# Studies of Residual and Pull Distributions

Pelham Keahey

SLAC SULI-08

# Program Purpose

- Develop analysis software to analyze track finding performance, this includes:
- Track finding efficiency
- Fake track rates
- Reasonable error estimation
- Optimization

# Helix Parameters

- Curvature (C, Omega) curvature of the track
- $\Phi_0$ (azimuthmal angle of the momentum at the dca)
- $\delta$ (distance of closest approach)
- $\tan \lambda$ (dz/ds, slope in the SZ plane)
- $z_0$ (z position at the dca)

# Calculations

- ω

$$\rangle_\tau = \rangle * \sigma\iota\nu\lceil = .0003 * B * P$$
$$\rceil = 1/P$$

- δ

$$x_c = x + R \sin \phi$$
$$y_c = y - R \cos \phi$$
$$\delta = R - (x_c^2 + y_c^2)^{1/2}$$

- $\Phi_0$

$$\text{atan2}(x_c/(R-\delta), -y_c/(R-\delta))$$

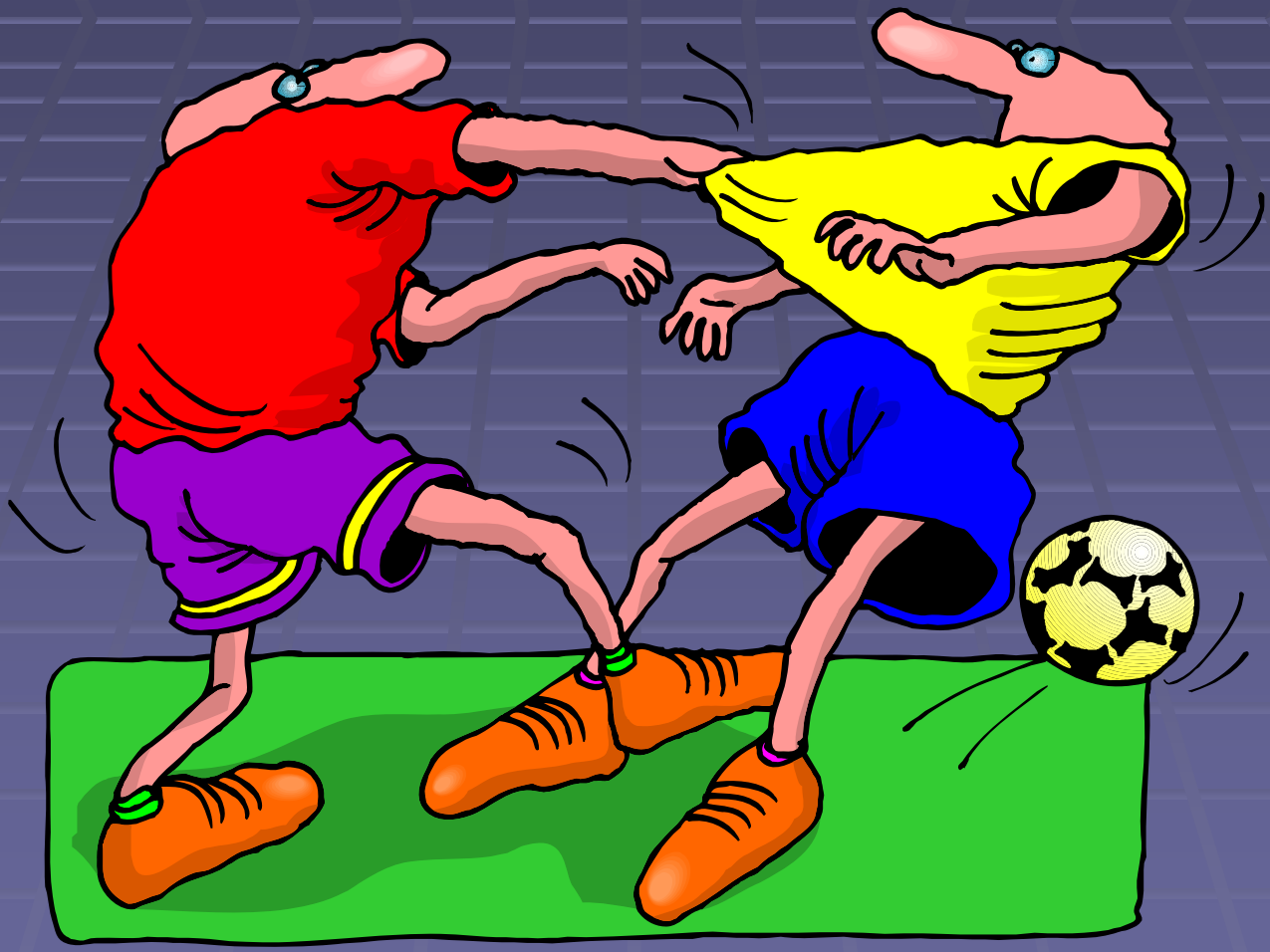
- tan λ

$$\tan \lambda = \rangle_z / (\rangle_x^2 + \rangle_y^2)^{1/2}$$

- z0

$$z = z_0 + s \tan\lambda$$

# MC particle vs HelicalTrackFit

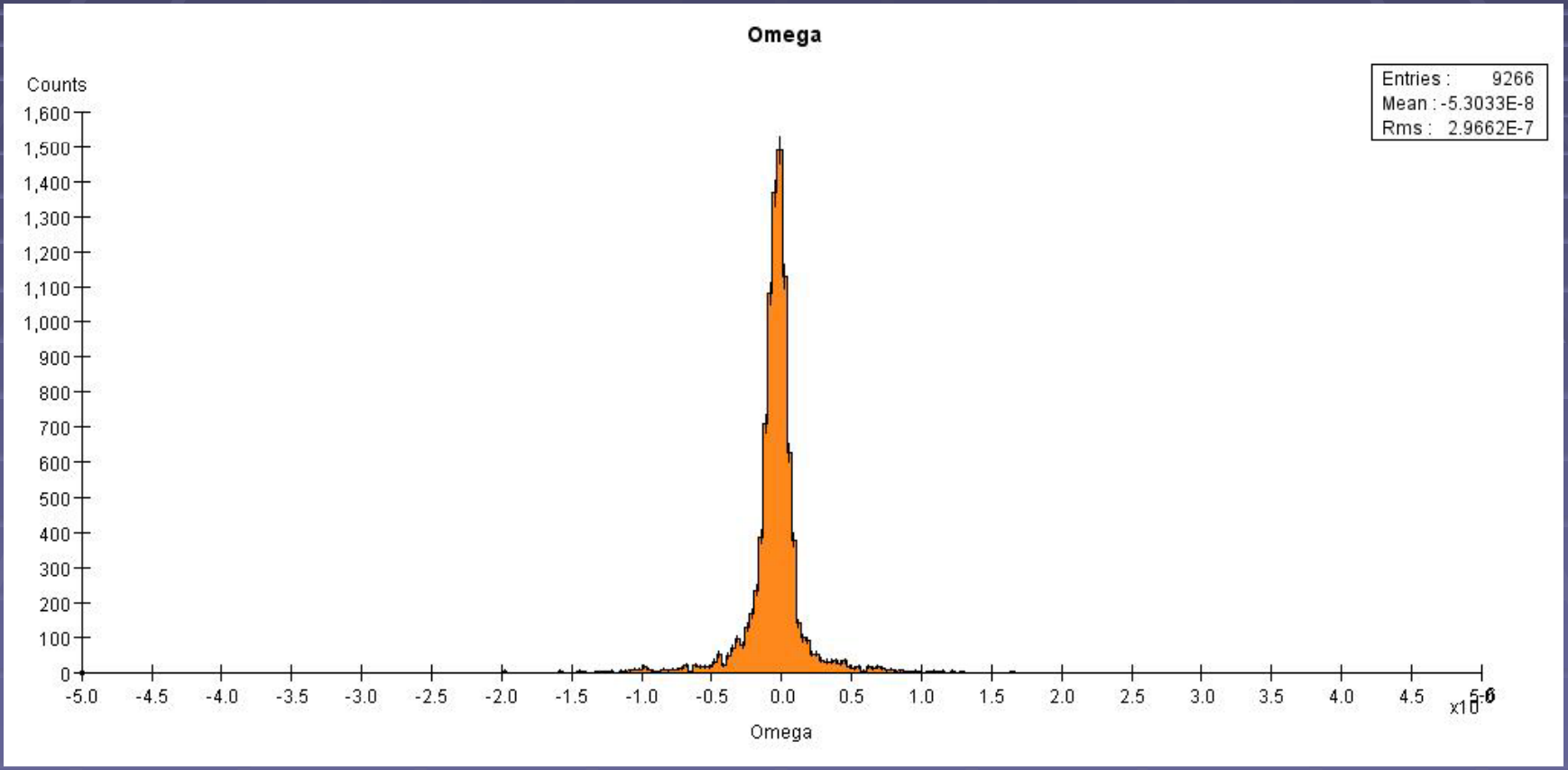


# More Numbers…

- For our residual graphs we plotted:
  - Fit - MC
- The Pull graphs where the same residual divided by the error retrieved from the fit method's covariance matrix

