Tracking and Vertexing

Norman Graf (SLAC)

HPS Collaboration Meeting @ JLab
May 4, 2017

The Global View

- Glass half full:
  - The tracking and vertexing works…
    - Omar’s Thesis
    - Sho’s Thesis
    - Unblinded 2015 Bump Hunt Analysis
- Glass half empty:
  - We could be doing (much) better
    - Better & faster simulation
    - Better & faster reconstruction
    - Better tracking efficiency
    - Track and vertexing resolutions

Tracking Simulation

- Definition of the SVT support structures could (should?) be improved
- Geometry description has gotten very complicated (just ask Matt Solt)
  - Can this be simplified/improved?
- Will need vastly more MC events for the 2018 running.
- Can we be smarter in our production?
  - Faster?
  - Better?
  - Use biasing to study edge cases (e.g. WAB conv.)

Fast Monte Carlo

- Takashi has a simple detector simulation which has proven to be very useful in quickly answering a number of questions related to acceptance and occupancies
  - Geant3-based
  - Uses Z scoring planes to record particle positions at rough location of SVT sensors and ECal
- It would be useful to incorporate this functionality more closely into our software/analysis framework
  - Tighter coupling of existing framework with HPS geometry and EDM
  - Reproduce functionality within hps-java

Track Reconstruction Software

- Track finding and fitting were adapted from software developed for generic collider detectors
- Adoption of this software allowed rapid development during the design phase of HPS but required a few compromises
  - Use of a generic geometry definition and pattern-recognition system.
    - Fast for development, not optimized for production.
  - Rotation of our coordinate system to spoof a solenoidal field
  - Use of track parameters not natural for a fixed-target geometry.

Software CPU Performance

- Our tracking software is SLOW!
  - Not currently an issue, but will definitely become critical during long 2018 run
  - Have detailed profiling data, but there has been no appreciable action to-date
  - Overall CPU budget dominated by tracking, primarily track-finding/fitting, followed by raw hit-fitting
    - Fix what we have
    - Start over
- Fitting readout samples to determine hit time and pulse height
  - Currently using generic minuit fit
  - Need to evaluate possible gains from a dedicated fitter
  - Fit once in pass0, don't refit in later passes.

Pattern Recognition

- Two nice talks by Matt and Holly addressing track finding efficiency.
- Currently begin by creating 3D spacepoints from stereo pairs and finding tracks *ab initio* via strategies.
- Revisit our track-finding strategies to incorporate new L0 in the upgrade
- Support “4-layer” tracking, both to identify WABs and recoil electron tracks

Pattern Recognition

- Possible improvements:
  - Improved axial/stereo matching (L4-L6)
  - Improved and/or more strategies using 3D points
  - Cluster-seeded tracking
    - ECal cluster position and energy define a trajectory which originates from the beam-spot (HPS Note 2015-006).
    - Find tracks consistent with that hypothesis.
  - Implement pattern recognition based on 1D strip hits.
    - No “ghost” hits, or parallax issues
    - Could see increased efficiency by not requiring hits in both axial and stereo layers per station.

Adaptive Methods

- Track and vertex reconstruction: From classical to adaptive methods

  Are Strandlie and Rudolf Frühwirth

  Rev. Mod. Phys. 82, 1419 (2010)

"*Adaptive methods have been developed to meet the experimental challenges at high-energy colliders, in particular, the CERN Large Hadron Collider. They can be characterized by the obliteration of the traditional boundaries between pattern recognition and statistical estimation, by the competition between different hypotheses about what constitutes a track or a vertex, and by a high level of flexibility and robustness achieved with a minimum of assumptions about the data.*"

Adaptive Methods

- Kalman Filter
- Deterministic Annealing Filter (DAF).
- Cellular Automaton
- Neural Network (e.g. Hopfield)
- …

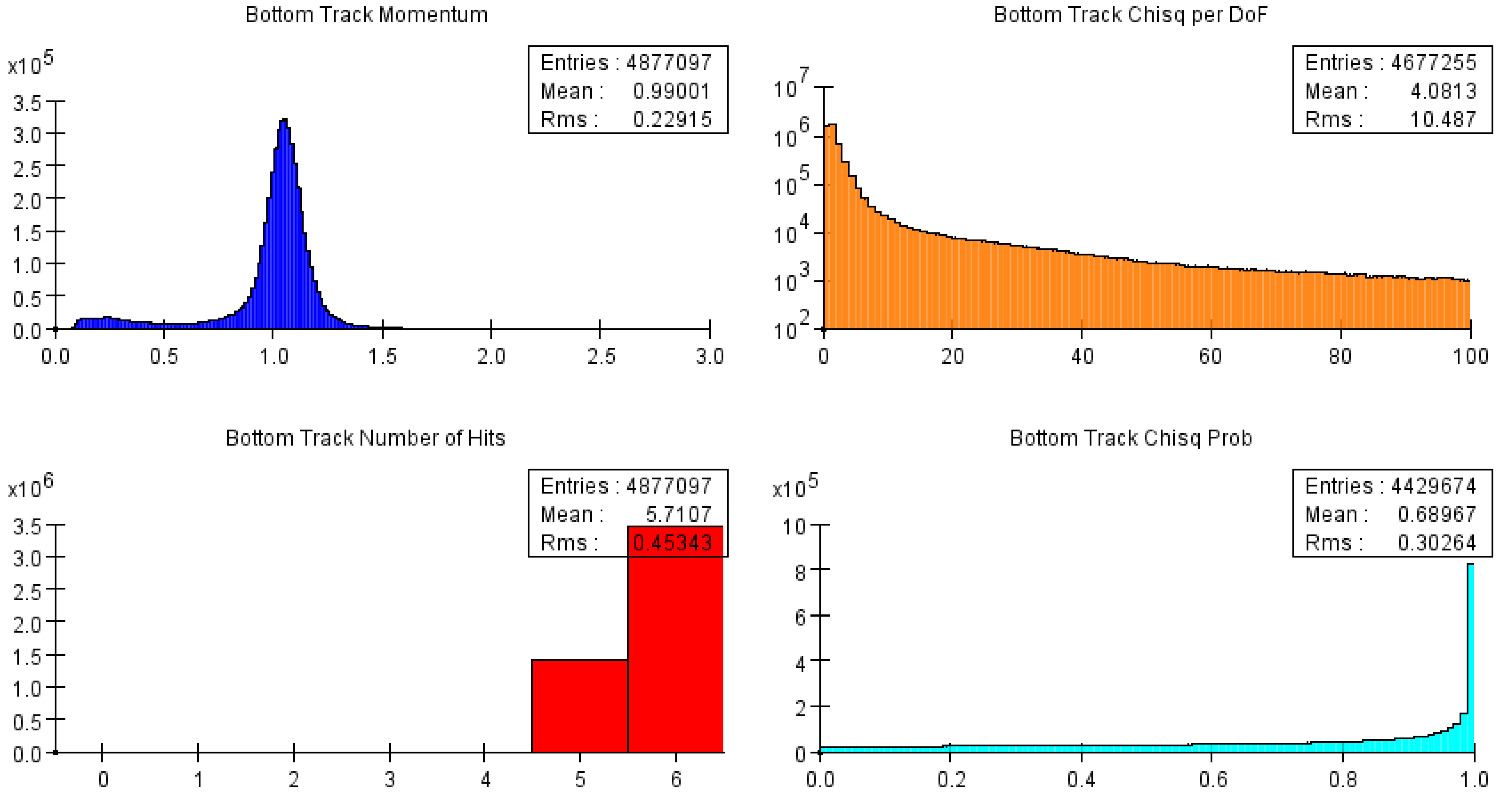
- Lots of interesting ideas

Track Fitting

- Track fit quality is not chi-squared distributed
- Discrepancy between data and MC
- Resolution of issues complicated by:
  - Strip cluster position
  - Module position (alignment)
  - Track extrapolation (non-uniform field)
  - Multiple scattering and energy loss
- GBL refit could benefit from external review
- Whole chain needs better documentation
  - Javadoc on what the code is expected to be doing
  - Documentation on procedure, algorithm, math

