Track Reconstruction: the trf toolkit

Norman Graf (SLAC)

ILC-ACFA Meeting, Beijing
February 6, 2007

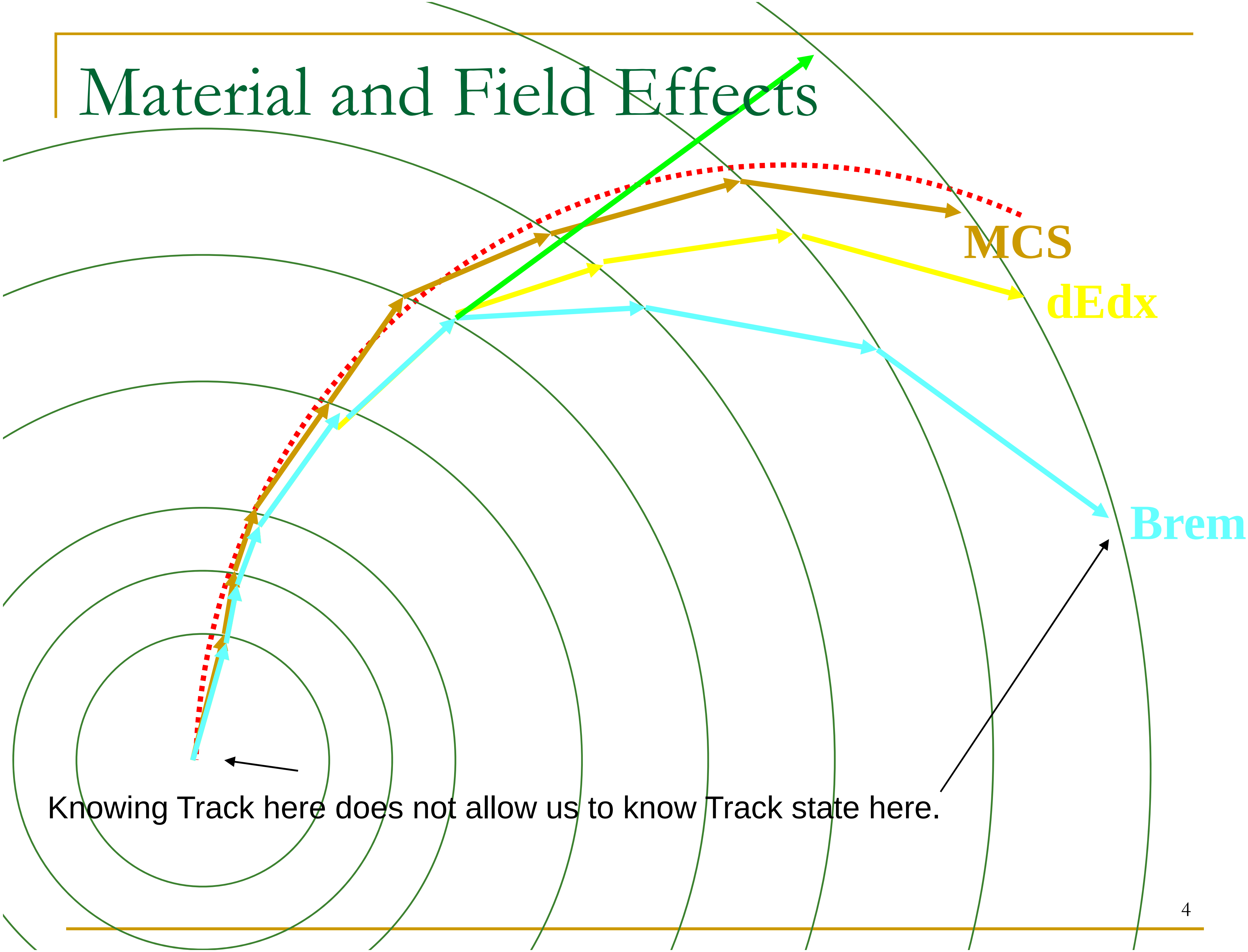
What is a track?

- Ordered association of digits, clusters or hits (finder)
  - Digit = data read from a detector channel
  - Cluster = collection of digits
  - Hit = Cluster (or digit) + calibration + geometry
    - Provides a measurement suitable to fit a track
    - E.g. a 1D or 2D spatial measurement on a plane
- Trajectory through space (fitter)
  - Space = 6D track parameter space
    - 3 position + 2 direction + 1 curvature
  - 5 parameters and error matrix at any surface
- Track is therefore only piecewise helical.
  - default is to break track down by measurement layers.
  - could increase granularity for inhomogeneous fields

Track Definition

- Six parameters are required to determine a charged particle’s ideal path in a magnetic field.
- However, knowing these parameters at a single point (*e.g.* the distance of closest approach to the beam, **dca**) is insufficient for precision fits due to material effects (dE/dx, MCS, bremsstrahlung) and field inhomogeneities.
  - No global functional form for the fit.
- Current LCIO Track interface definition is too simplistic by not allowing for these effects.

Material and Field Effects



Knowing Track here does not allow us to know Track state here.

Infrastructure components

- Hit
  - Defined at a surface.
  - Provides a measurement and associated error
  - Provides a mechanism to predict the measurement from a track fit
  - Provides access to underlying cluster and/or digits

TrackerHit

- Current TrackerHit interface only accommodates three dimensional hits.
- Many tracking subdetectors only provide one dimensional measurements (silicon microstrips) or two dimensional hits (such as silicon pixels).
- Furthermore, using Cartesian coordinates is not always the most natural for individual subdetectors.
- Cylinder:
  - 1D Axial: ϕ
  - 1D Stereo: $\phi+\kappa z$
  - 2D Combined: (ϕ, z)
- XYPlane:
  - 1D Stereo: $w_v{*}v + w_z{*}z$
  - 2D Combined: (v, z)
- ZPlane:
  - 1D Stereo: $w_x{*}x + w_y{*}y$
  - 2D Combined: (x,y)

trf Hits

- trfcyl:
  - HitCylPhi : a phi measurement on a cylinder.
  - HitCylPhiZ : stereo measurement on a cylinder.
    - phiz = phi + stereo*z.
  - HitCylPhiZ2D : measurement of both phi and z on a cylinder.
- trfxyp:
  - HitXYPlane1 : one dimensional v-z measurement on a XYPlane.
    - avz = wv*v + wz*z
  - HitXYPlane2 : two dimensional (v,z) measurement on an XYPlane
- trfzp:
  - HitZPlane1 : one dimensional xy measurement on a ZPlane.
    - axy = wx*x + wy*y
  - HitZPlane2 : two dimensional (x,y) measurement on a ZPlane

Surfaces

- Surfaces generally correspond to geometric shapes representing detector devices.
- They provide a basis for tracks, and constrain one of the track parameters.
- The track vector at a surface is expressed in parameters which are “natural” for that surface.

