Status of silicon tracking.

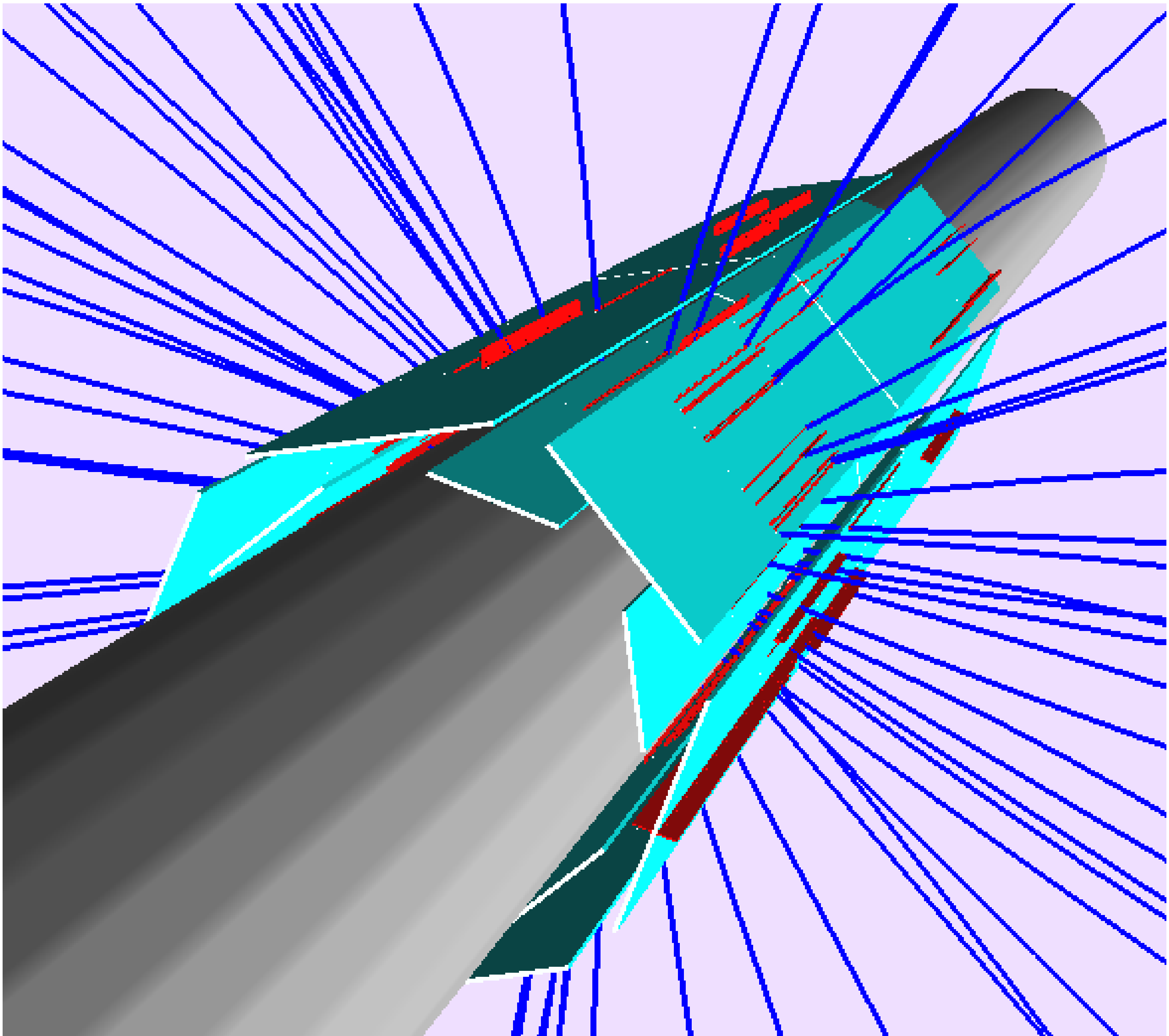
* Offline has recently added Layer 00 into the silicon tracking.

* New geometry was easy to incorporate, but...

* The change was challenging because the indexing of the entire silicon system was changed==> Numbering now is 0-7 in offline.

* Many people have been working to make this a reality: Rick, Pasha, Matt, Petar, Patrizia, Joe.