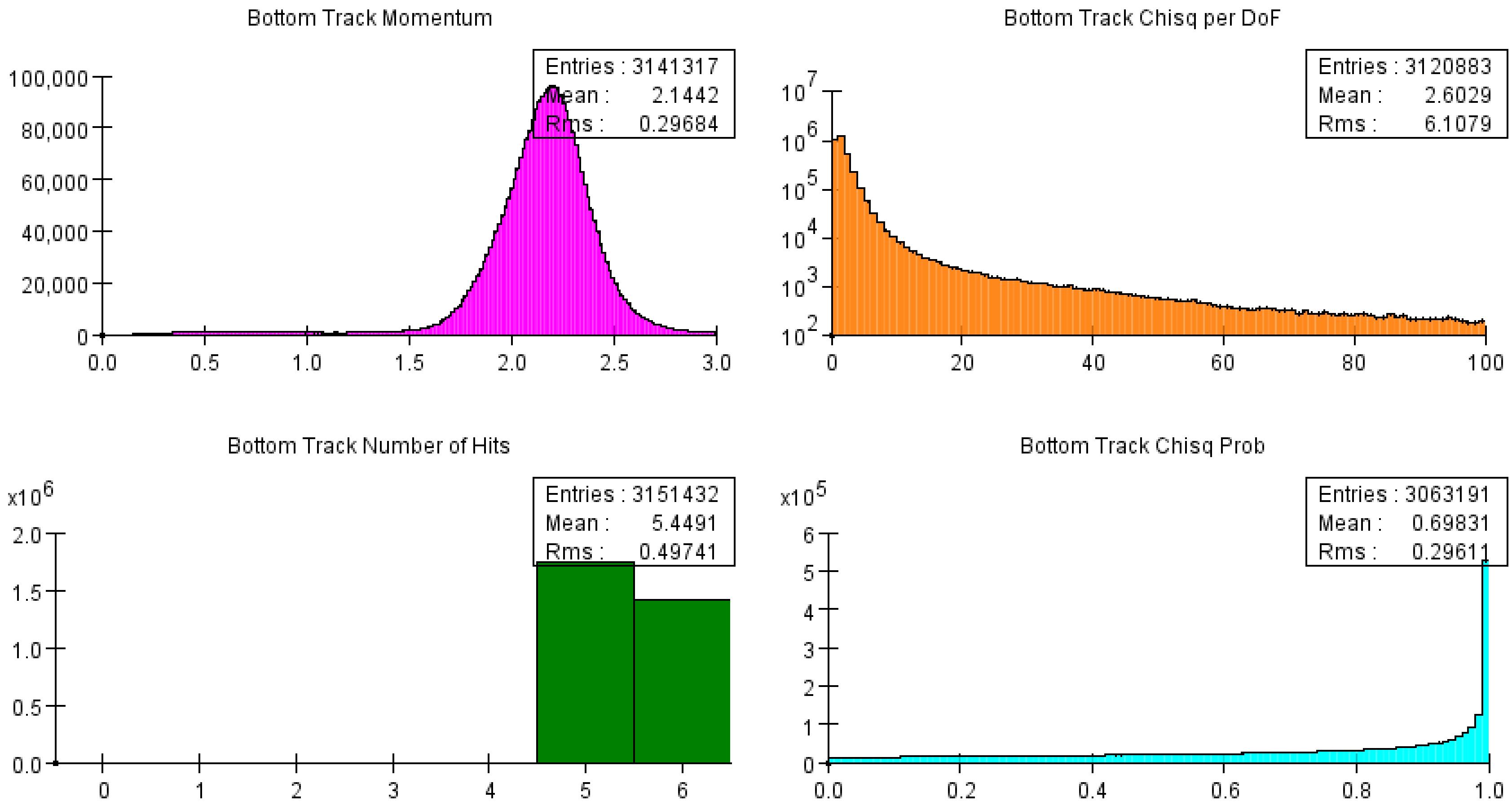
Track Quality 101

- FEE candidates from 2015



Track Quality 101

- FEE candidates from 2016



Track Chi-squared

- Multiple sources:
  - Incorrect Hit positions (alignment)
  - Incorrect Hit uncertainties
  - Incorrect handling of Multiple Scattering/Energy Loss
  - Incorrect handling of Track Propagation
  - Bugs in the code
- Start at the beginning
  - Just heard a very nice presentation by Alessandra on internal alignment of the SVT modules using the data.
  - Still a lot of work ahead.

Alignment

- Currently use GBL track fits fed to millipede II
  - Hold some layers constant, float others to minimize χ^2
  - Provides internal alignment of Si sensors
  - Alessandra just gave a great talk with all the details
- Need to tie this both to the ECal and Lab coordinate system
- Need to constrain weak modes

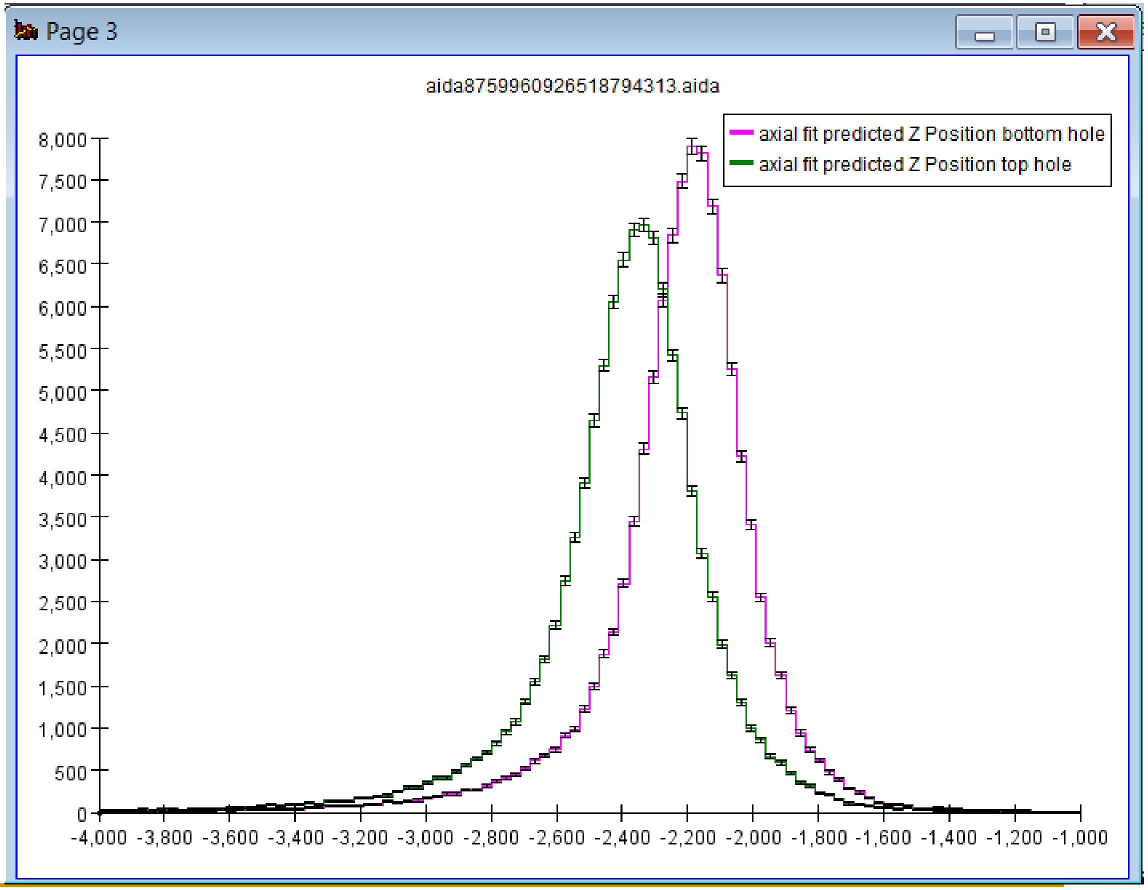
Alignment Moving Forward

- Include beam spot (and ECal?) into alignment procedure using single-track
- Include vertex constraint for multiple track events
- Include vertex and mass constraint for Møller events