1.) Cylinder

- Surface defined coaxial with z, therefore specified by a single parameter r.
- Track Parameters: $(\phi, z, \alpha, \tan\lambda, q/p_T)$
- Bounded surface adds z_{min} and z_{max}.
- Supports 1D and 2D hits:
  - 1D Axial: ϕ
  - 1D Stereo: $\phi+\kappa z$
  - 2D Combined: (ϕ, z)

2.) XY Plane

- Surface defined parallel with z, therefore specified by distance u from the z axis and an angle ϕ of the normal with respect to x axis.
- Track Parameters: (v, z, dv/du, dz/du, q/p)
- Bounded surface adds polygonal boundaries.
- Supports 1D and 2D hits:
  - 1D Stereo: $w_v * v + w_z * z$
  - 2D Combined: (v, z)

3.) Z Plane

- Surface defined perpendicular to z, therefore specified by single parameter z.
- Track Parameters: (x, y, dx/dz, dy/dz, q/p)
- Bounded surface adds polygonal boundaries.
- Supports 1D and 2D hits:
  - 1D Stereo: $w_x*x + w_y*y$
  - 2D Combined: (x,y)

4.) Distance of Closest Approach

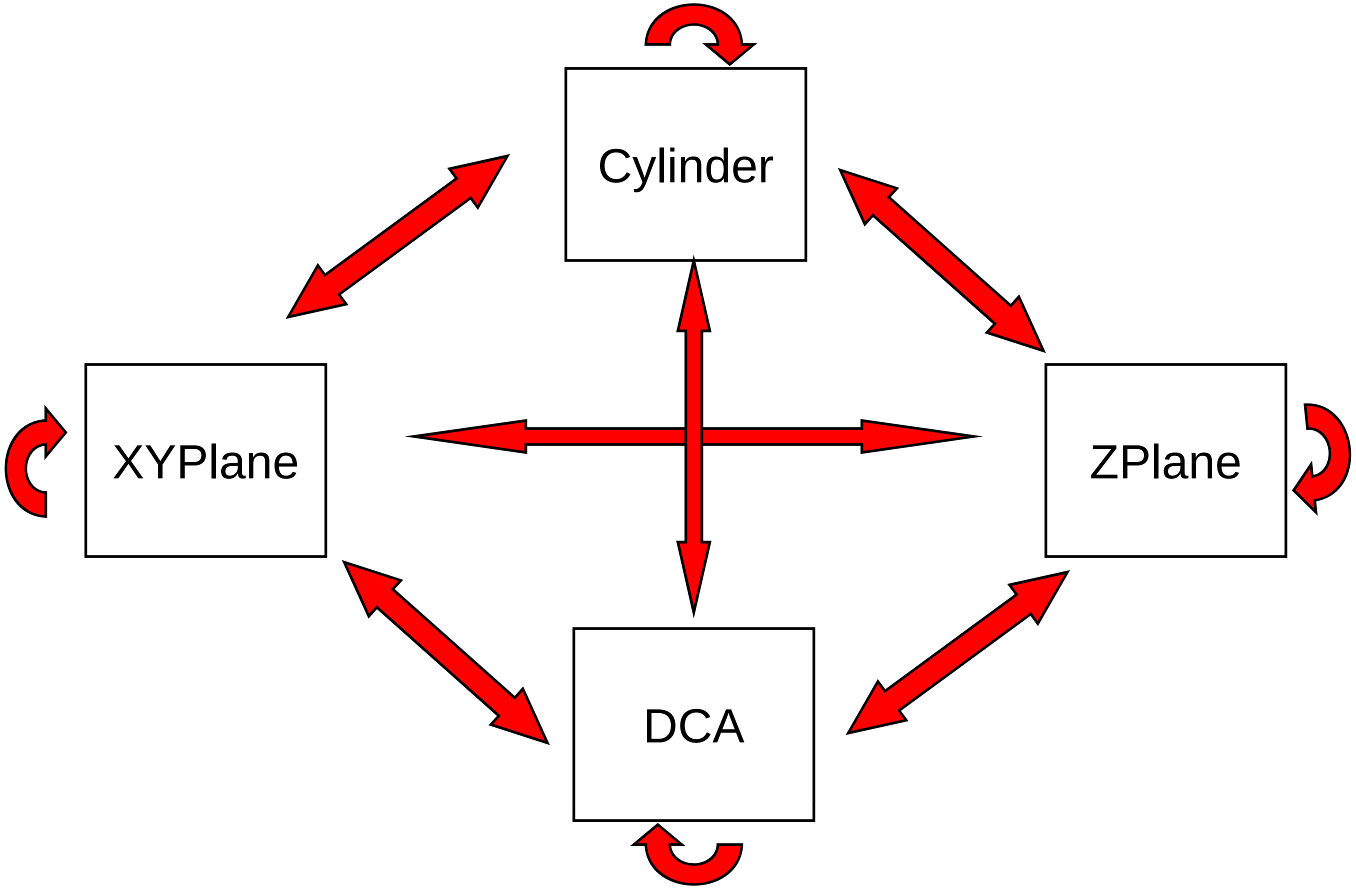
- DCA is also a 5D **Surface** in the 6 parameter space of points along a track.
- It is **not** a 2D surface in 3D space.
- Characterized by the track direction and position in the (x,y) plane being normal; $\alpha=\pi/2$.
- Track Parameters: $(r, z, \phi_{dir}, \tan\lambda, q/p_T)$

Propagator

- Propagators propagate a track (and optionally its covariance matrix) to a new surface.
- A propagator returns an object of type PropStat which describes the status of the attempted propagation:
  - *i.e.* whether it was successful and, if so, in which direction the track was propagated (forward or backward).
- Interacting Propagators modify the track and its covariance matrix (in case of energy loss), or just the covariance matrix (thin multiple scattering.)

Propagators

- Propagators are defined for all combinations of surfaces:



Interactors

- Describes the interface for a class which modifies a track. Examples are:
- Multiple Scattering
  - ThickCylMS
  - ThinXYPlaneMS
  - ThinZPlaneMS
- Energy Loss
  - CylELoss

Detector

- Use compact.xml to create a tracking Detector composed of surfaces, along with interacting propagators to handle track vector and covariance matrix propagation, as well as energy loss and multiple scattering.
  - Silicon pixel and microstrip wafers modeled as either xyplane or zplane.
  - TPC modeled as cylindrical layers (corresponding to pad rows).
  - Currently using thin multiple scattering approximations.
  - Using pure solenoidal field propagators
    - Runge-Kutta propagators available when needed.

Track Finding

- Using a conformal mapping technique
  - Maps curved trajectories onto straight lines
  - Simple link-and-tree type of following approach associates hits.
  - Once enough hits are linked, do a simple helix fit
    - circle in r-phi
    - straight line in s-z
    - simple iteration to make commensurate
  - Use these track parameters to predict track into regions with only 1-D measurements & pick up hits.
  - Outside-in, inside-out, cross-detector: completely flexible as long as concept of *layer* exists.
    - Runtime control of finding details.
  - Simple fit serves as input to final Kalman fitter.