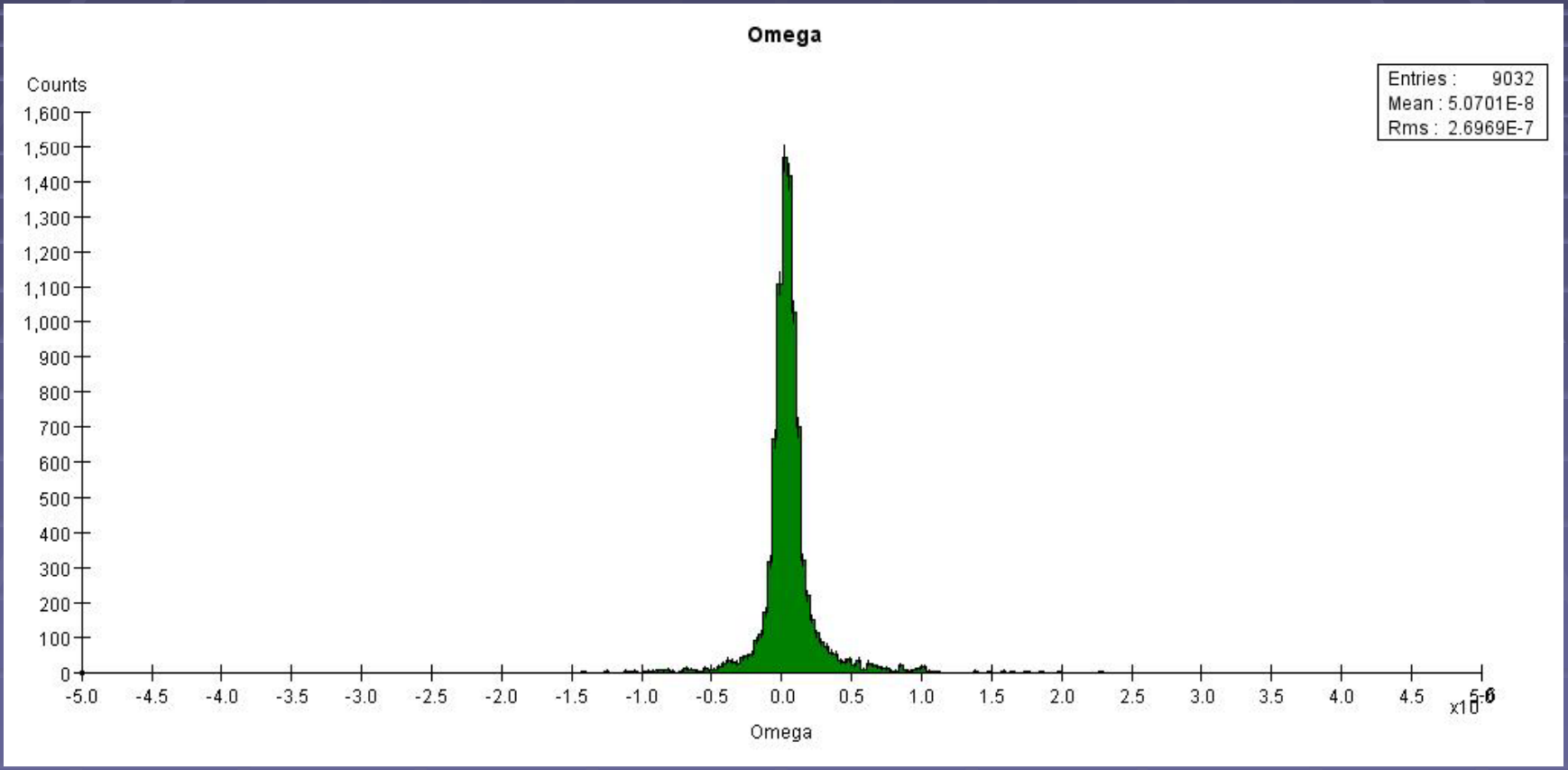
Math.sqrt(fit.covariance().e(HelicalTrackFit.curvatureIndex, HelicalTrackFit.curvatureIndex))

