Tracking Resolutions via Residual and Covariance Methods

Matt Posik
Temple University

Outline

- ❑ Extracting angular resolutions
  - ➢ Residual Method
  - ➢ Covariance Matrix
- ❑ Summary

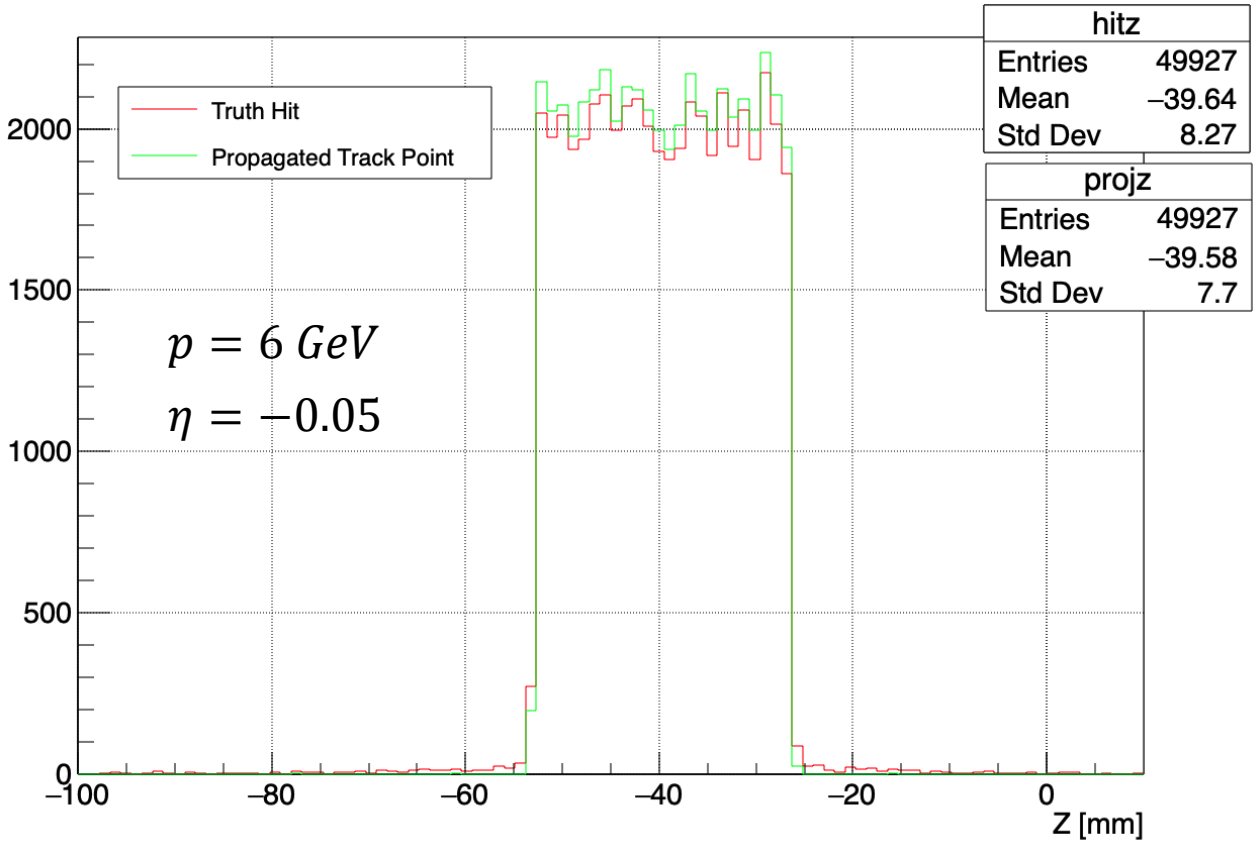
Extracting Angular Resolutions: Residual Approach