Application to $t\bar{t} \rightarrow$ six jets events

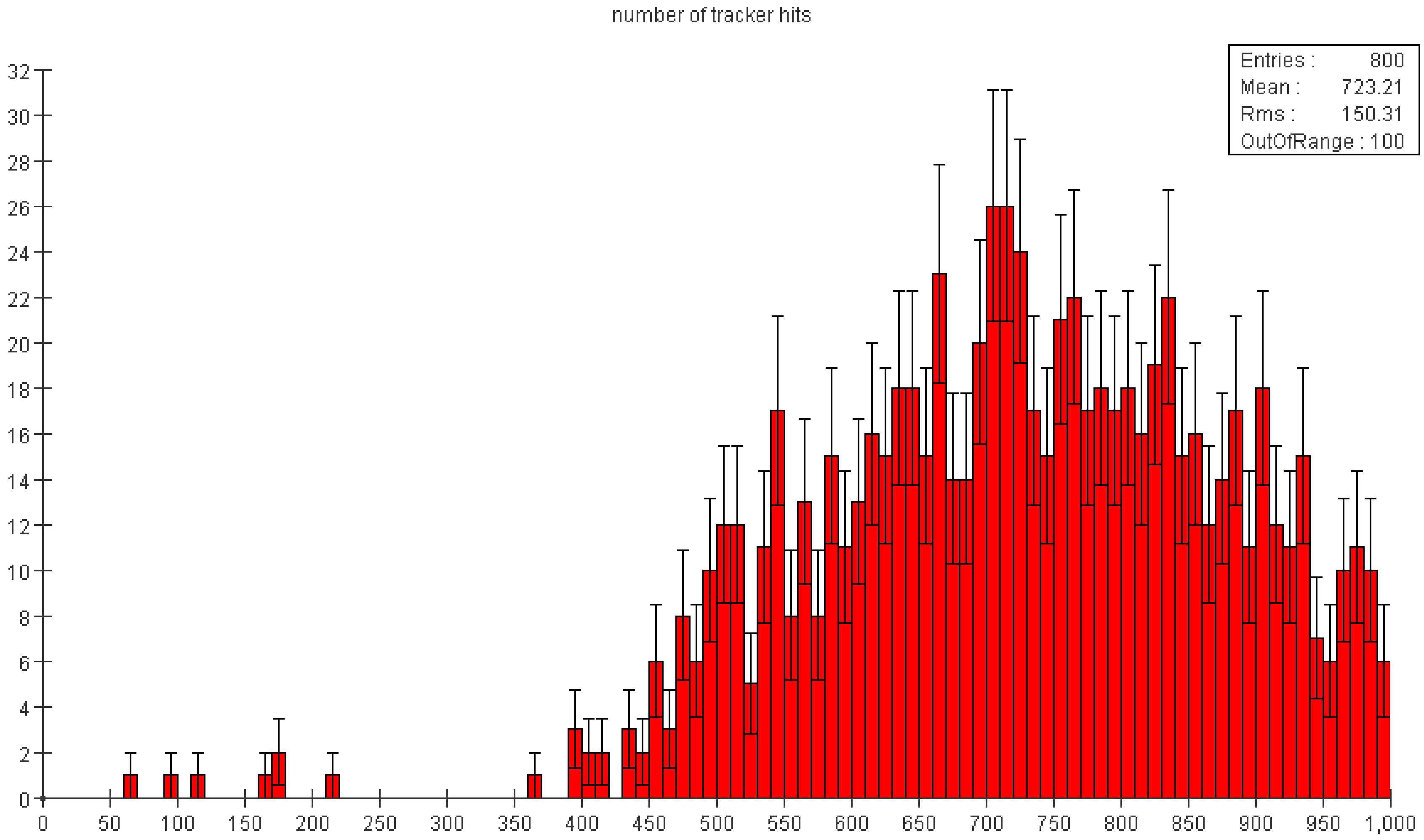
- Generate $e^+ e^- \rightarrow t\bar{t}$, $t\bar{t} \rightarrow$ six jets.
- Simulate response of silicon detector using full GEANT simulation (slic).
- Convert SimTrackerHits in event into:
  - 1-D phi measurements in Central Tracker Barrel
  - 2-D phi-z measurements in Vertex Barrel (pixel)
  - 2-D x-y measurements in forward disks (assume stereo strips)
  - 2-D phi-z measurements in TPC (place hits on cylinders in middle of readout pads)
  - Simple smearing being used
    - NO digitization ∴ NO ghosts, NO merging, NO fakes … yet.

Application to $t\bar{t} \rightarrow$ six jets events

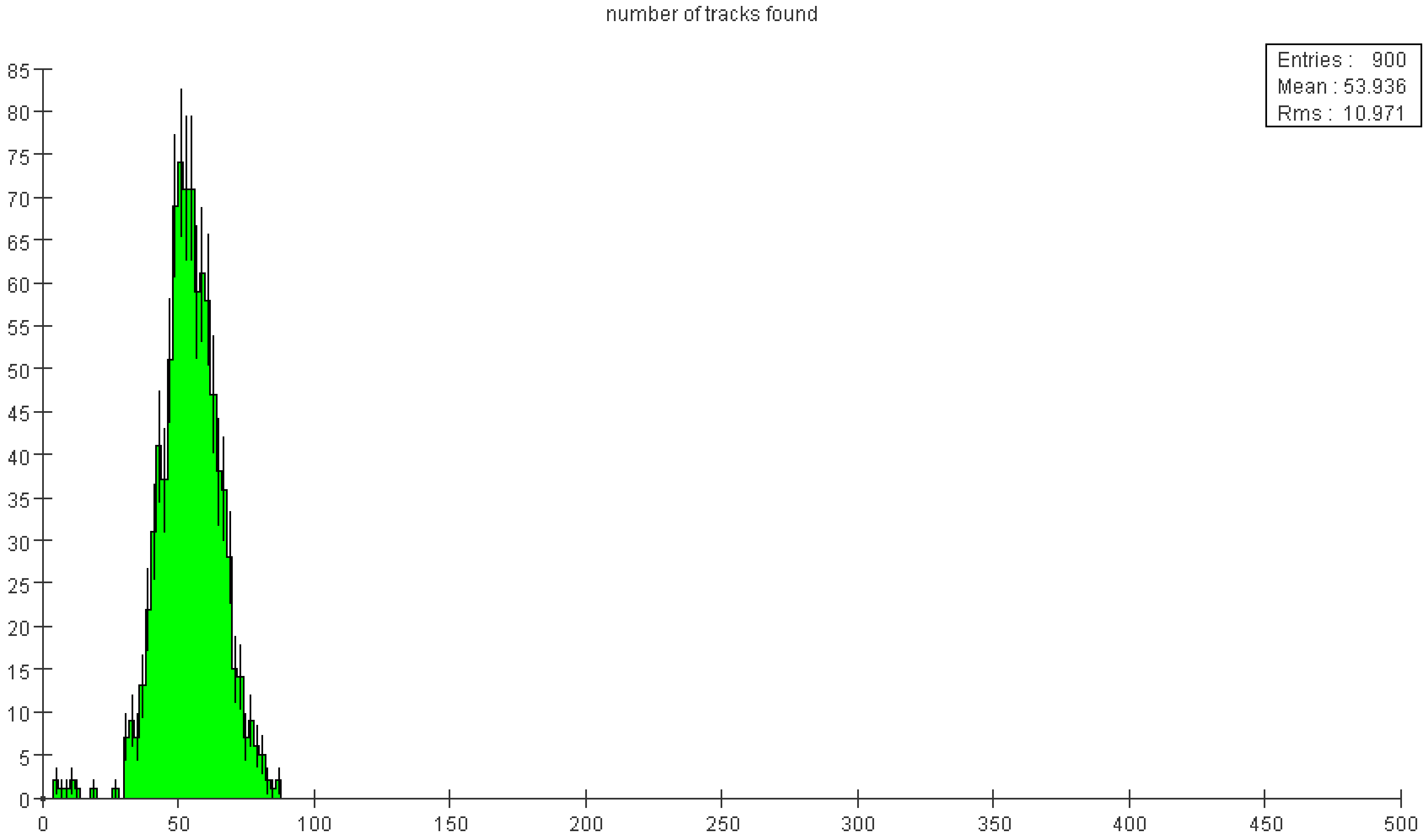
- Open event, read in data.
- Create tracker hits.
- Find tracks & fit with simple helix.
- Fit tracks with Kalman filter, MCS, dEdx.
- Analyze tracks.
- Write out histograms.

- Takes 3min to fully analyze 900 events on 1.7GHz laptop.