- Ties SVT coordinate system to HPS lab system
- Couples top and bottom halves of detector
- Constrains weak (momentum) mode
- (Some ad-hoc corrections to deal with some of these issues were introduced into tweakpass6)

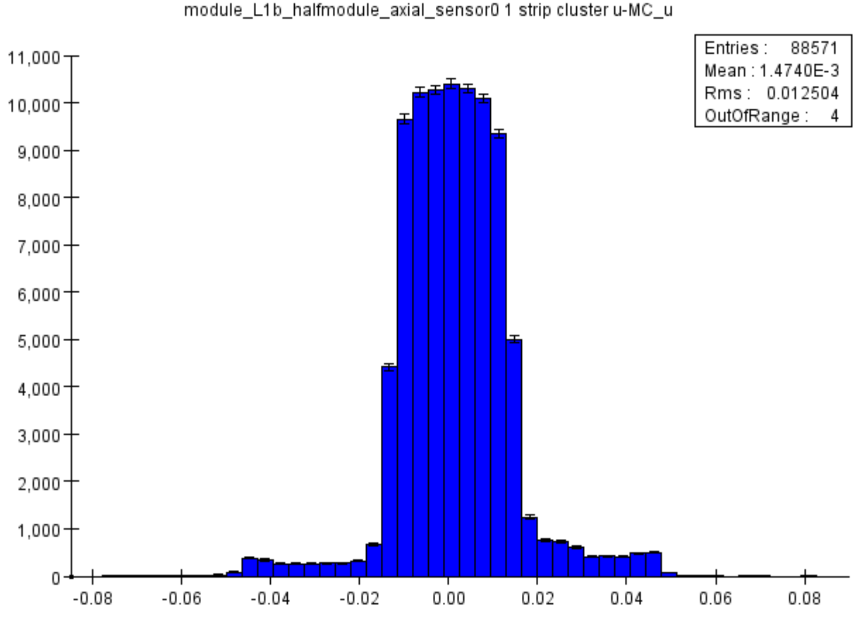
Field-Off Straight Tracks

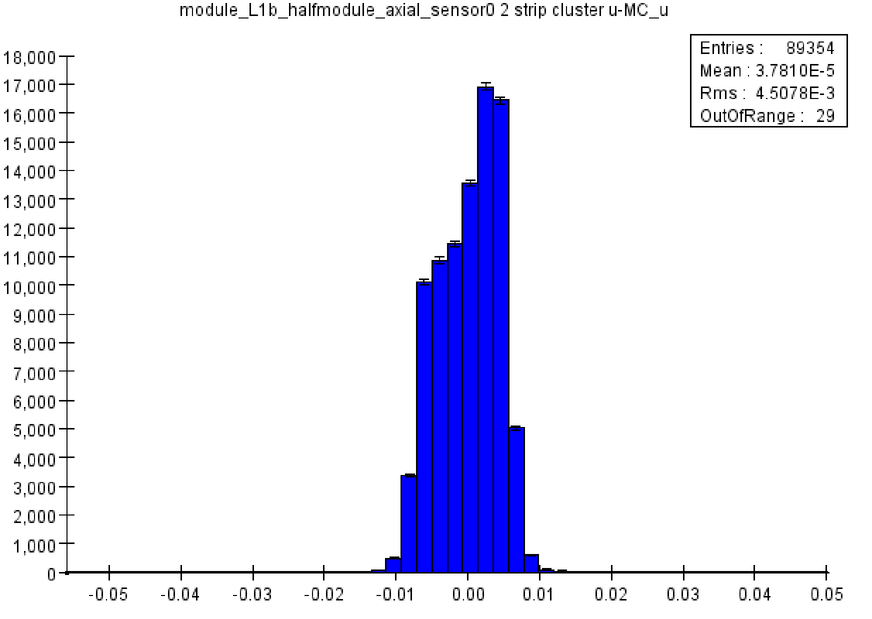
- Project tracks found in both top and bottom SVT back to intersection with Y-Z plane (y=0)
- Plot value of Z at intercept
- Top z = -2346
- Bottom z = -2180
- σ ~ 150



Track Chi-squared

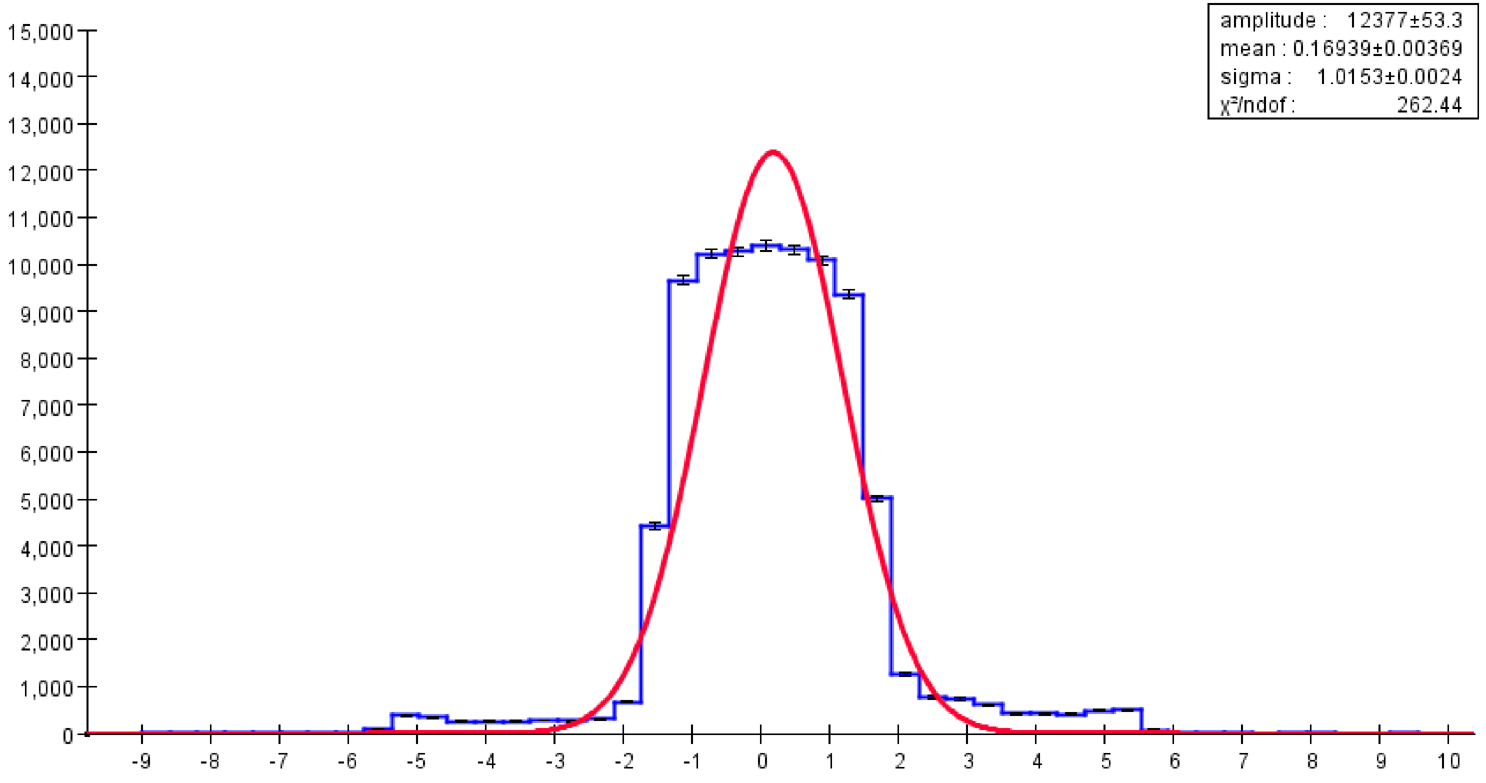
- Start at the beginning:
  - Using MC, look at residuals and pulls for Cluster Hits compared to known position of SimTrackerHits
  - No misalignment.