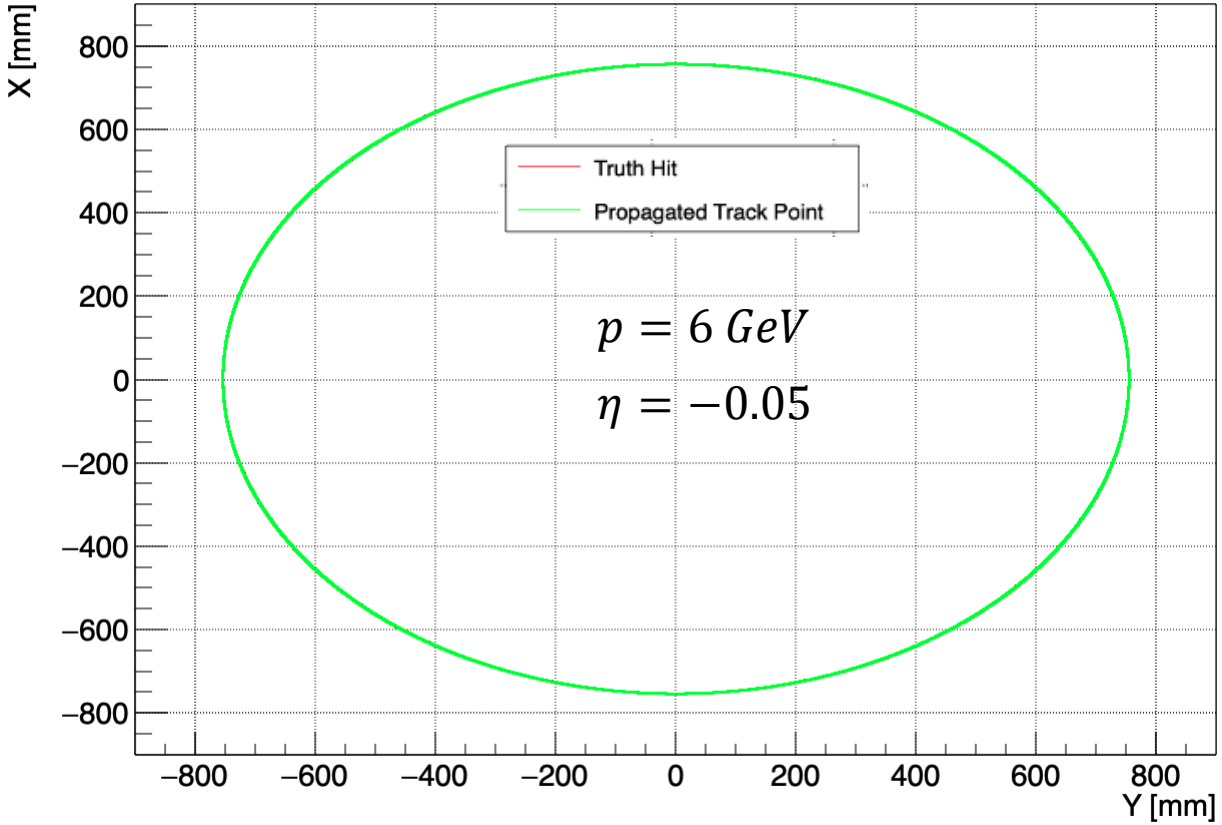
- Define low mass DIRC reference surface in DD4HEP to record truth hit information

  ePIC Branch, main code file

- Define ACTS surface that matches D44HEP DIRC reference surface to propagate trajectories to

  EICrecon Branch, main code files





Extracting Angular Resolutions: Residual Approach

- Loop over trajectories and propagate them to the ACTS surface

Trajectory loop snippet

```
auto trajectories = event->Get<ActsExamples::Trajectories>("CentralCKFTruthSeededActsTrajectories");

// Iterate over trajectories
m_log->debug("Propagating through {} trajectories", trajectories.size());
for (size_t traj_index = 0; traj_index < trajectories.size(); traj_index++) {
    auto &trajectory = trajectories[traj_index];
    m_log->trace(" -- trajectory {} --", traj_index);

    std::unique_ptr<edm4eic::TrackPoint> proj_DIRC_point;
    try {
        // >>> try to propagate to surface <<<
        proj_DIRC_point = m_propagation_algo.propagate(edm4eic::Track{}, trajectory, m_dirc_surf);
    }
    catch(std::exception &e) {
        throw JException(e.what());
    }

    if(!proj_DIRC_point) {
        m_log->trace("   could not propagate!", traj_index);
        std::cout<<" could not propagate! traj_index " << traj_index << std::endl;
        continue;
    }
```

Propagate trajectories to ACTS surface

Code Structure

Trajectory Loop {

Sim Reference Hit Loop {

Keep trajectory that is closest to ref hit*

...}

... }

Propagated track point information

```
auto DIRC_proj_pos   = proj_DIRC_point->position;
auto DIRC_proj_len   = proj_DIRC_point->pathlength;
auto DIRC_proj_mom   = proj_DIRC_point->momentum;
auto DIRC_proj_theta = proj_DIRC_point->theta;
auto DIRC_proj_phi   = proj_DIRC_point->phi;
auto DIRC_proj_theta_theta_error = proj_DIRC_point->directionError.xx;
auto DIRC_proj_phi_phi_error     = proj_DIRC_point->directionError.yy;
auto DIRC_proj_theta_phi_error   = proj_DIRC_point->directionError.xy;
TVector3 proj_pos_vector(DIRC_proj_pos.x, DIRC_proj_pos.y, DIRC_proj_pos.z);
TVector3 proj_mom_vector(DIRC_proj_mom.x, DIRC_proj_mom.y, DIRC_proj_mom.z);
```

$$^{*}\ \Delta R = \sqrt{\left(x_{prop} - x_{ref}\right)^2 + \left(y_{prop} - y_{ref}\right)^2 + \left(z_{prop} - z_{ref}\right)^2}$$

