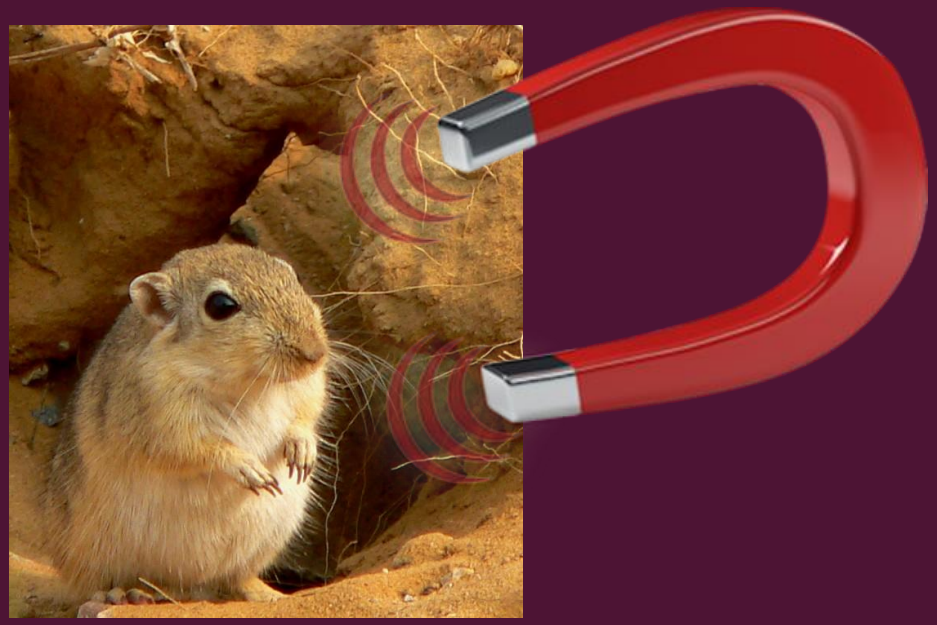
STUDIES WITH CORRECT FIELDMAPS

MIRIAM DIAMOND

OCT 15 2018



- EXTRAPOLATOR VALIDATION
- TRACK-CLUSTER MATCHING
- UPDATED VERTEXING

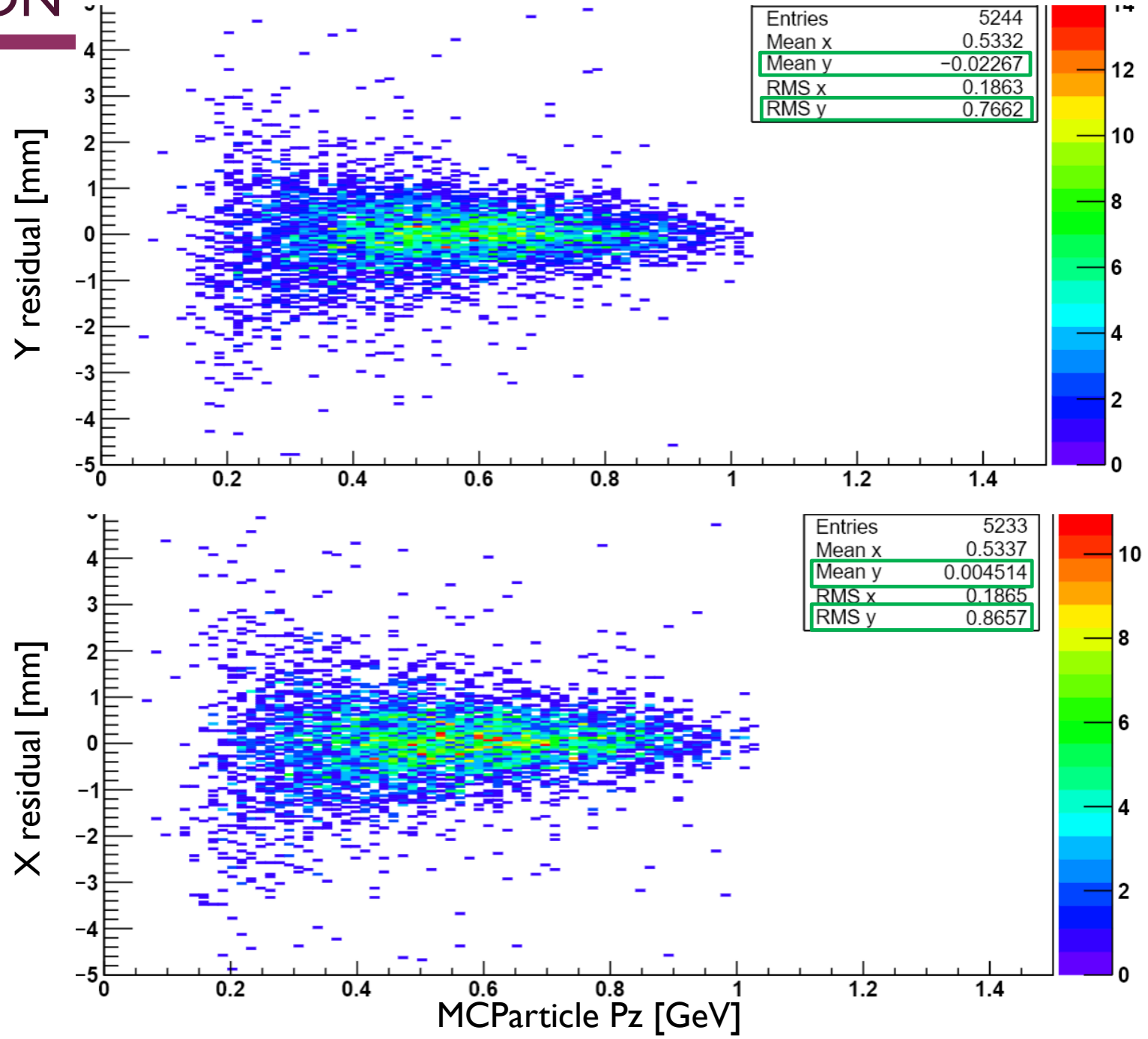
EXTRAPOLATOR VALIDATION

- github iss327: uses all components of full field-map, handles tilted planes
  - Based on Robert's RK4 integrator

- Tested using MC truth info
  - Look at each MCParticle passing some basic selection requirements
  - Take position&momentum of its SimTrackerHit in last SVT layer, pass to extrapolateTrackUsingFieldMapRK to extrapolate to ECal
  - Compare extrapolation result to particle's SimTrackerHit in TrackerHitsECal collection (simulated hit on ECal scoring plane)
- Often have multiple TrackerHitECal entries assigned to same MCParticle
  - Usually backsplash from calo shower… so select TrackerHitECal entry with earliest time
- Residual = extrapolated position – TrackerHitECal position

EXTRAPOLATOR VALIDATION

- https://confluence.slac.stanford.edu/download/attachments/236487741/track%20extrap.pdf?version=2&modificationDate=1528155770000&api=v2 showed strange residuals with old fieldmap and old (By only) extrapolator
  - Momentum dependence
  - Non-zero mean
  - Large RMS
- Now these anomalies are gone ☺



TRACK-CLUSTER MATCHING

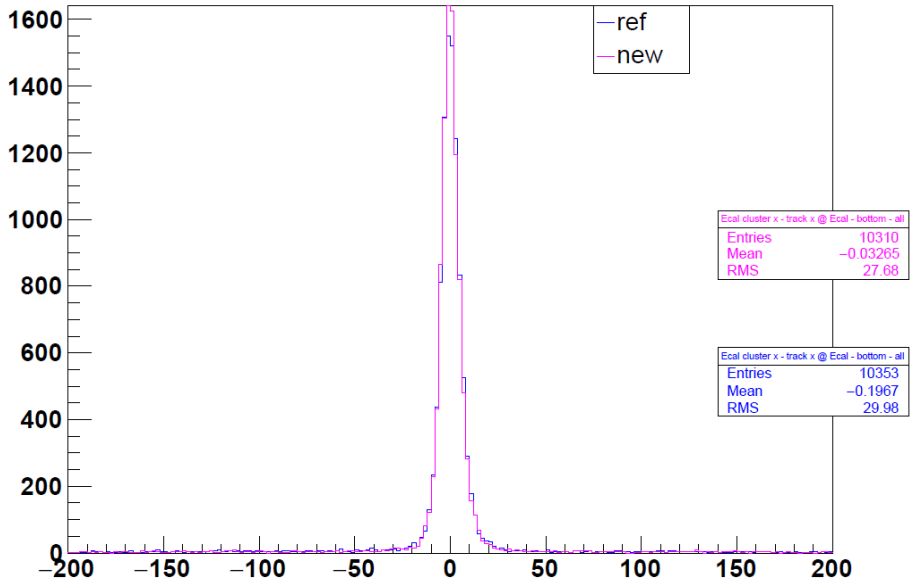
- Extrapolation to ECal for GBL tracks done by TrackDataDriver
- Track-cluster matching (TrackClusterMatcher, called by ReconParticleDriver) uses TrackState@ECal created by TrackDataDriver
- Matching uses cluster-track residuals parameterization
- Results don't change with new extrapolator/fieldmap, because matching criterion is so loose
- But, could consider tightening criterion and re-doing parameterization (again)