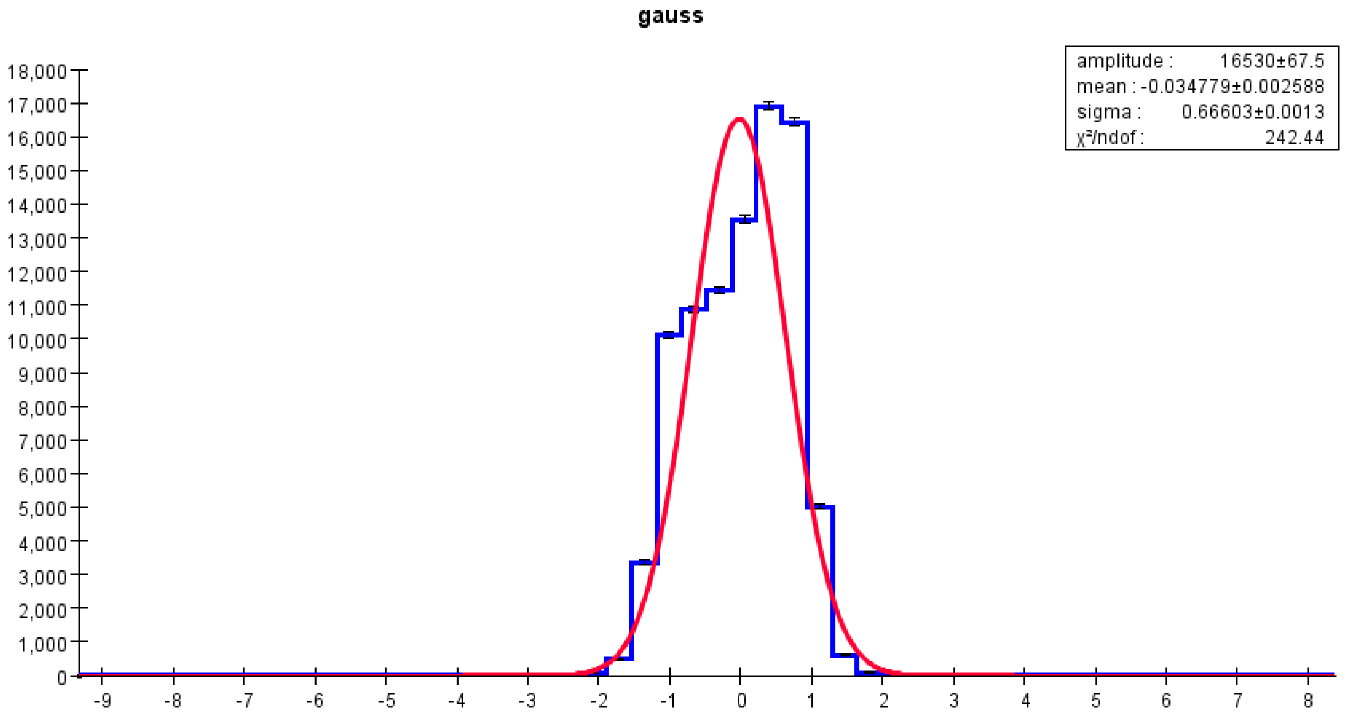
1 Strip Clusters

- See expected box distribution.
- Gaussian fitted to pull gives sigma ~1

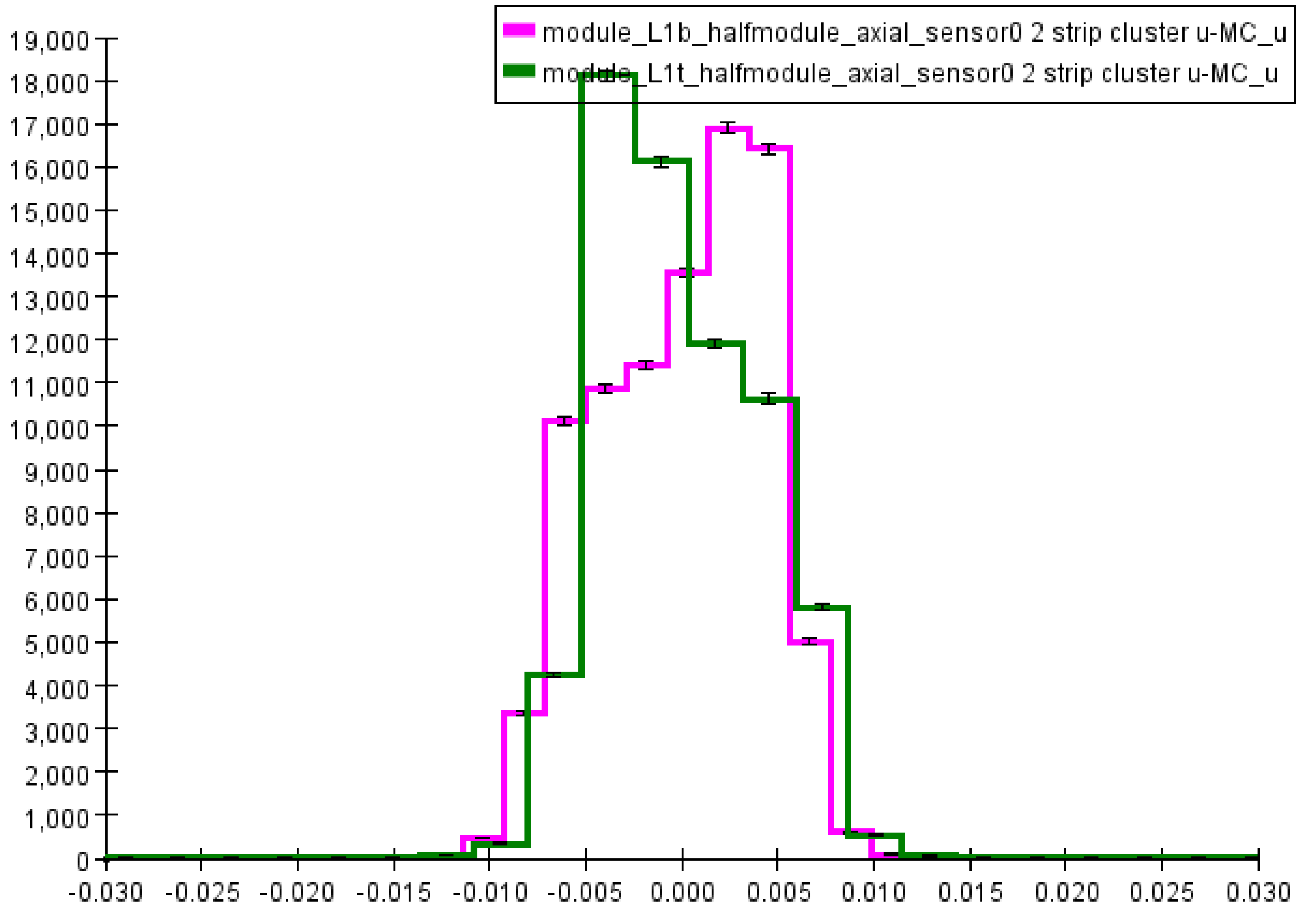


2 Strip Clusters

- Asymmetric residuals, Gaussian fits to pull give sigma ~ 0.6
- Puzzling
- Work in progress