# (Residual) Omega-Pos



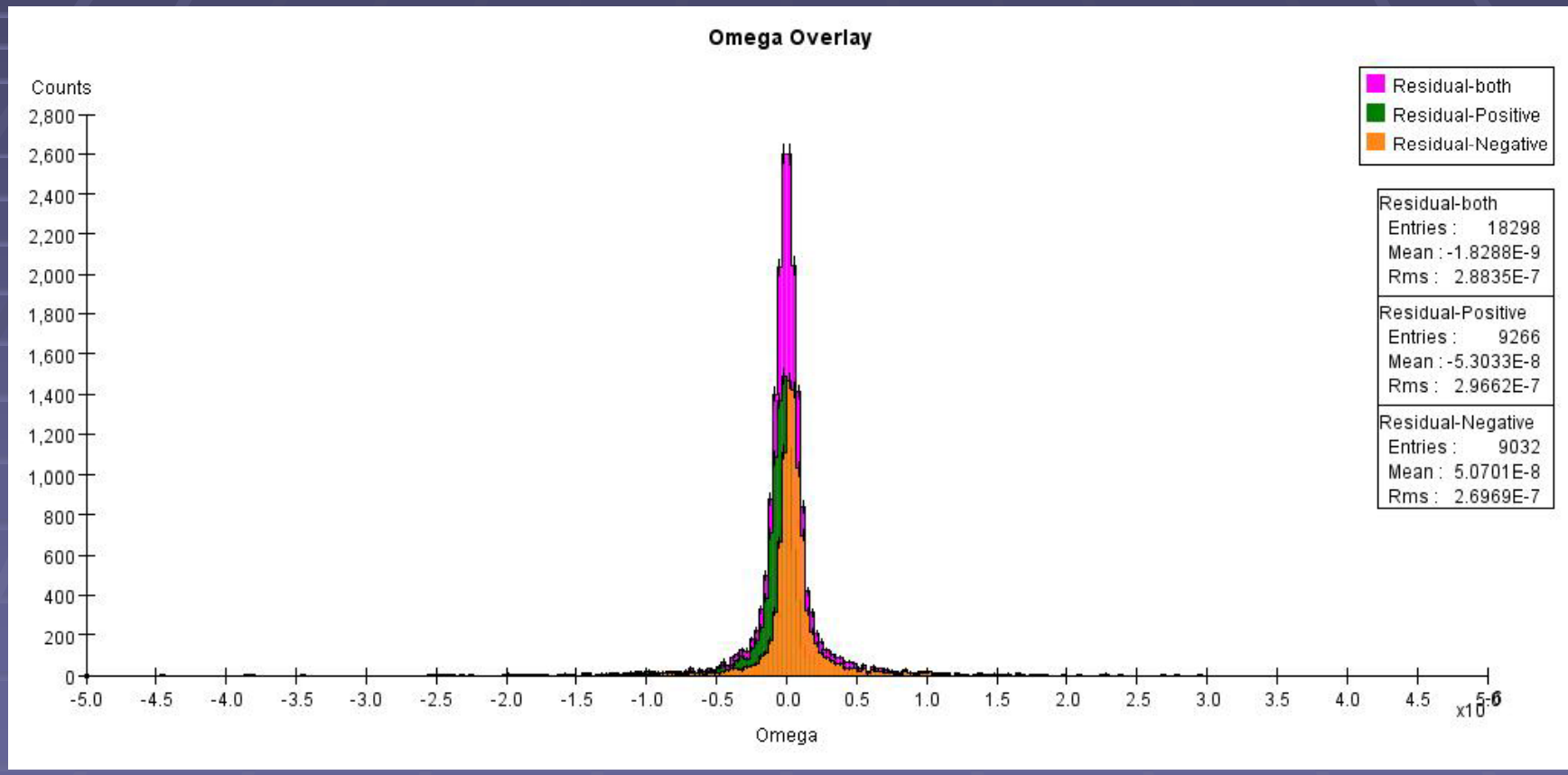

muon_Theta1-179_100GeV

# (Residual) Omega-Neg




muon_Theta1-179_100GeV

# (Residual) Omega-Overlay

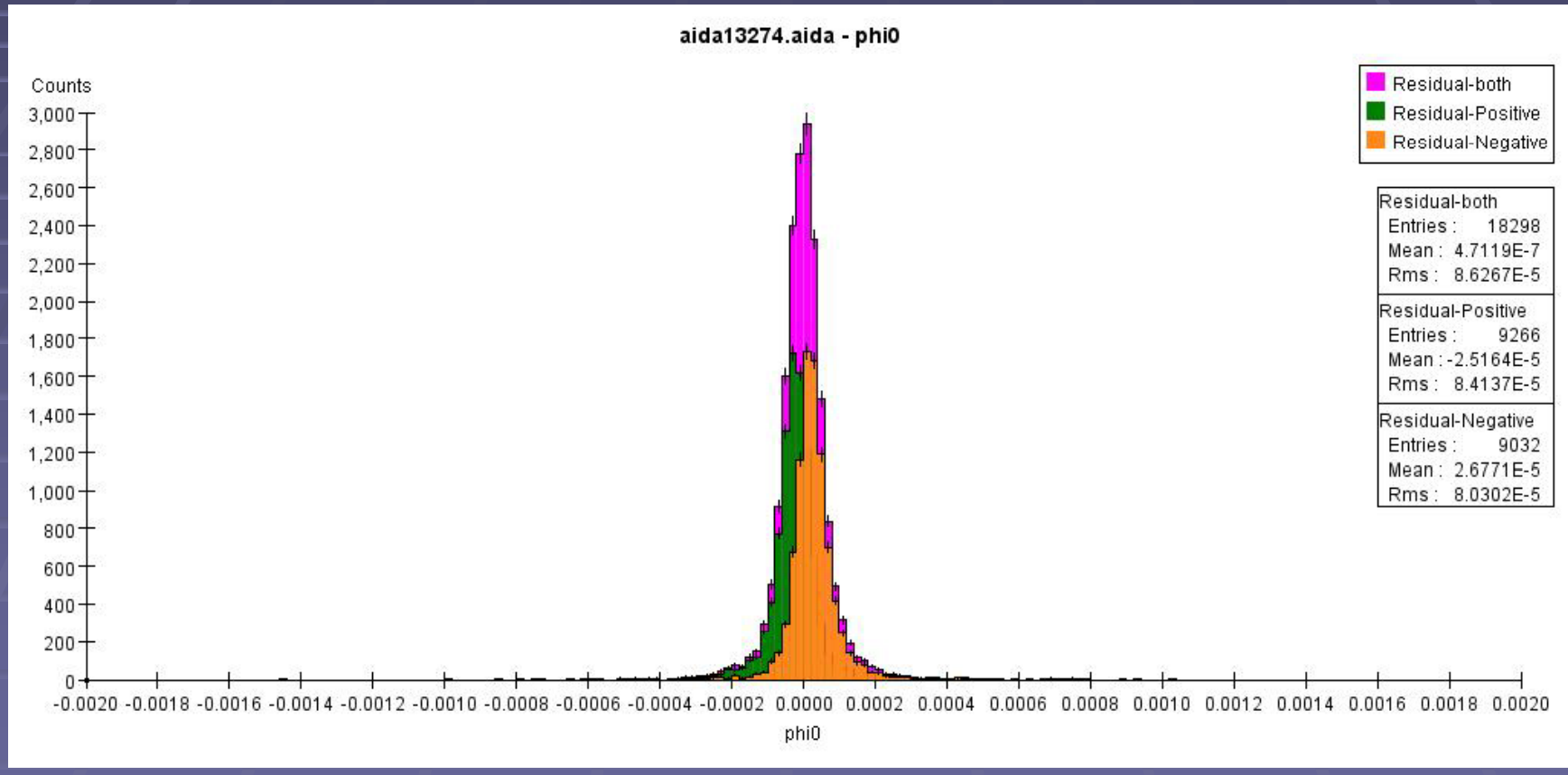



muon_Theta1-179_100GeV

# Graph +-charges seperately

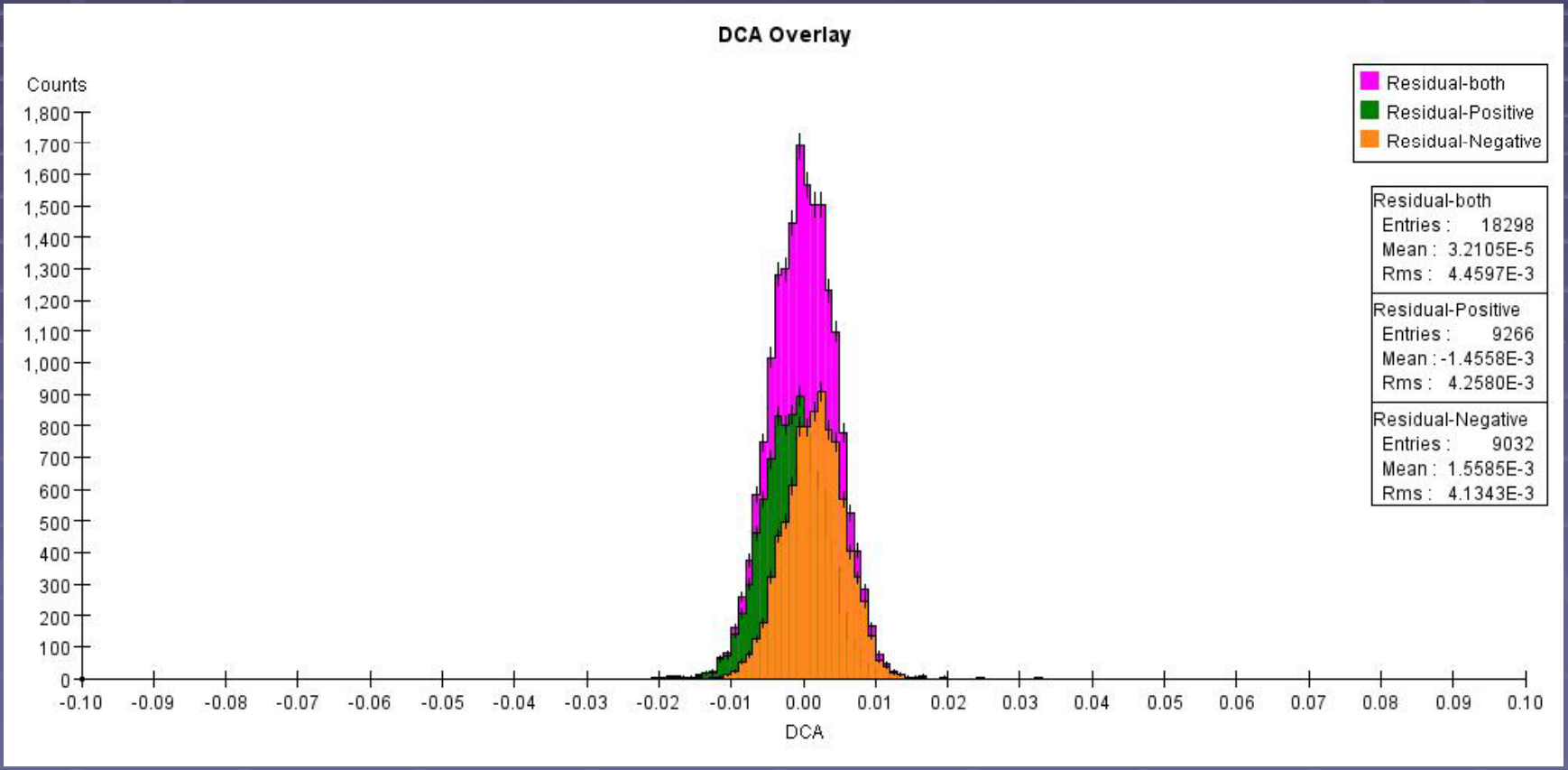
- The analysis driver is set to graph positive and negative particles separately, why?
  - There is a slight variance if you graph both on the same graph, refer to “Residual Omega Overlay”
- Analyzing them separately allows for greater accuracy
- Lower energy simulations show a greater energy loss of the particle dependent on charge, thus separating the analysis is key

# (Residual) $\Phi_0$ -Overlay




muon_Theta1-179_100GeV

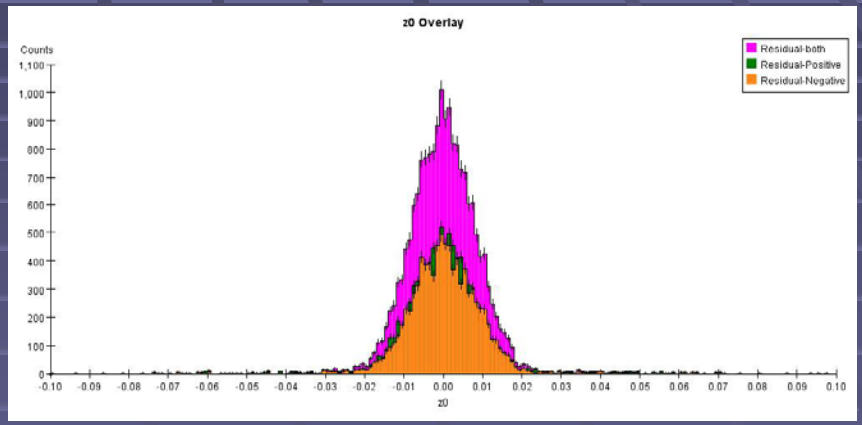
# (Residual) δ -Overlay

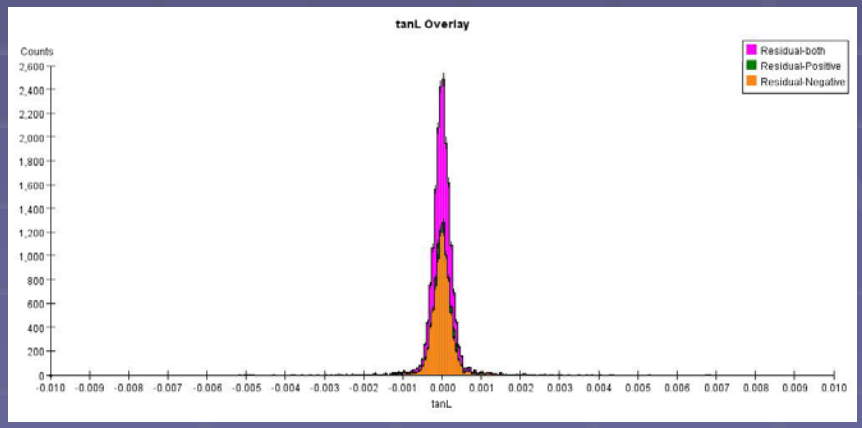



muon_Theta1-179_100GeV

# Charge sign and the parameters

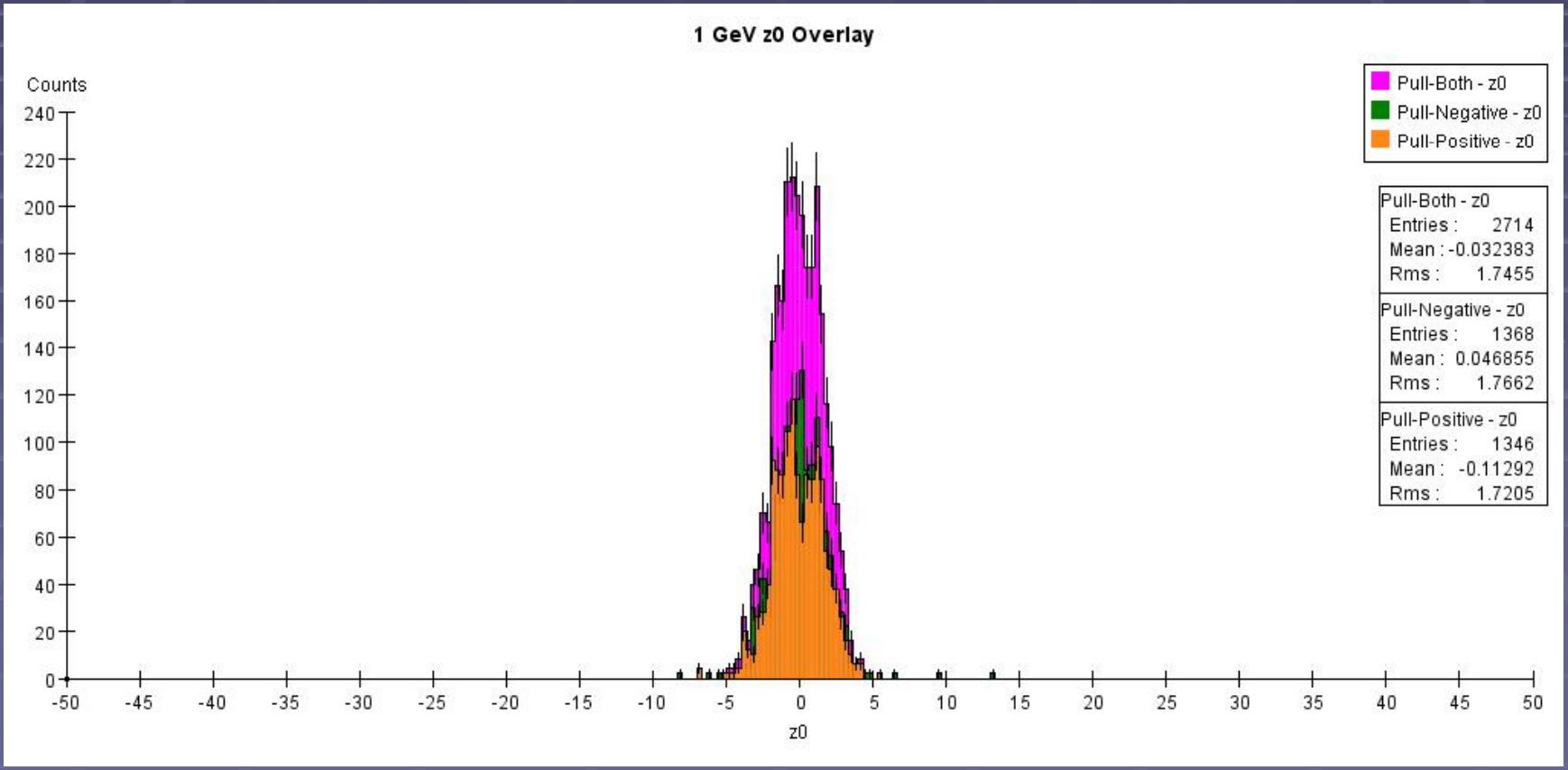
- tan λ, and $Z_0$ exhibit the same distribution regardless of charge, this remained the same for lower energy simulations as well