hps-recon ▸ src/main/java ▸ org.hps.recon.utils ▸ TrackClusterMatcher ▸

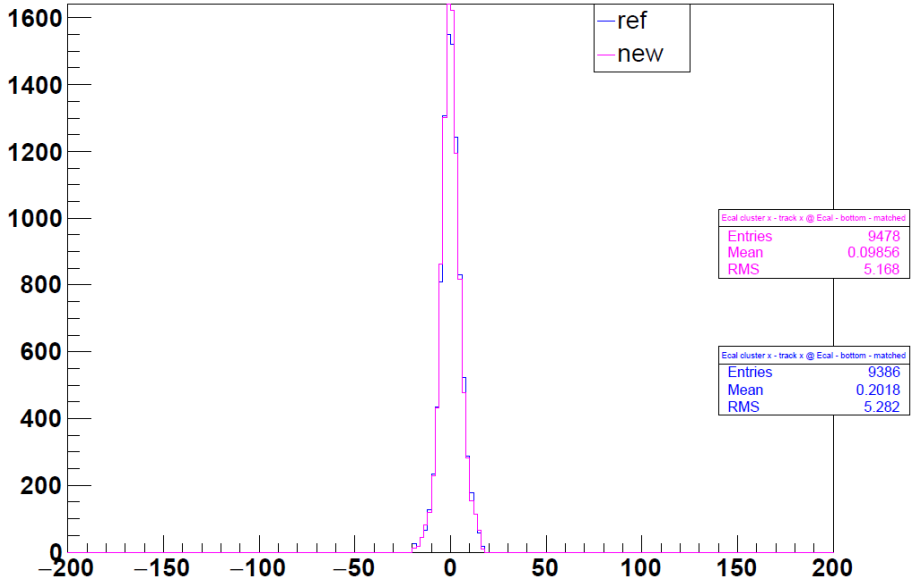
```
    /**
     * These cuts are set at +/- 4 sigma extracted from Gaussian fits to the
     * track-cluster residual distributions. The data used to determine these
     * limits is a pass 2 test file (t2.6) using run 5772.
     */
    private double topClusterTrackMatchDeltaXLow = -14.5; // mm
    private double topClusterTrackMatchDeltaXHigh = 23.5; // mm
    private double bottomClusterTrackMatchDeltaXLow = -19.5; // mm
    private double bottomClusterTrackMatchDeltaXHigh = 16.5; // mm

    private double topClusterTrackMatchDeltaYLow = -21.5; // mm
    private double topClusterTrackMatchDeltaYHigh = 28; // mm
    private double bottomClusterTrackMatchDeltaYLow = -28; // mm
    private double bottomClusterTrackMatchDeltaYHigh = 24; // mm

    /**
     * Rafo's parameterization of cluster-seed x/y position residuals as function of energy.
     *
     * Derived using GBL/seed tracks, non-analytic extrapolation, uncorrected cluster positions,
     * and EngRun2015-Nominal-v4-4-fieldmap detector.
     *
     *  f = p0+e*(p1+e*(p2+e*(p3+e*(p4+e*p5))))
     */
    private static final double dxMeanTopPosiGBL[] = { 6.67414,-9.57296, 5.70647, 27.4523,-28.110
    private static final double dxSigmTopPosiGBL[] = { 52.6437,-478.805, 1896.73,-3761.48, 3676.7
    private static final double dxMeanBotPosiGBL[] = { 4.13802, 15.8887,-74.2844,-9.78944, 308.54
    private static final double dxSigmBotPosiGBL[] = { 37.6513,-294.851, 1002.15,-1639.08, 1228.0
```

TRACK-CLUSTER MATCHING X

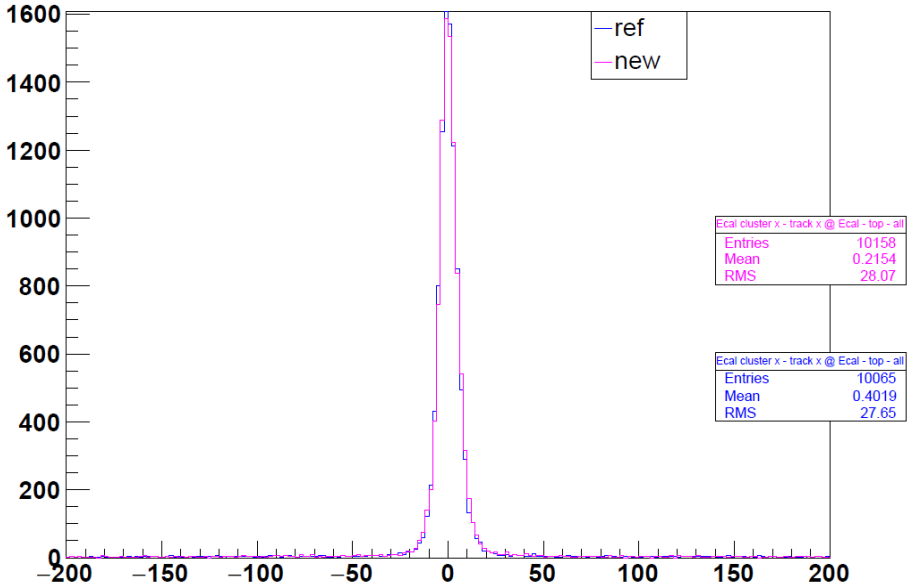
Ecal cluster x - track x @ Ecal - bottom - all



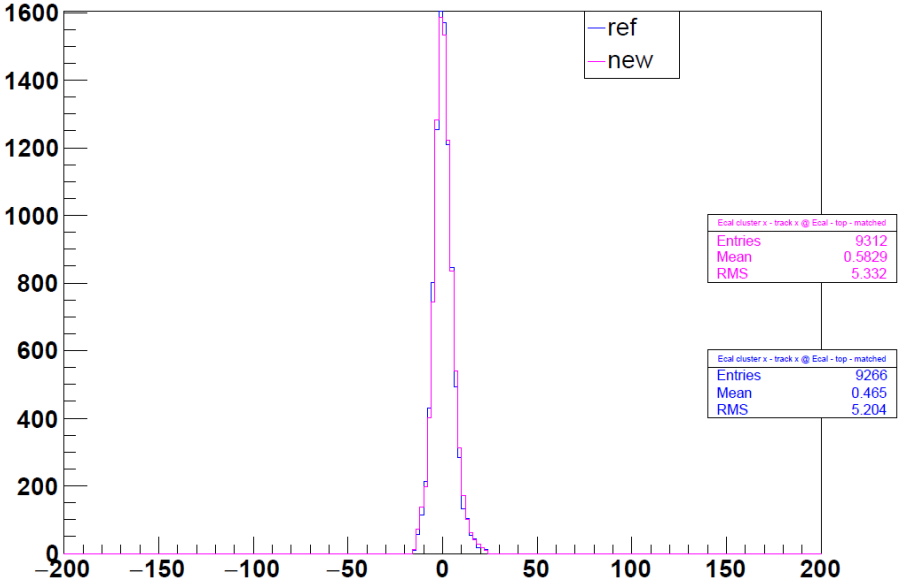
Ecal cluster x - track x @ Ecal - bottom - matched



Ecal cluster x - track x @ Ecal - top - all

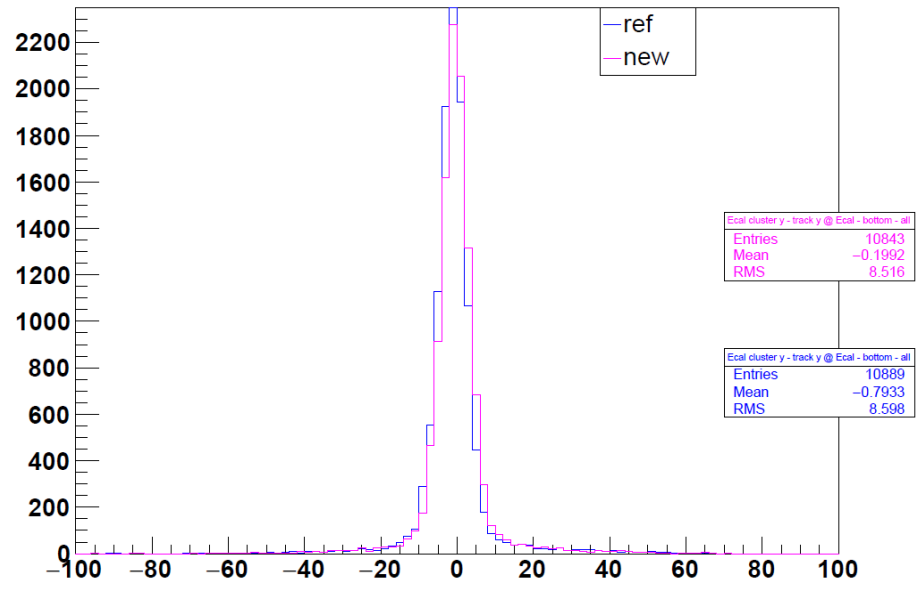


Ecal cluster x - track x @ Ecal - top - matched

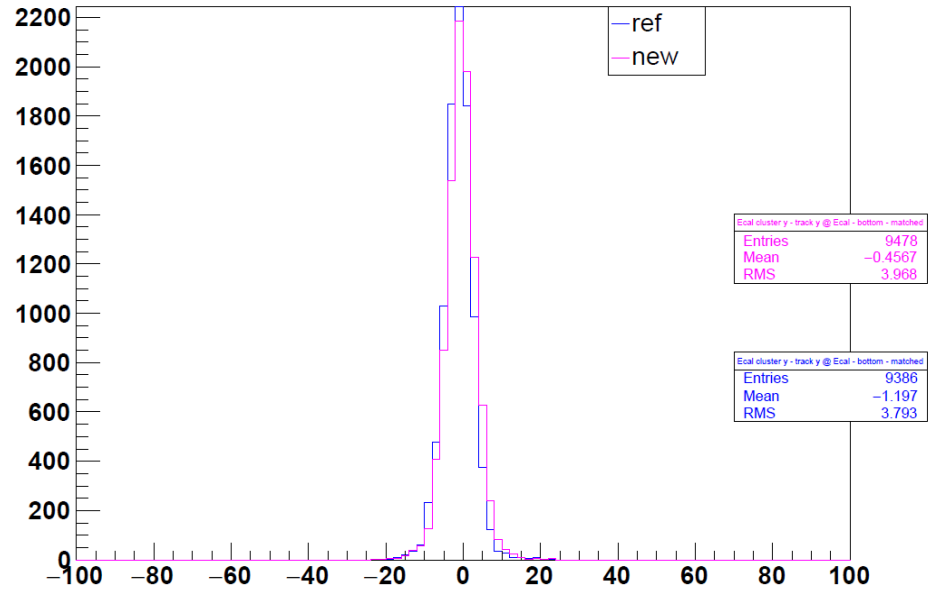


TRACK-CLUSTER MATCHING Y

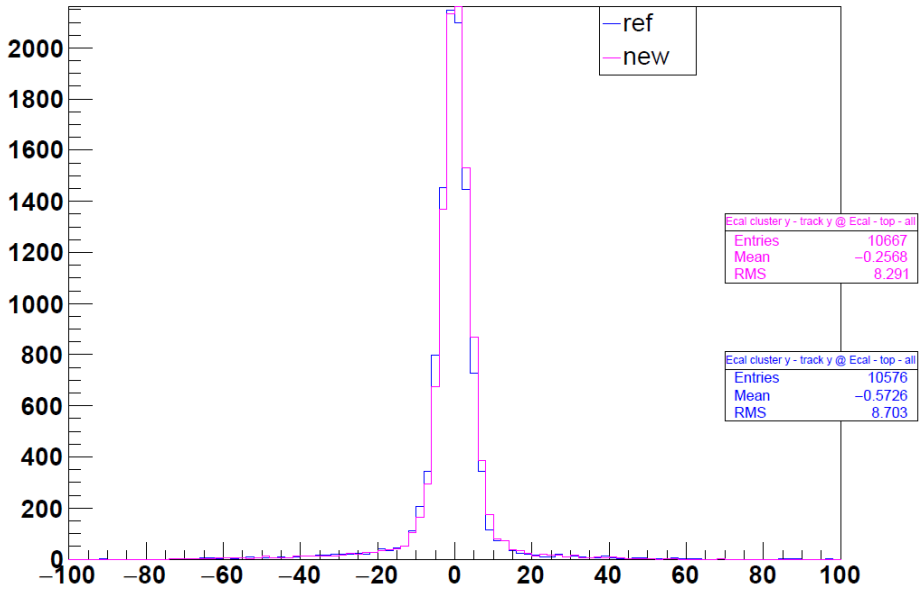
Ecal cluster y - track y @ Ecal - bottom - all



