TRACK EXTRAPOLATION

MIRIAM DIAMOND

NOV 20 2017

CURRENT EXTRAPOLATION IN NTUPLE

- Track extrapolation in hps-analysis ▸ src/main/java ▸ org.hps.analysis.tuple ▸ TupleDriver

```
for (HpsSiSensor sensor : sensors) {
    double zPos = sensor.getGeometry().getPosition().z();
    Hep3Vector extrapPos = TrackUtils.extrapolateTrack(track, zPos);
```

hps-tracking ▸ src/main/java ▸ org.hps.recon.tracking ▸ TrackUtils

```
public static Hep3Vector extrapolateTrack(Track track, double z) {
    return extrapolateTrack(track.getTrackStates().get(0), z);
}
```

- Room for improvement:
  - Use trackstate-at-sensor from GBL, instead of trackstate at IP
  - Calculate proper intercept with sensor, instead of plane at middle of sensor

    hps-tracking ▸ src/main/java ▸ org.hps.recon.tracking ▸ TrackUtils ▸ getHelixPlaneIntercept

  - Use fieldmap

    hps-tracking ▸ src/main/java ▸ org.hps.recon.tracking ▸ TrackUtils ▸ extrapolateTrackUsingFieldMap

Can't do both

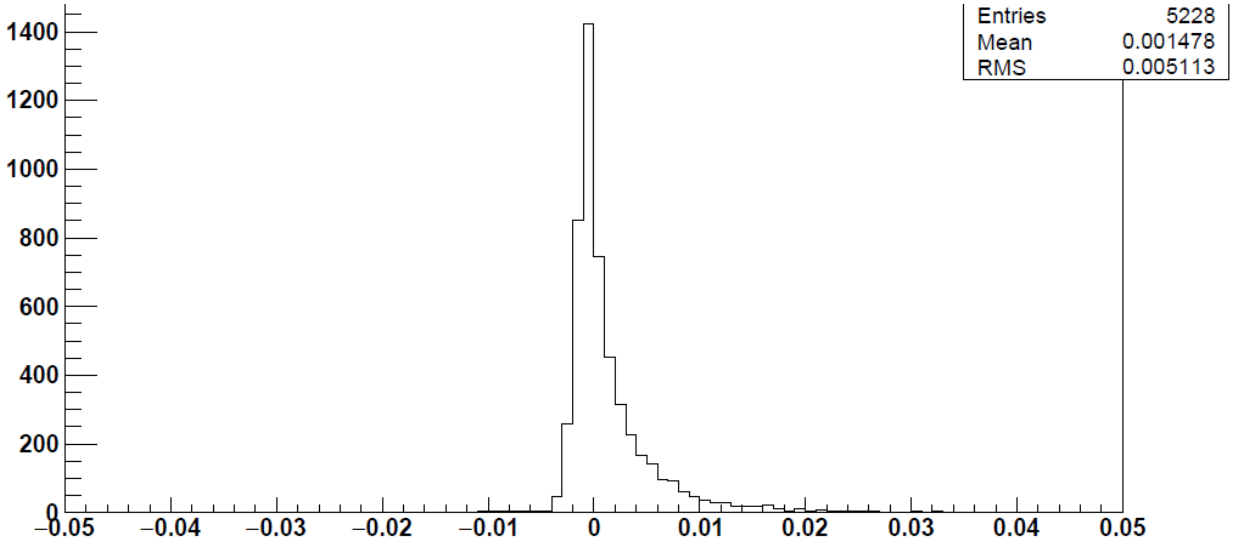
ASSESSING EXTRAPOLATION

- We have trackstate-at-sensor from GBL *only if track has hit at that sensor*
  - Then get proper intercept of this trackstate with sensor ("ideal" scheme)
  - Fieldmap effects negligible here

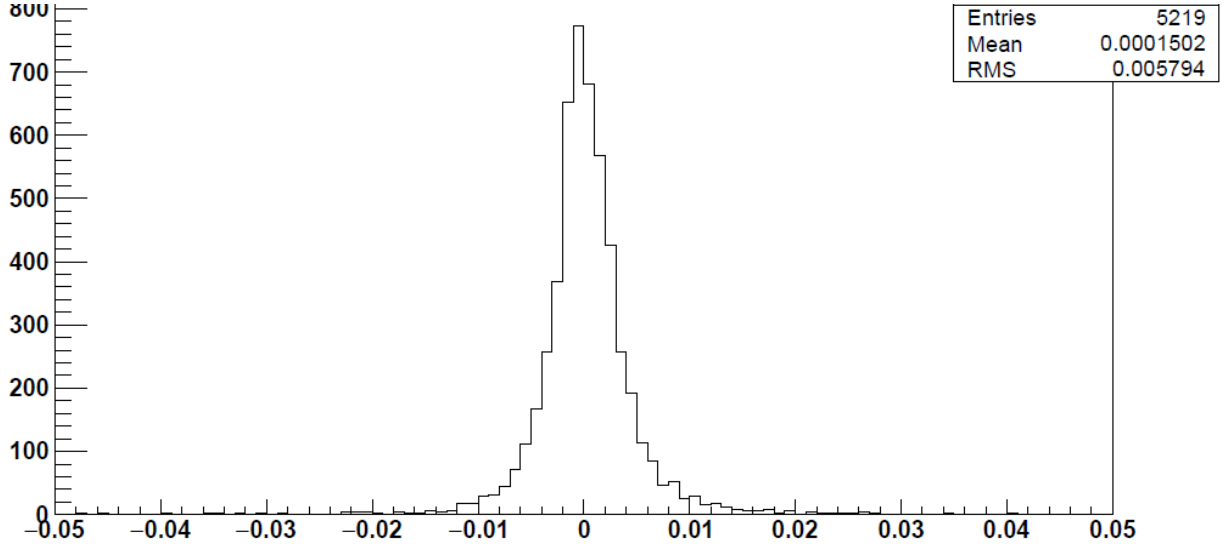
- Extrapolation often used for tracks missing Layer 1 hit, to determine whether they're in Layer 1 acceptance
- Prompt A' MC Study: try current ntuple extrapolation on track that actually has a Layer 1 hit, and compare result to ideal scheme (using trackstate at Layer 1)