Here's what this looks like now:



PULLS FOR OUTSIDE IN TRACKING + Z HITS

PT = 0.5 GeV

Cot Theta	sigma:	0.90
Curvature	sigma:	1.30
Z0	sigma:	0.92
D0	sigma:	0.95
Phi0	sigma:	1.08

Pull is defined to be:

Deviation / Error

PT = 1.0 GeV

Cot Theta	sigma:	0.83
Curvature	sigma:	1.05
Z0	sigma:	0.84
D0	sigma:	0.87
Phi0	sigma:	0.94

PT = 2.0 GeV

Cot Theta	sigma:	0.90
Curvature	sigma:	1.09
Z0	sigma:	0.86
D0	sigma:	0.90
Phi0	sigma:	1.03

PT = 4.0 GeV

Cot Theta	sigma:	0.82
Curvature	sigma:	1.07
Z0	sigma:	0.81
D0	sigma:	0.83
Phi0	sigma:	0.93

PT = 8.0 GeV

Cot Theta	sigma:	0.81
Curvature	sigma:	1.02
Z0	sigma:	0.80
D0	sigma:	0.83
Phi0	sigma:	0.98

PT = 40.0 GeV

Cot Theta	sigma:	0.76
Curvature	sigma:	0.99
Z0	sigma:	0.75
D0	sigma:	0.81
Phi0	sigma:	0.91

PULLS FOR STANDALONE TRACKING:

PT = 0.5 GeV

Cot Theta	sigma:	0.97
Curvature	sigma:	1.39
Z0	sigma:	0.84
D0	sigma:	0.99
Phi0	sigma:	1.11

PT = 1.0 GeV

Cot Theta	sigma:	0.90
Curvature	sigma:	1.14
Z0	sigma:	0.88
D0	sigma:	0.87
Phi0	sigma:	0.98

PT = 2.0 GeV

Cot Theta	sigma:	0.94
Curvature	sigma:	1.15
Z0	sigma:	0.83
D0	sigma:	0.97
Phi0	sigma:	1.04

PT = 4.0 GeV

Cot Theta	sigma:	0.90
Curvature	sigma:	1.11
Z0	sigma:	0.90
D0	sigma:	0.85
Phi0	sigma:	0.97

PT = 8.0 GeV

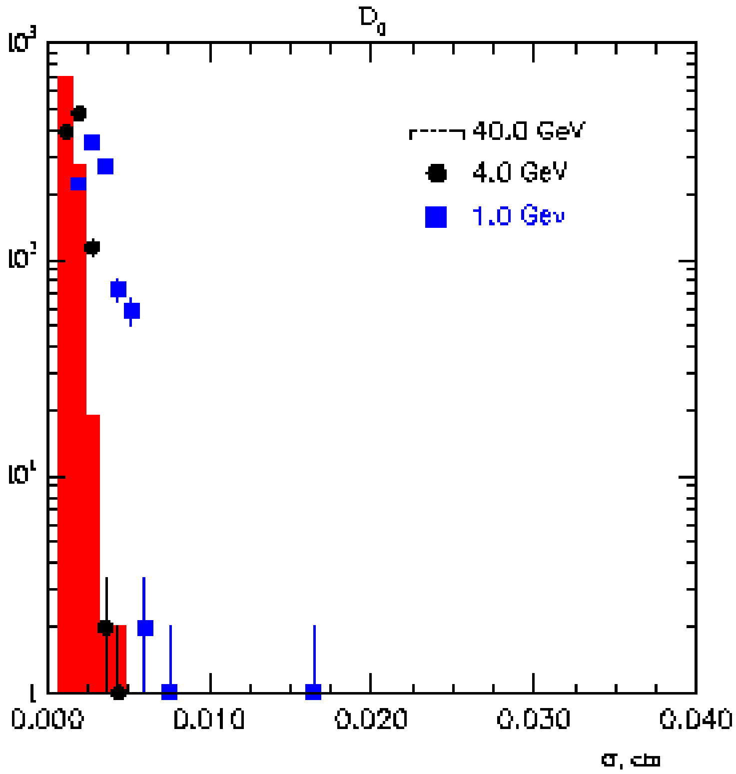
Cot Theta	sigma:	0.83
Curvature	sigma:	1.12
Z0	sigma:	0.85
D0	sigma:	0.87
Phi0	sigma:	1.04