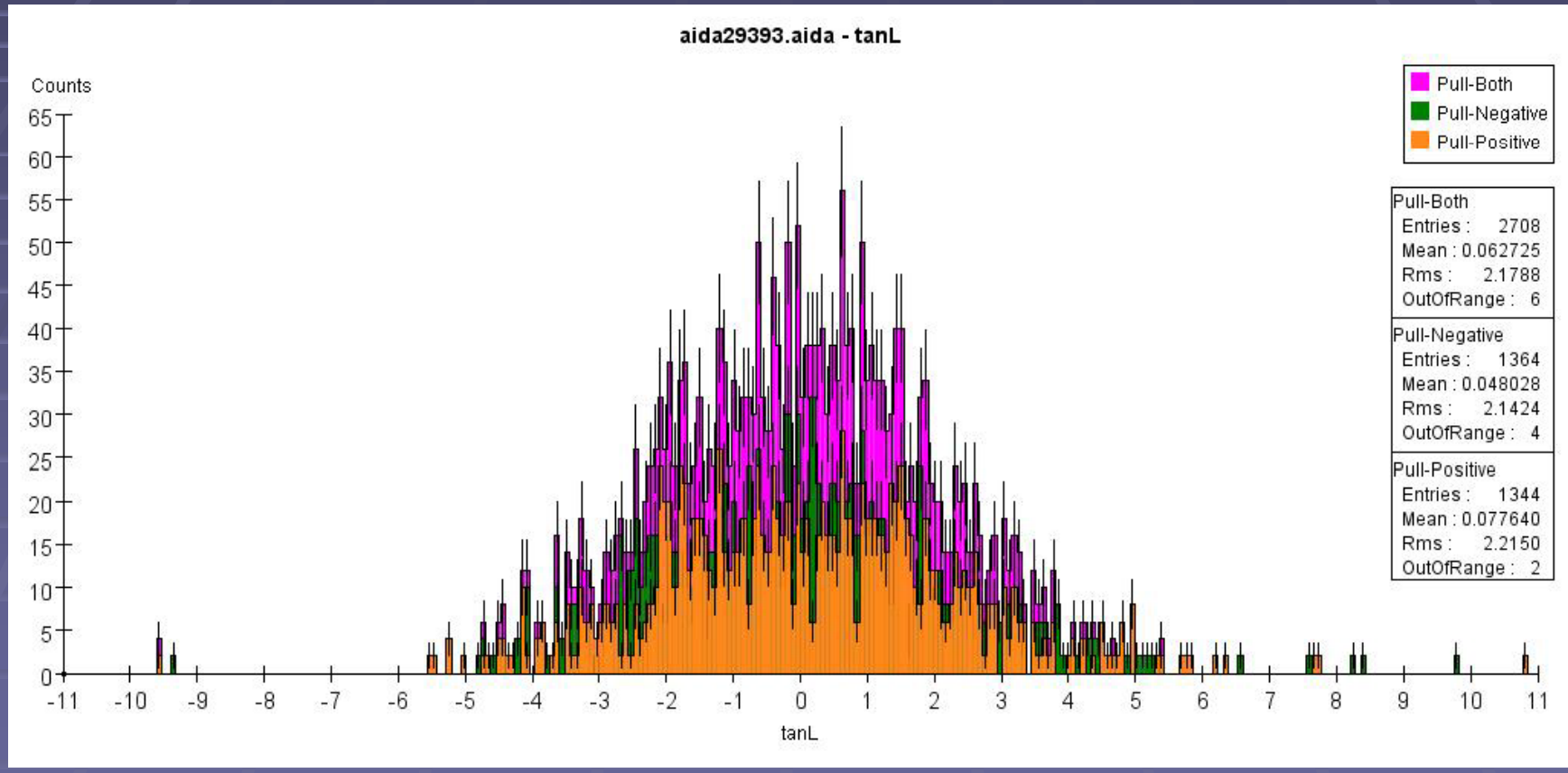

muon_Theta1-179_100GeV

# 1 GeV z0 Pull



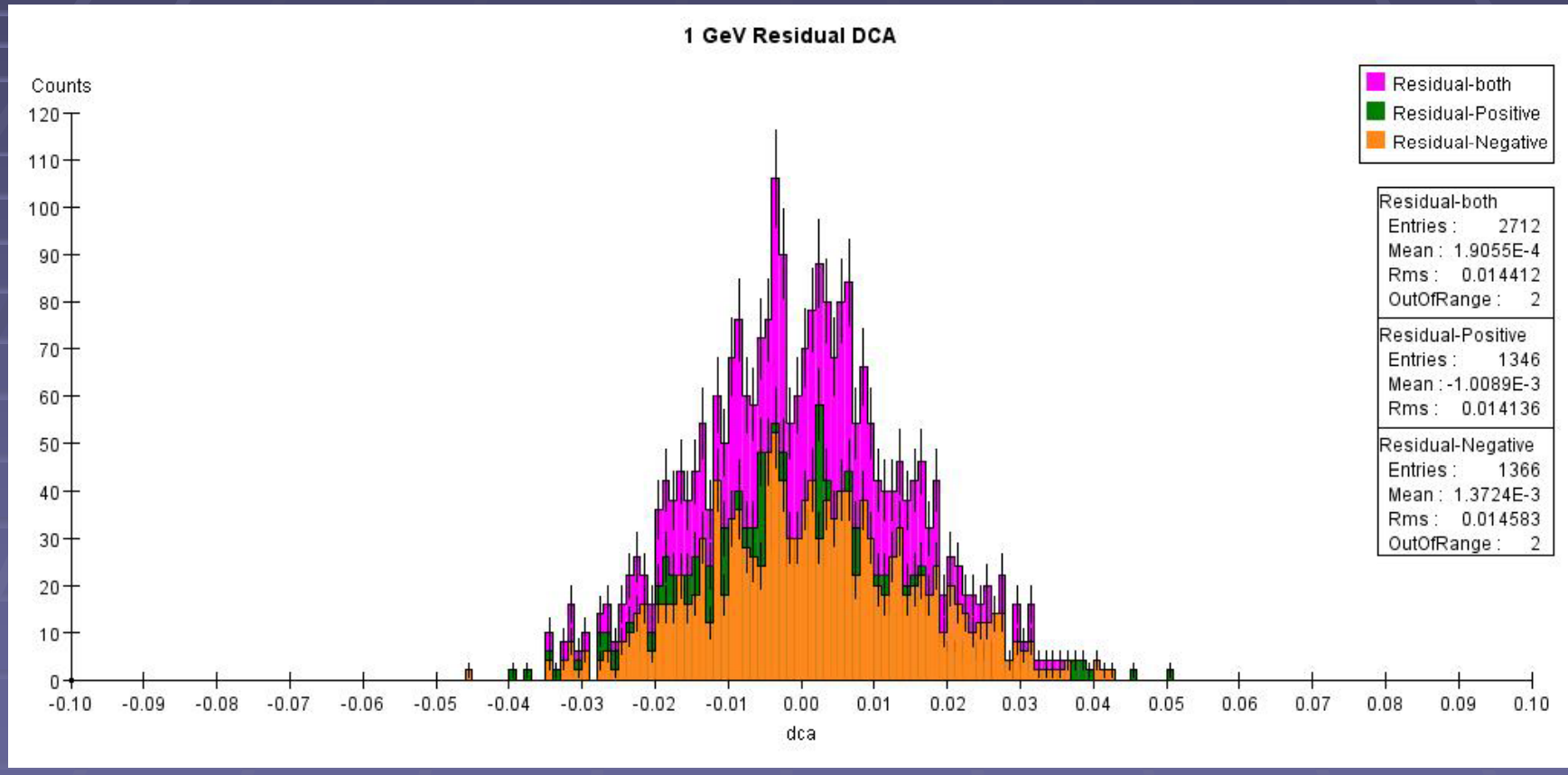

muon_Theta1-179_1GeV

# 1 GeV tan λ Pull




muon_Theta1-179_1GeV

# 1 GeV Residual DCA



# Fitted Helix compared to MC hits

- Comparing the r, Φ, z calculated by each method
- In this analysis set we have to account for each layer as they greatly differ in the value for each parameter.
- Layers: Tracker Barrel, Vertex Barrel, Endcaps