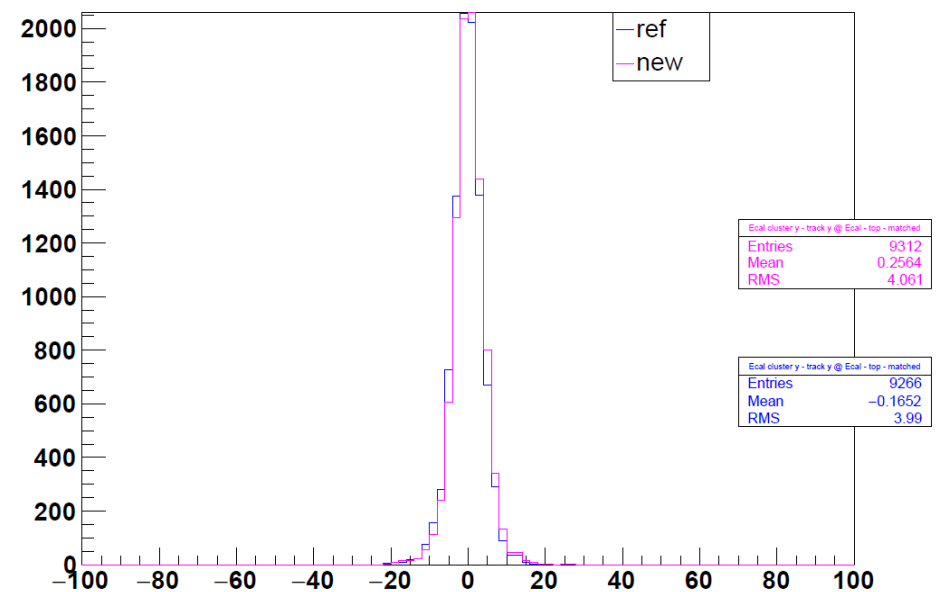
Ecal cluster y - track y @ Ecal - bottom - matched



Ecal cluster y - track y @ Ecal - top - all



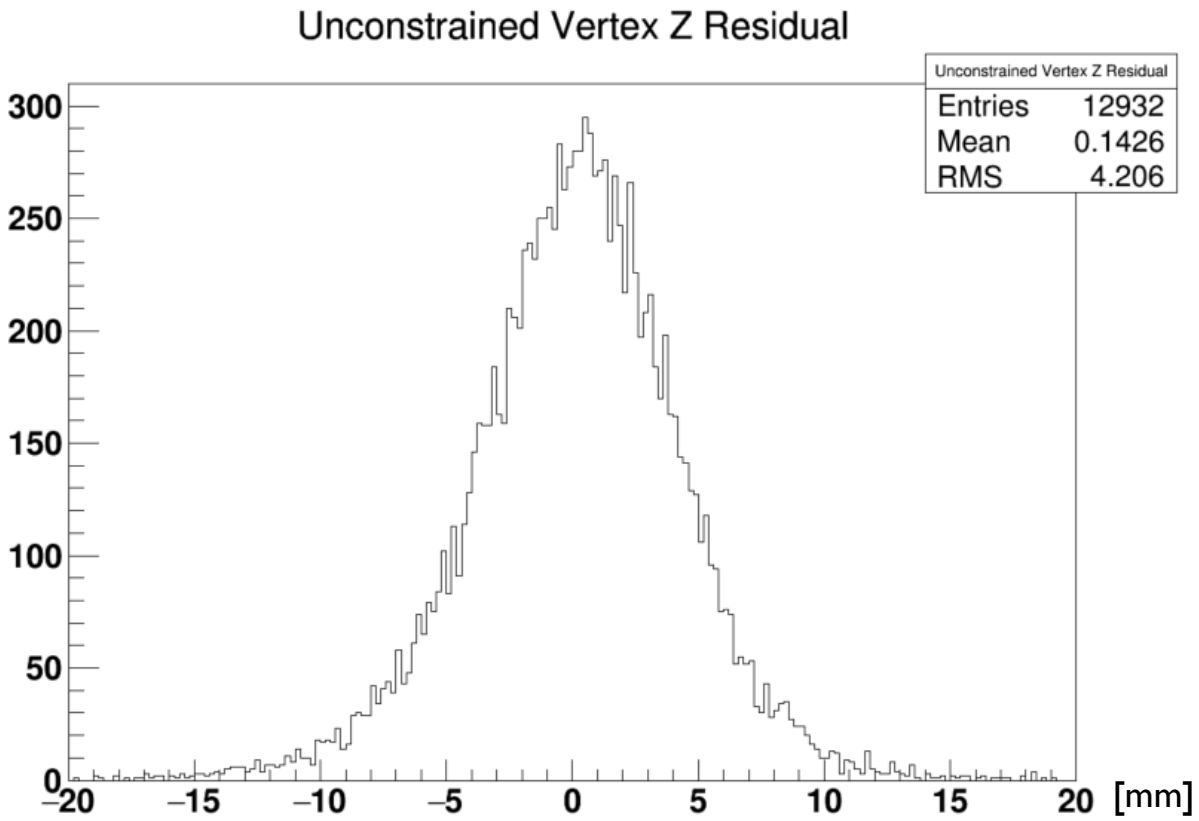
Ecal cluster y - track y @ Ecal - top - matched

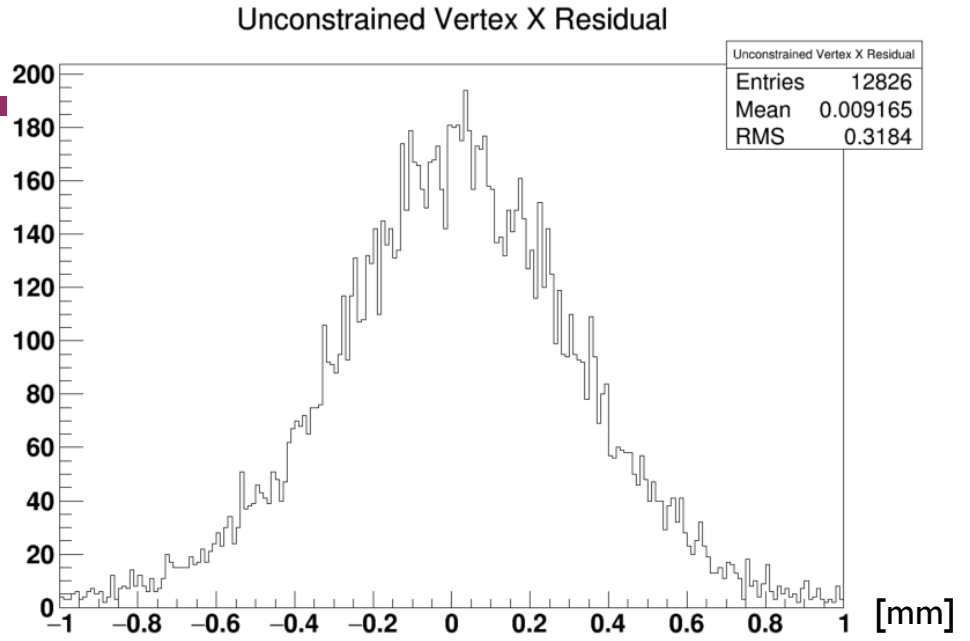


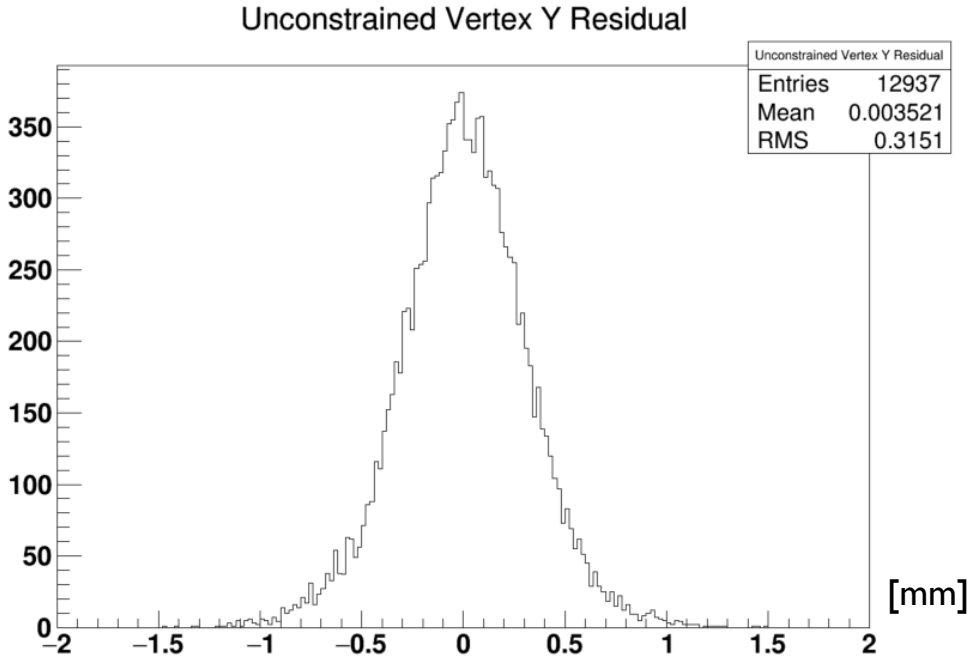
VERTEXING UPDATES?

- Vertexing currently doesn't take into account changing B-field from target to L1
- Does this need fixing?
  - Applied MOUSE cuts to reconstructed V0s in new Prompt A' sample
    - target at z=0.5mm
  - Plotted residuals and pulls of Unconstrained, Beamspot-Constrained, and Target-Constrained V0 positions, reconstructed – MC
    - vs vertex Pz
  - What other plots/studies would be useful?