PT = 40.0 GeV

Cot Theta	sigma:	0.84
Curvature	sigma:	1.09
Z0	sigma:	0.77
D0	sigma:	0.89
Phi0	sigma:	0.99

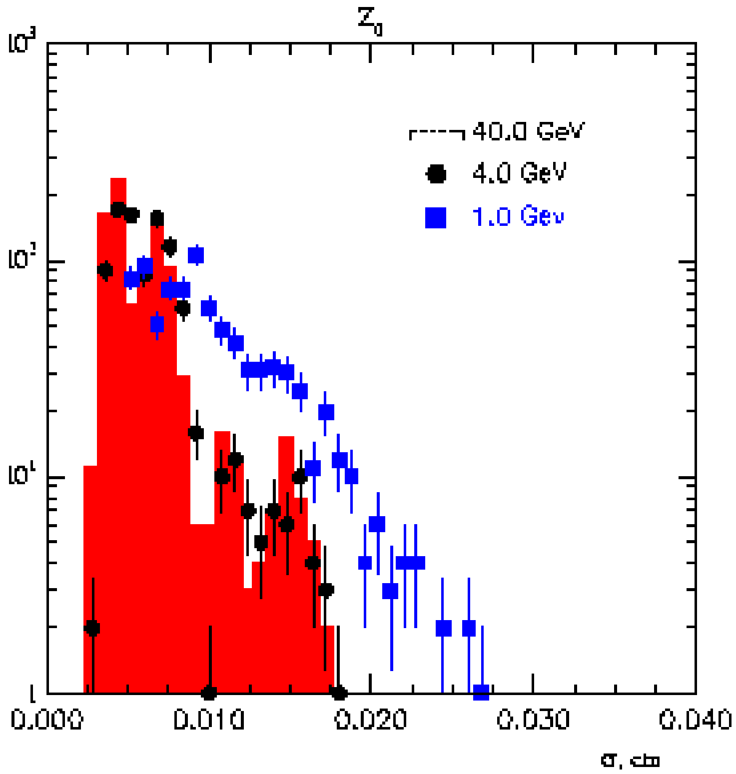
Resolutions for outside-in tracking. These will be written up in a note when they are studied a little more.

D0 Resolution



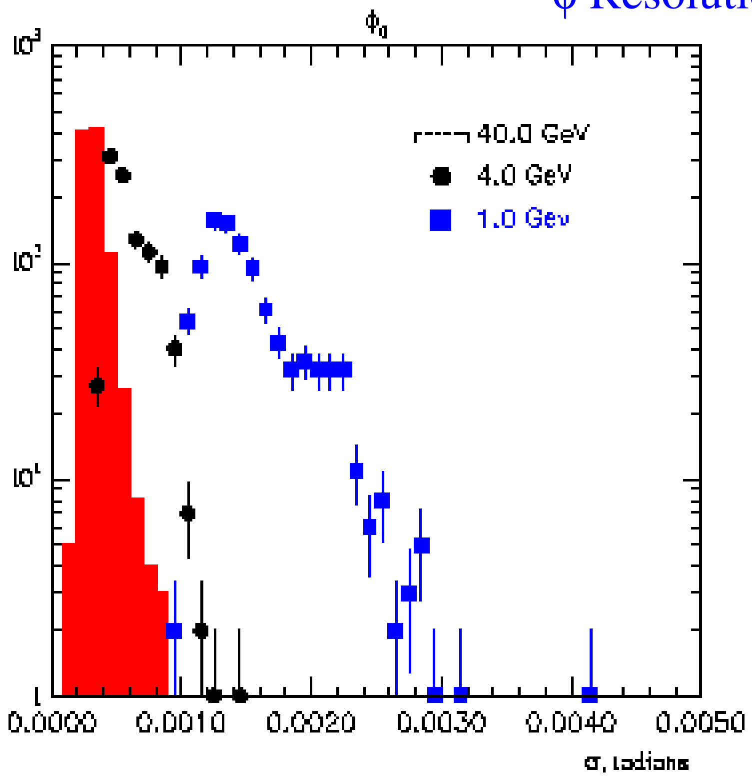
σ (cm)