# Calculations

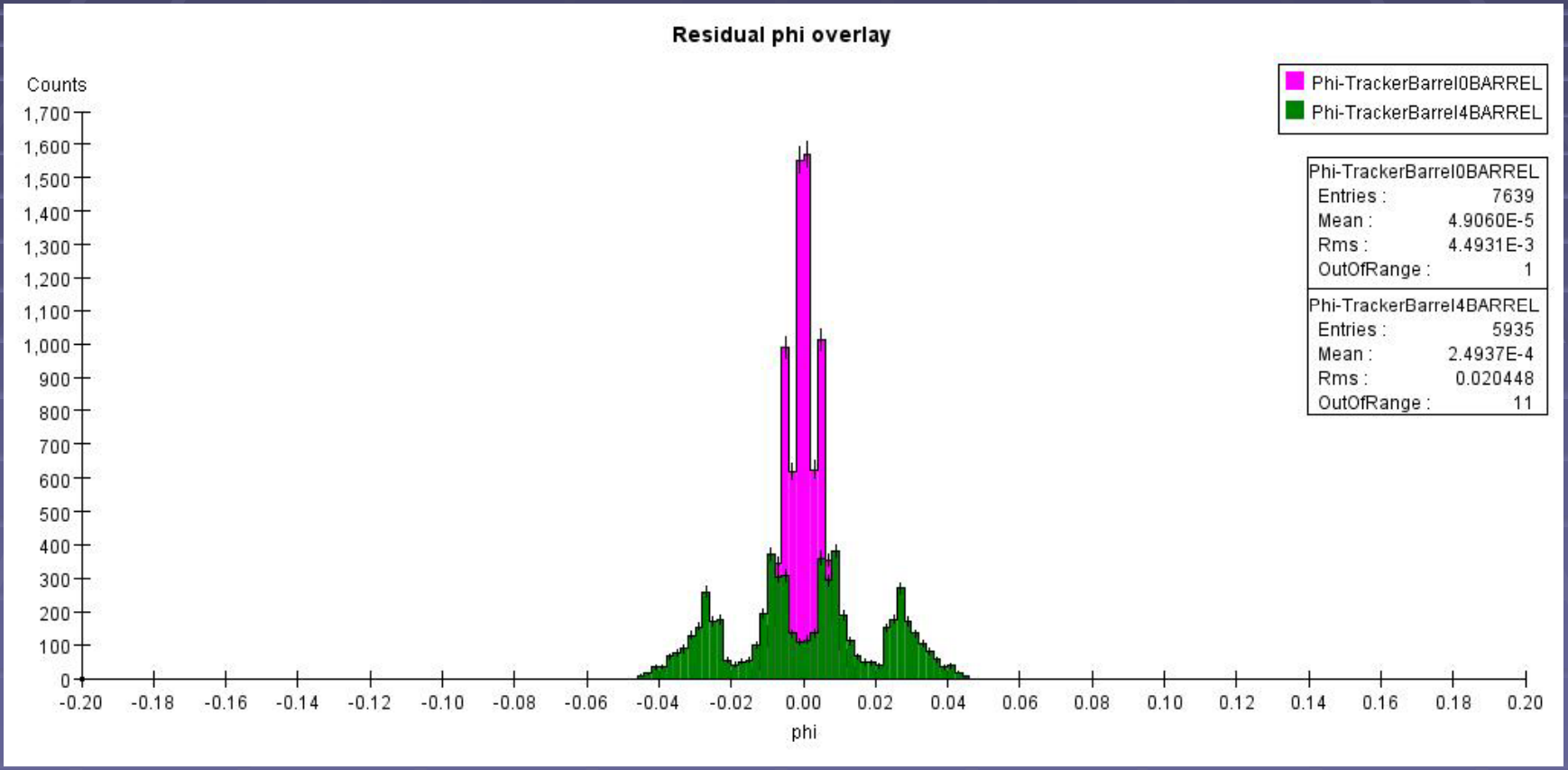
$$x = x_0 + s\ \mathbf{sinc}\left(\frac{Cs}{2}\right)\cos\left(\phi_0 - \frac{Cs}{2}\right)$$
$$y = y_0 + s\ \mathbf{sinc}\left(\frac{Cs}{2}\right)\sin\left(\phi_0 - \frac{Cs}{2}\right)$$

$$\phi = \textbf{\textit{atan2}}\left(\sin\phi_0 - \mathrm{C}\ (\mathrm{x} - \mathrm{x}_0), \cos\phi_0 + \mathrm{C}\ (\mathrm{y} - \mathrm{y}_0)\right)$$