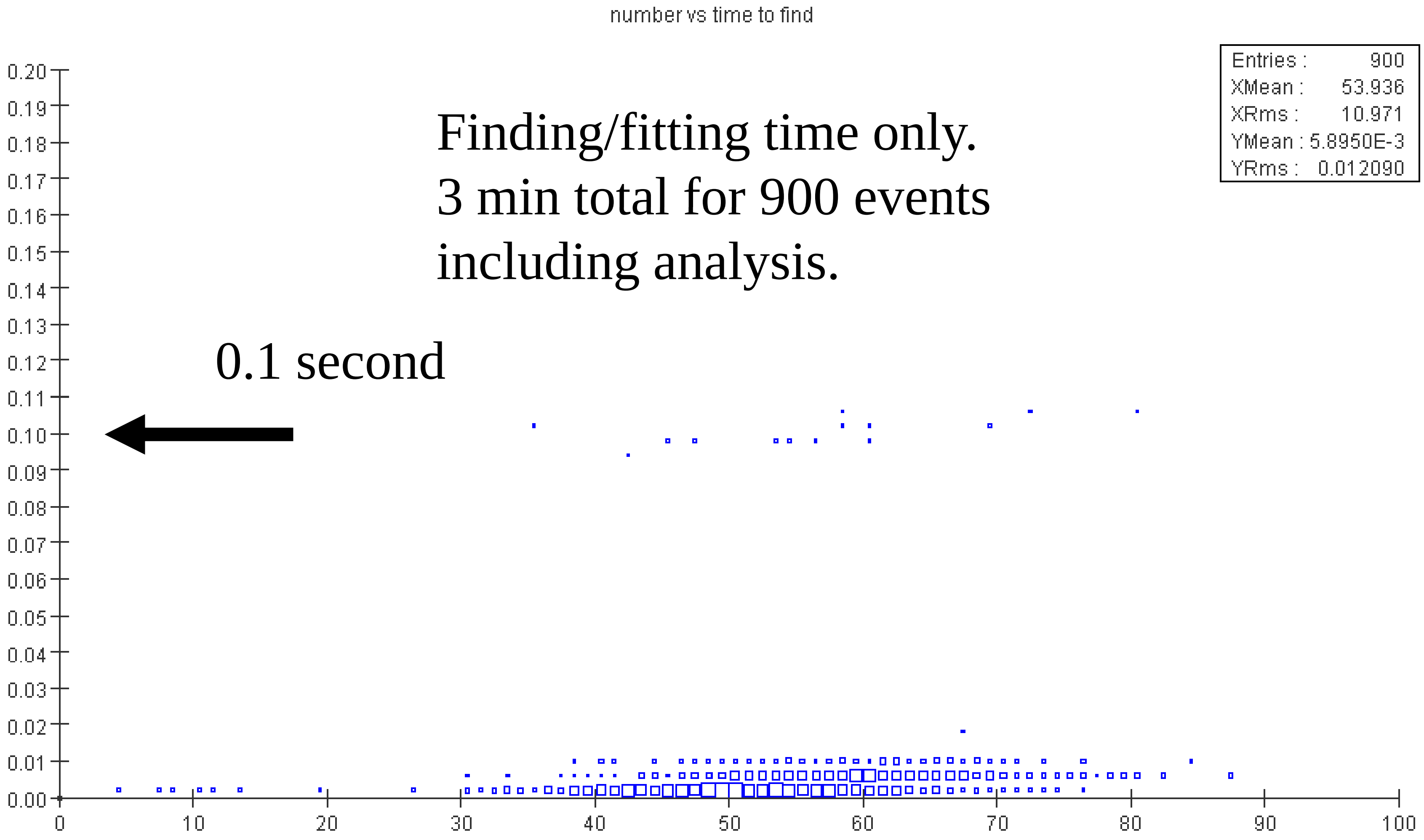
$t\bar{t} \rightarrow$six jets # of Hits



of tracks found



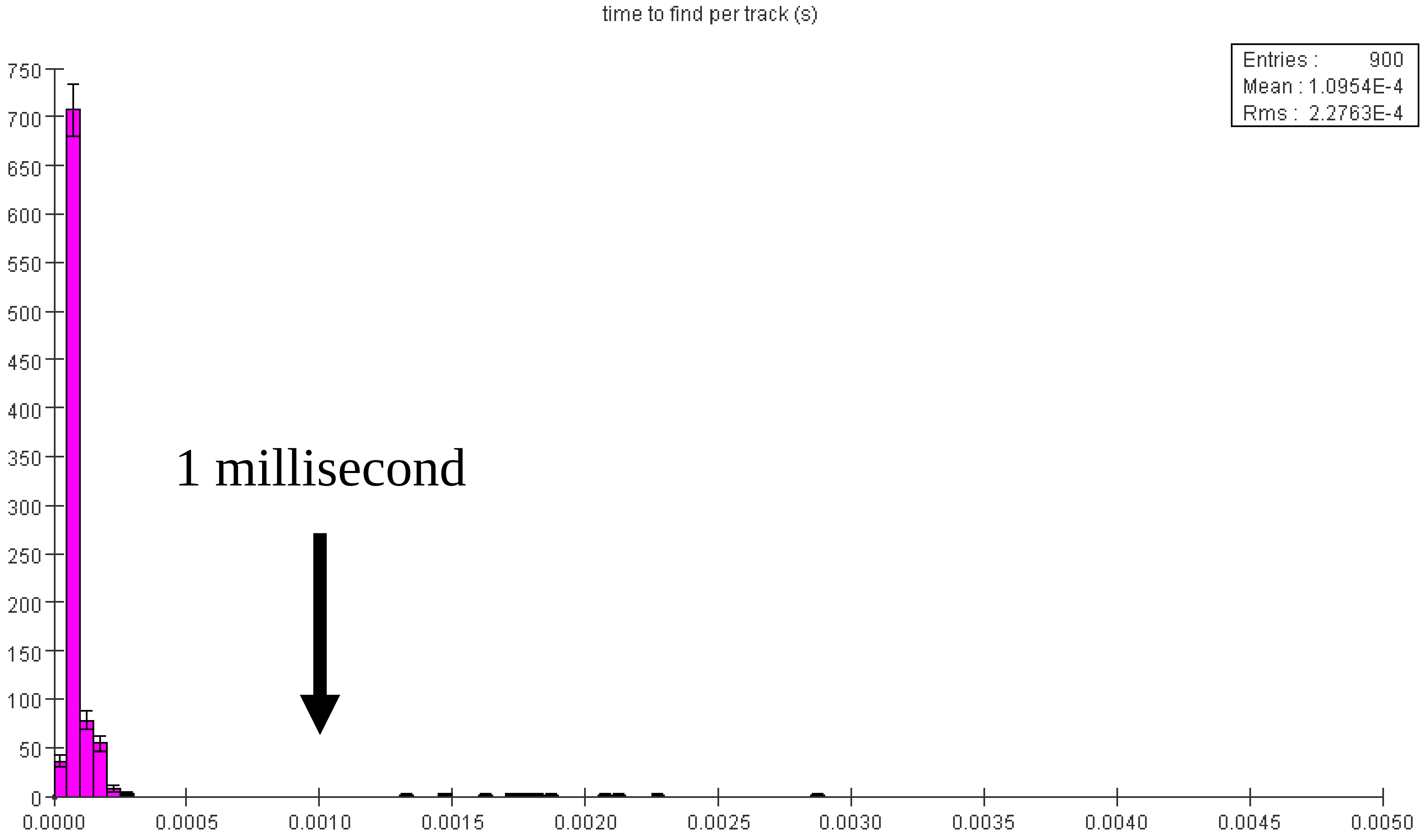
time (s) vs # tracks (1.7GHz)



Finding/fitting time only.
3 min total for 900 events
including analysis.

