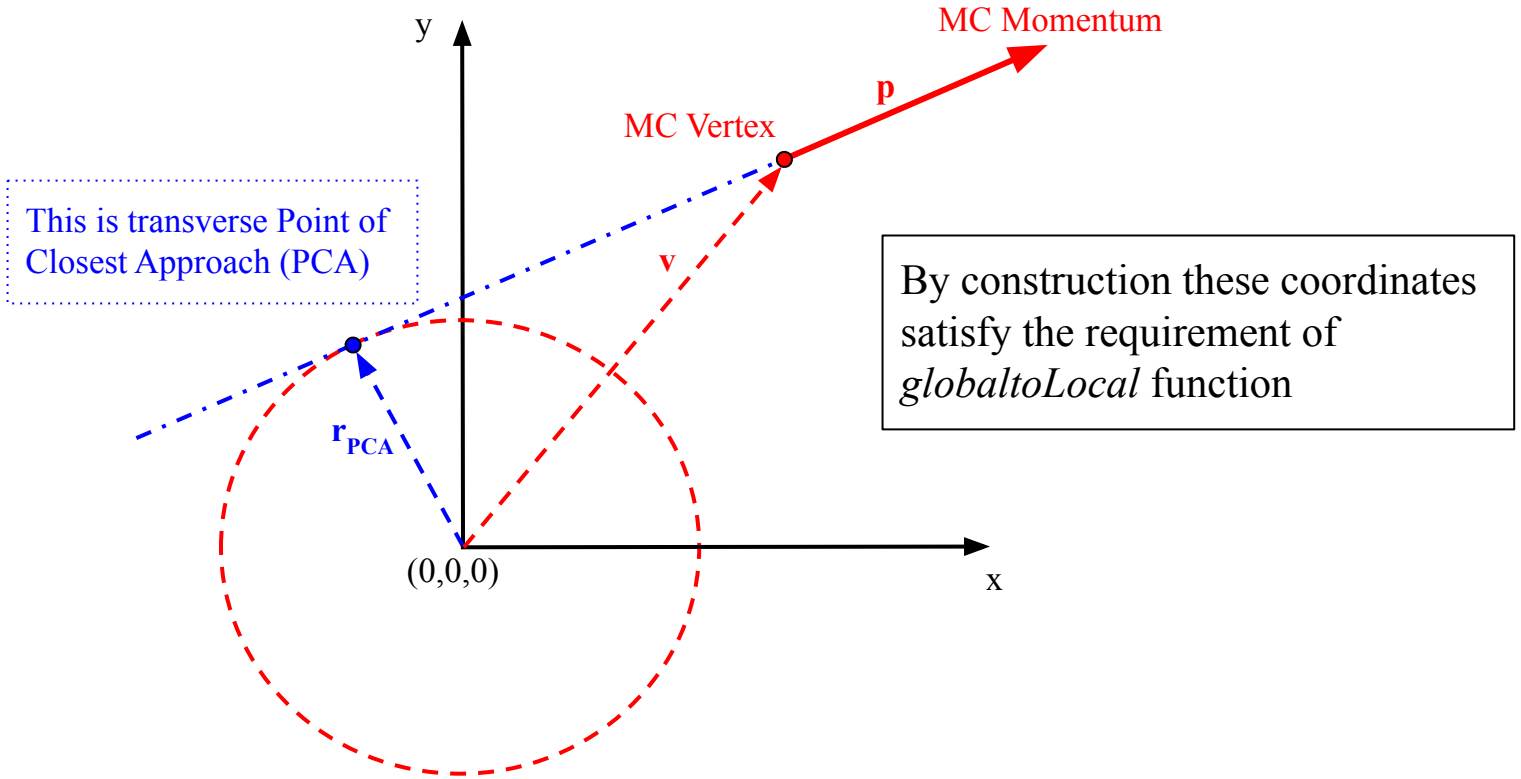
# Proposed Fix: Point of Closest Approach (PCA) Definition

In ACTS, a point on the trajectory in xy-plane that is closest to the beamline axis is called transverse PCA.