ASSESSING EXTRAPOLATION

[ntuple extrapolation scheme] – ["ideal" extrapolation scheme] , Layer 1



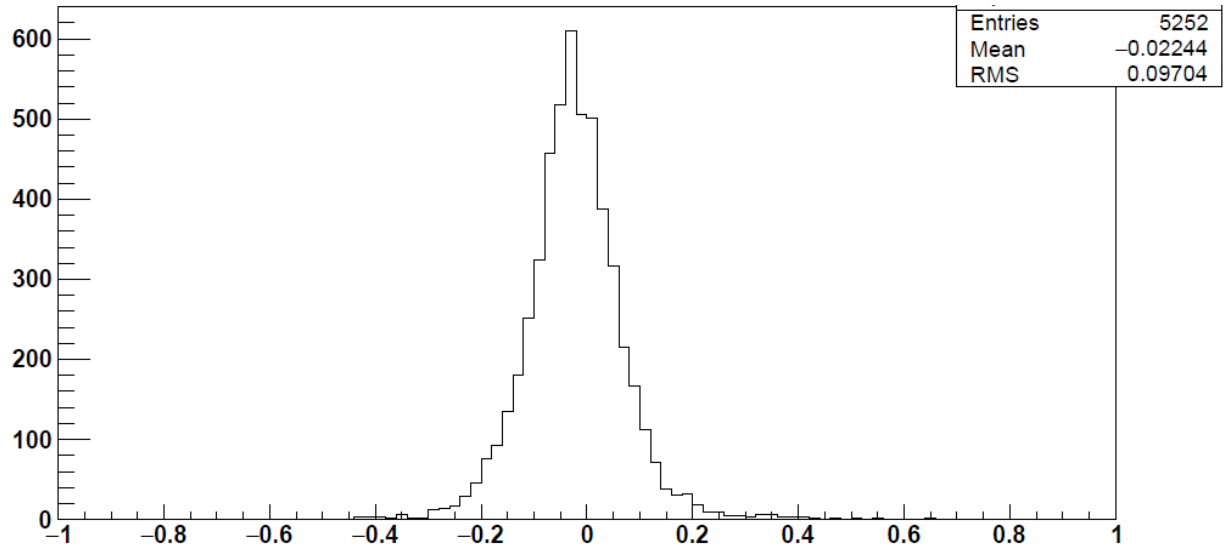
Extrapolation Position Difference, x [mm]



Extrapolation Position Difference, y [mm]

ASSESSING EXTRAPOLATION

[ntuple extrapolation scheme] – ["ideal" extrapolation scheme] , Layer 1



Extrapolation Position Difference, z [mm]

- How might this affect analysis results?
  - Is this type of study worth trying on various types of MC and/or data?
  - For other layers?
  - Apparently, old ntuples actually extrapolated to average z plane (halfway between axial and stereo sensors) for each layer?

EXTRAPOLATION BUG

hps-tracking ▸ src/main/java ▸ org.hps.recon.tracking ▸ TrackUtils ▸ getHelixPlaneIntercept(HelicalTrackFit, Hep3Vector, Hep3Vector, double, double) : Hep3Vector

also called by:

- lcsim-tracking ▸ src/main/java ▸ org.lcsim.recon.tracking.seedtracker ▸ HelixFitter ▸ FitCandidate
  - hps-tracking ▸ src/main/java ▸ org.hps.recon.tracking ▸ MultipleScattering ▸ FindScatters
    - hps-tracking ▸ src/main/java ▸ org.hps.recon.tracking ▸ MultipleScattering ▸ FindHPSScatterPoints
      - hps-tracking ▸ src/main/java ▸ org.hps.recon.tracking ▸ MultipleScattering ▸ getHelixIntersection

- hps-tracking ▸ src/main/java ▸ org.hps.recon.tracking.gbl ▸ MakeGblTracks ▸ makeStripData
- hps-tracking ▸ src/main/java ▸ org.hps.recon.tracking.gbl ▸ GBLOutput ▸ printGBL

Only getHelixIntersection contains:

```
// TODO Catch special cases where the incidental iteration procedure seems to fail
if (Math.abs(helix.R()) < 2000 && Math.abs(helix.dca()) > 10.0) {
    if (_debug) {
        System.out.printf("%s: momentum is low (p=%f,R=%f,B=%f) and d0 is big (d0=%f), skip the iterative calculation\n"
```

Absence of this check = root cause of https://github.com/JeffersonLab/hps-java/issues/243 "Matrix is Singular" exception?