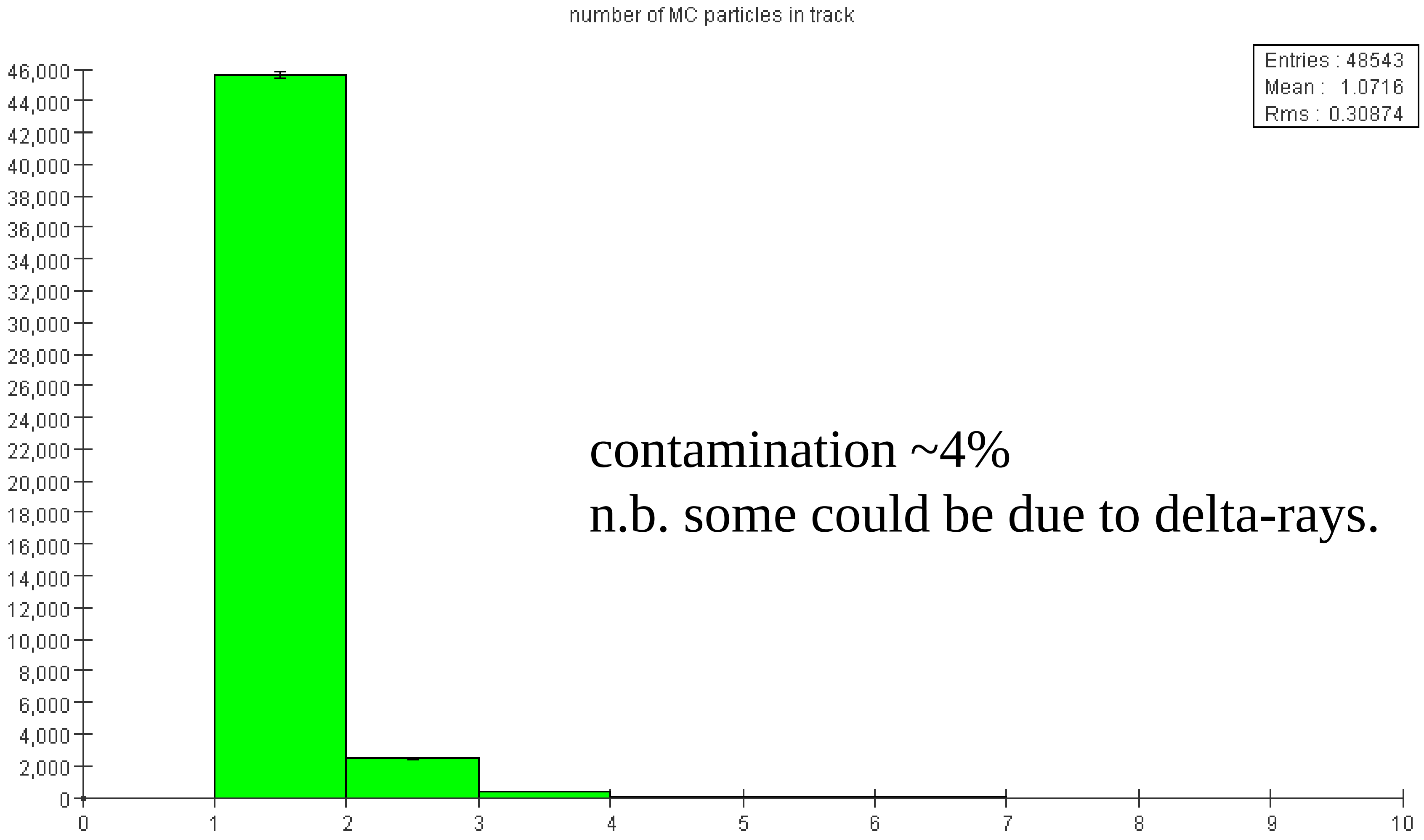
0.1 second

time(s) per track (1.7GHz)