Source: https://acts.readthedocs.io/en/latest/tracking.html

# Proposed Fix: PCA calculation using straight line approximation



# Proposed Fix: PCA Calculation

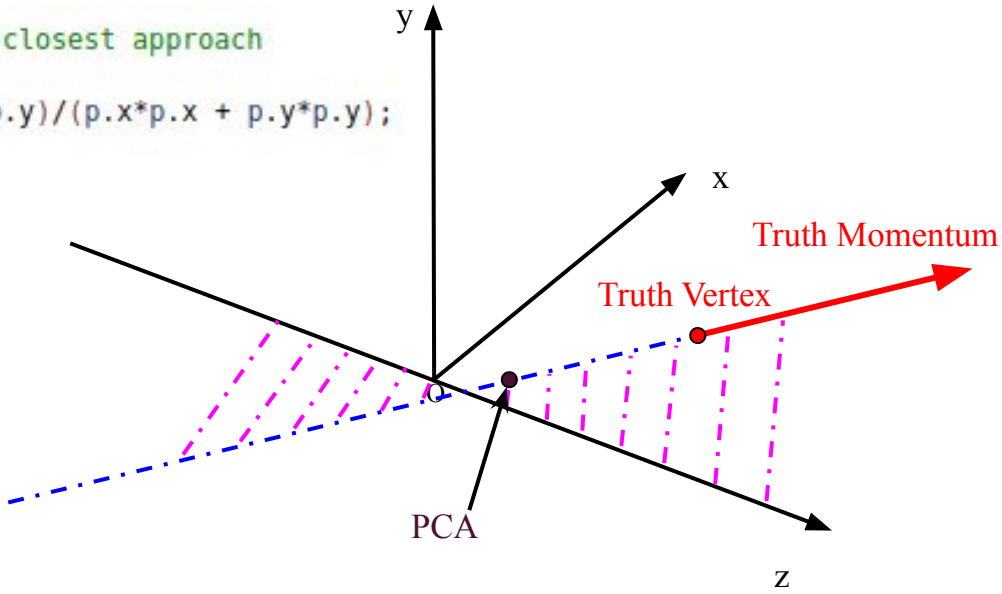
The coordinates of PCA will be where dotted magenta line distance is minimum from z axis

```
// track particle back to transverse point-of-closest approach
// with respect to the defined line surface
auto linesurface_parameter = -(v.x*p.x + v.y*p.y)/(p.x*p.x + p.y*p.y);

auto xpca = v.x + linesurface_parameter*p.x;
auto ypca = v.y + linesurface_parameter*p.y;
auto zpca = v.z + linesurface_parameter*p.z;
```

Truth Vertex Components

Truth Momentum Components



# Proposed Fix: PCA Calculation

PR : https://github.com/eic/EICrecon/pull/1291

```
// define line surface for local position values
auto perigee = Acts::Surface::makeShared<Acts::PerigeeSurface>(Acts::Vector3(0,0,0));

// track particle back to transverse point-of-closest approach
// with respect to the defined line surface

auto linesurface_parameter = -(v.x*p.x + v.y*p.y)/(p.x*p.x + p.y*p.y);


auto xpca = v.x + linesurface_parameter*p.x;
auto ypca = v.y + linesurface_parameter*p.y;
auto zpca = v.z + linesurface_parameter*p.z;

Acts::Vector3 global(xpca, ypca, zpca);

// convert from global to local coordinates using the defined line surface
Acts::Vector2 localpos;
Acts::Vector3 direction(sin(theta)*cos(phi), sin(theta)*sin(phi), cos(theta));
auto local = perigee->globalToLocal(m_geoSvc->getActsGeometryContext(), global, direction);

if(!local.ok())
{
    continue;
}

localpos = local.value();

// Insert into edm4eic::TrackParameters, which uses numerical values in its specified units
auto track_parameter = track_parameters->create();
track_parameter.setType(-1); // type --> seed(-1)
track_parameter.setLoc({(float)localpos(0), (float)localpos(1)}); // 2d location on surface [mm]
```