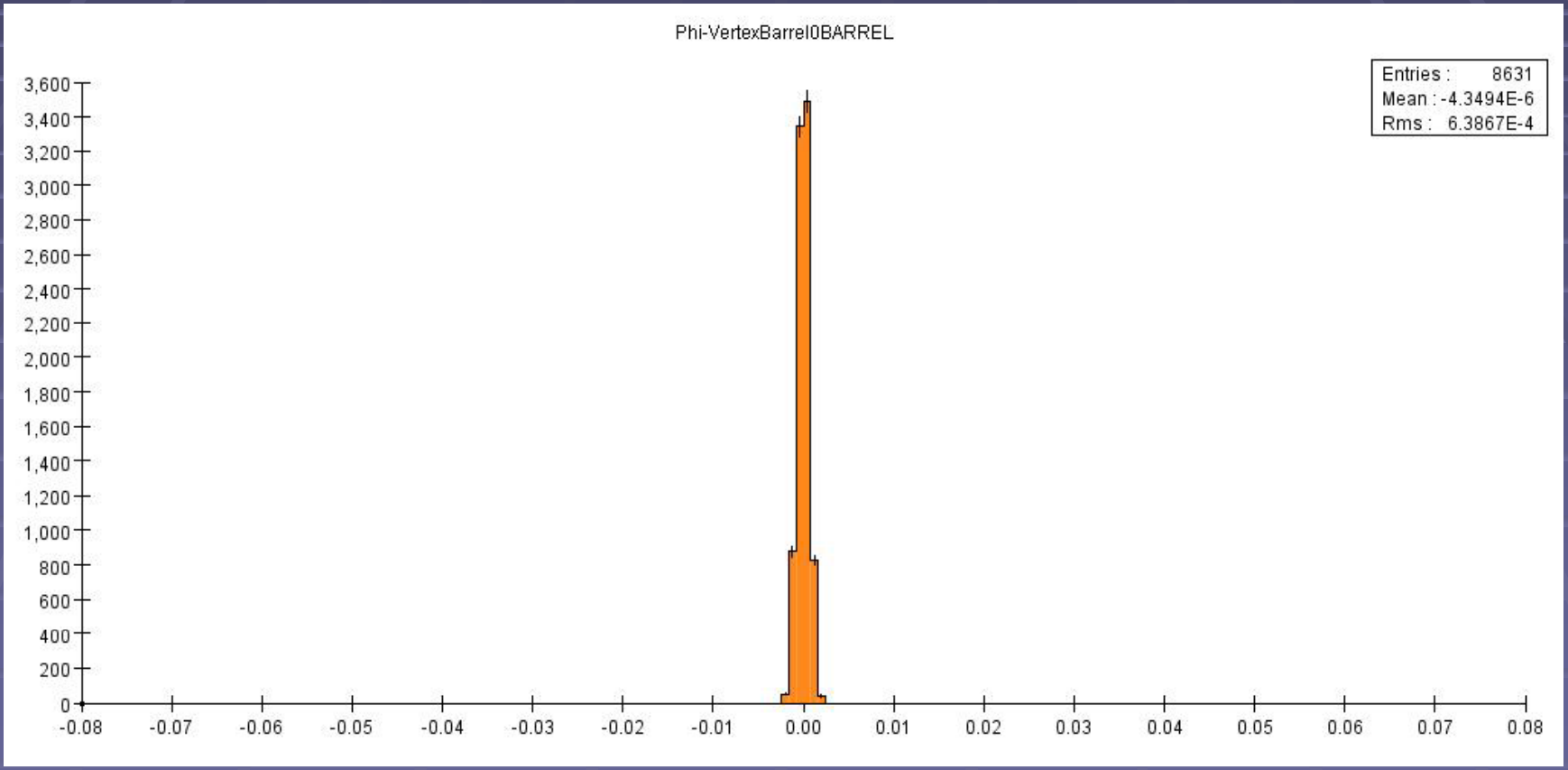
$$z = z_0 + s\ \tan\lambda$$

$$(x - x_c)^2 + (y - y_c)^2 = R^2$$

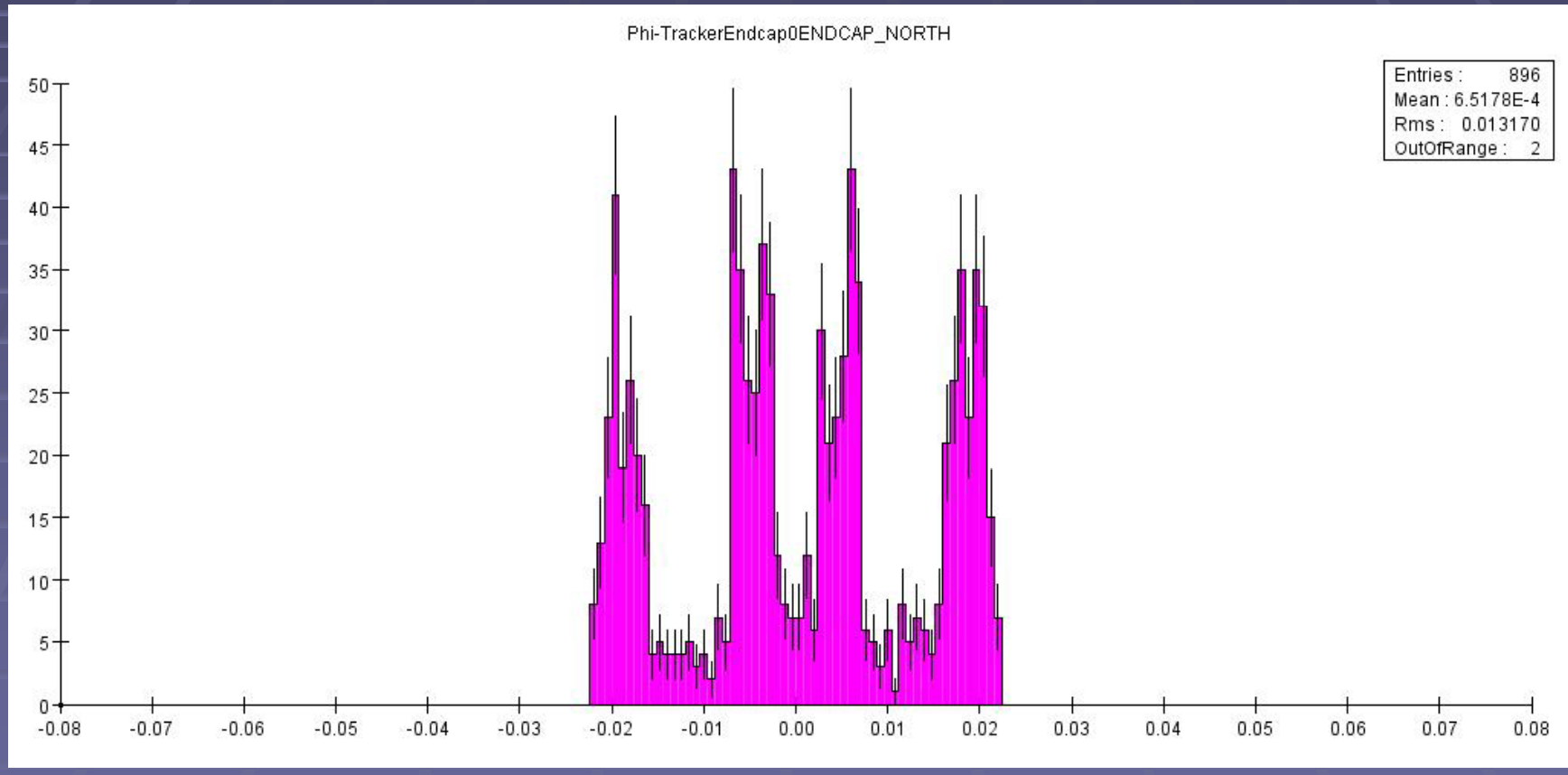
# Tracker Barrel - Φ



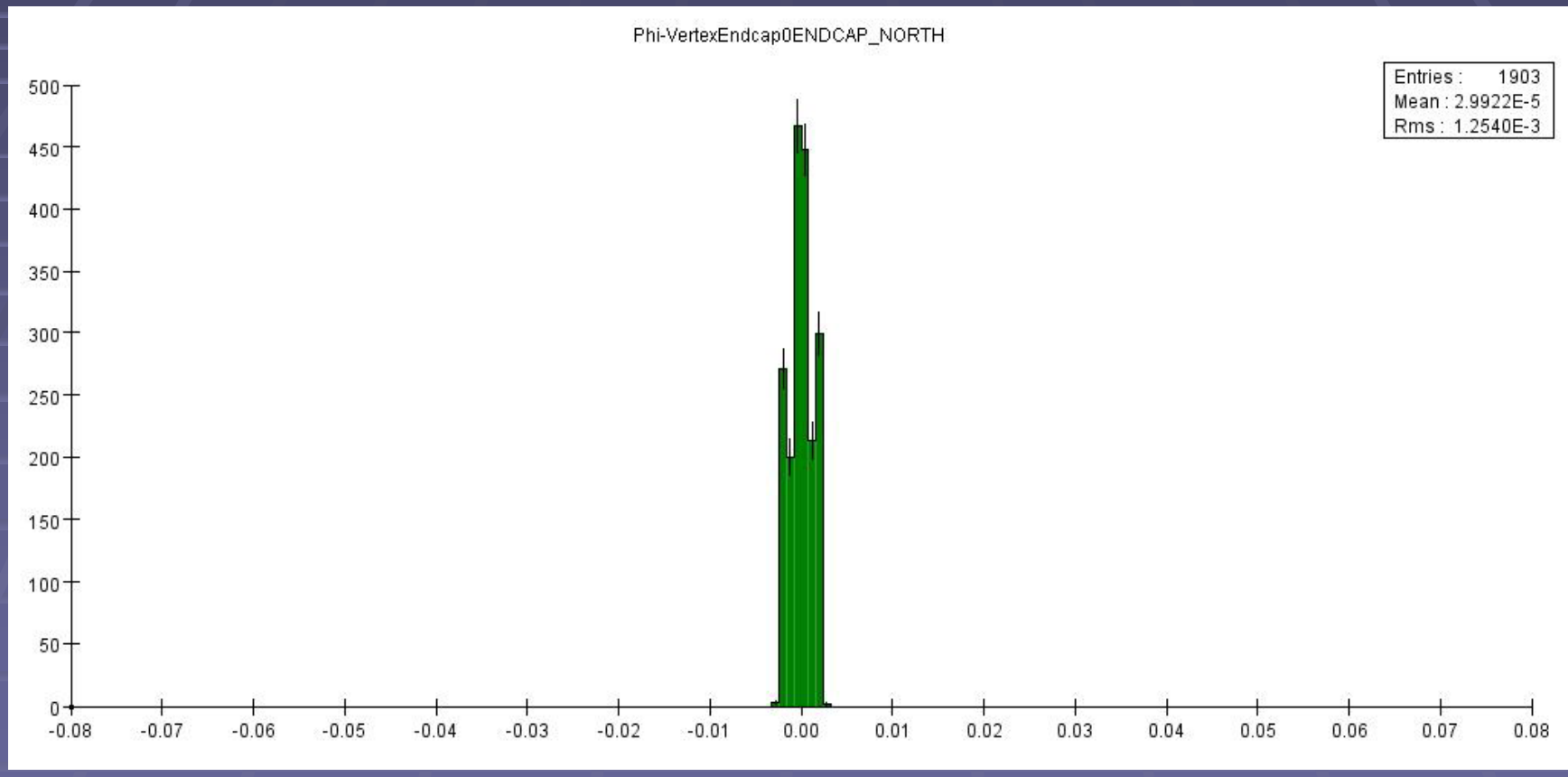