Extracting Angular Resolutions: Residual Approach

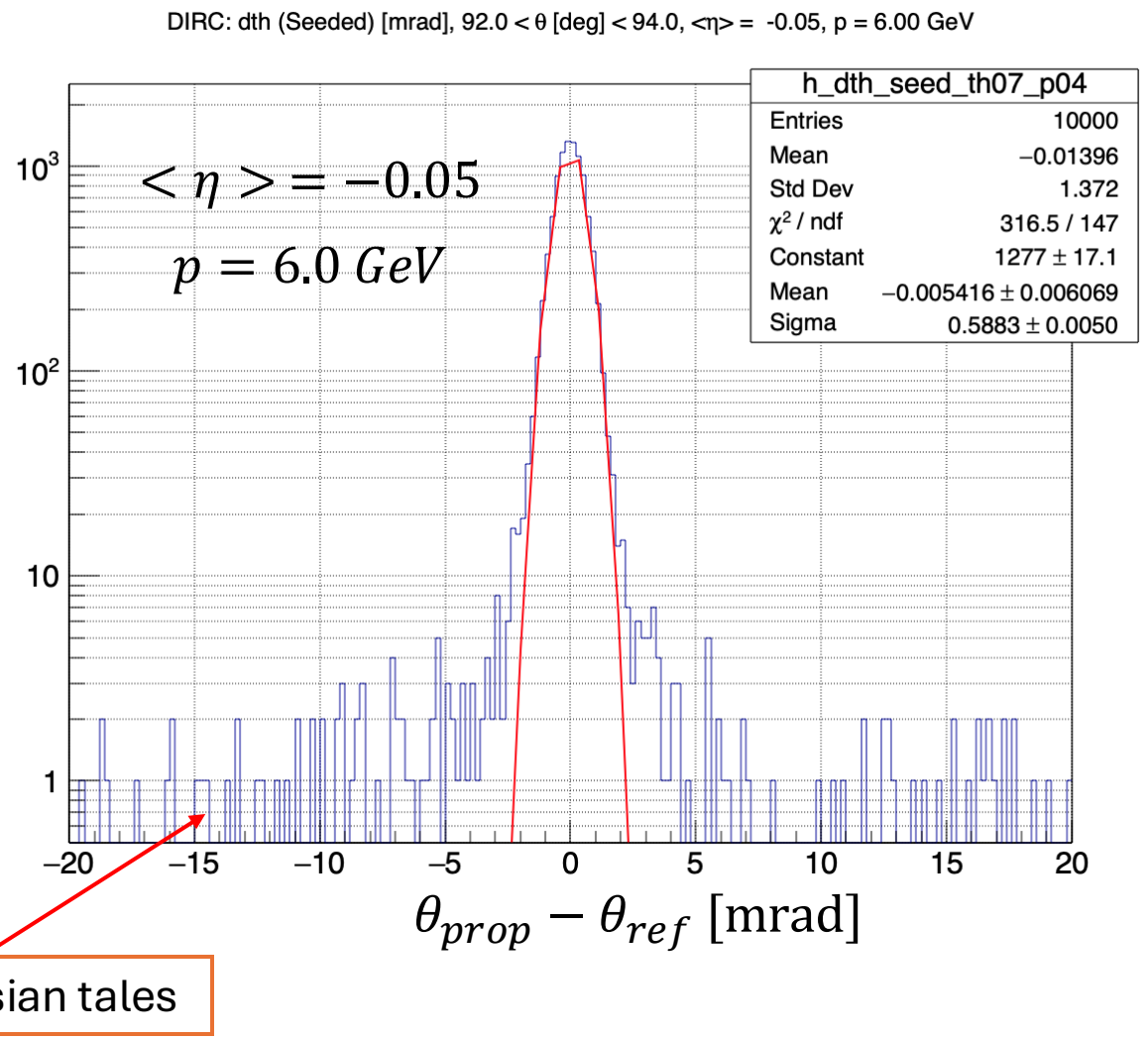
❑ Angles determined from reference and propagated trajectory hit via TVector

