Track Find and Fitting

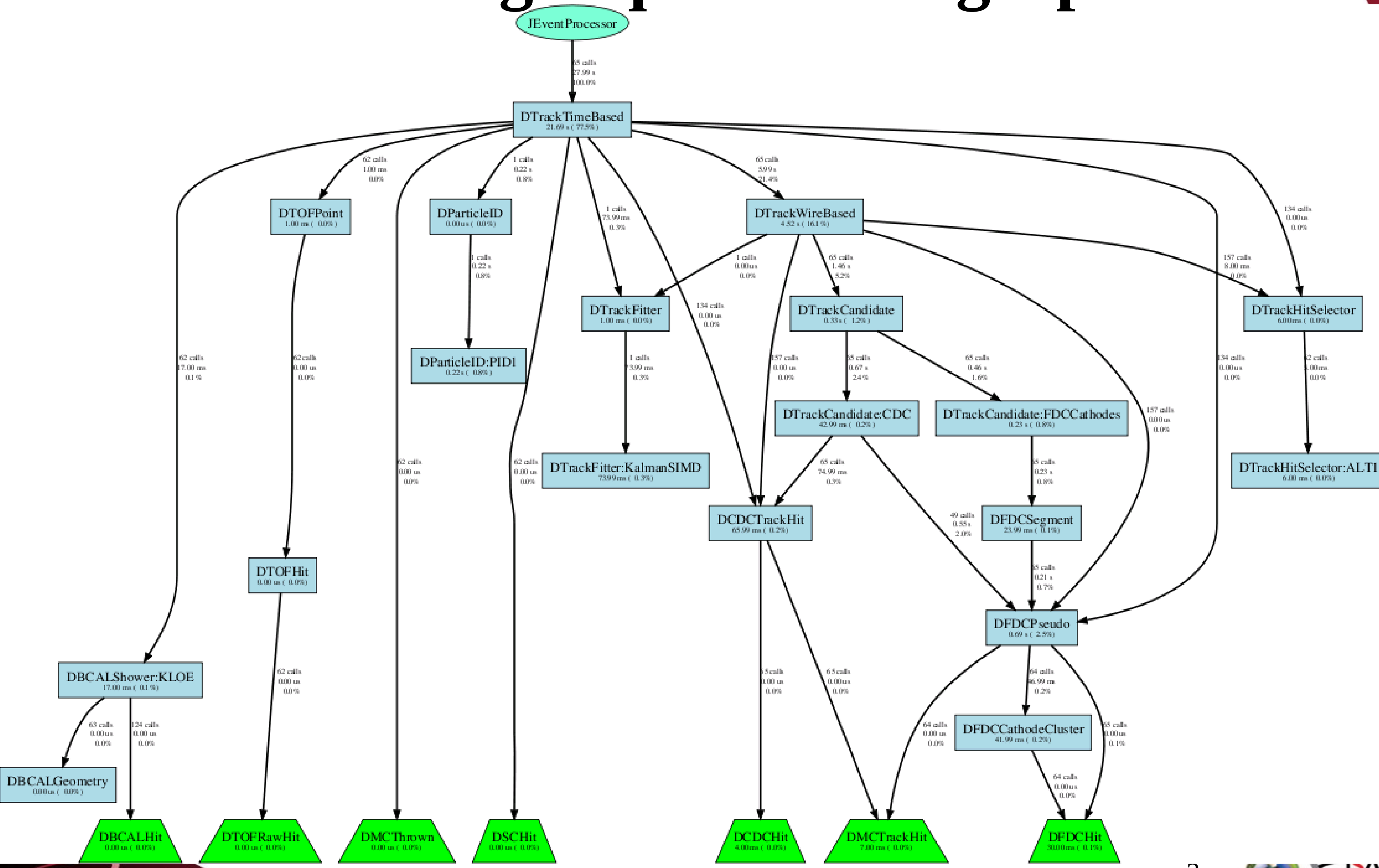
Simon Taylor /JLab

- Track finding
- Wire-based fitting
- Time-based fitting

Tracking Sequence in words

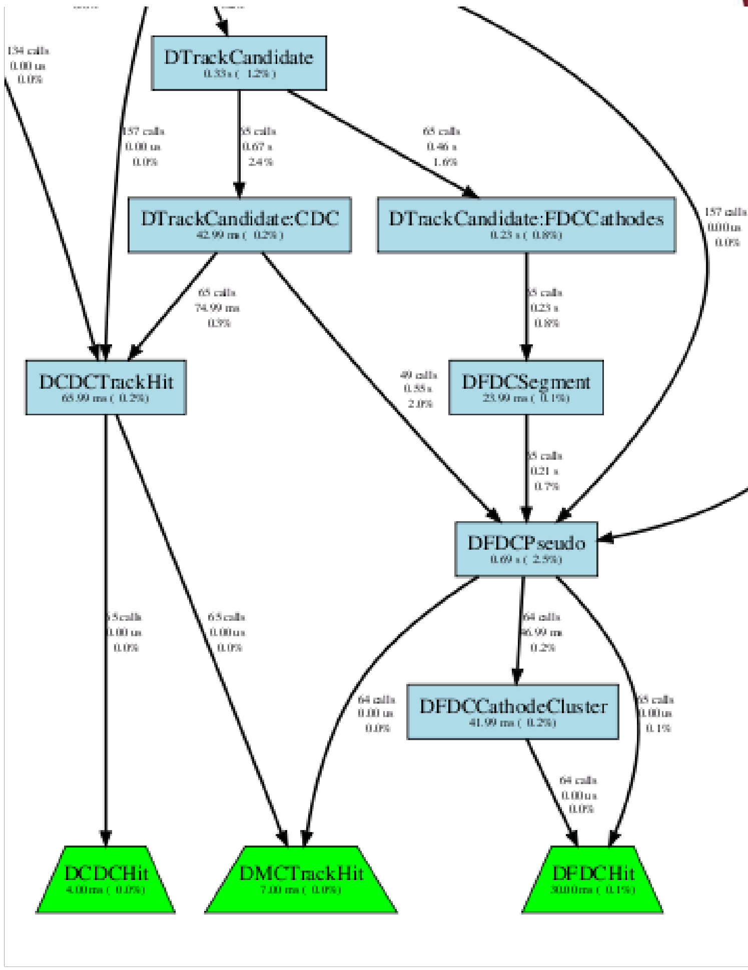
- Step 1: Track finding
  - Helical model – assume B-field is constant
  - Look for clusters of hits close in position in adjacent (or close by) layers to form segments
    - No drift time information used at this time
  - Link segments together to form "track candidates"
  - Fit with Riemann Helical Fit to obtain estimates for tracking parameters (momentum and position at the vertex)
- Step 2: Wire-based fitting
  - Fit the track candidates using the full knowledge of the non-uniform magnetic field but still without drift time data
  - Fit multiple times with different mass hypotheses
    - {Proton, π^+} for positive particles, π^- for negative particles
    - Take into account energy loss (important for low-momentum particles)
- Step 3: Time-based fitting
  - Refit wire-based tracks using drift time information

Tracking sequence call graph



Step 1: Track Finding

- Find tracks in FDC; →*DTrackCandidate:FDCCathodes*
- Find tracks in CDC: →*DTrackCandidate:CDC*
- Add lists together, merging candidates common to FDC and CDC: →*DTrackCandidate*
  - To select hits, code looks for nearest neighbors in adjacent layers within some radius
  - Preliminary fit with helical model to find initial guess for track parameters