L1t vs L1b 2-strip Cluster Resolution



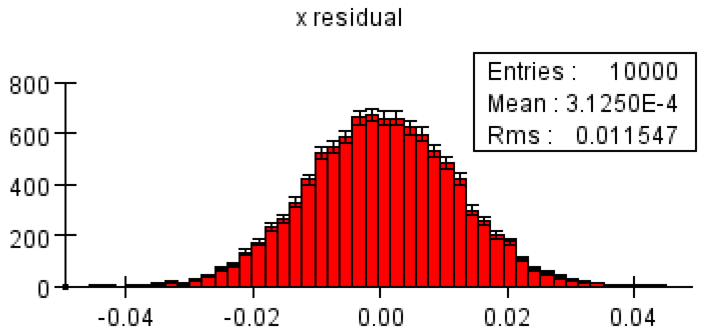
Tracking Fast MC Goals

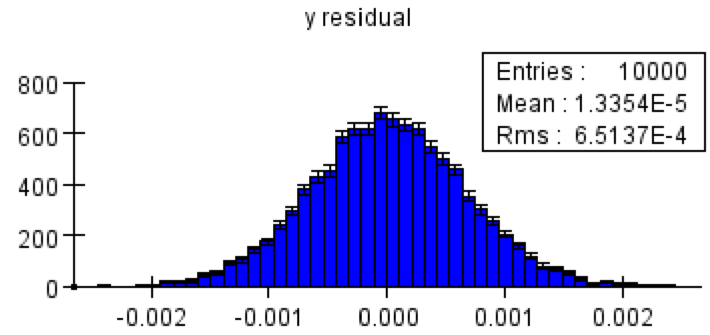
- Fast MC Short Term Goals
  - Use as tool to debug/understand track fitting
  - Create Tracks by swimming through vacuum and recording postion at scoring planes.
  - Smear MC position with Gaussian to produce hits.
  - Use MC info for pattern recognition.
  - Start with simple detector and check mechanics of track fit, then increase complexity of model (multiple scattering, magnetic field map, etc.) to see where problems arise.
- Fast MC Longer Term Goals
  - Use as tool to debug/understand vertex fitting
  - Use as tool to debug/understand alignment strategies
  - Use as tool to generate large statistics samples of events for analyses

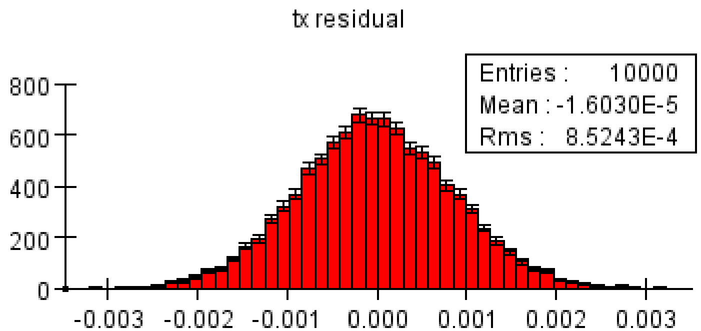
Fast MC Track Fitting

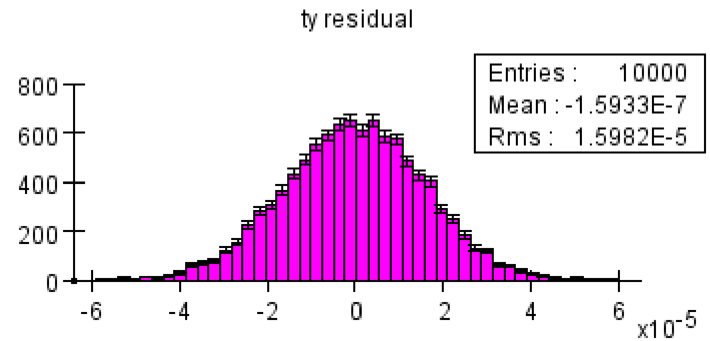
- Using Fast MC output to check track fitting, ensure no bugs or unexpected features.
- Generate single particle tracks
  - 1 GeV, (0.2, 0.3, -5.0), smeared (theta, phi)
- Swim to planes located nominally where HPS SVT sensors are located.
- Smear MC hit positions by Gaussian with fixed width.
- No material interactions, no scattering.
- Use Kalman Filter to fit list of hits, analyze resolution, pulls, chisquared.