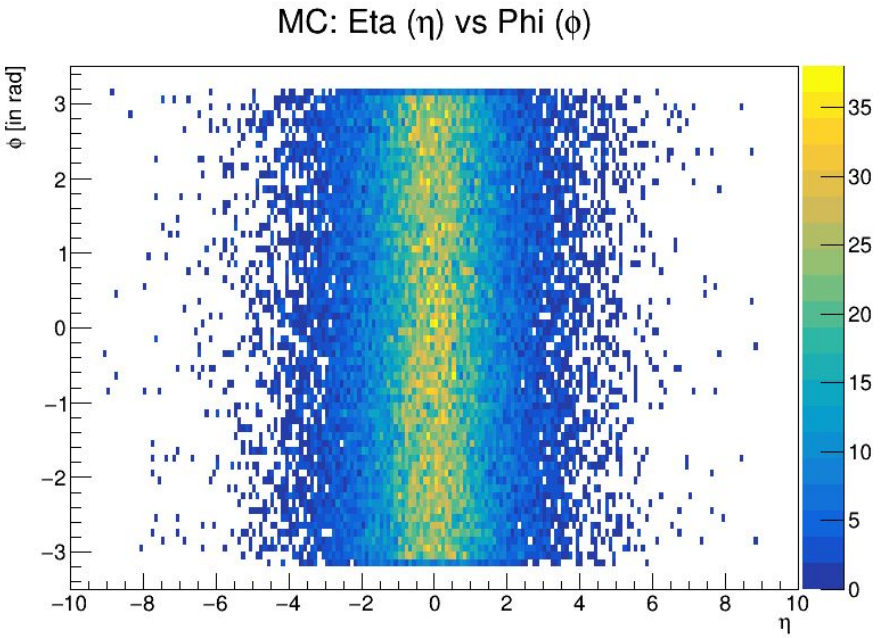
# Testing the fix:

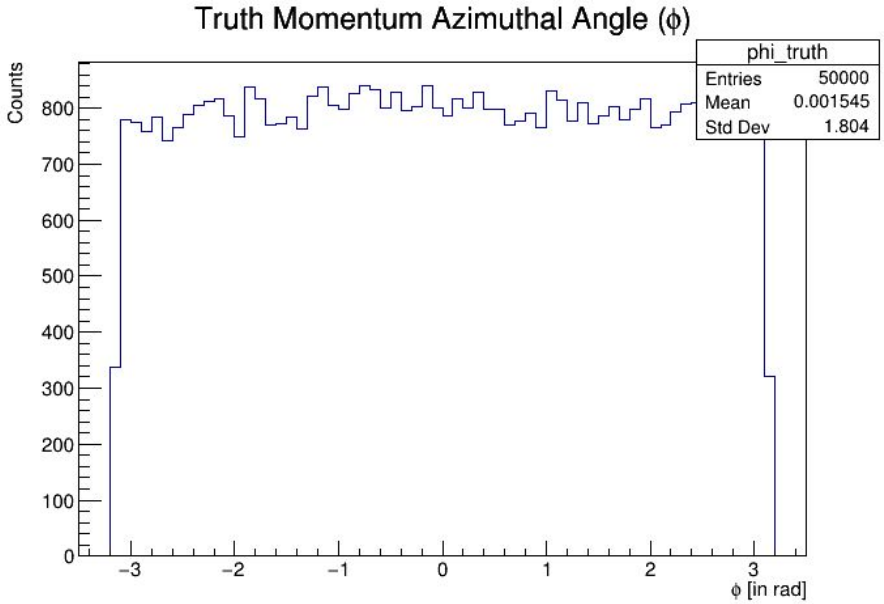
The *ddsim* output file used to analyse the effect of changes in *EICrecon* had the following configurations:

- Particle Thrown: Muon
- Number of Events: 5000
- Gun Multiplicity i.e. Muons thrown per event: 10
- Distribution used: Uniform, so it will be flat in theta & phi
- Min. Muon Momentum: 0 GeV
- Max. Muon Momentum: 10 GeV
- Gun Direction: (0.000 0.000 1.000) [in mm] //default
- Gun Position: (3.000 4.000 5.000) [in mm] *vertex position*
- Compact File: $DETECTOR_PATH/$DETECTOR_CONFIG.xml