# Vertex Barrel - Φ



# Tracker Endcap - Φ



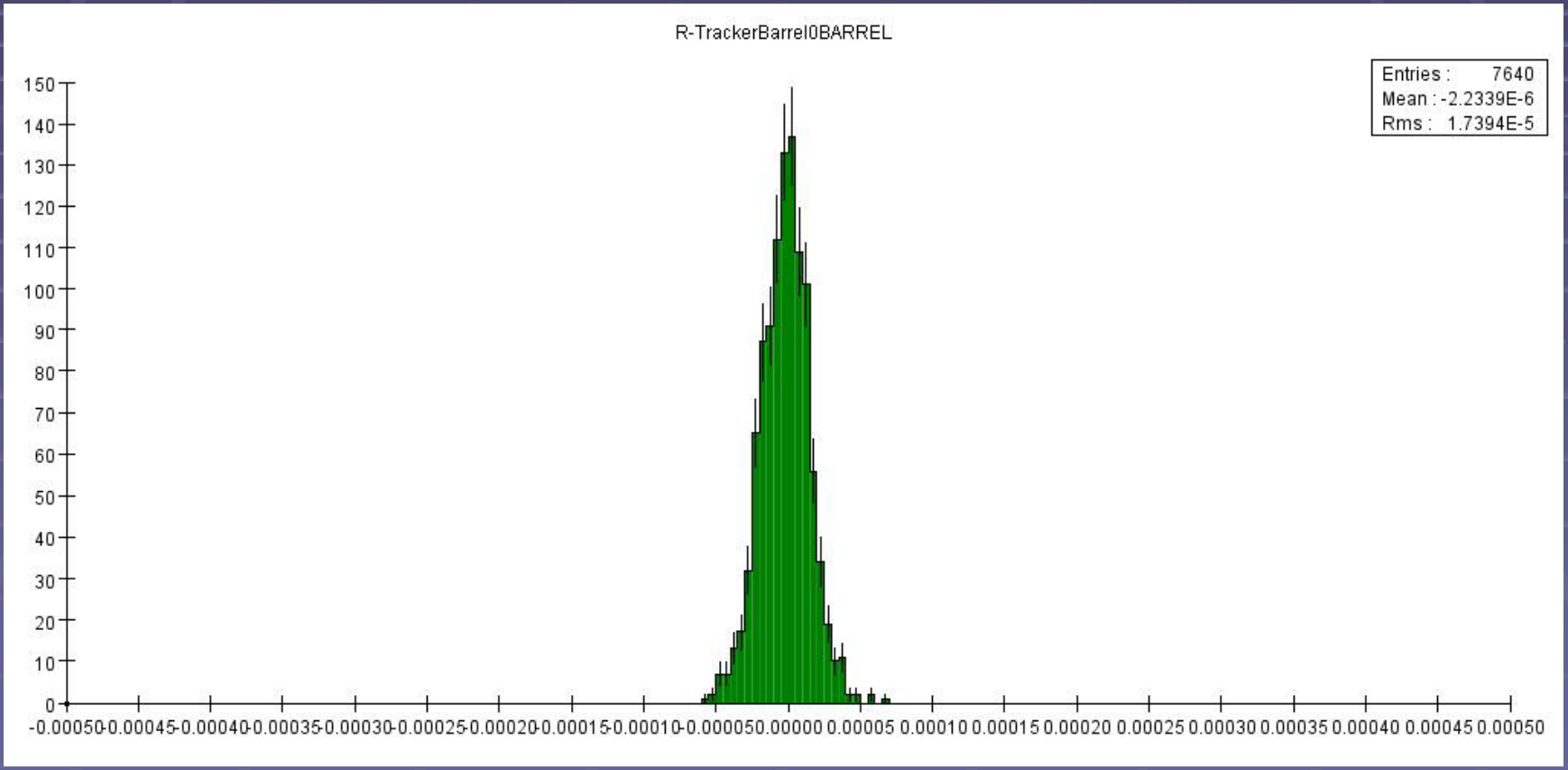
# Vertex Endcap - Φ



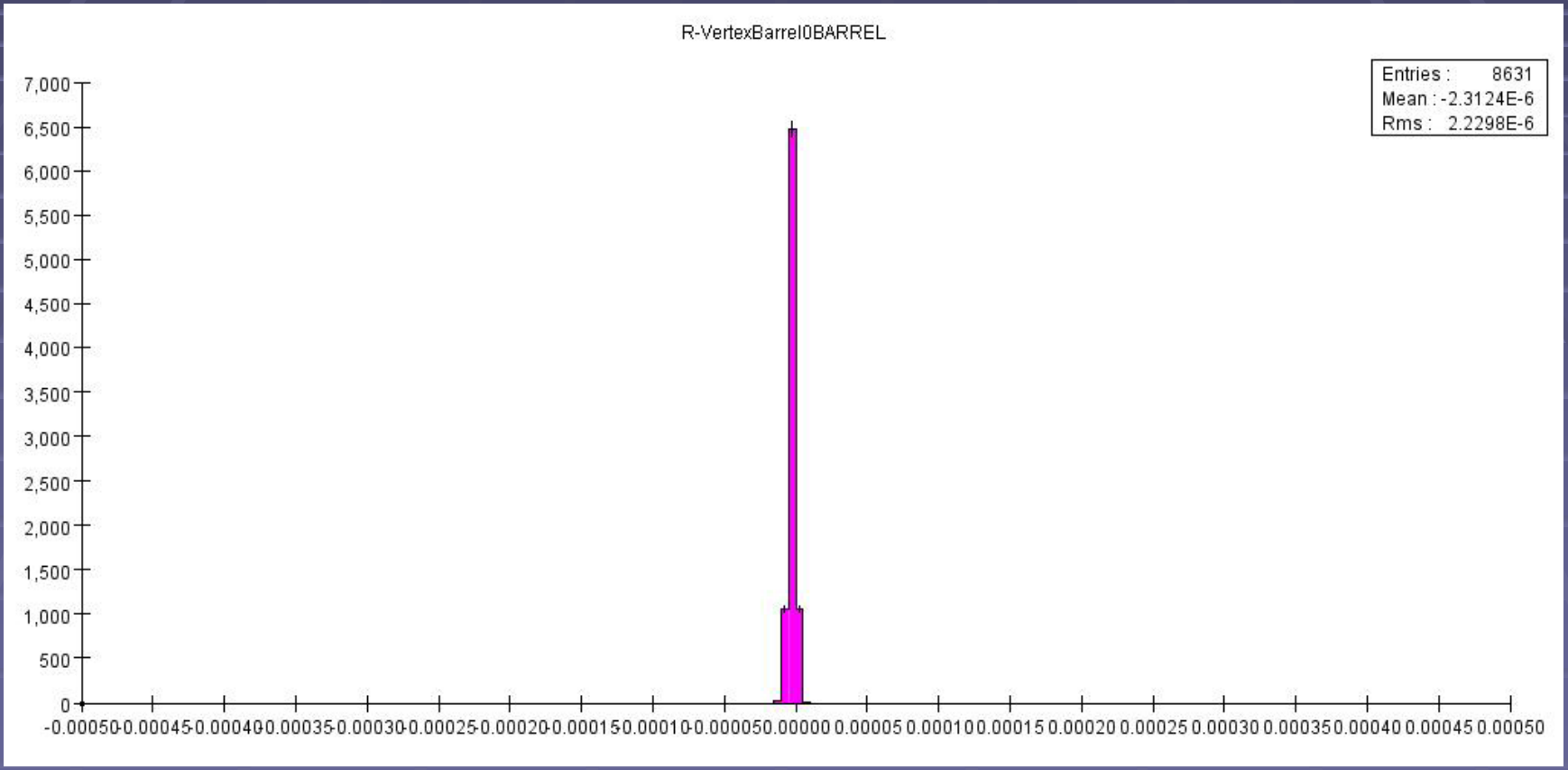
# Residual

- The residual histograms show that the tracker isn’t as concise as the vertex barrel/endcaps