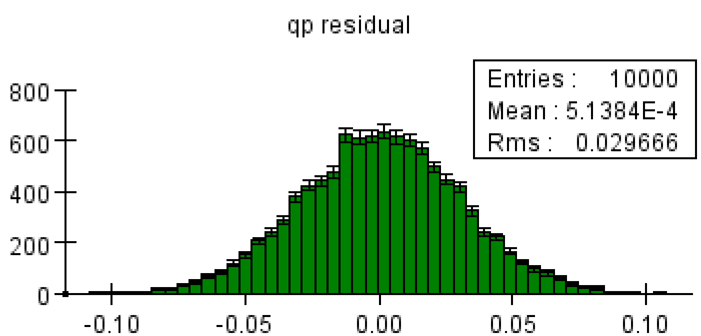
Track Parameter Residuals @ Target



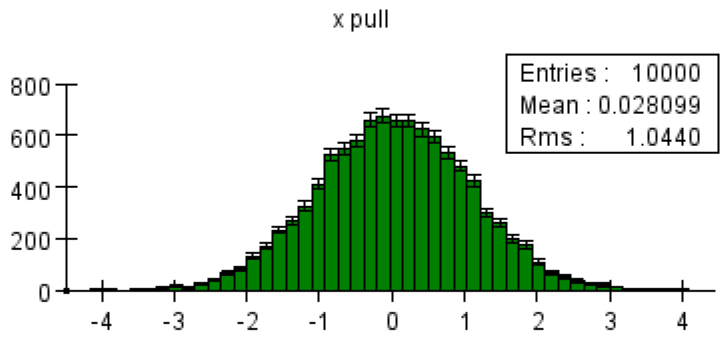


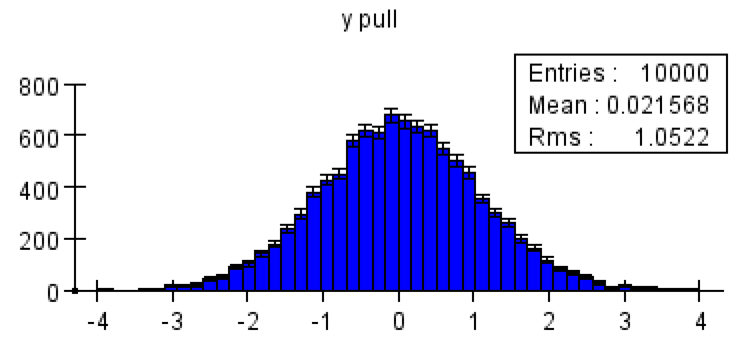


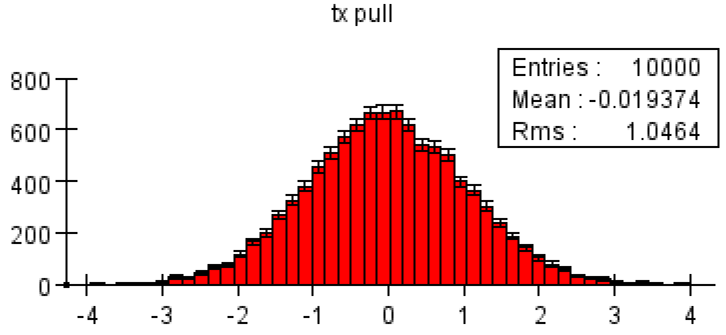


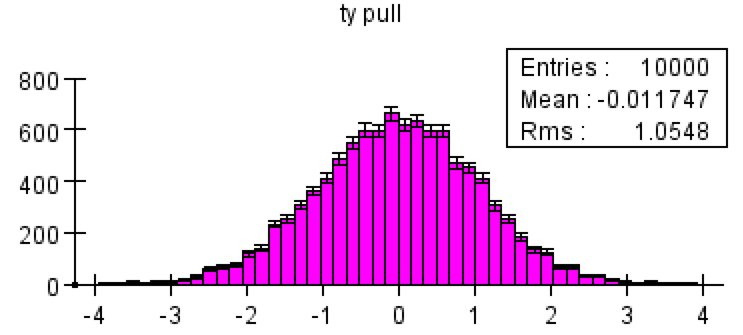


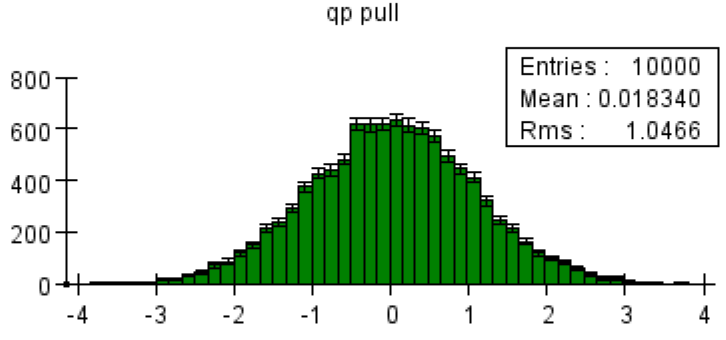
Track Parameter Pulls @ Target



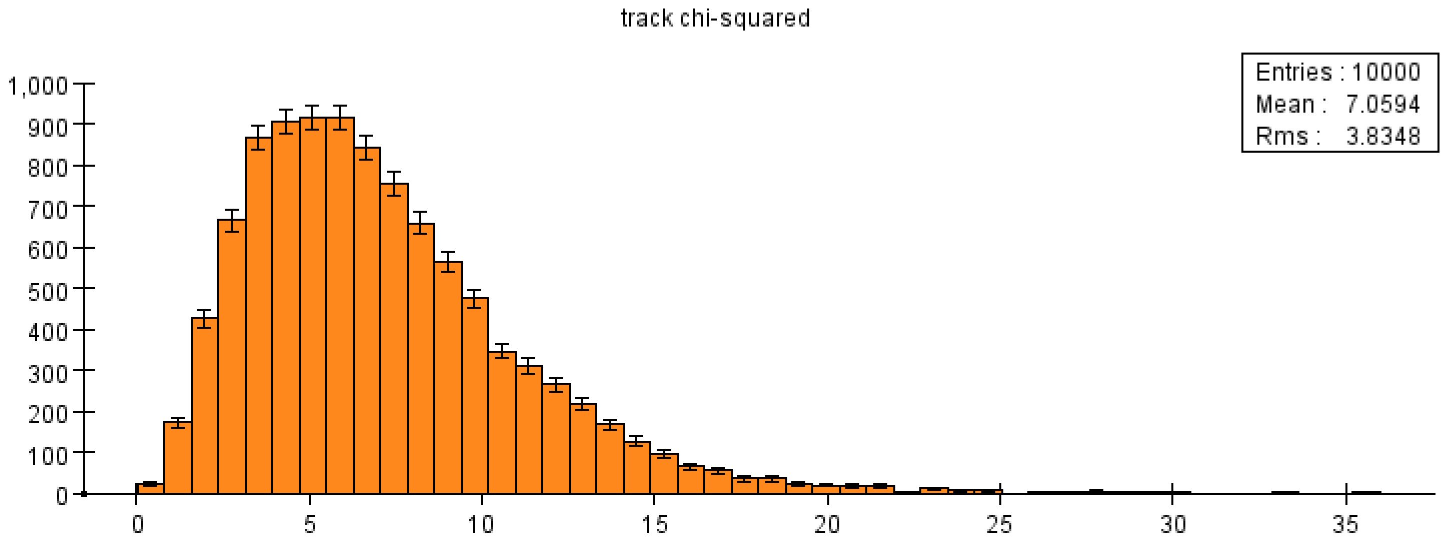


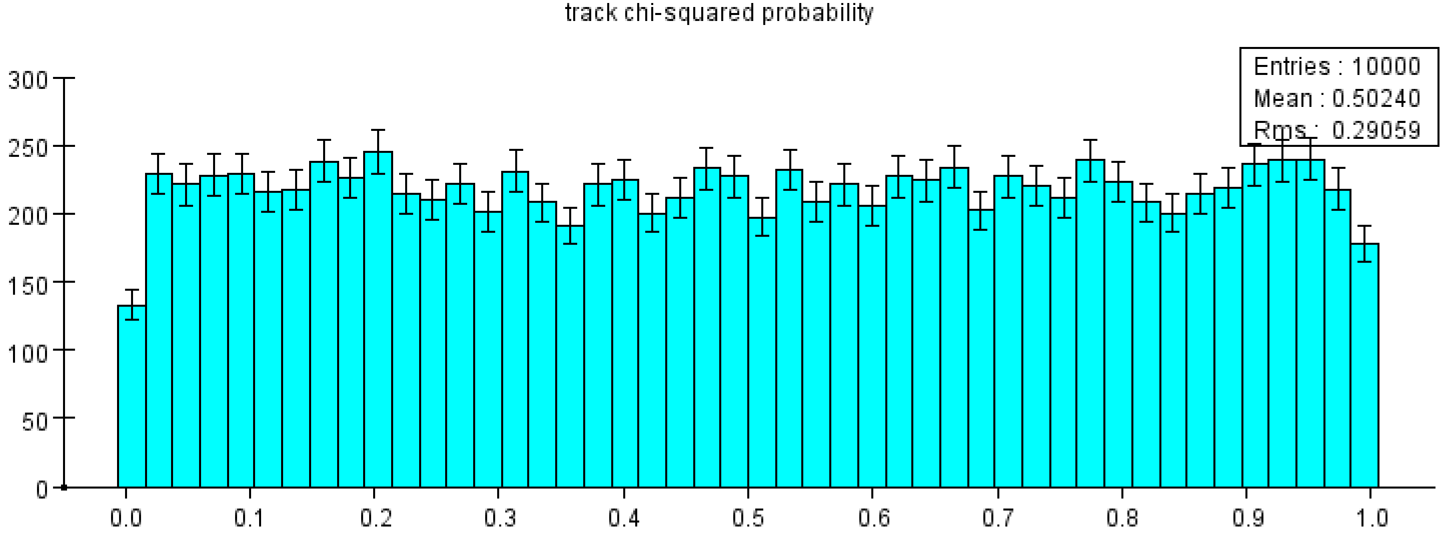






Track Fit Chisquared





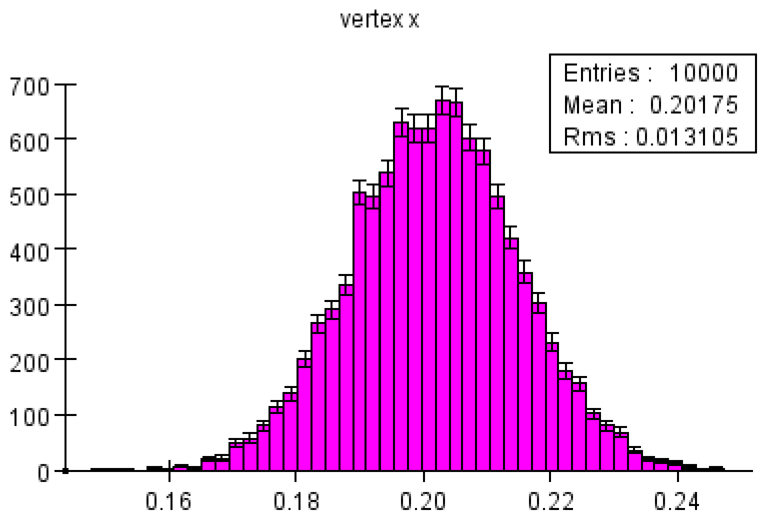
Vertex Fitting

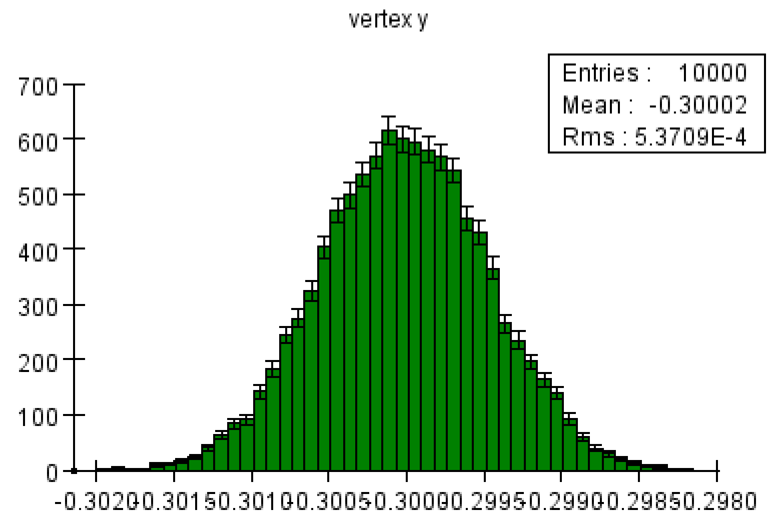
- Currently using Billoir's implementation of a Kalman fit using perigee parameters.
  - Presupposes that track states have been defined "close" to the actual vertex.
    - Our track states defined at z=0