Track finding in CDC

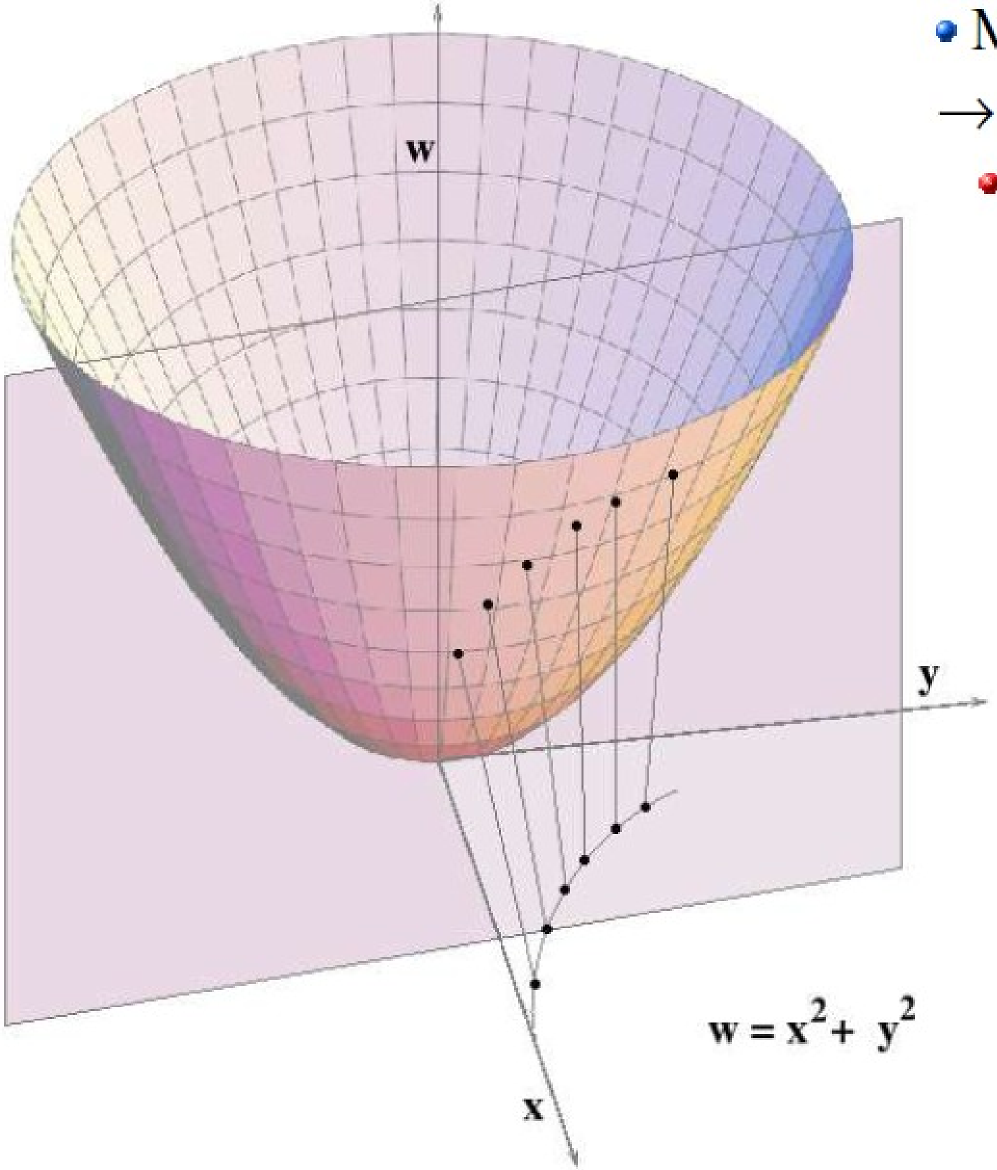
- Link hits together in each axial layer to form “seeds”
- Link seeds between axial layers together
- Find intersections between axial layers and stereo layers
- Use helical model to form track candidates → *DTrackCandidate:CDC*

Track finding in FDC

- Associate adjacent cathode hits into clusters $\rightarrow$ *DFDCCathodeClusters*
- Find centroids in cathode planes and match strips with wires to form space points $\rightarrow$ *DFDCPseudo*
- Link space points in adjacent layers in each package into track segments $\rightarrow$ *DFDCSegment*
- Link segments by swimming through the field from package to package to form track candidates $\rightarrow$ *DTrackCandidate:FDCCathodes*

Obtain preliminary momentum and position parameters for input to wire-based fit using Riemann Helical Fit

Riemann Helical Fit



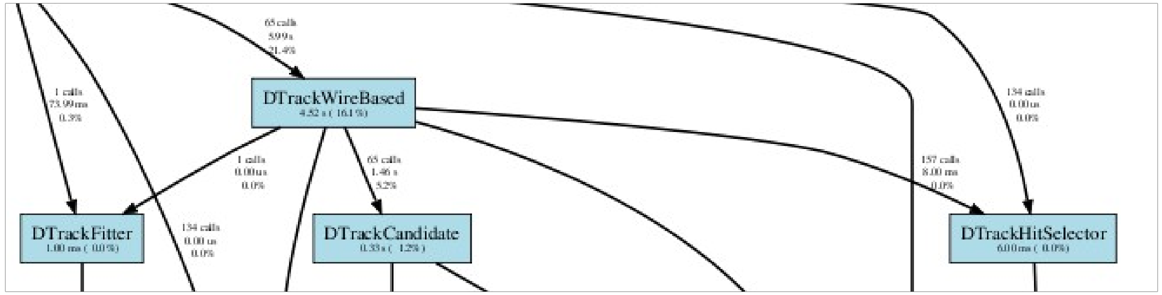
- Map points on circle in *xy*-projection
→ points on circular paraboloid surface
  - Finding center, radius of circle → fitting plane in *xyw*-space

- Extension to helical path:
  - Compute arc lengths *s* from point to point in *xyz*-space
  - Linear regression of *s* on *z* → tangent of dip angle *tan* λ

Class= *DHelicalFit*

R. Fruhwirth, et al., NIM **A490**(2002)366-378.

Step 2: Wire-based track fitting



- Starting with parameters from *DTrackCandidate*, swim out to select hits (using *DTrackHitSelector*) belonging to the track to be used in the fit
- Fit (with the algorithm chosen for *DTrackFitter*) using the wire positions for coordinates of the CDC hits and the position along the wire computed from the cathodes for the FDC$\rightarrow$ *DTrackWireBased*