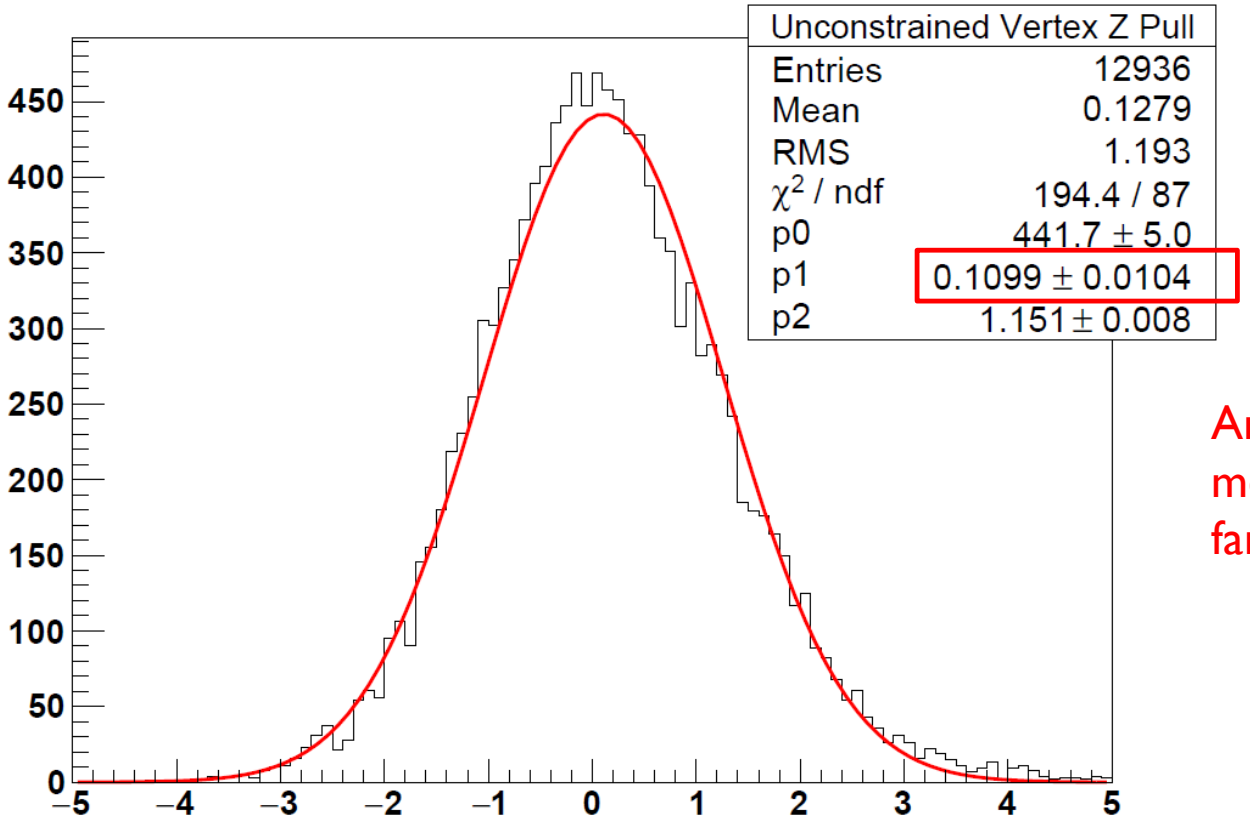
UNCONSTRAINED V0



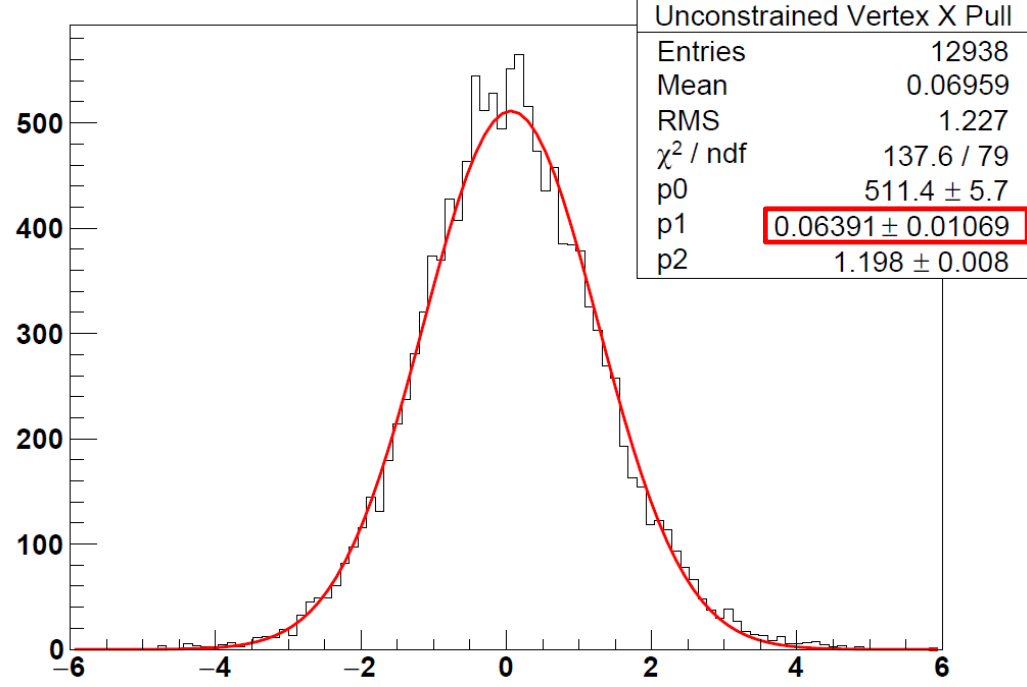


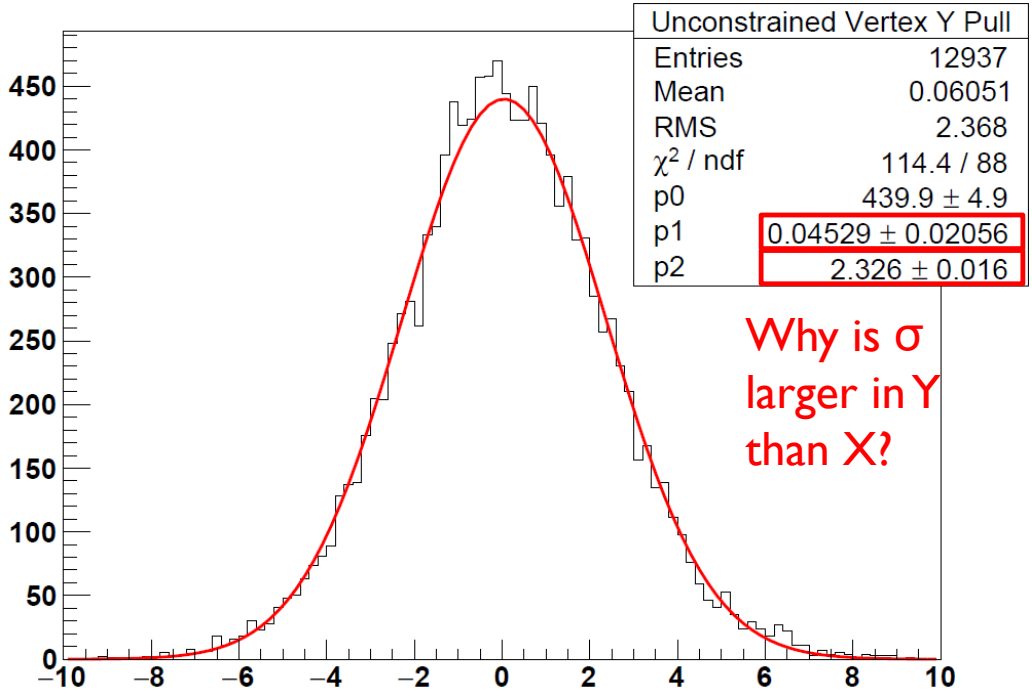


UNCONSTRAINED V0



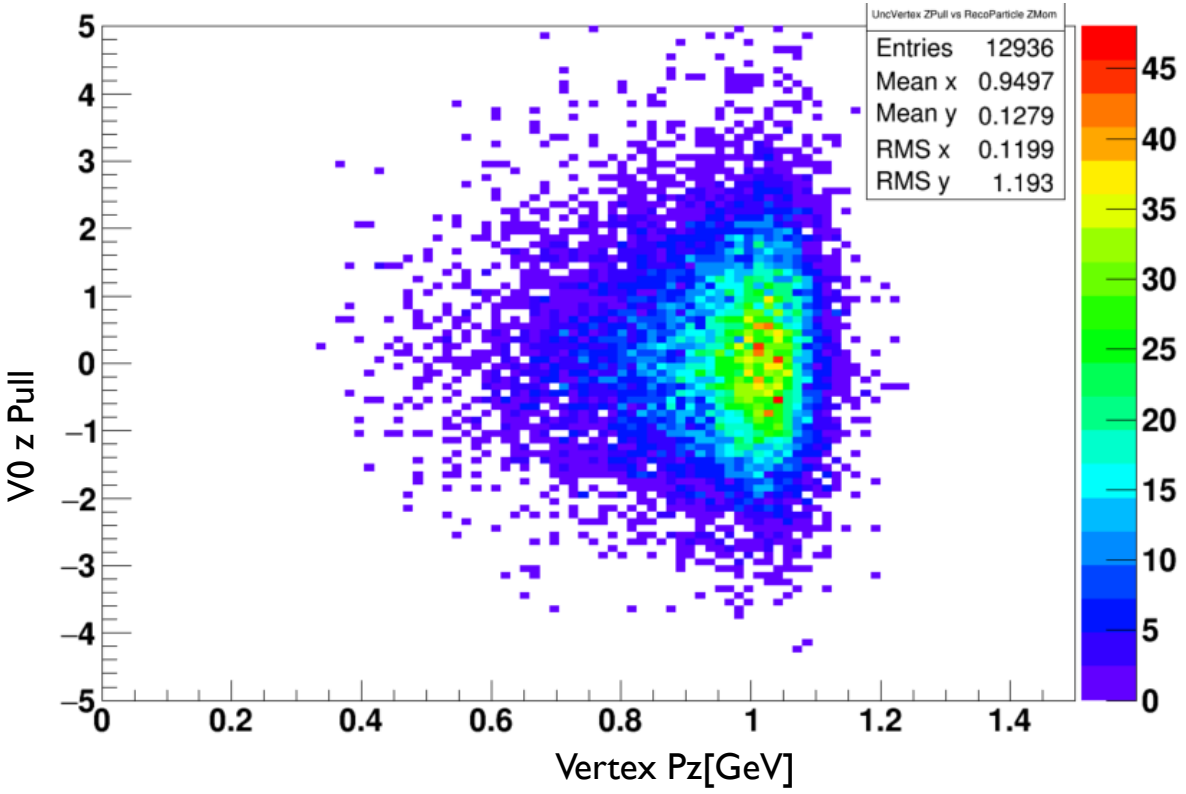
Are the means too far from 0?