# Analysis: Truth/MC Particles Distributions

Vertex: (3,4,5) mm



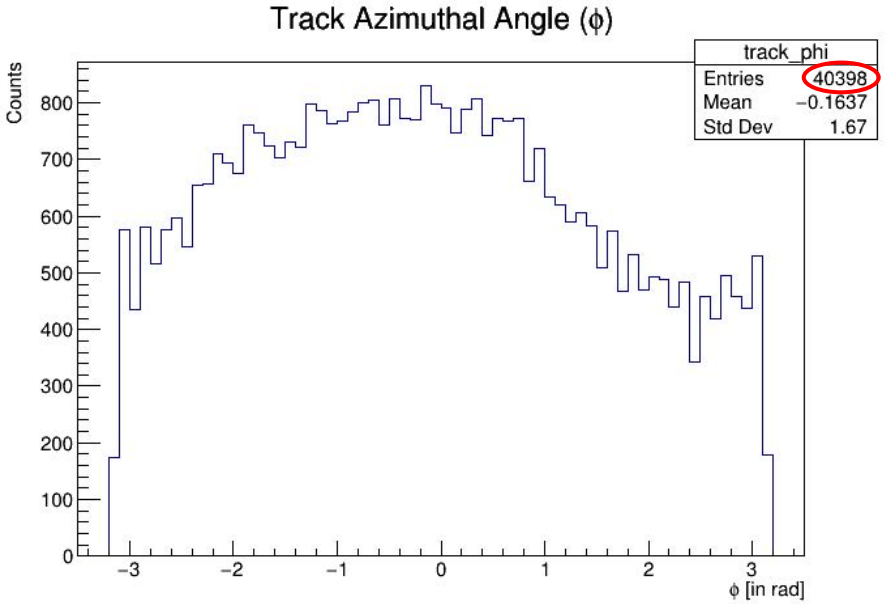
# Results: Track Phi

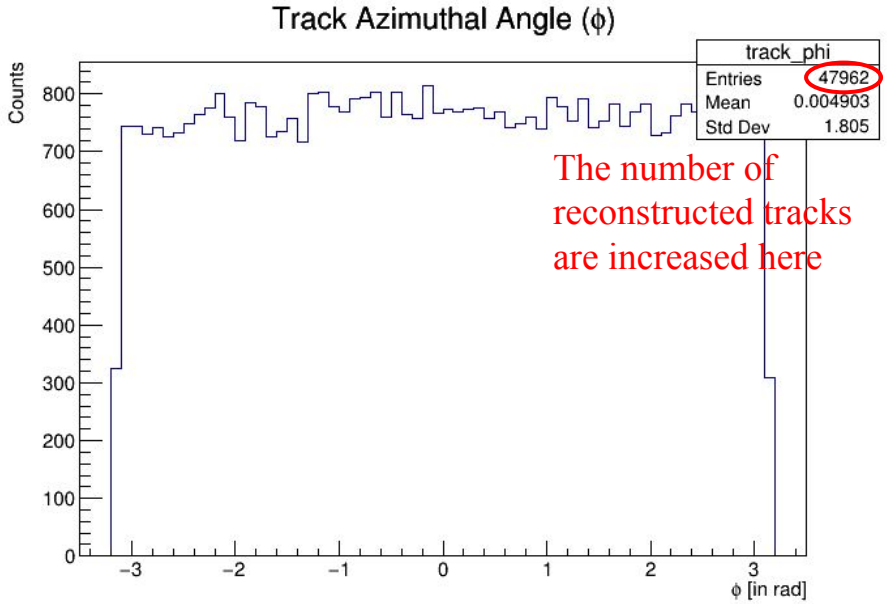
Vertex: (3,4,5) mm

default

50000 -ve muon tracks

changed



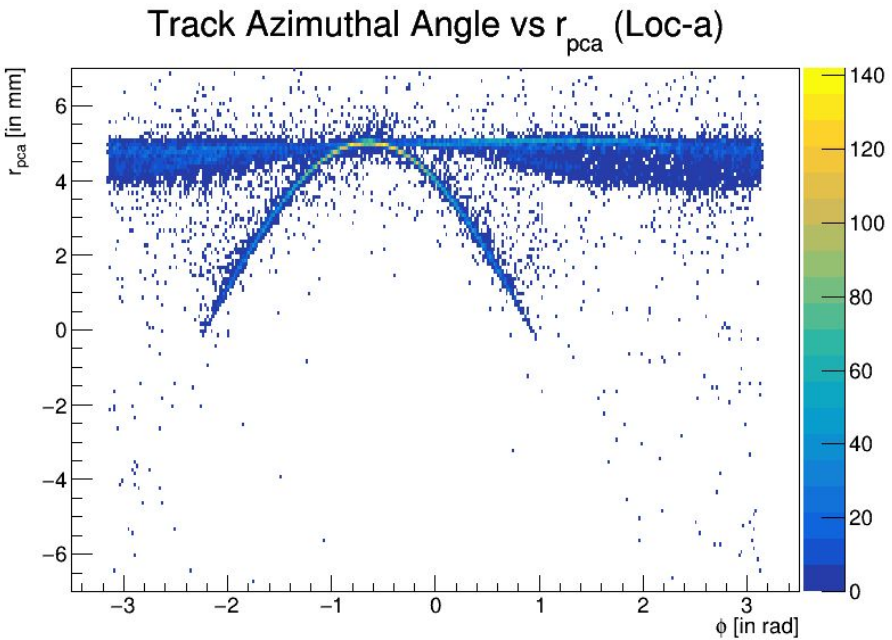




The number of reconstructed tracks are increased here

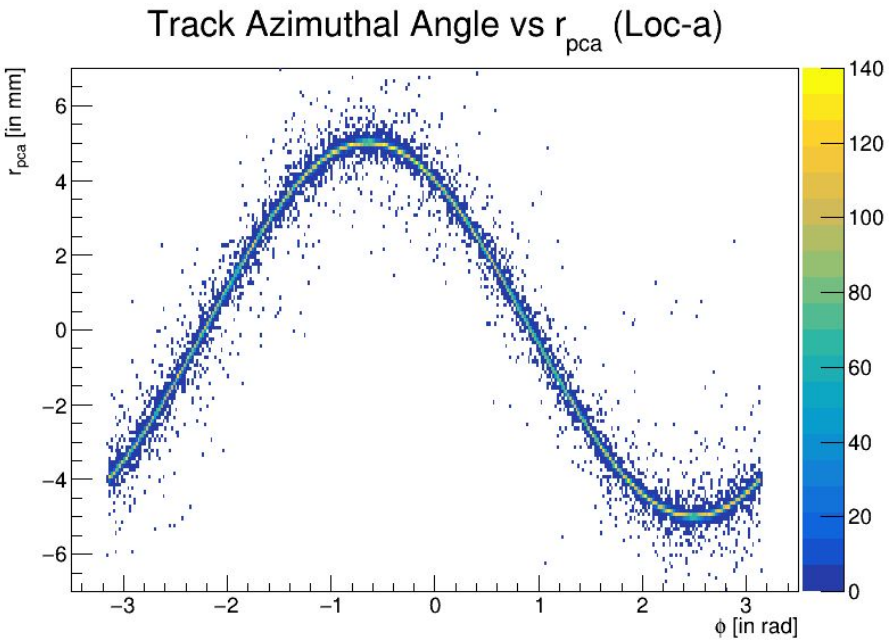