  - Fitting assumes Gaussian uncertainties with correct covariance matrices
    - Current track parameter pulls not normally distributed
    - $\therefore$ no surprise that vertex resolution is off
    - $\therefore$ no surprise that vertex pulls not normally distributed
- Implementing another algorithm would be very useful as a cross-check.
- Documentation needed

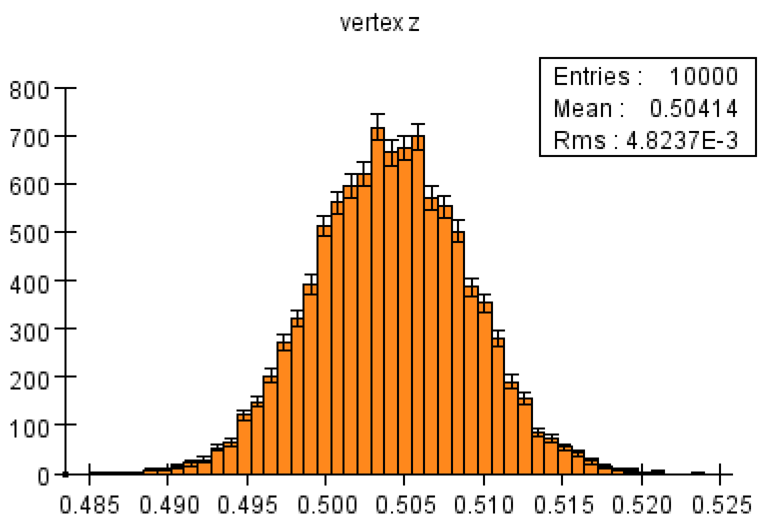
Fast Vertexing study

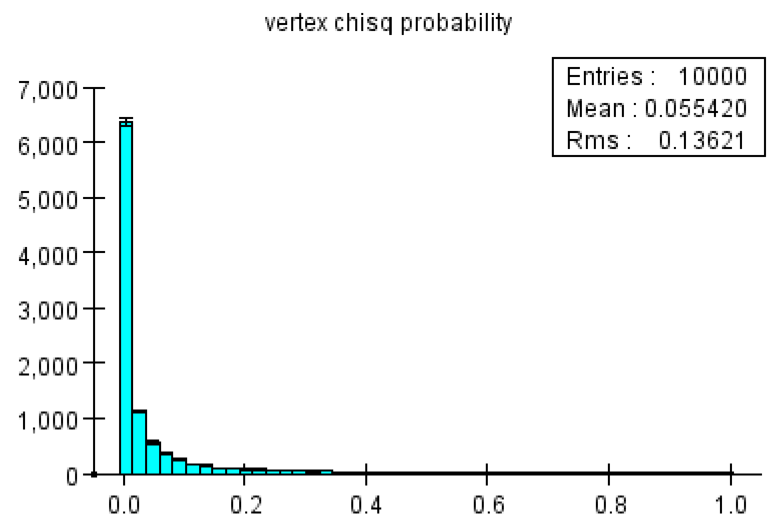
- Have fast Tracking MC that produces well-fit tracks. Do a quick study of vertexing displaced vertices.
- Generate events with displaced vertices.
- Use fastMC to simulate and fit tracks
- Vertex tracks
- Swim tracks to found vertex
- Re-vertex tracks