Track Fitting Algorithm

DTrackFitter:KalmanSIMD_ALT1

- Kalman Filter is engine of Wire-based and Time-based fitting
- Track described by 5-parameter state vector at each point along its path
- State vector propagated step-by-step toward target
    - Account for multiple scattering and energy loss at each step
    - Does not require inversions of large matrices (at most 2x2 for FDC hits)
    - Jacobian matrix elements computed analytically
- State vector treated as a perturbation to an initial guess
    - Generate reference trajectory by swimming with initial parameters from the target
- Iterate up to 20 times, checking for χ^2 convergence
    - Regenerate reference trajectory based on results of previous pass

Kalman Filter Algorithm

Start with guess from track finding or wire-based stages

Seed S^0

Propagate state S and covariance C through magnetic field

$$S_k^{k-1} = S_k^{k-1,0} + F_k(S_{k-1}^{k-1} - S_{k-1}^{k-1,0})$$
$$C_k^{k-1} = F_{k-1} C_{k-1}^{k-1} F_{k-1}^T + Q_{k-1}$$

Multiple scattering
Energy loss

Use measurements to update state vector

$$K_k = C_k^{k-1} H_k^T \left[V_k + H_k C_k^{k-1} H_k^T\right]^{-1}$$
$$S_k^k = S_k^{k-1} + K_k \left(m_k - h(S_k^{k-1})\right)$$
$$C_k^k = C_k^{k-1} + K_k H_k C_k^{k-1}$$

Use result (S) of iteration as new seed

Forward Tracking

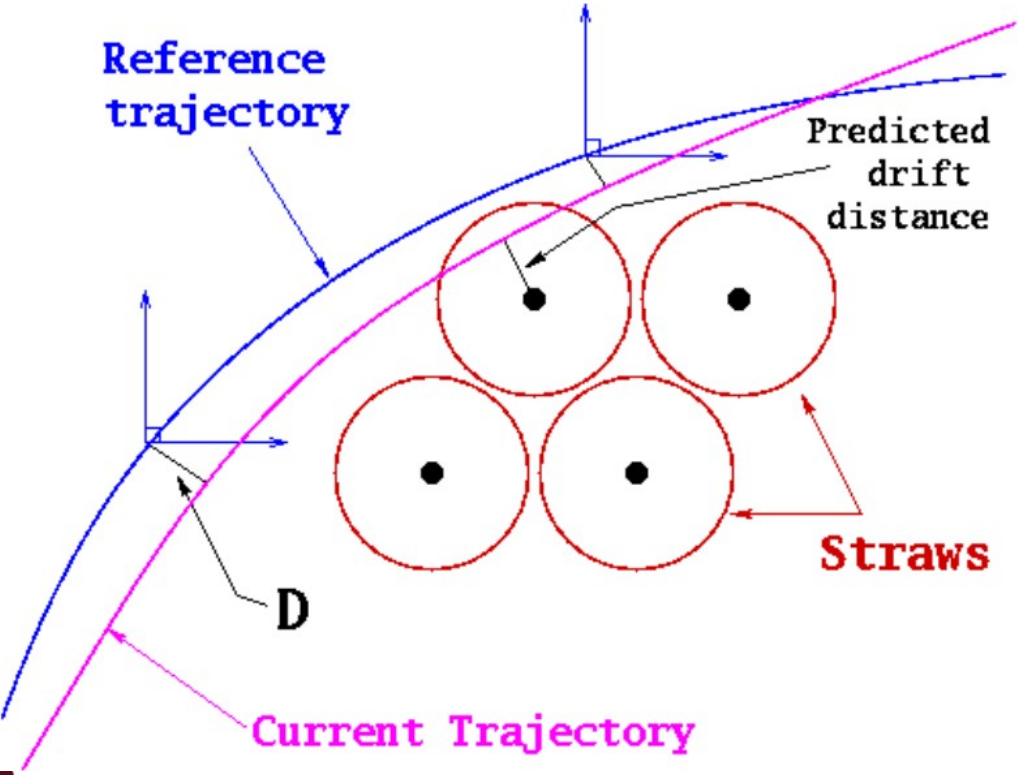
- State vector $\{x, y, t_x = dx/dz, t_y = dy/dz, q/p\}$
  - "Fitted" state vector considered as small perturbation relative to a seed
  - Seed determined from list of track candidates using helical model
- First step: create reference trajectory from seed, swimming from vertex to most downstream FDC hit
  - Take into account multiple scattering and energy loss when stepping through the field – only do this for the reference trajectory...
- Measurements added one by one, starting with most downstream hit
- Iterate up to 20 times
  - Generally only 2 or 3 iterations needed for convergence
  - CDC hits included without changing state vector

- Initial guess for covariance matrix (off-diagonal elements=0)