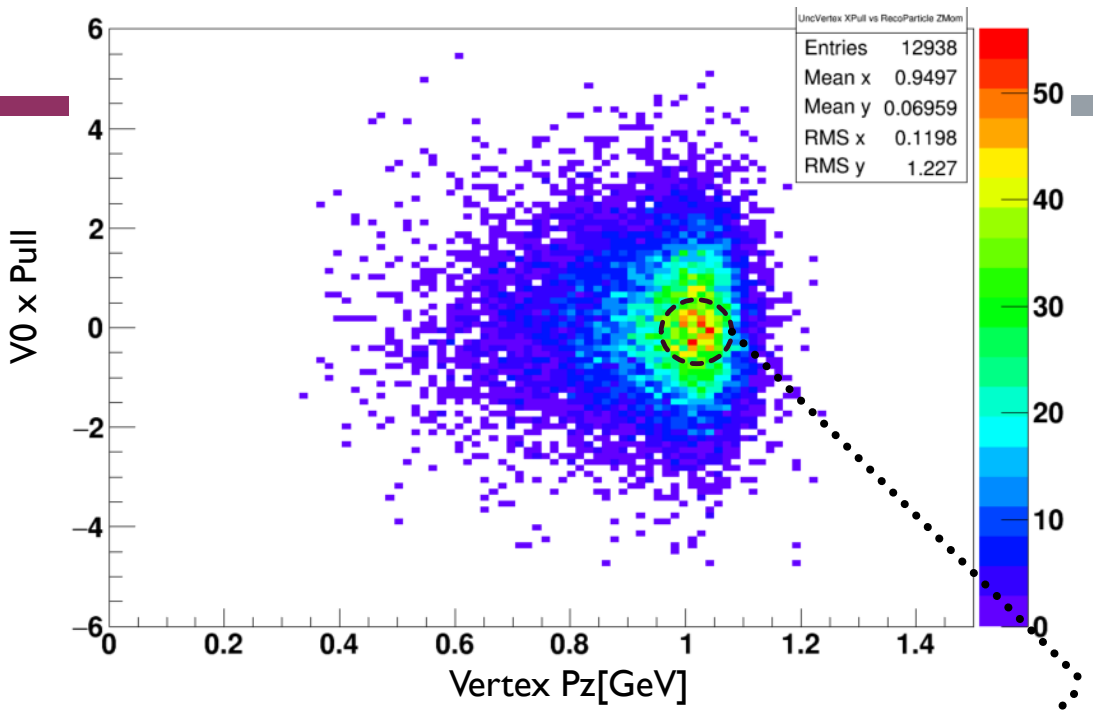


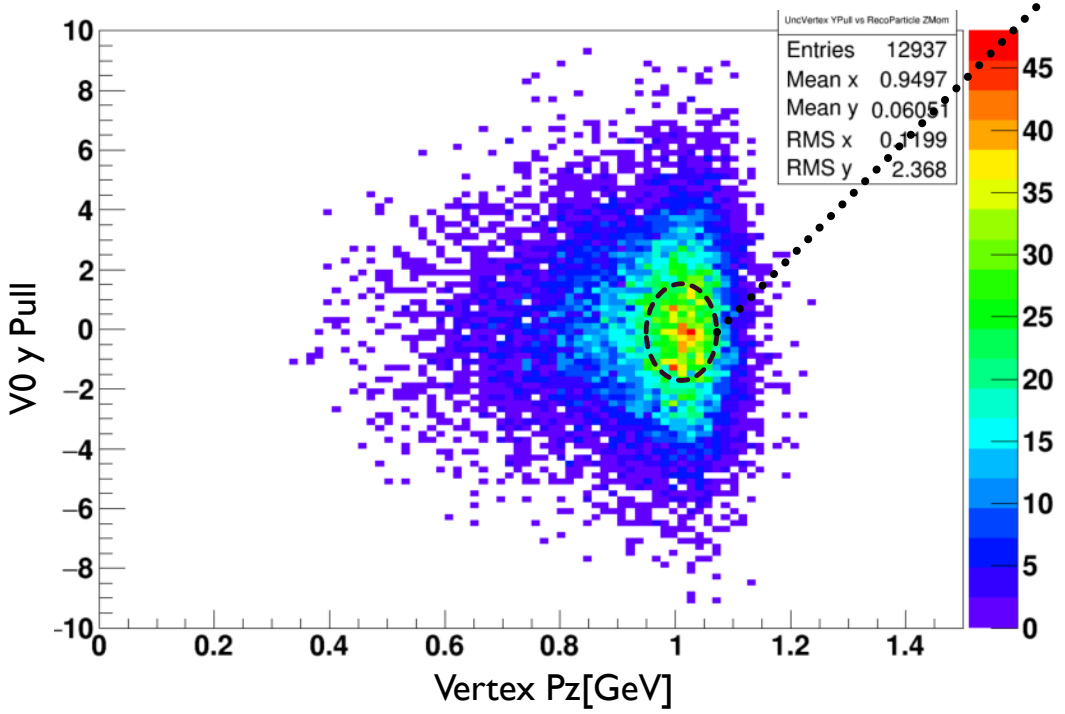


Why is σ larger in Y than X?

UNCONSTRAINED V0

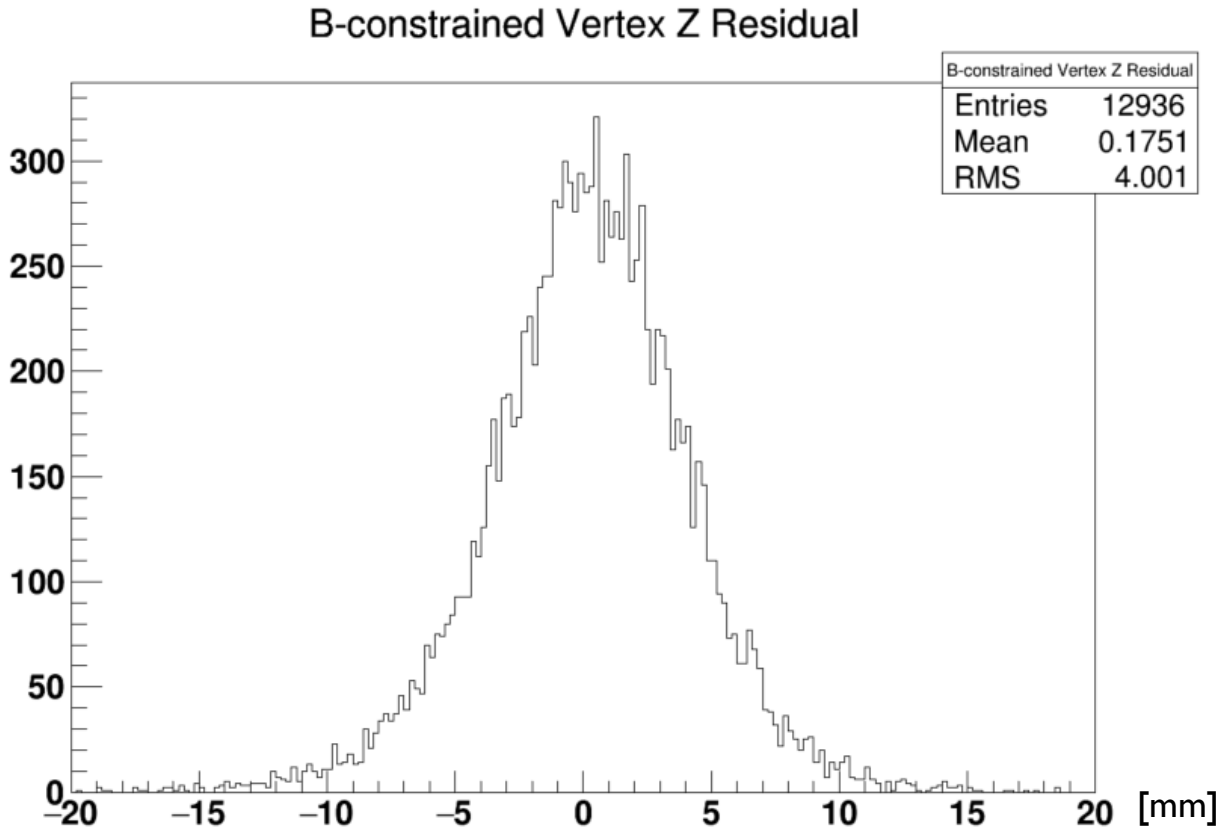


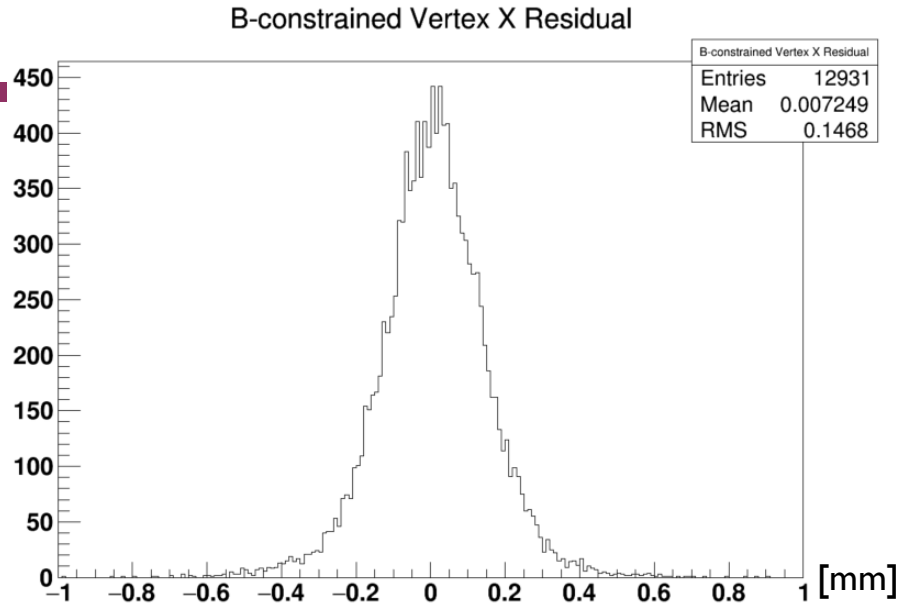


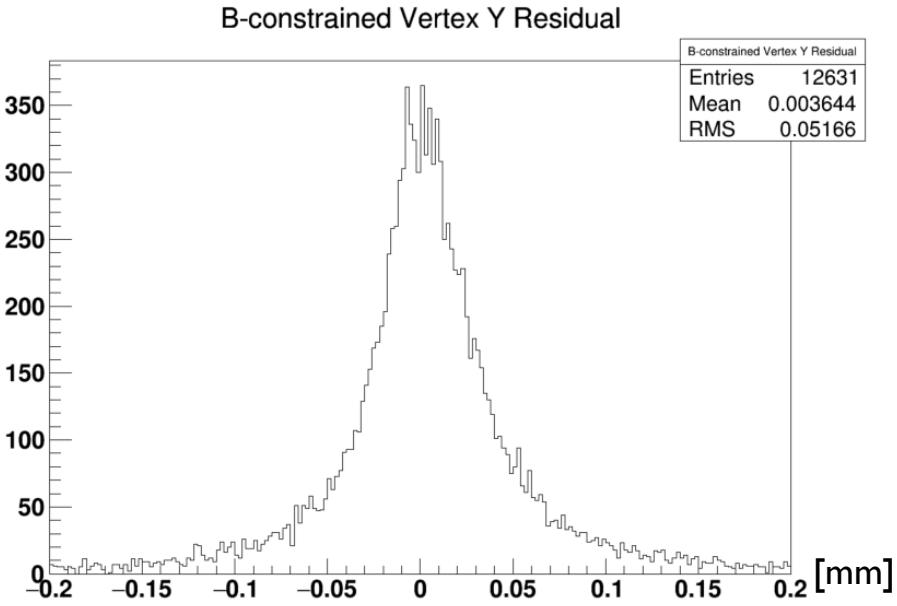


Pz dependence in core?