Z0 Resolution

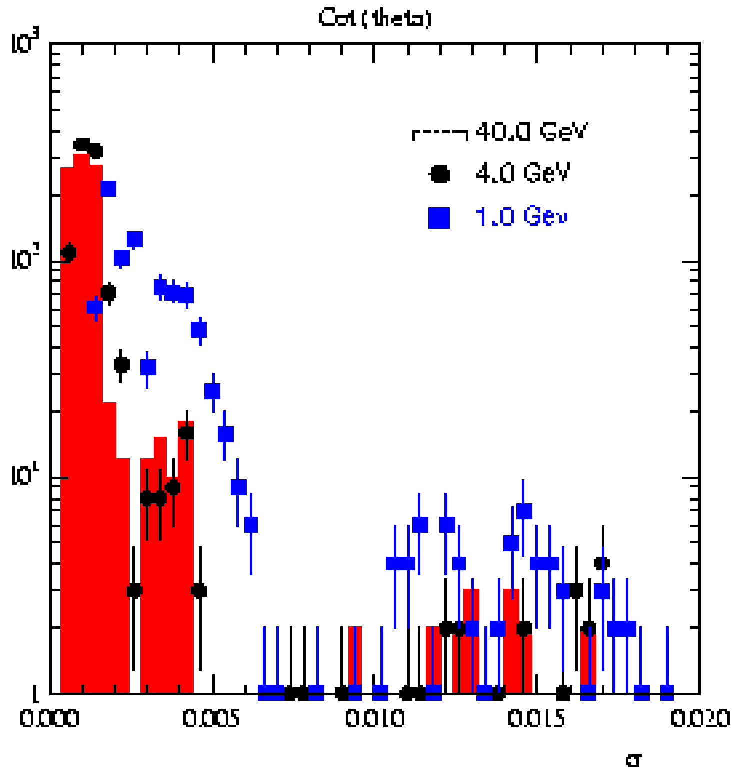


σ (cm)

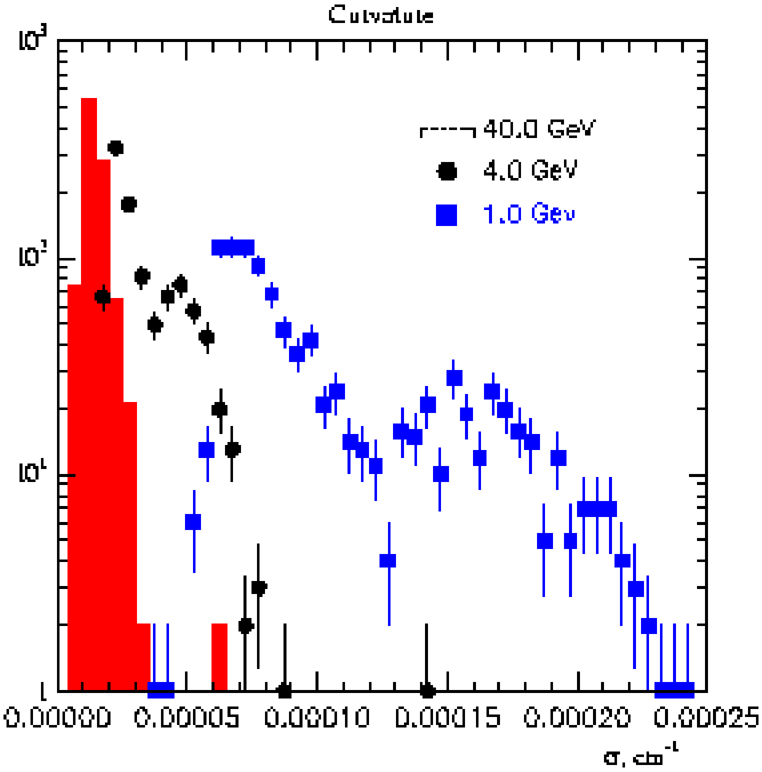
φ Resolution



Cot (θ) resolution



Curvature Resolution



Status of the Pattern Recognition algorithms after the change is unknown, but should not be adversely affected.

Charge deposition model:

You want correct decay lengths.

==> Need correct track resolutions

==> Need correct multiple scattering

==> Also need correct hit resolutions.