$\sigma(\delta p/p) = 10\%$, $\sigma_y = 3.3$ mm, $\sigma_x = 3.3$ mm, $\sigma_{tx} = 100$ mrad, $\sigma_{ty} = 100$ mrad

Central Tracking

- Start out by swimming a reference trajectory as for forward tracking



- Use central parameters: $\{q/p_T, \phi, \tan\lambda, D, z\}$
  - D=distance of closest approach to reference trajectory at a given time along the particle's path
  - Origin of coordinate system for D moves from point to point along reference trajectory
  - $\lambda = \pi/2 - \theta$ ("Dip angle")

- Initial guess for covariance matrix (off-diagonal elements=0)

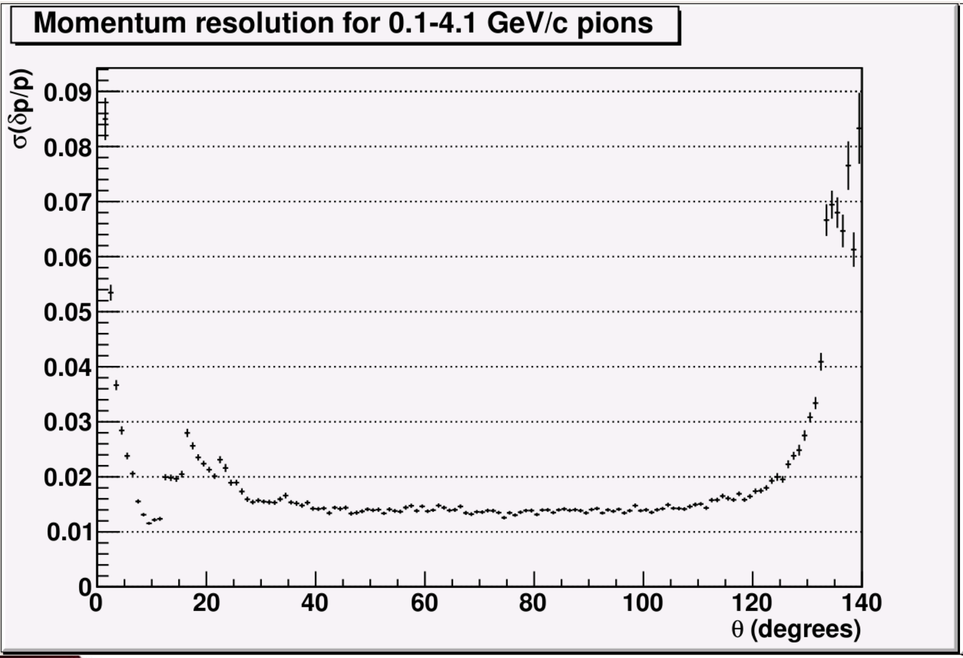
$\sigma(\delta p_T/p_T)$ = 10%, σ_D=3.3 mm, σ_z = 3.3 mm, σ_λ=100 mrad, σ_ϕ=100 mrad