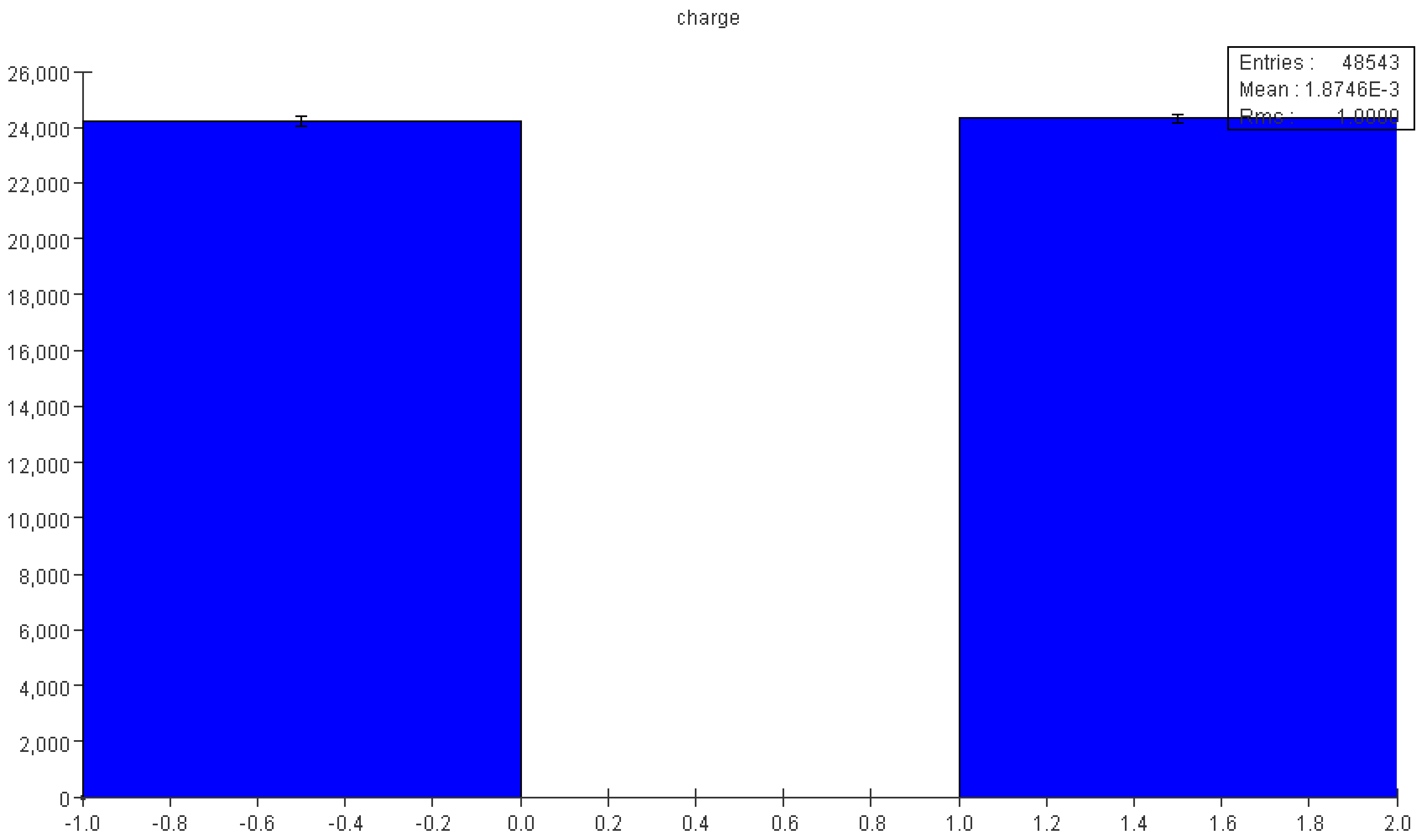
1 millisecond

of MCParticles/track

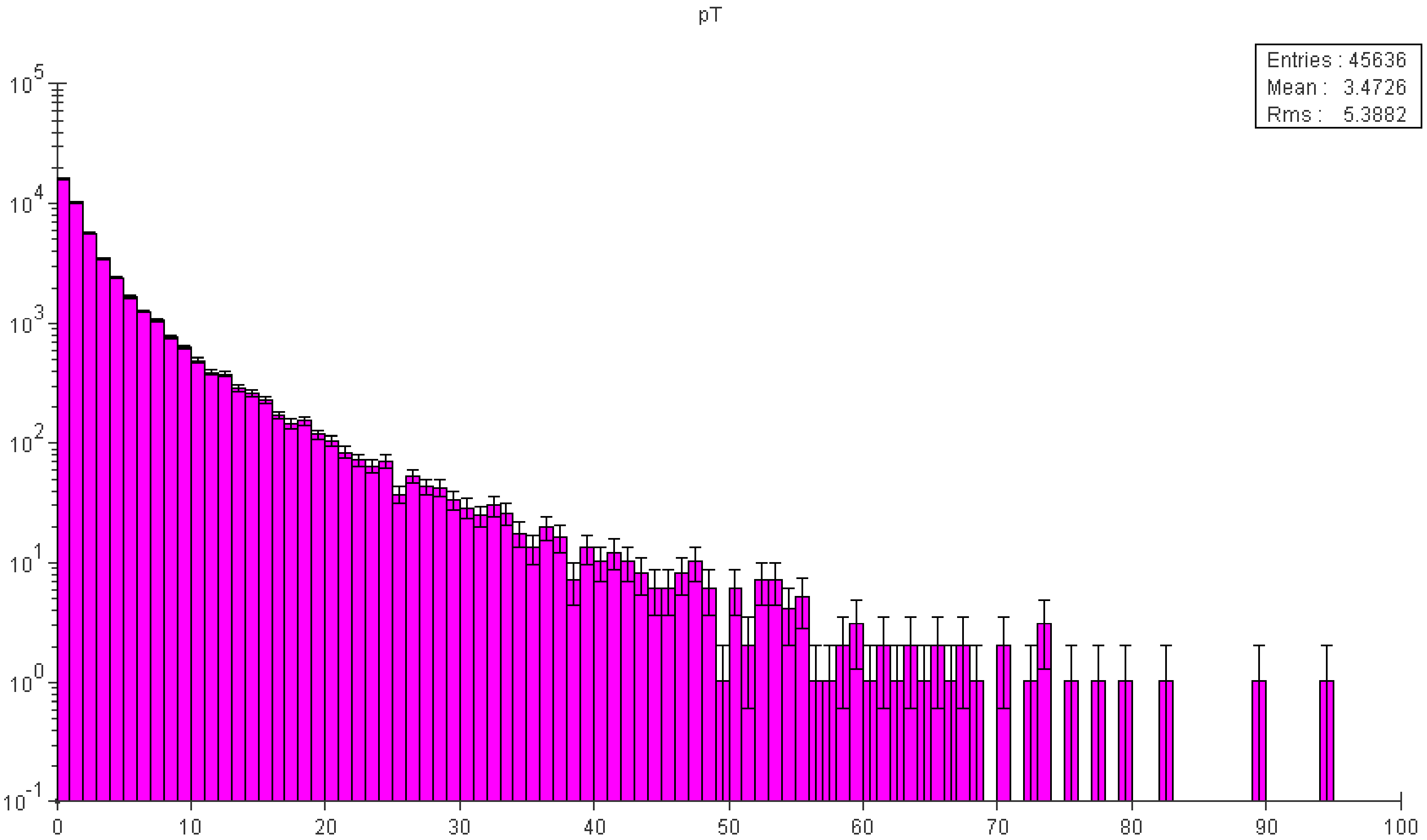


contamination ~4%
n.b. some could be due to delta-rays.

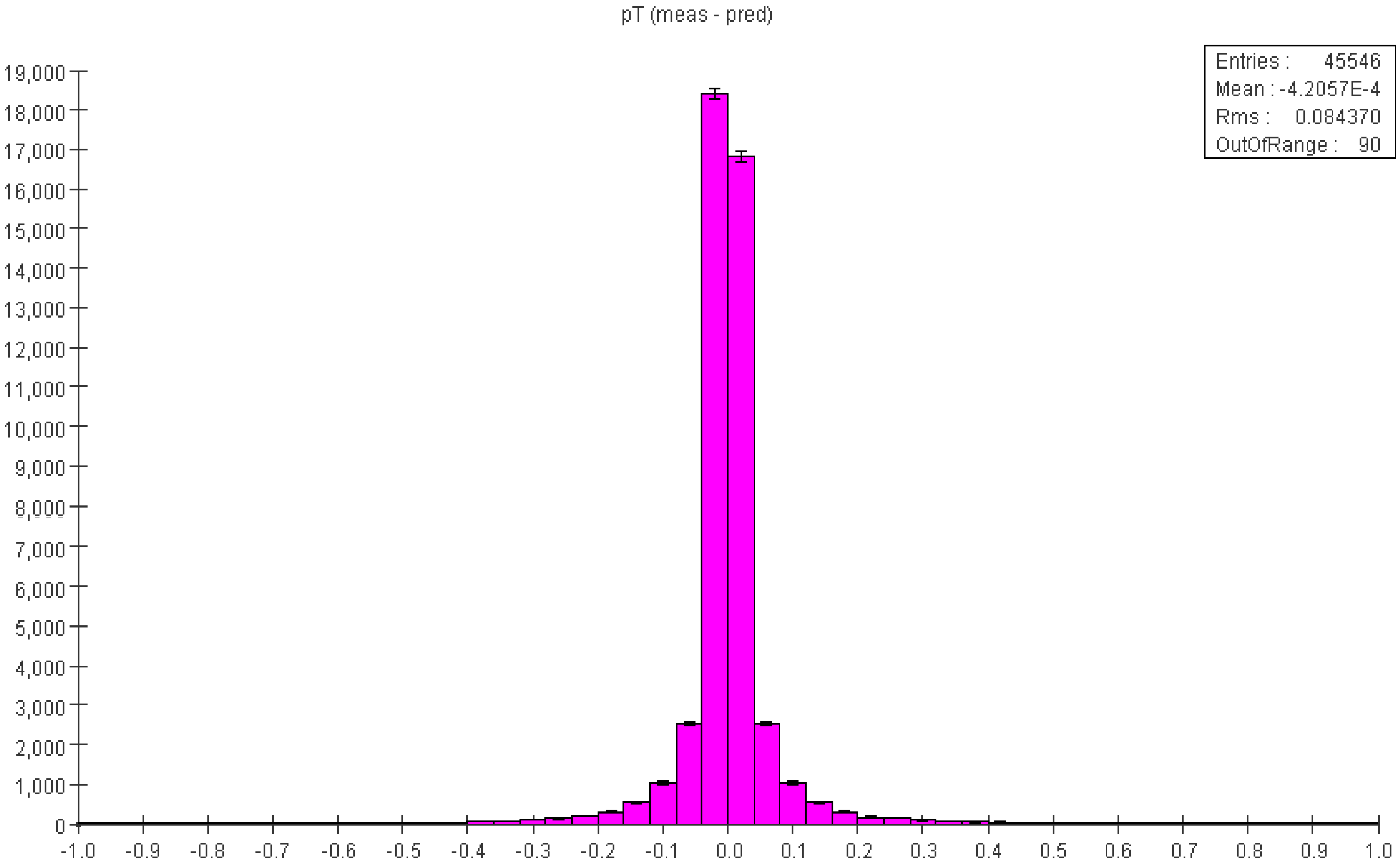
charge


charge
Entries : 48543
Mean : 1.8746E-3
Rms : 1.0000
26,000
24,000
22,000
20,000
18,000
16,000
14,000
12,000
10,000
8,000
6,000
4,000
2,000
0
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0