$\theta_{ref} = (x_{ref}, y_{ref}, z_{ref})$.Theta()
$\phi_{ref} = (x_{ref}, y_{ref}, z_{ref})$.Phi()

$\theta_{prop} = (x_{prop}, y_{prop}, z_{prop})$.Theta()
$\phi_{prop} = (x_{prop}, y_{prop}, z_{prop})$.Phi()

$\Delta\theta = \theta_{prop} - \theta_{ref}$

$\Delta\phi = \phi_{prop} - \phi_{ref}$



❑ Take difference of truth hit and propagated track point

- Resolution is given by Gaussian sigma
  - ➢ $\sigma_\theta = 0.59$ mrad

Non-Gaussian tales

Extracting Angular Resolutions: Covariance Matrix Approach

- Using the tracking covariance matrix is an alternative approach to using the residual approach
- No need to implement low mass DIRC reference surface in DD4HEP to record truth hit information
- Angular resolutions extracted from edm4eic::TrackPoint::directionError

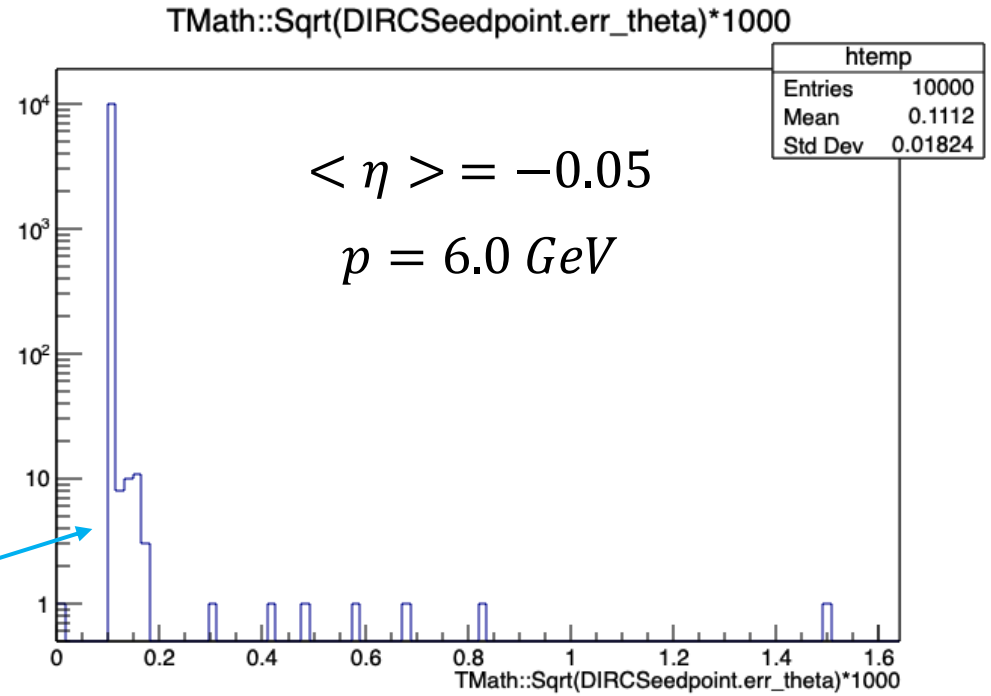
Covariance Matrix

```
// Direction
const float theta(parameter[Acts::eBoundTheta]);
const float phi(parameter[Acts::eBoundPhi]);
const decltype(edm4eic::TrackPoint::directionError) directionError{
        static_cast<float>(covariance(Acts::eBoundTheta, Acts::eBoundTheta)),
        static_cast<float>(covariance(Acts::eBoundPhi, Acts::eBoundPhi)),
        static_cast<float>(covariance(Acts::eBoundTheta, Acts::eBoundPhi))
};
```

TrackPropagation.cc

```
auto DIRC_proj_pos   = proj_DIRC_point->position;
auto DIRC_proj_len   = proj_DIRC_point->pathlength;
auto DIRC_proj_mom   = proj_DIRC_point->momentum;
auto DIRC_proj_theta = proj_DIRC_point->theta;
auto DIRC_proj_phi   = proj_DIRC_point->phi;
auto DIRC_proj_theta_theta_error = proj_DIRC_point->directionError.xx;
auto DIRC_proj_phi_phi_error     = proj_DIRC_point->directionError.yy;
auto DIRC_proj_theta_phi_error   = proj_DIRC_point->directionError.xy;
TVector3 proj_pos_vector(DIRC_proj_pos.x, DIRC_proj_pos.y, DIRC_proj_pos.z);
TVector3 proj_mom_vector(DIRC_proj_mom.x, DIRC_proj_mom.y, DIRC_proj_mom.z);
```