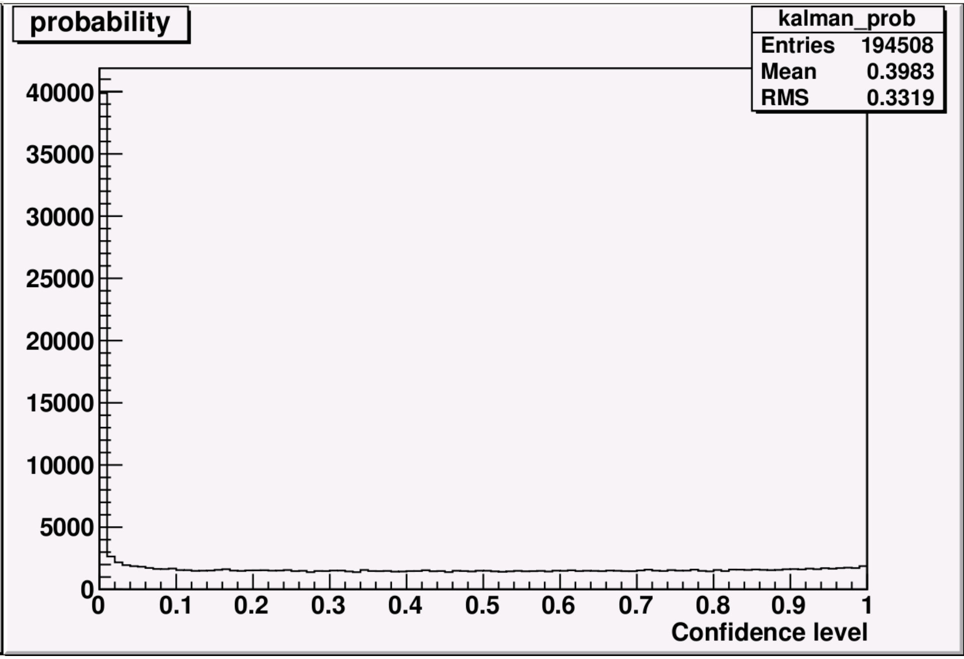
Step 3: Time-based fitting

- Match wire-based track to Start Counter (or TOF or BCAL if no hit in SC) and propagate corrected SC (TOF, BCAL) time to "vertex" $\rightarrow T_0$
    - Code for matching in *DParticleID* class
- Swim out from "vertex" and reselect hits belonging to track (using *DTrackHitSelector*)
- Refit using drift time information → *DTrackTimeBased*
    - Initial seed for parameters = result of wire-based fit

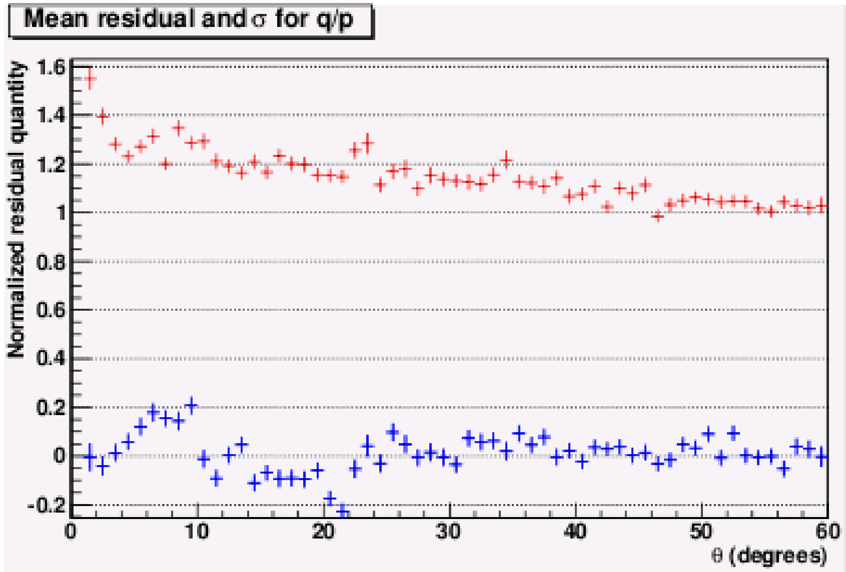
Fit quality

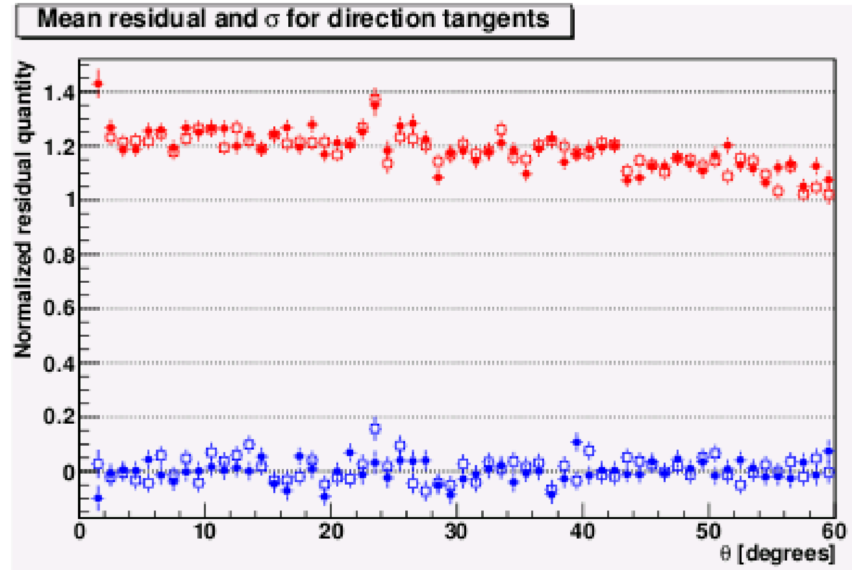
- Generated 0.1-4.1 GeV/c pions from center of target
  - Multiple-scattering and energy loss ON
  - Decays, secondary particles, hadronic interactions OFF