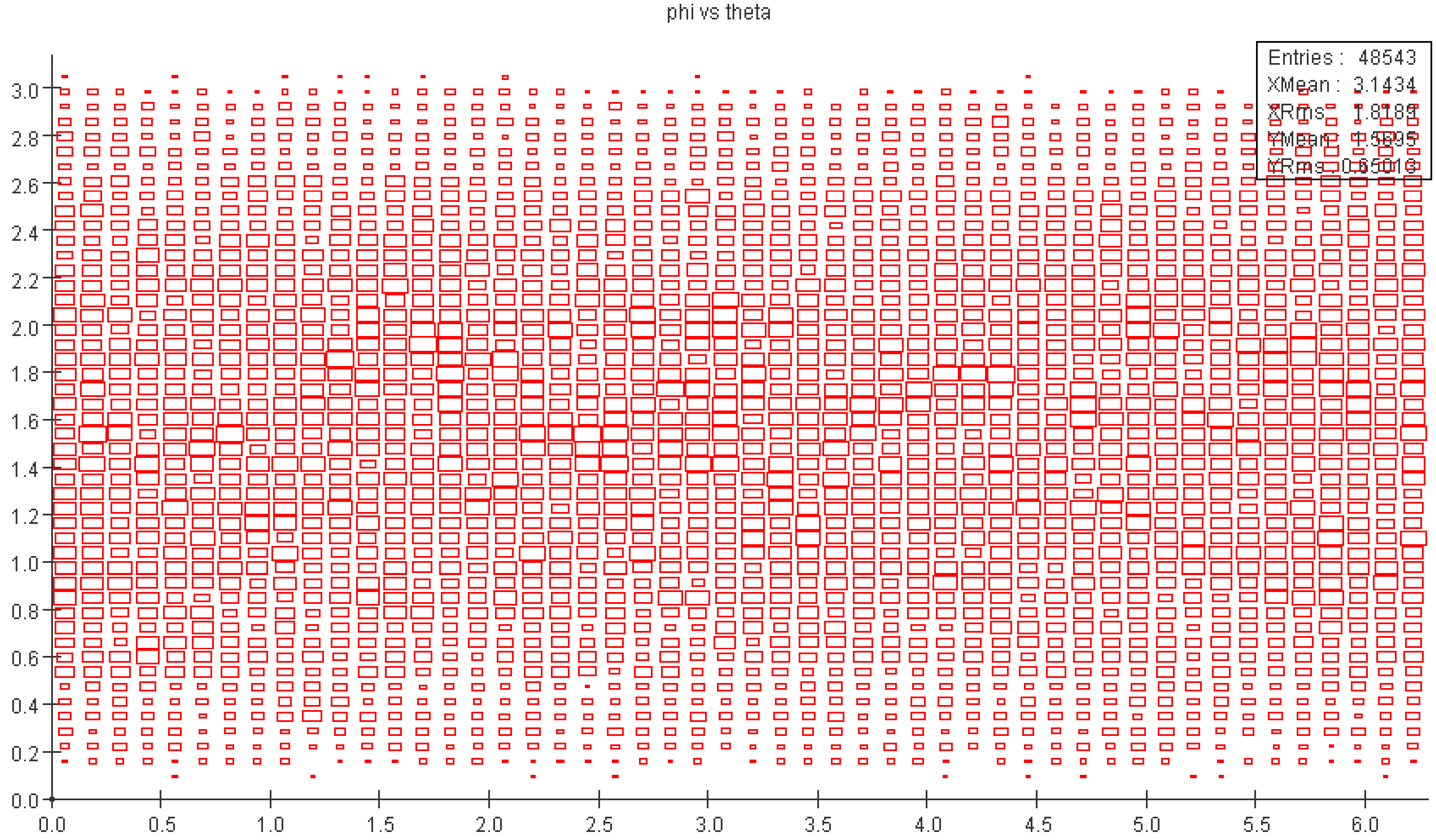
$t\bar{t}$ →six jets pT



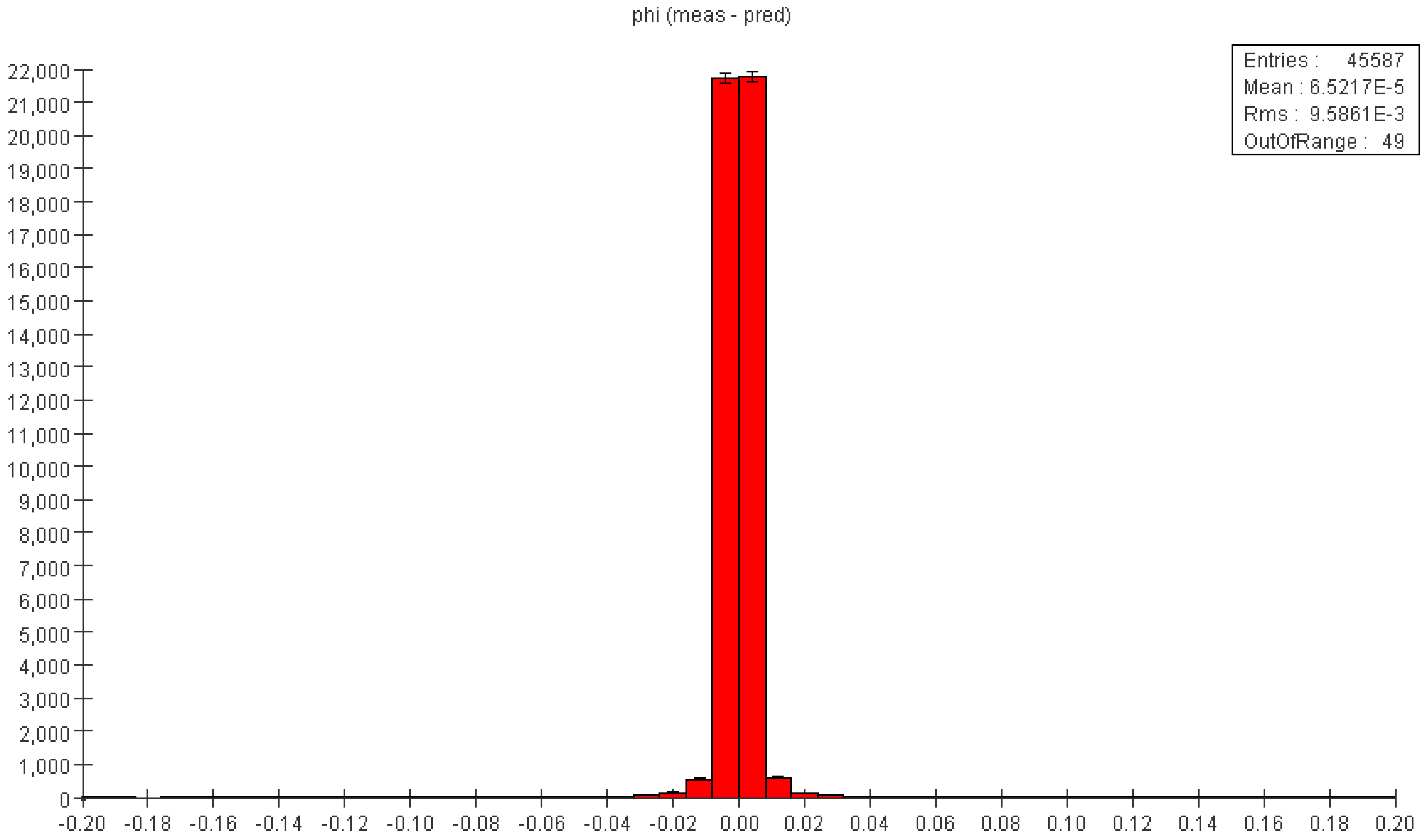
pT (meas-pred)


pT (meas - pred)
Entries : 45546
Mean : -4.2057E-4
Rms : 0.084370
OutOfRange : 90
19,000
18,000
17,000
16,000
15,000
14,000
13,000
12,000
11,000
10,000
9,000
8,000
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
-1.0
-0.9
-0.8
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