First Vertex Fit

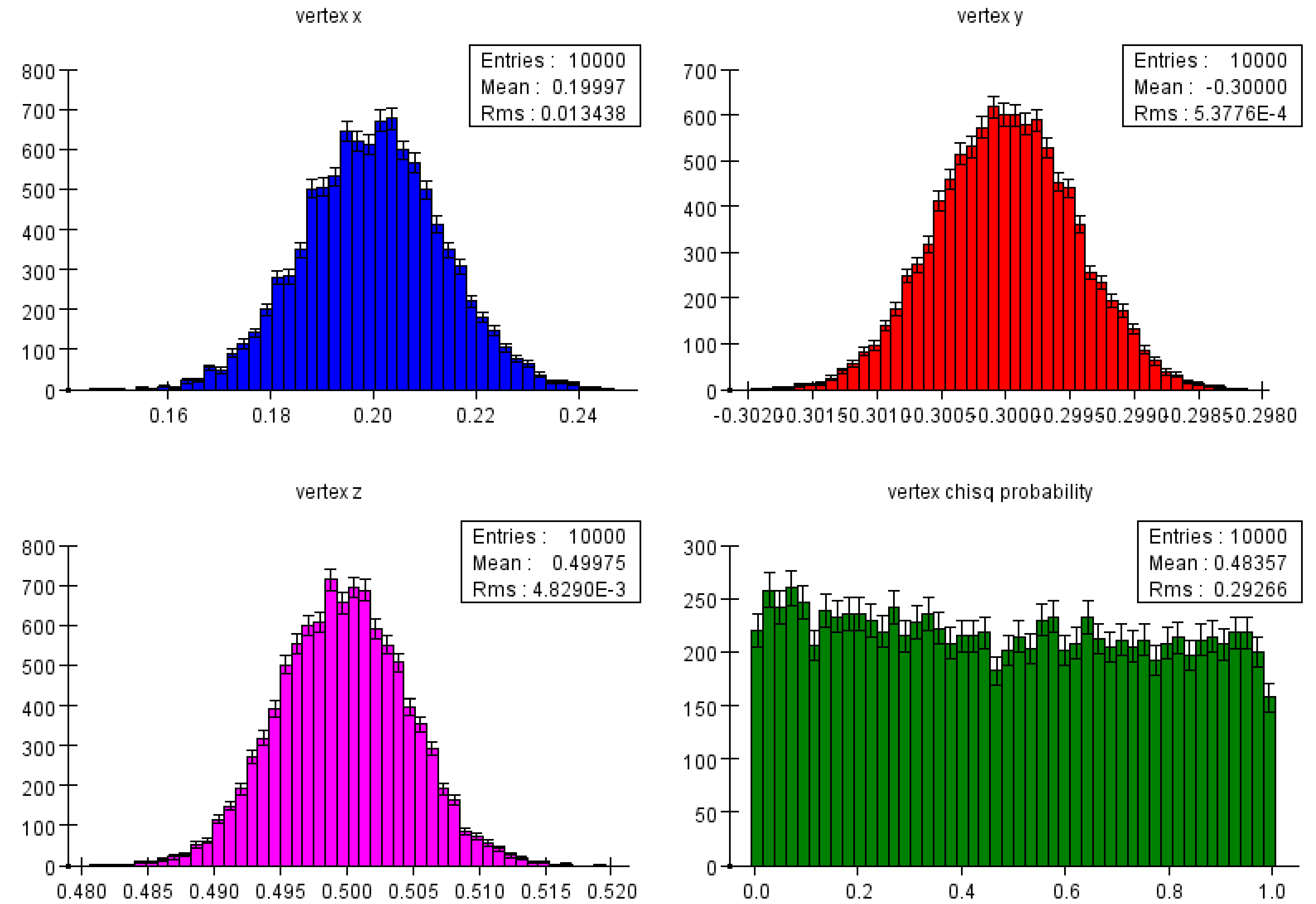




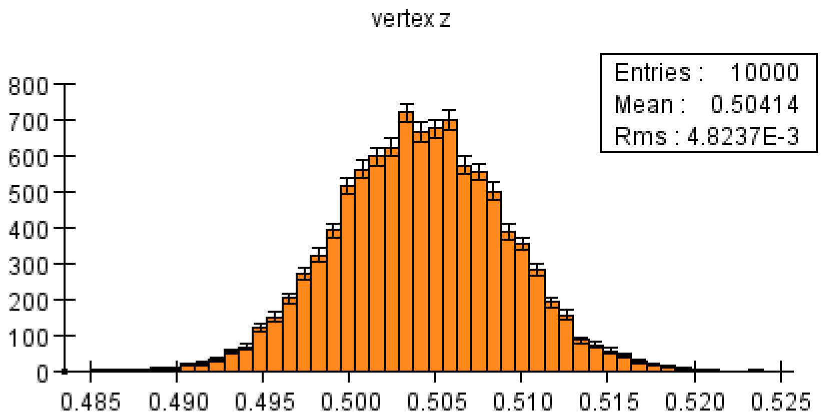


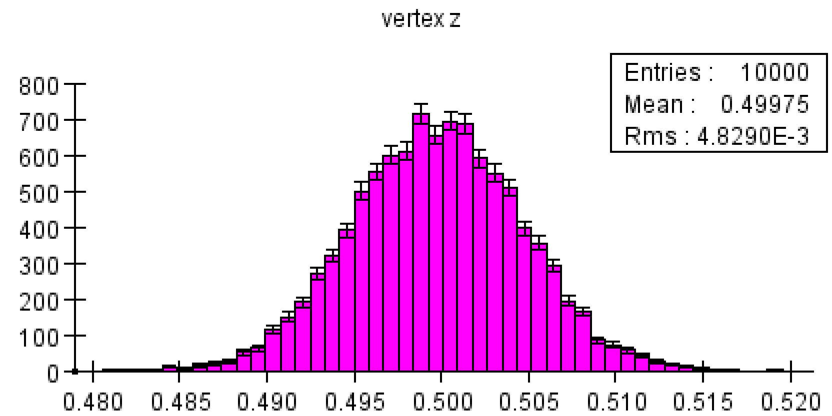


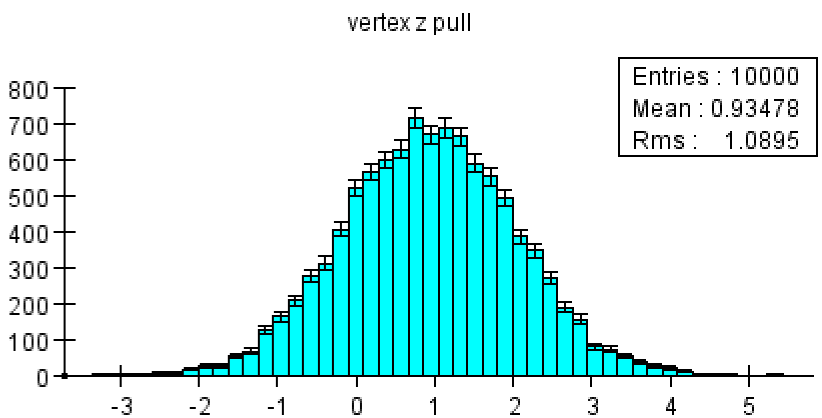
Refit Vertex Fit

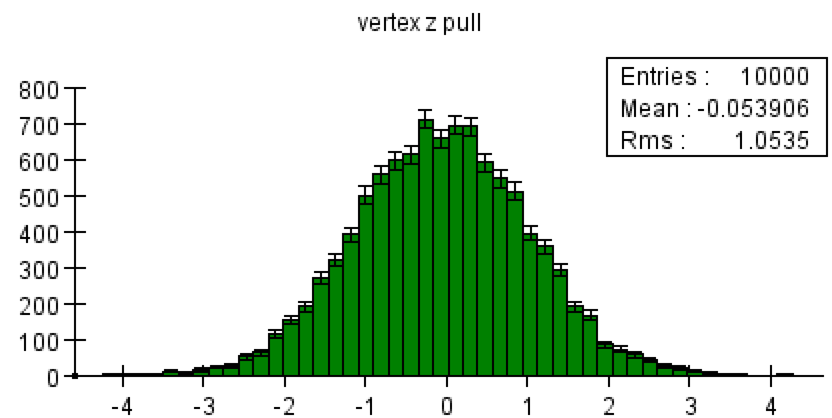


Vertex Pulls before and after refit









Manpower

- Many of the principal developers of the tracking/vertexing software have moved on
- Opportunities abound for individuals or institutions to contribute, either improving existing software or developing/implementing new code.

Summary

- Current code and algorithms are working, but…
- Improvements to the Tracking and Vertexing feed directly into improvements in the bump-hunt and vertex analyses and our discovery reach
- Major changes are unlikely for the analysis of data already taken, but will be necessary for physics run in 2018.
  - We have a little over a year to get ready.
    - + Much can be done in that time
    - - Much needs to get done in that time
- Lots of ideas for improvement
- Great opportunities for new contributors

Meetings & Results Pages

- Weekly meetings to discuss both hardware and software tracking issues
  - Subscribe to mailing list
- 2015 Performance Studies
  - Confluence page with task list and performance plots related to analysis of the 2015 data
- 2016 Performance Studies
  - Confluence page with task list and performance plots related to analysis of the 2015 data