Propagated track point information

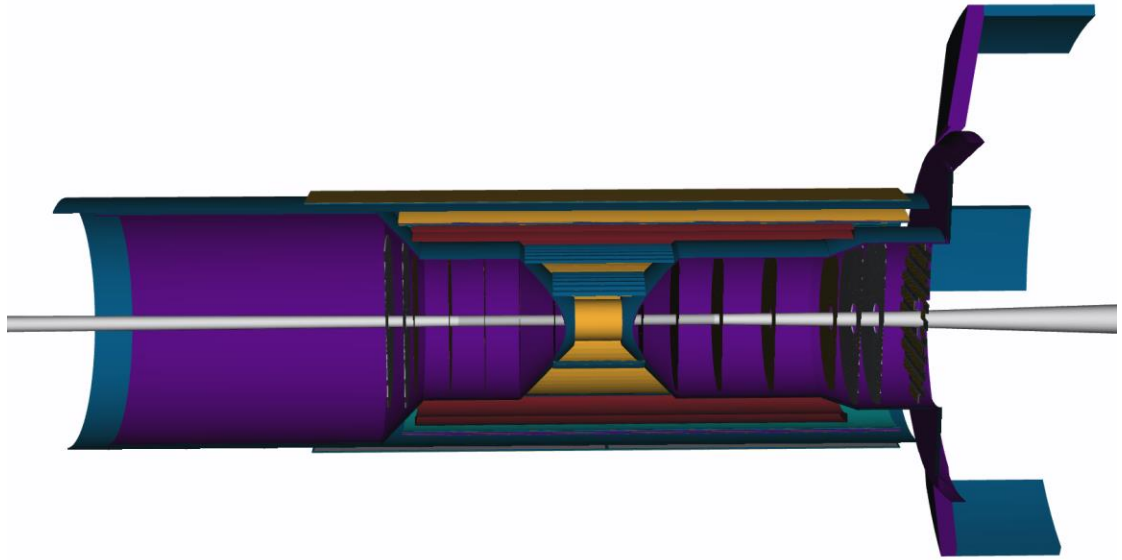


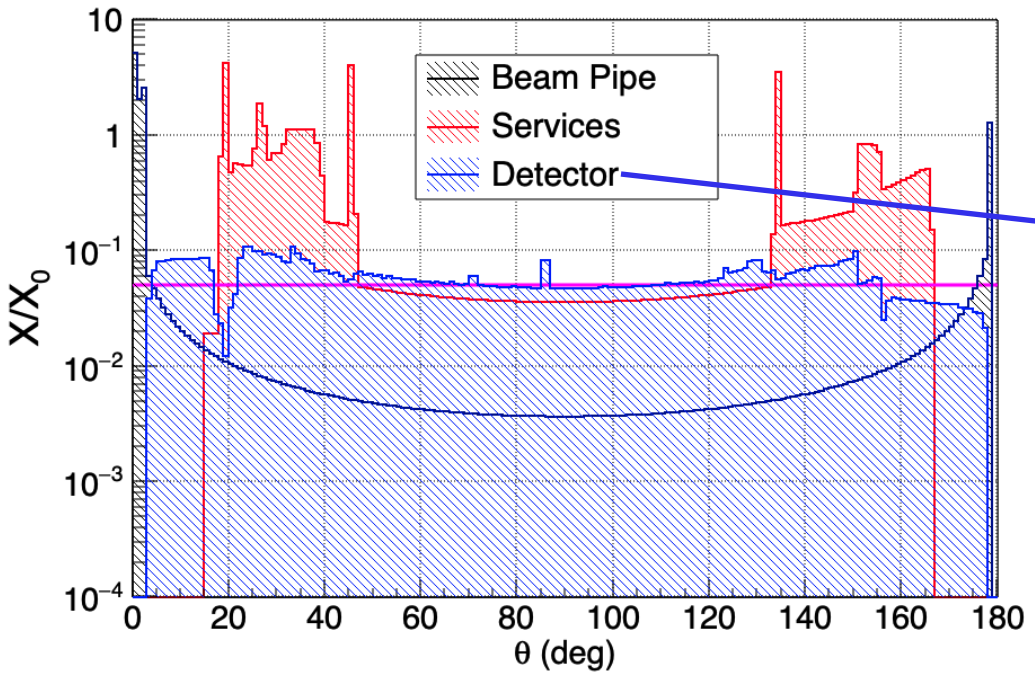
$\sqrt{\sigma_\theta^2}$ [mrad]