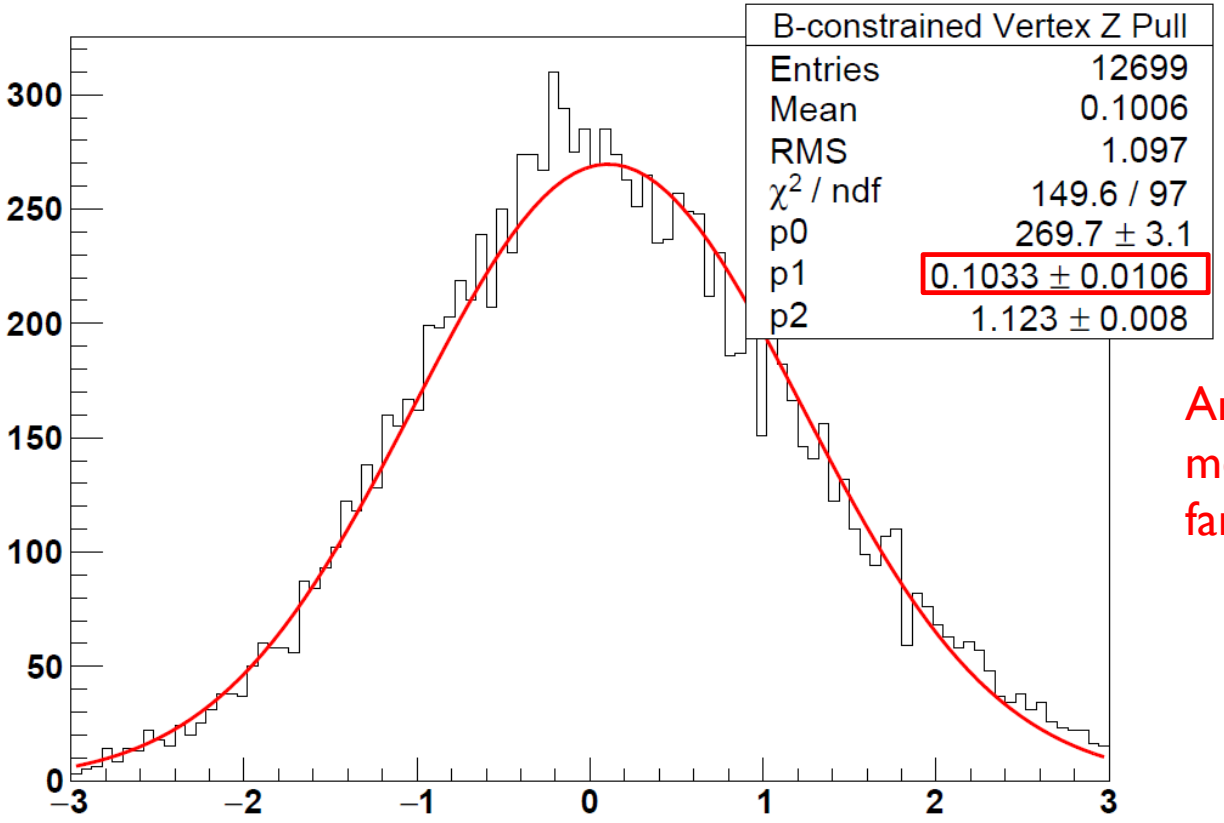
BEAMSPOT-CONSTRAINED V0



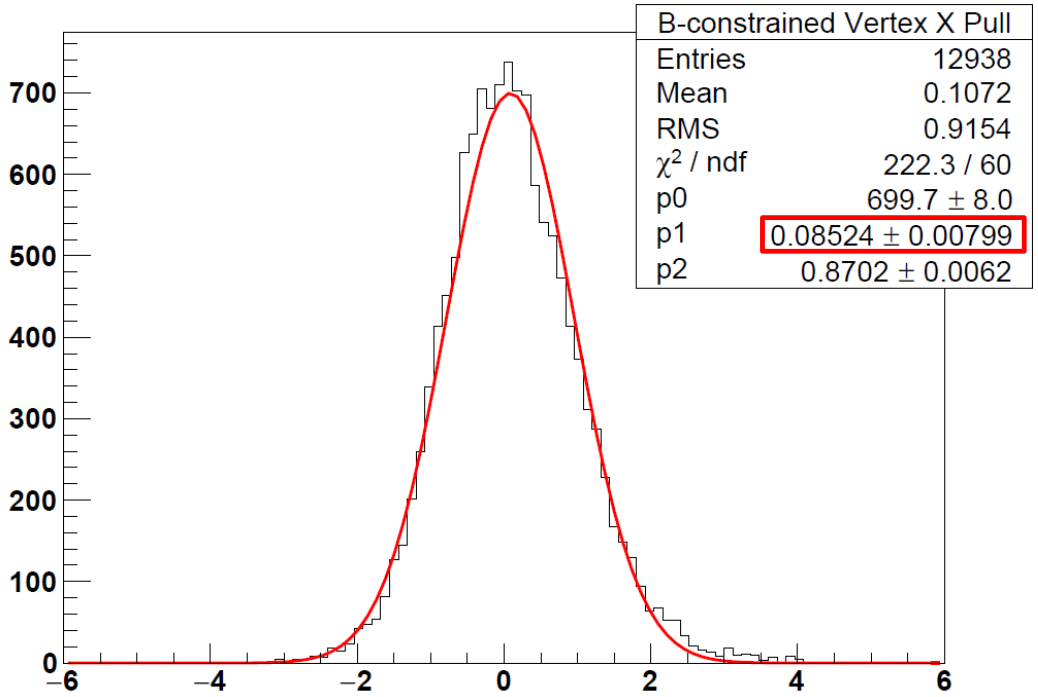


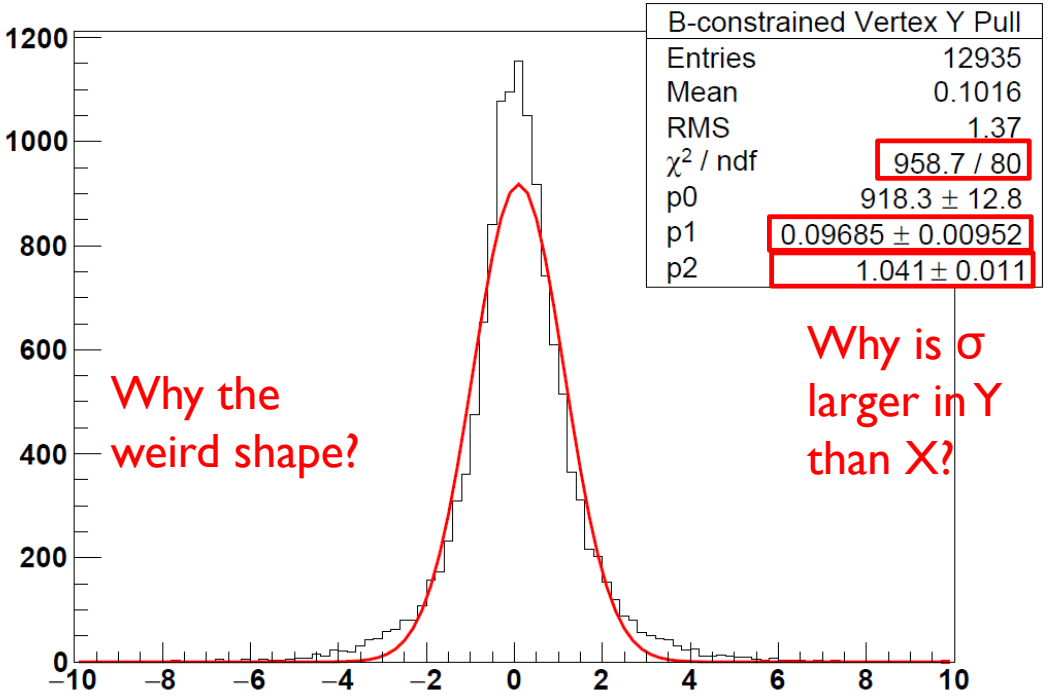


BEAMSPOT-CONSTRAINED V0



Are the means too far from 0?