# Results: Track phi vs $r_{pca}$

Vertex: (3,4,5) mm

default



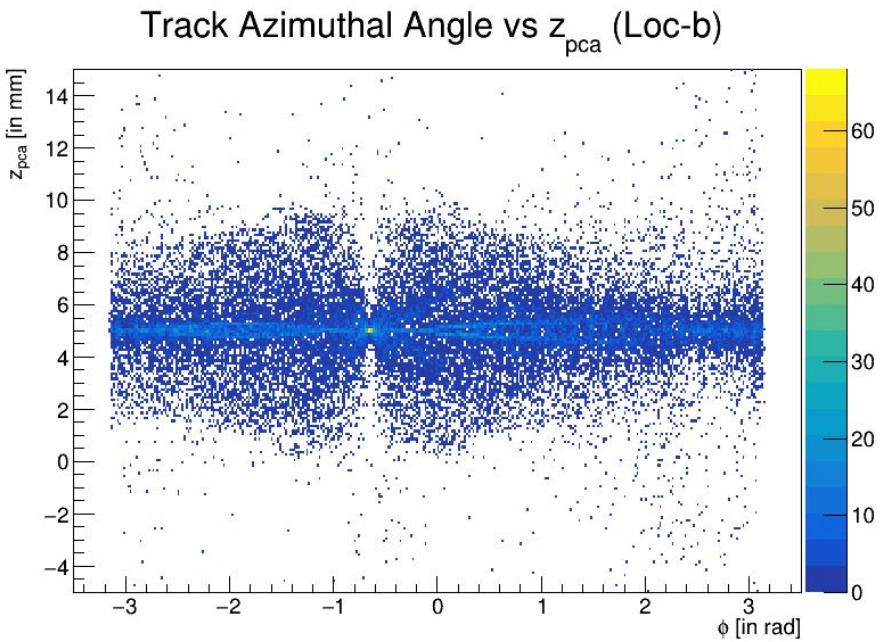

changed



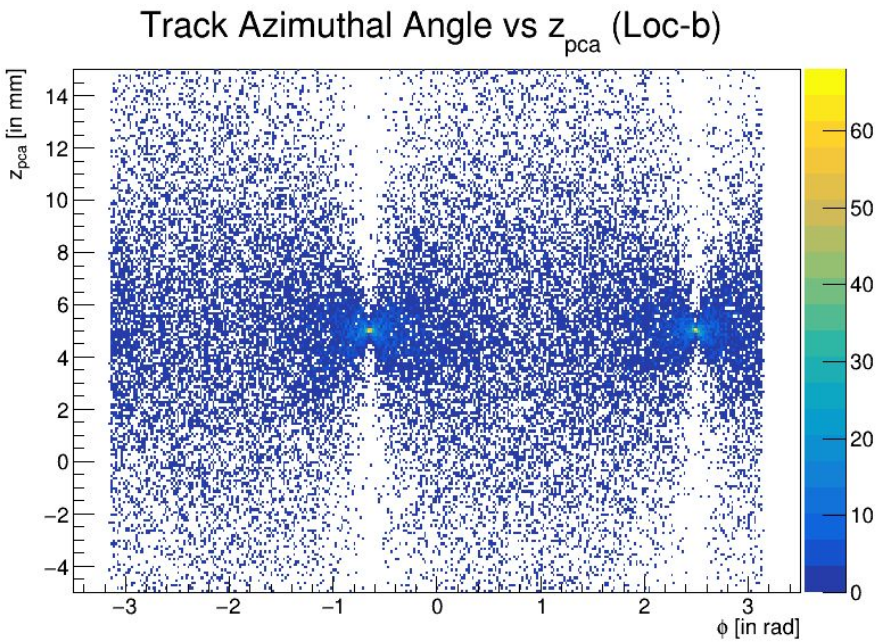
# Results: Track phi vs $z_{pca}$

Vertex: (3,4,5) mm

default




changed



# Summary & Discussion

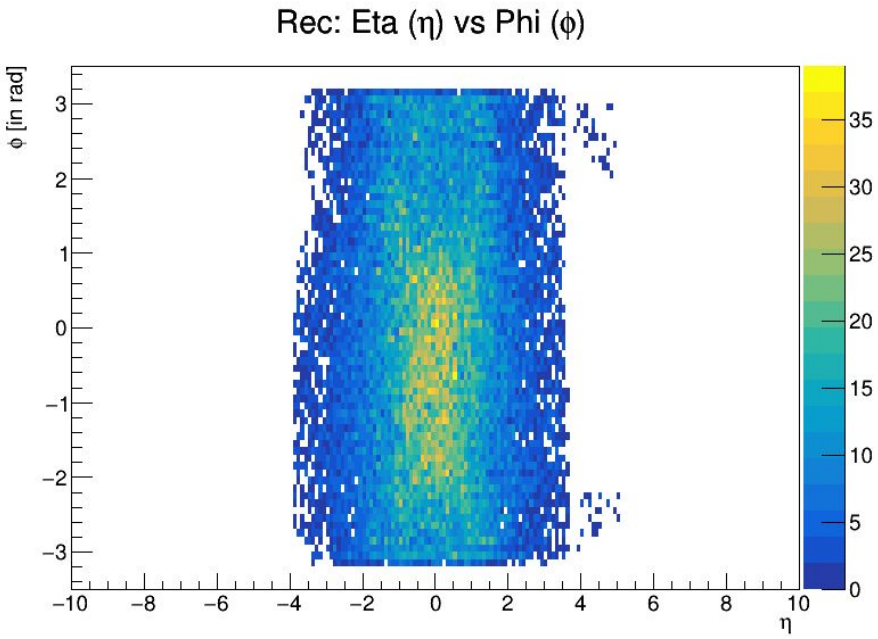
- Changed seed coordinates in truth tracking form vertex position to PCA position which is calculated by a straight line approximation.
- Initial tests show improvement in the truth seeded tracking performance for tracks generated off (x,y) = (0,0) i.e. z axis.
- Will continue further on this and detailed analysis of vertexing algorithm for off axis tracks with this fix is yet to be done.

# Backup Slide(s)

# Analysis: Reconstructed Particles Distributions

Vertex: (3,4,5) mm

default




changed