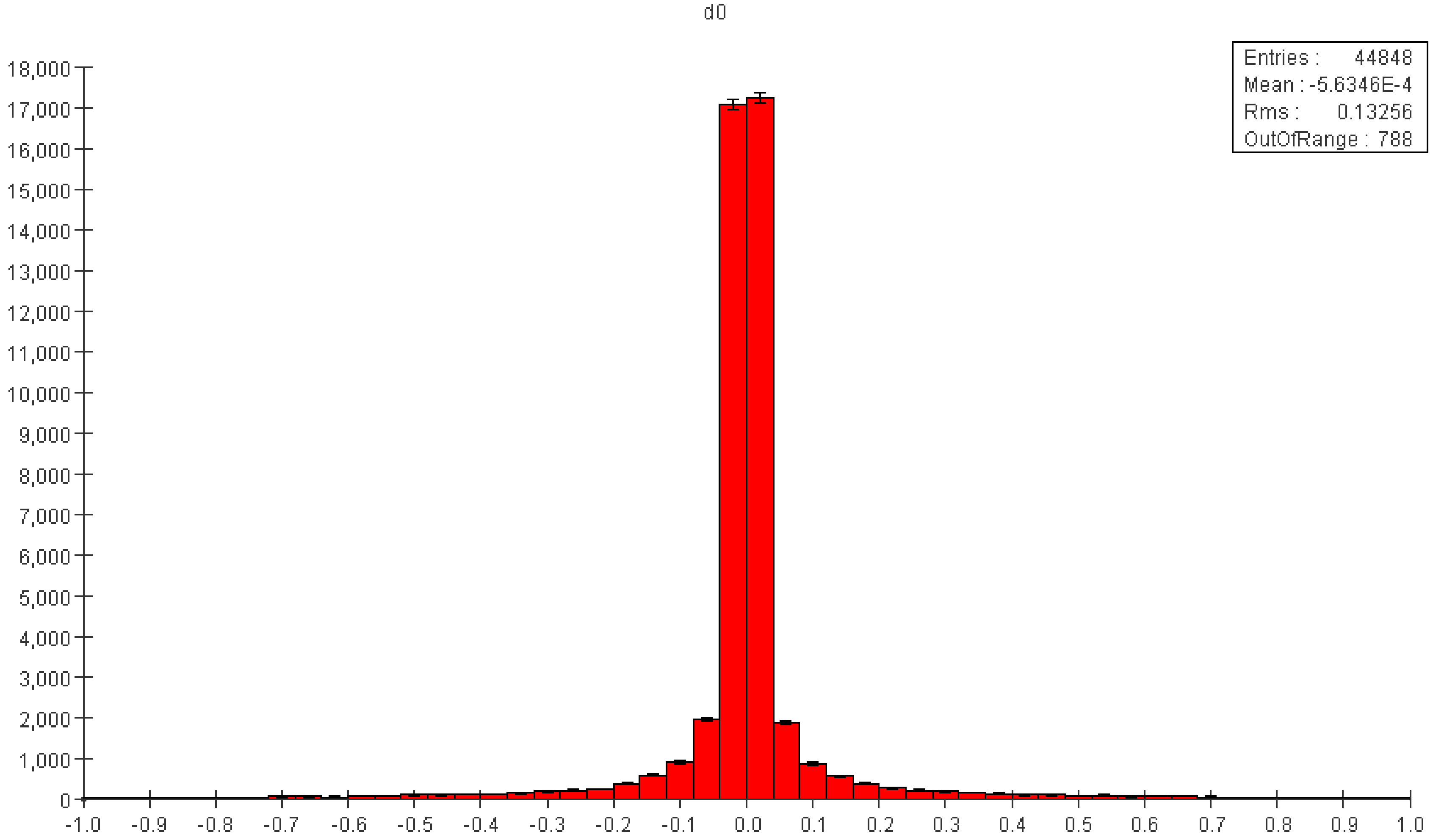
φ vs θ


phi vs theta
Entries : 48543
XMean : 3.1434
XRms : 1.8189
YMean : 1.5695
YRms : 0.65013
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0

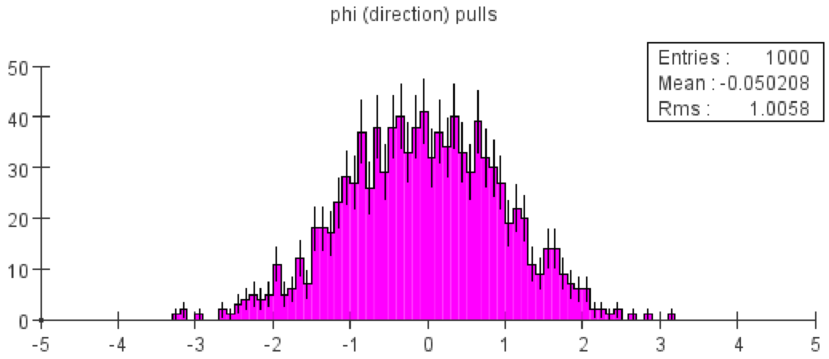
phi (meas – pred)

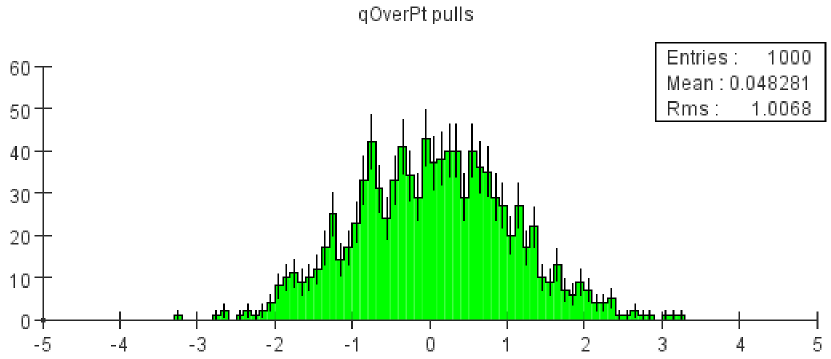


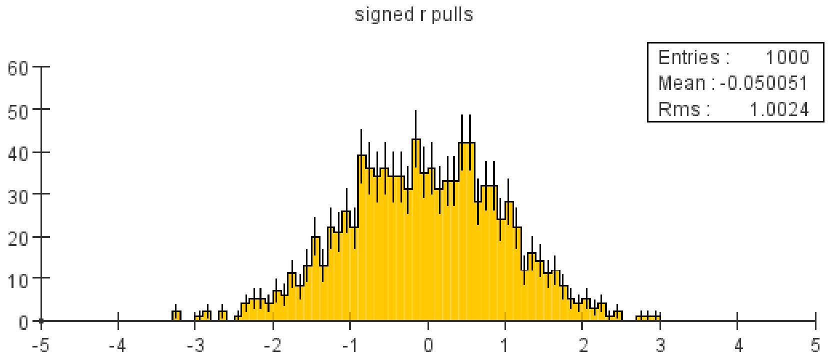
impact parameter

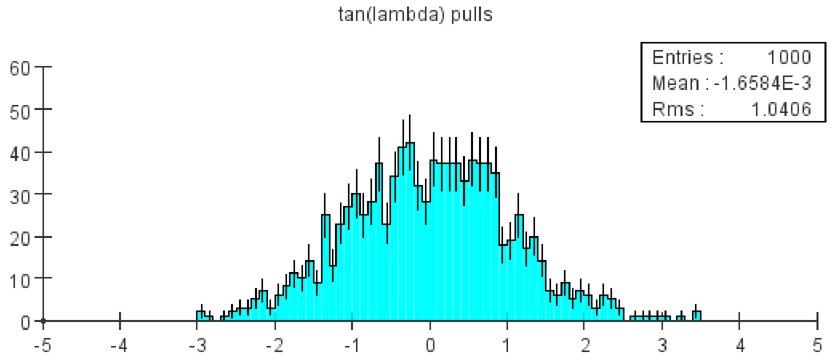


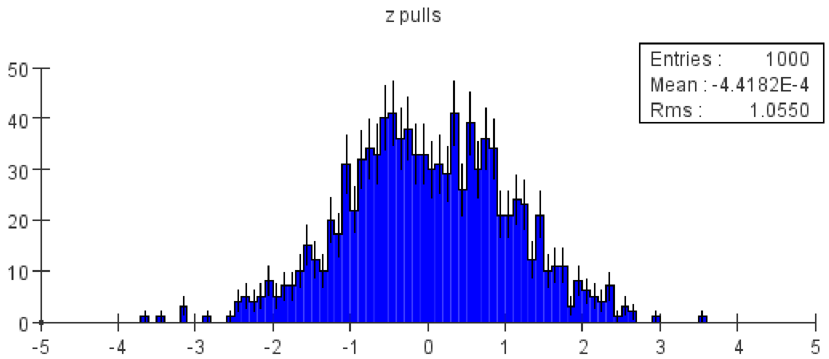
Full Kalman Fit pulls











Single 10GeV muons in central region (5 2D + 5 1D pts).

Test Detector w/ELoss and MCS

Summary

- Trf toolkit provides full infrastructure for defining detectors, hits, tracks as well as propagators, interactors and fitters.
  - Currently working on generic interface between compact detector description and tracking Detector.
  - Lot of effort being devoted to “smart” propagator.
- Available in Java (org.lcsim) as well as C++ (standalone).
- Pattern recognition based on 2-D measurements on surfaces is implemented for collider-type detectors.
- Fast, with high efficiency.
- Extrapolation into 1-D tracker and fitting with full Kalman filter for linear collider detectors.
- Lots of work ahead to characterize and improve.