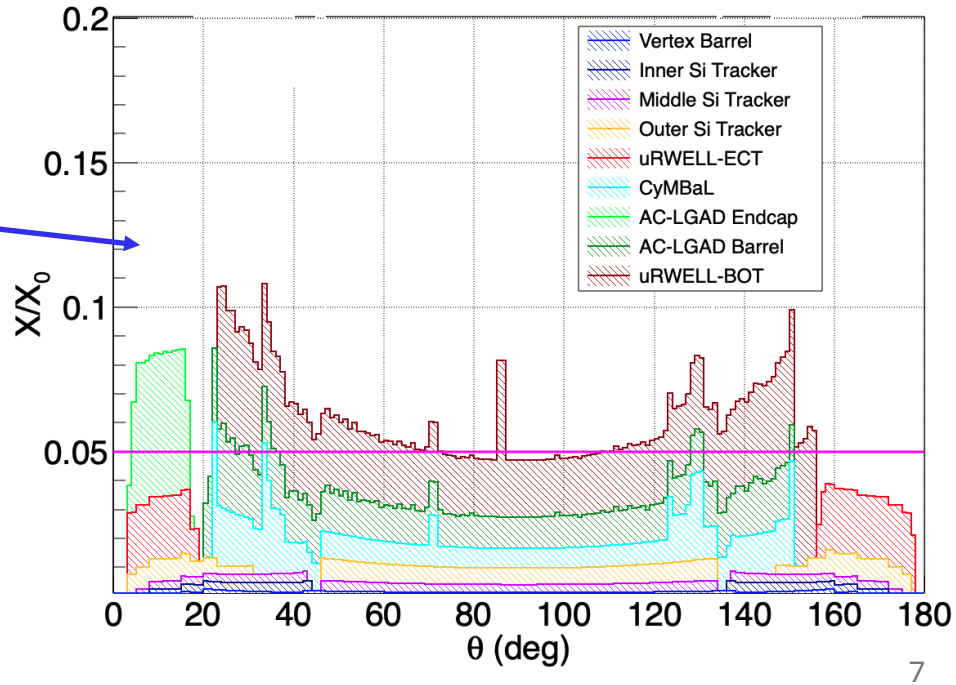
- Resolution is mean of histogram
  - $\sigma_\theta^{cov} = < \sqrt{\sigma_\theta^2} >$ = 0.11 mrad

Simulation Details

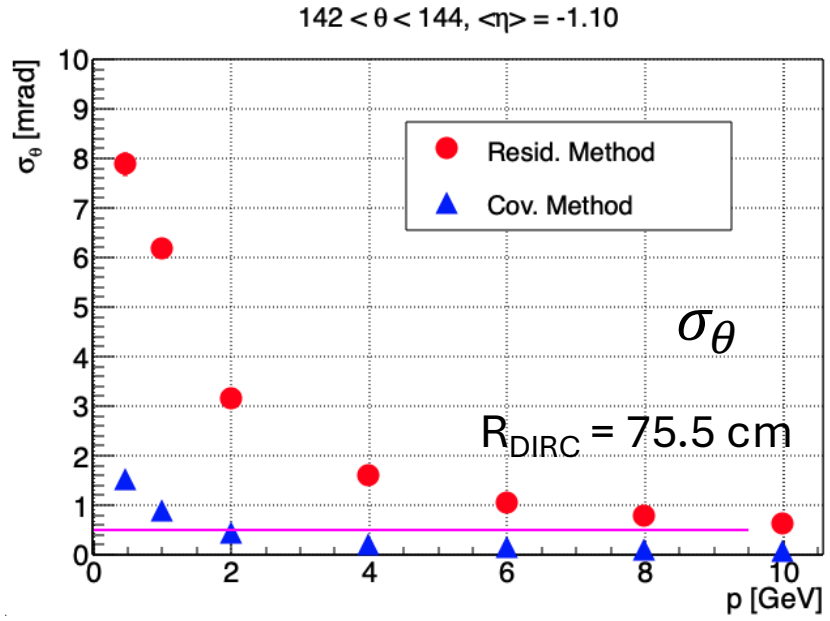
- ❑ ePIC: 24.11.1
- ❑ EICrecon: v1.19.0
- ❑ Single particle π^-
- ❑ Discrete momenta settings
- ❑ $0^o \leq \Delta\phi \leq 360^o$
- ❑ $\Delta\theta = 2^o$