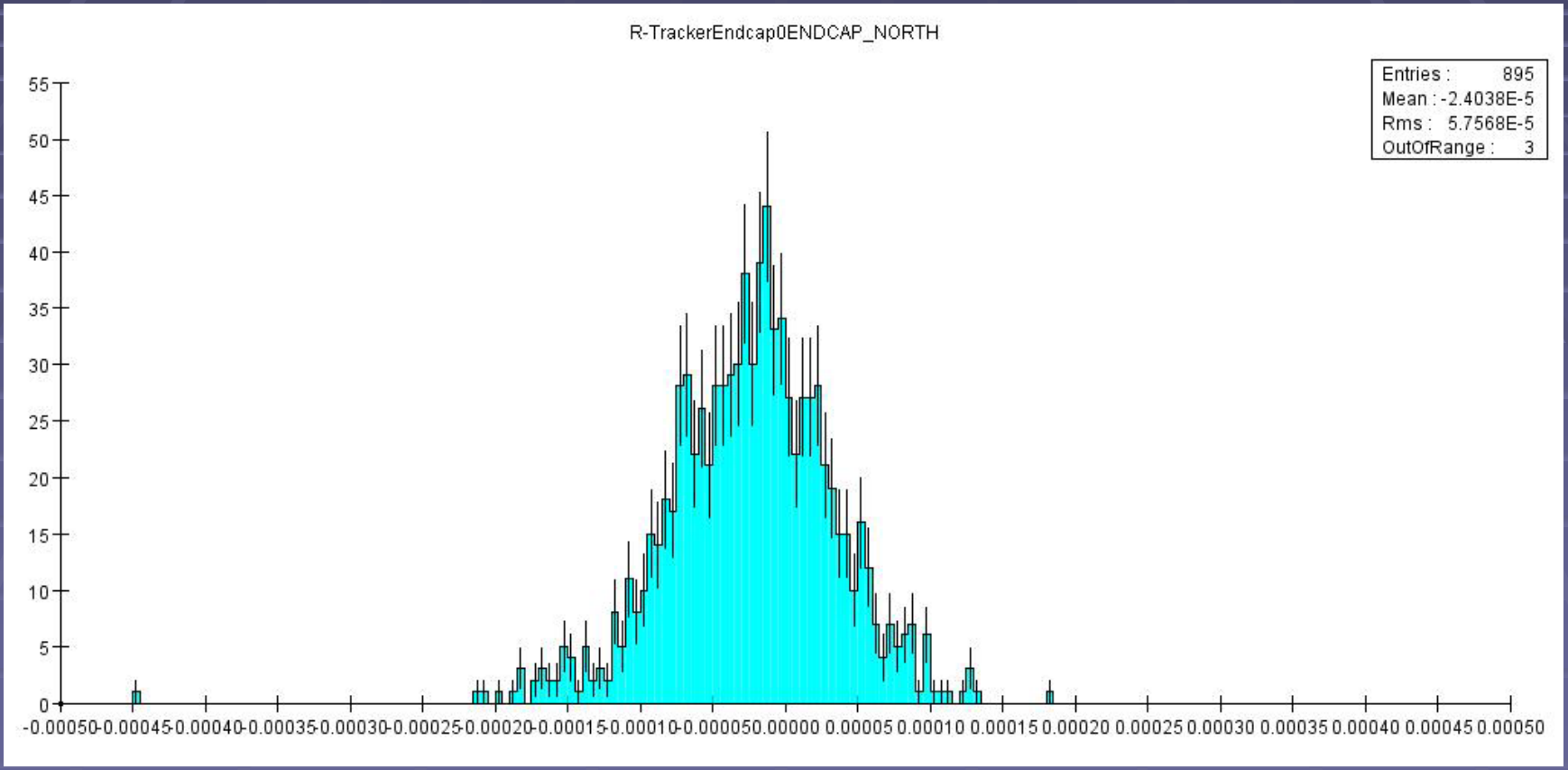
# Tracker Barrel r



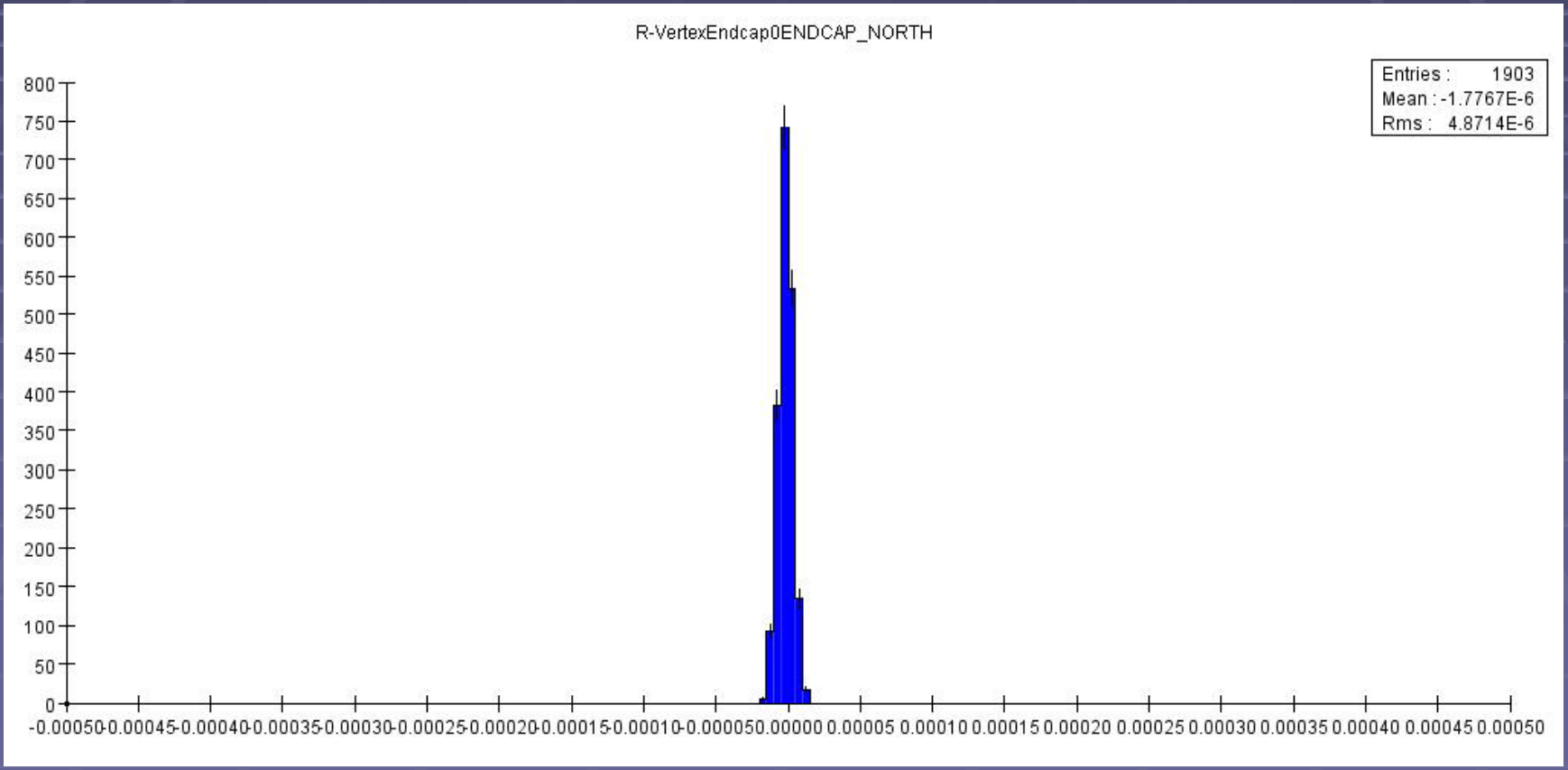
# Vertex Barrel r



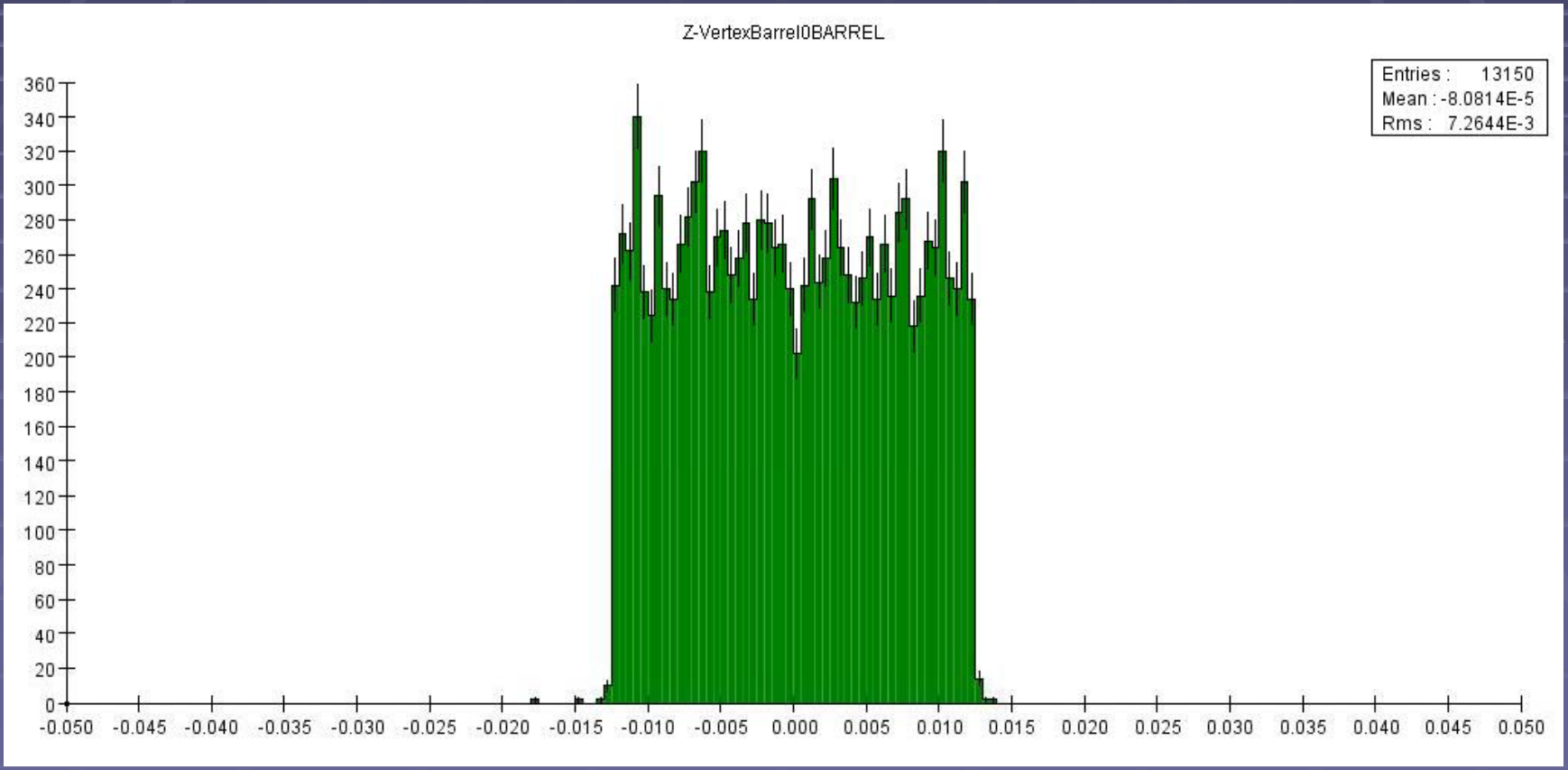
# Tracker Endcap r



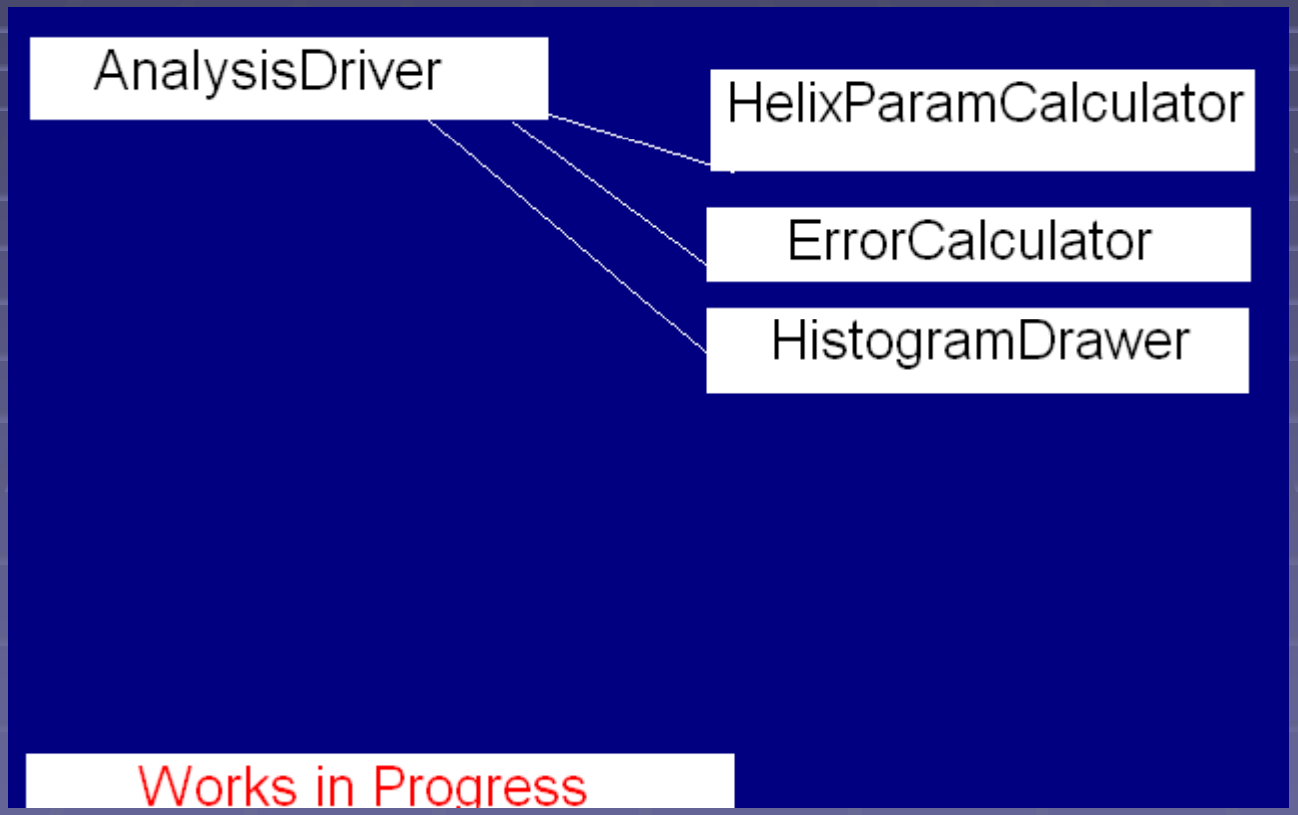
# Vertex Endcap r



# Vertex Barrel Z



# Implementation



# HelixParamCalculator

- public HelixParamCalculator(MCParticle,EventHeader)
- Calculates and returns the 5 helix parameters when fed an MCParticle and an event
- Can also return:
  - B-Field
  - $(x_0,y_0)$ coordinates of the particle at the DCA
  - momentum/transverse momentum

# HistogramDrawer

- public HistogramDrawer(HelicalTrackFit,MCParticle,EventHeader)
- This class currently will graph all of the helix parameters histograms, like the few shown before
- Work in progress, but currently you can chose between positive/negative particles and whether fit or the MCParticle class is used to calculate them (or both).

# ErrorCalculator

- public ErrorCalculator(HelicalTrackFit)
- Will calculate the error for any of the 5 helix parameters using the covmatrix in the fit class

# Next

- Complete the analysis on the tracker barrel, vertex barrel and endcaps
- Provide easy to use classes which can facilitate my own work but also be used by others to in the future.