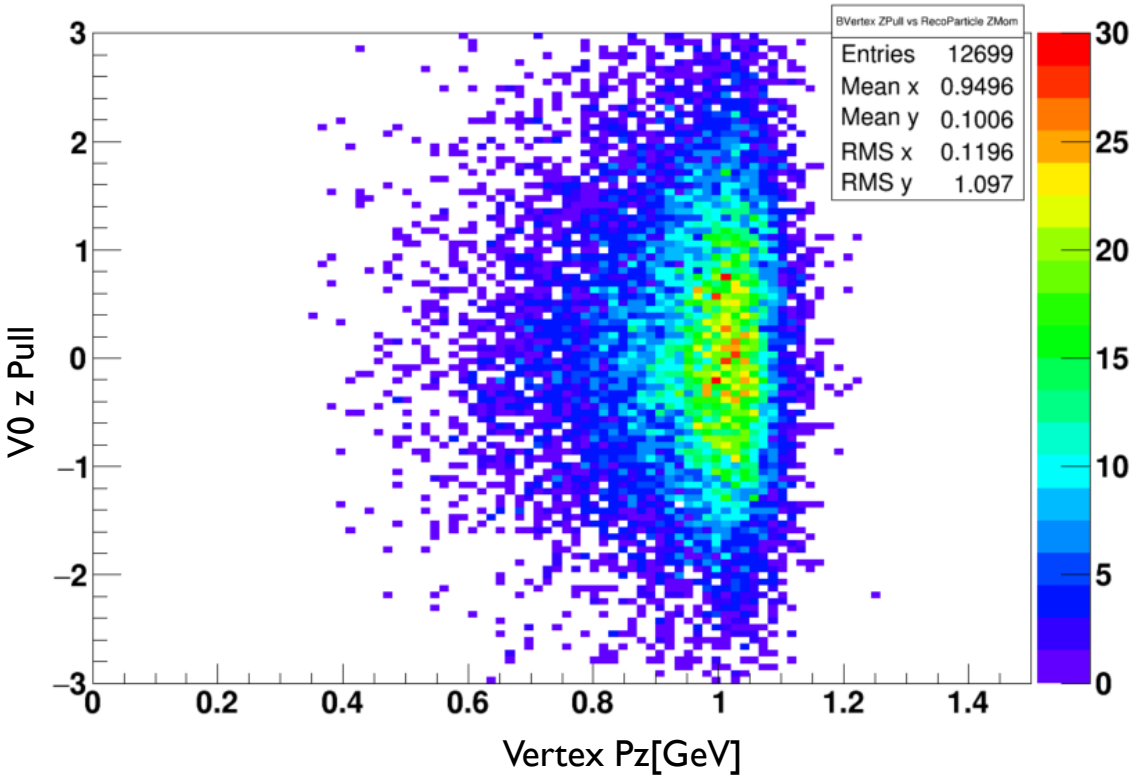


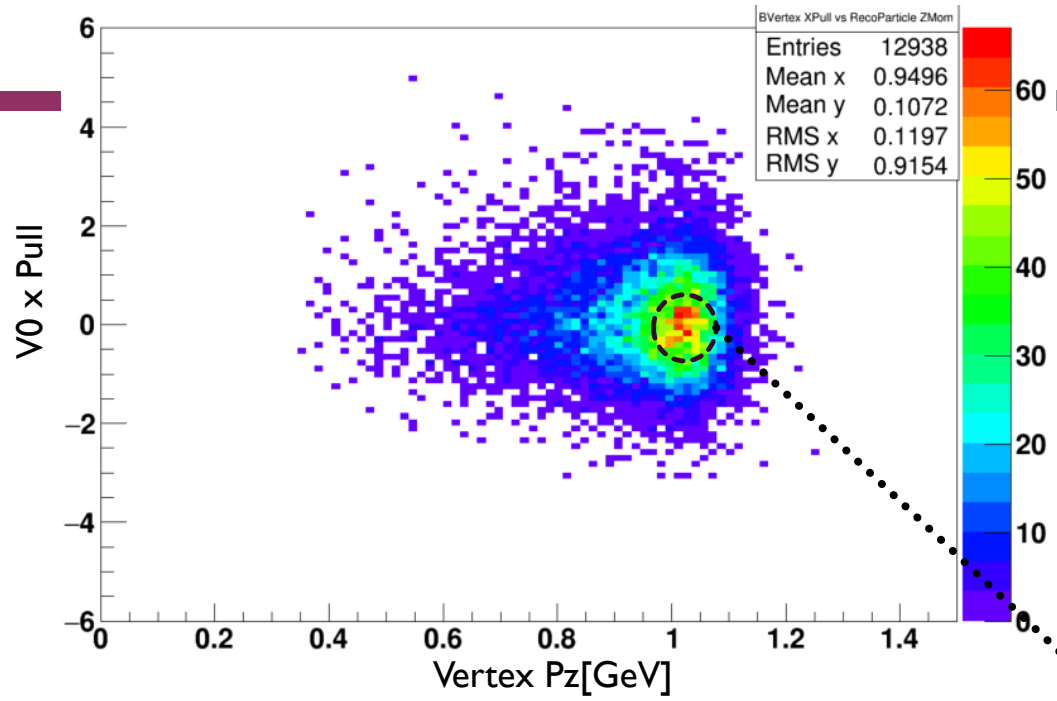


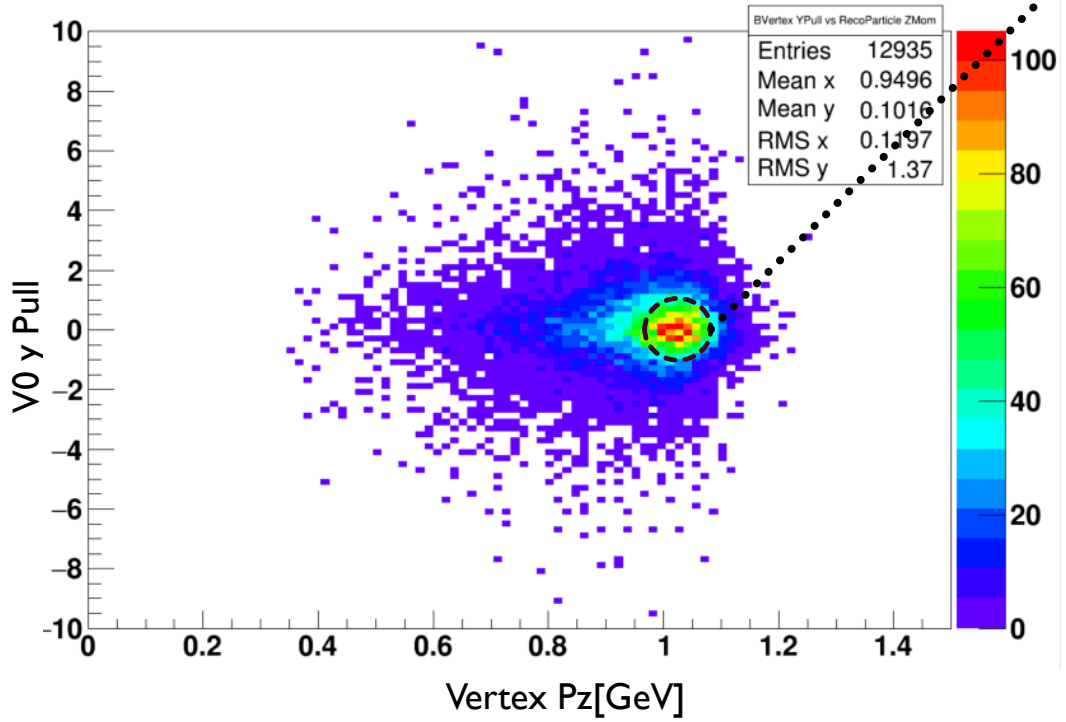
Why the weird shape?

Why is σ larger in Y than X?