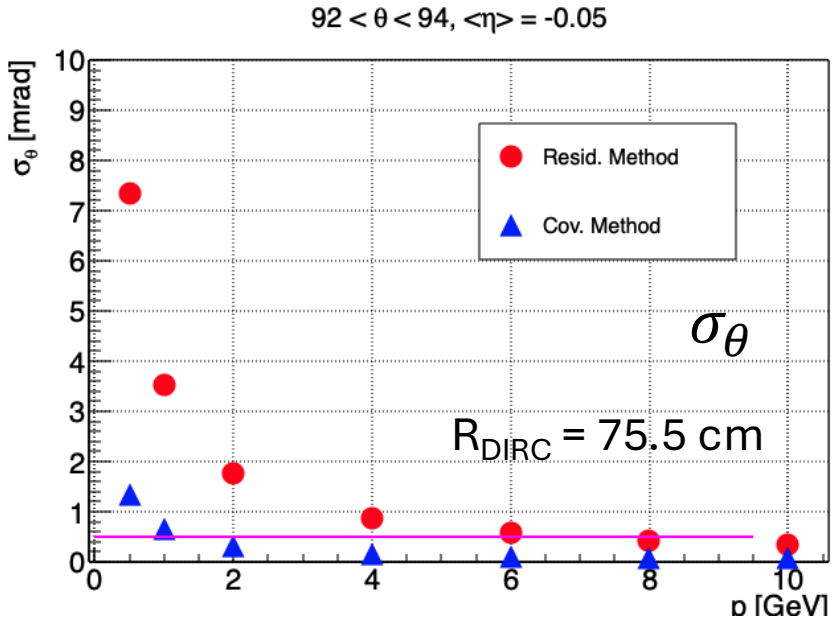


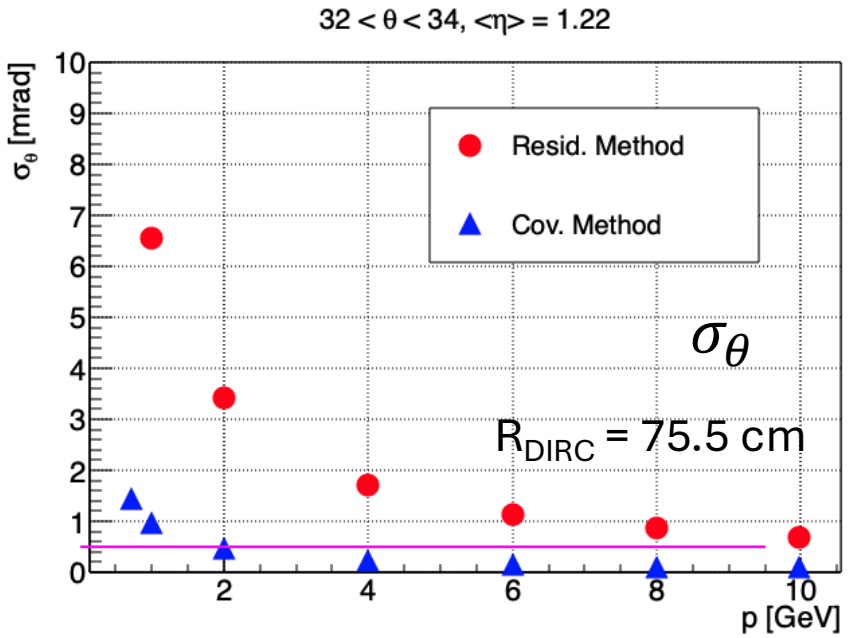


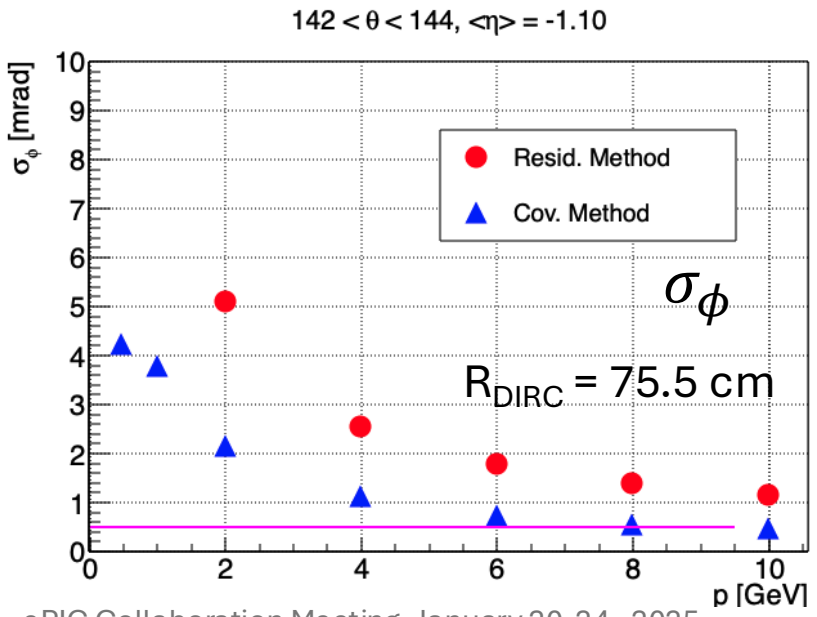


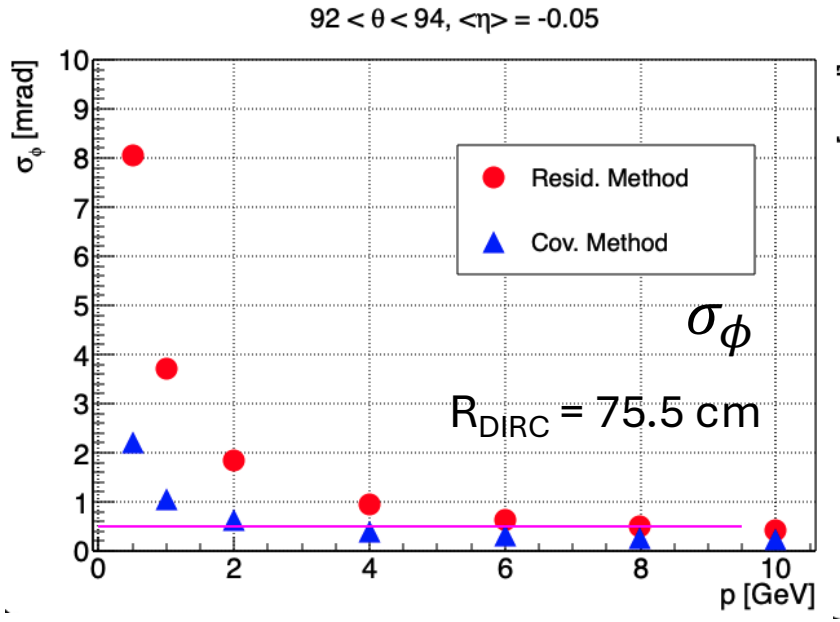
Angular Resolution Results

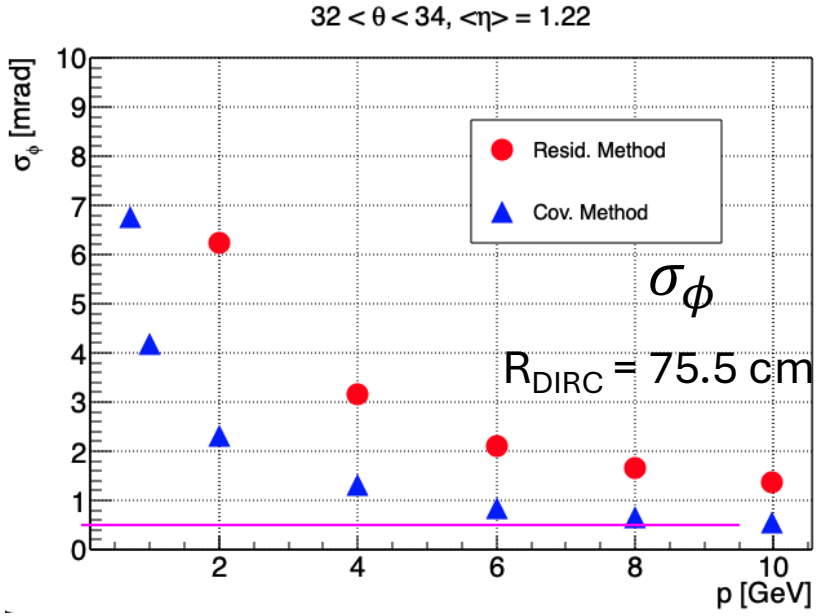












Summary

- Two methods were implemented to estimate the tracking angular resolution at the hpDIRC
  - Each method gives different results
- The Covariance matrix approach gives much better angular resolutions than the Residual method
  - Expected both methods to give similar results
- Generally, polar angle (θ) resolution is better than (ϕ) resolution