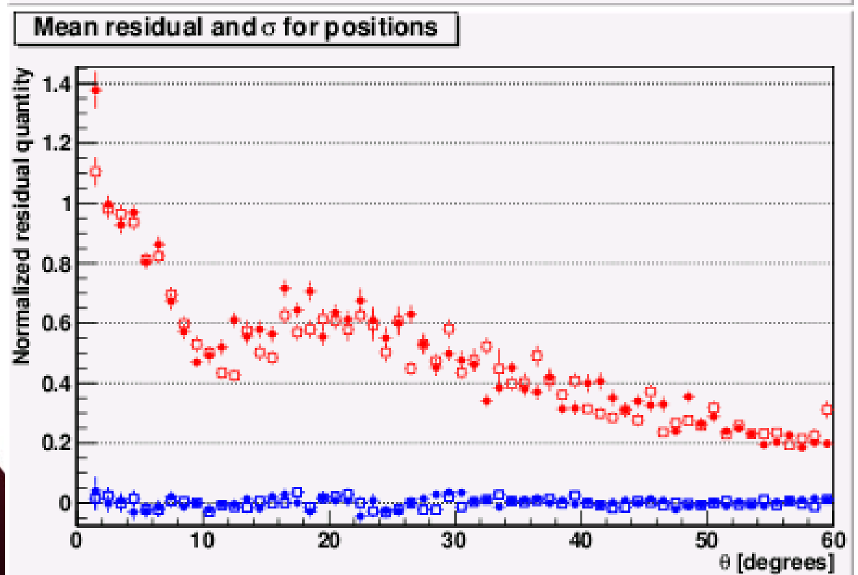




Normalized Residuals





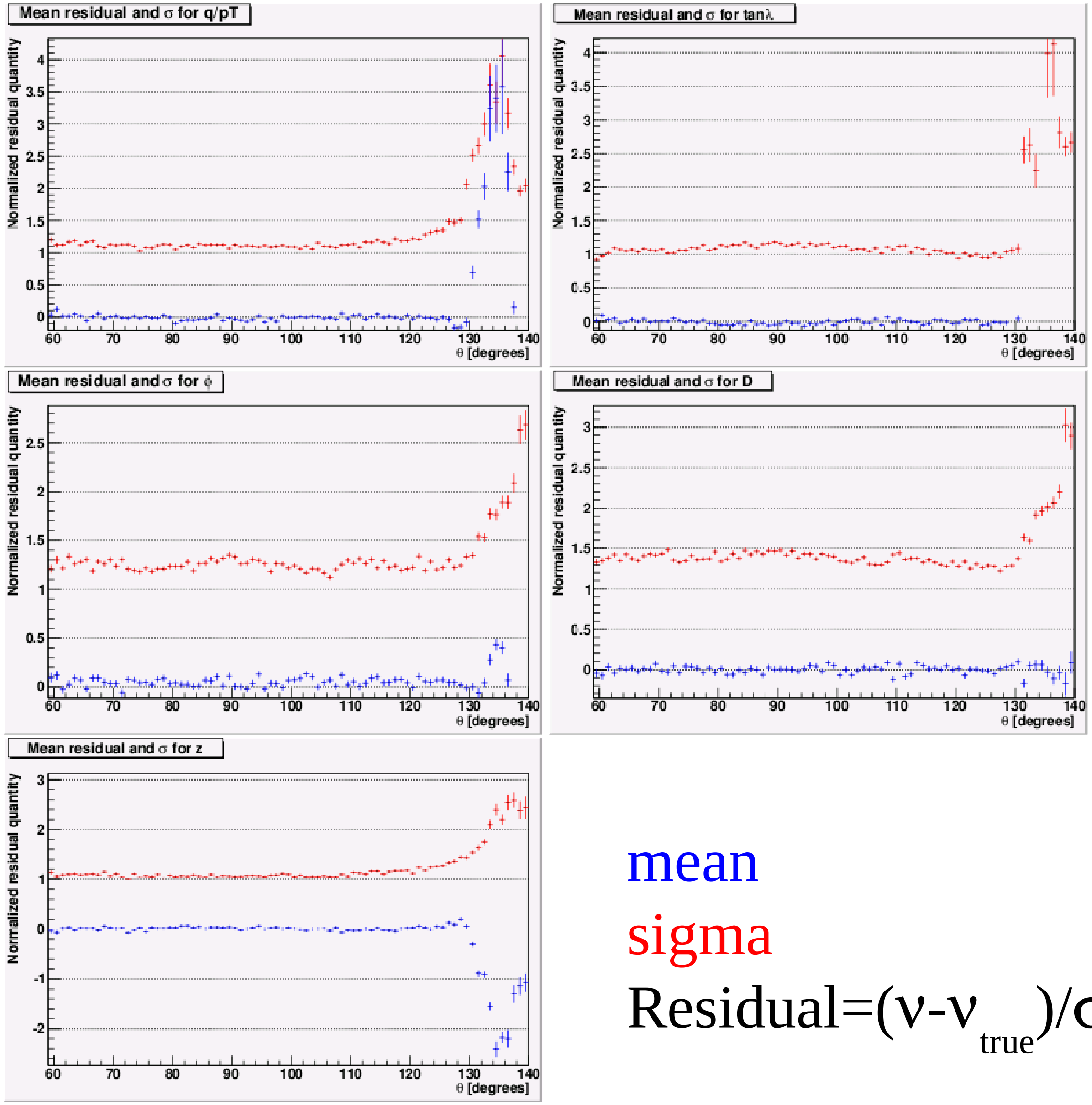


mean

sigma

Residual=$(\nu-\nu_{true})/\sigma_{\nu}$

Normalized Residuals

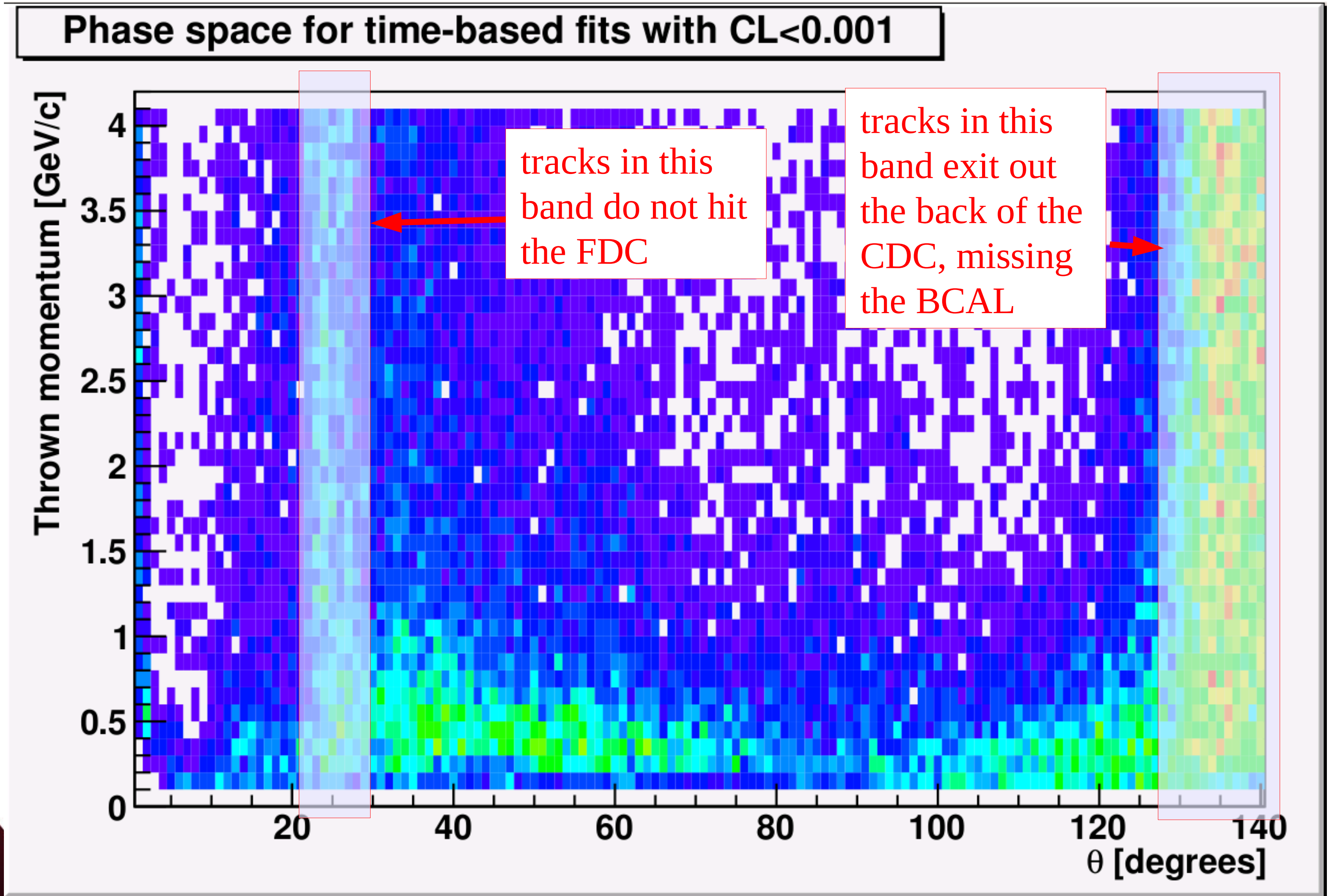


mean

sigma

Residual=$(\nu-\nu_{true})/\sigma_{\nu}$

Phase space for bad-quality fits



Issues

- Efficiency for single tracks is high but there are problem regions
- Poor quality fits leading to bands of low CL at certain angles and certain momenta need to be addressed
- Errors due to multiple scattering could be improved