BEAMSPOT-CONSTRAINED V0

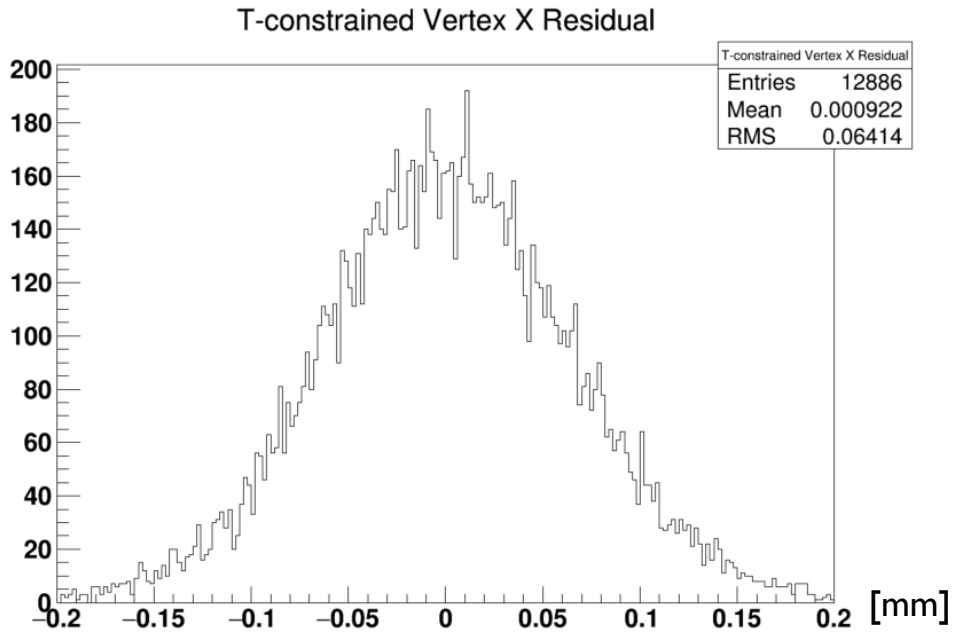


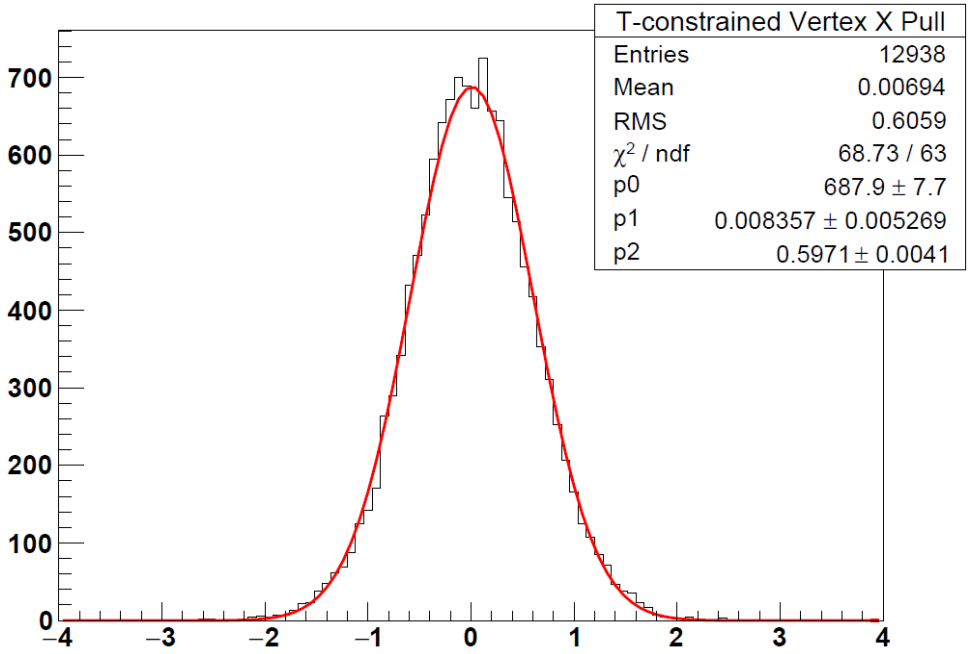


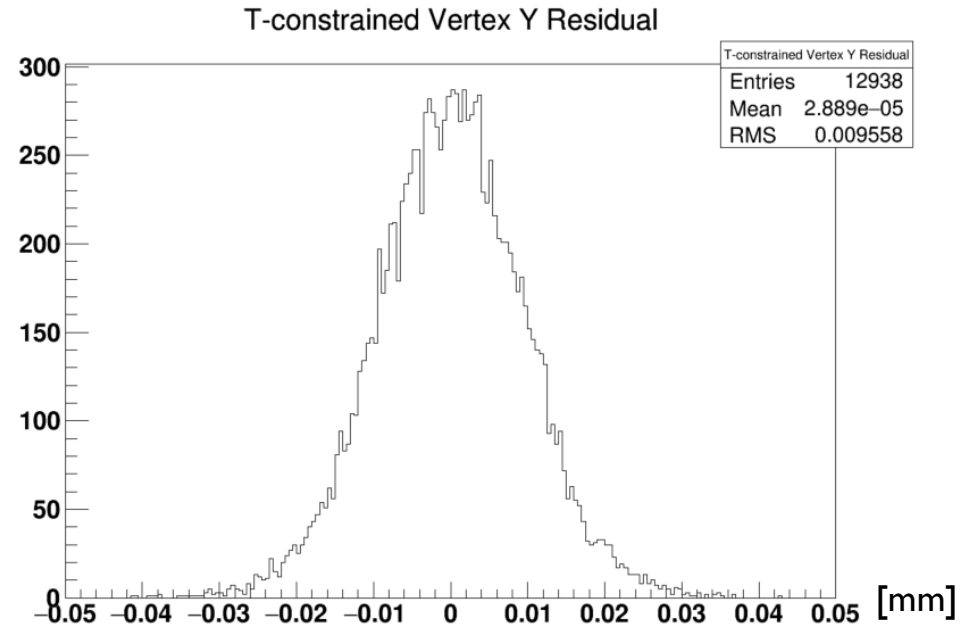


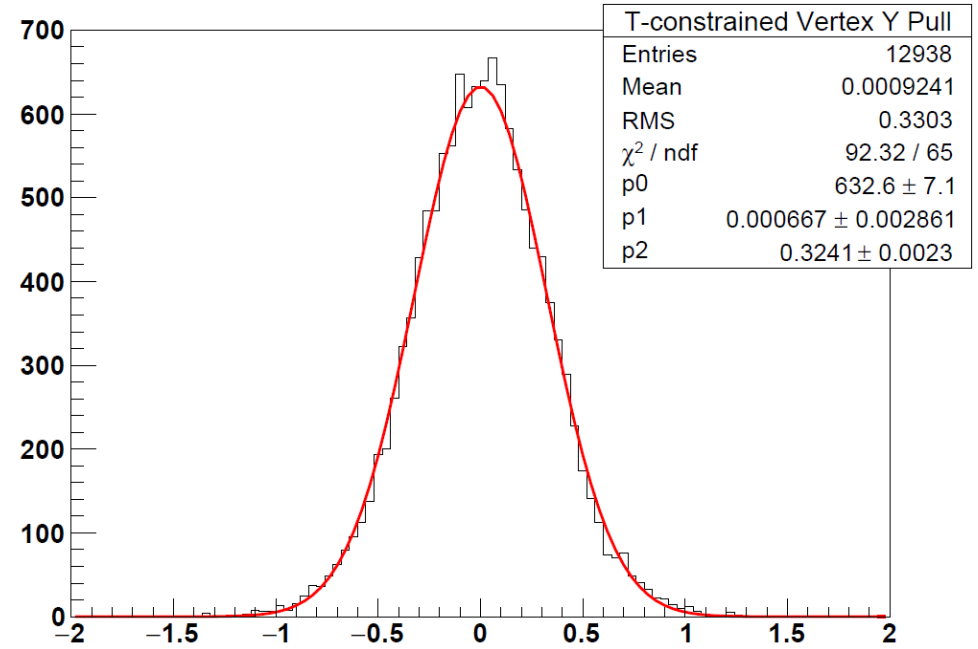
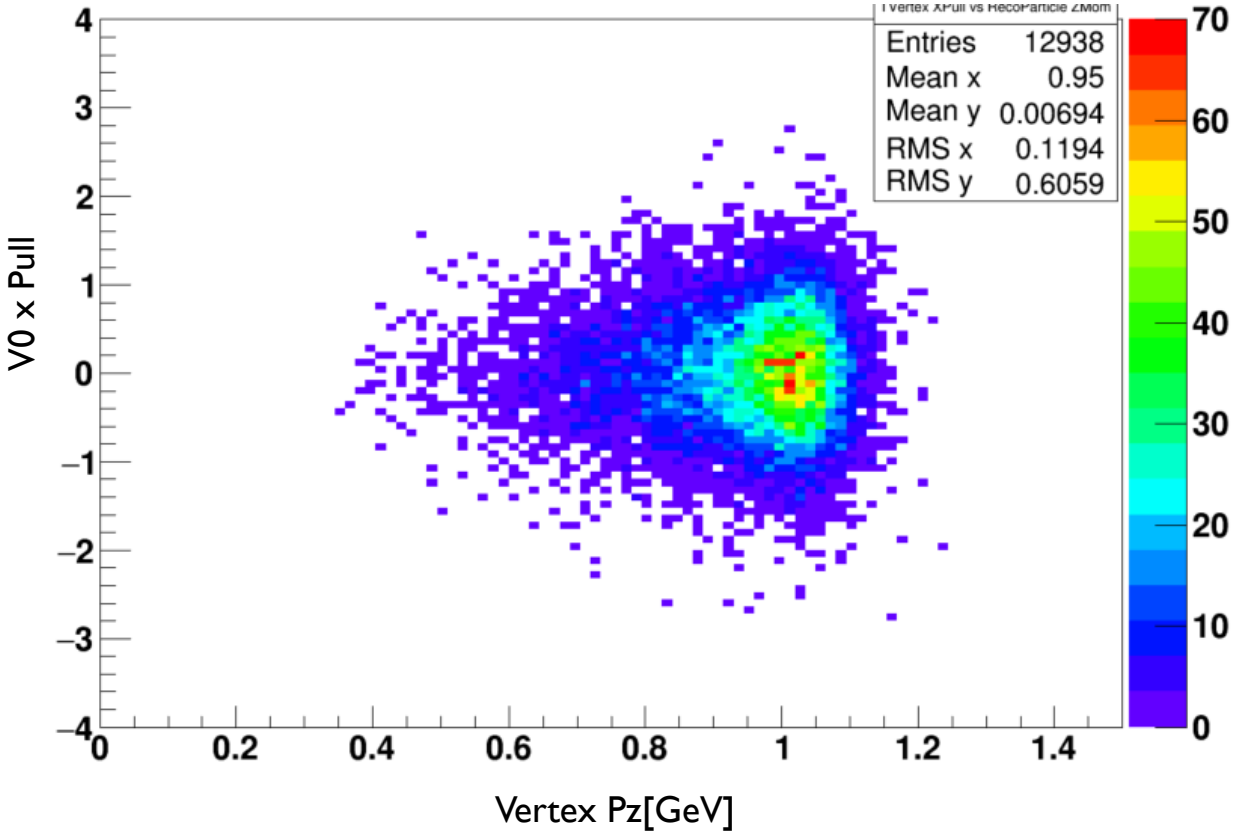
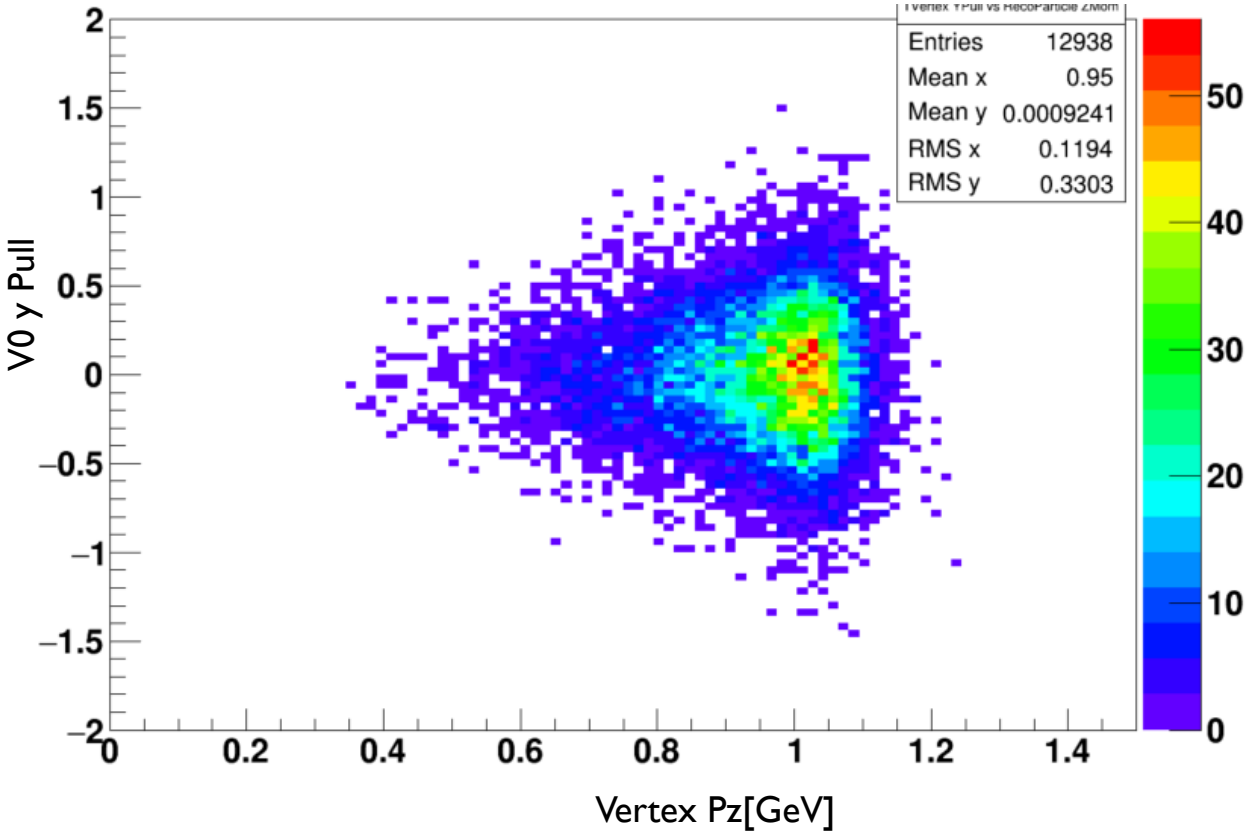
Weird core shape?

TARGET-CONSTRAINED V0









TARGET-CONSTRAINED V0