3 Models Available in Simulation:

* The Geometrical Model (Don't use this!)
* The Padua Model (Full model, at this time)
* The New Mexico Model (Parametrized model)

The ingredients of the Padova model:

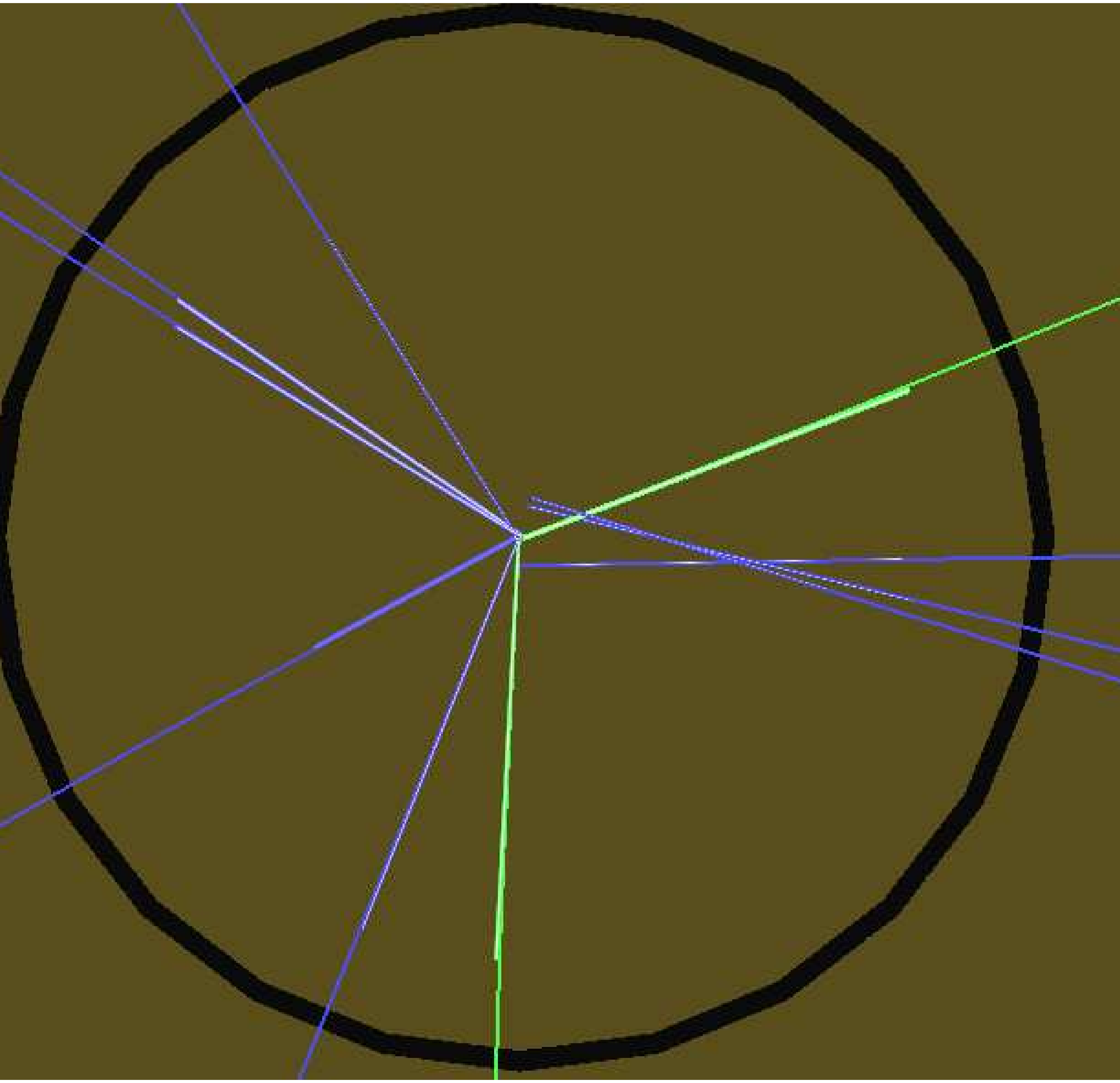
* Step track through Silicon @ 30 μ step size.
* Delta ray generation
* Include Hall effect
* Electron/hole drift with diffusion
* Can take sensor voltages into account.
* Does P/N charge correlation
* You must add detector noise on top, via talk-to
* Does floating strip in layer 00
* Compared with Run I data and SVXII beam test data.

The model runs slowly, but may be optimized.

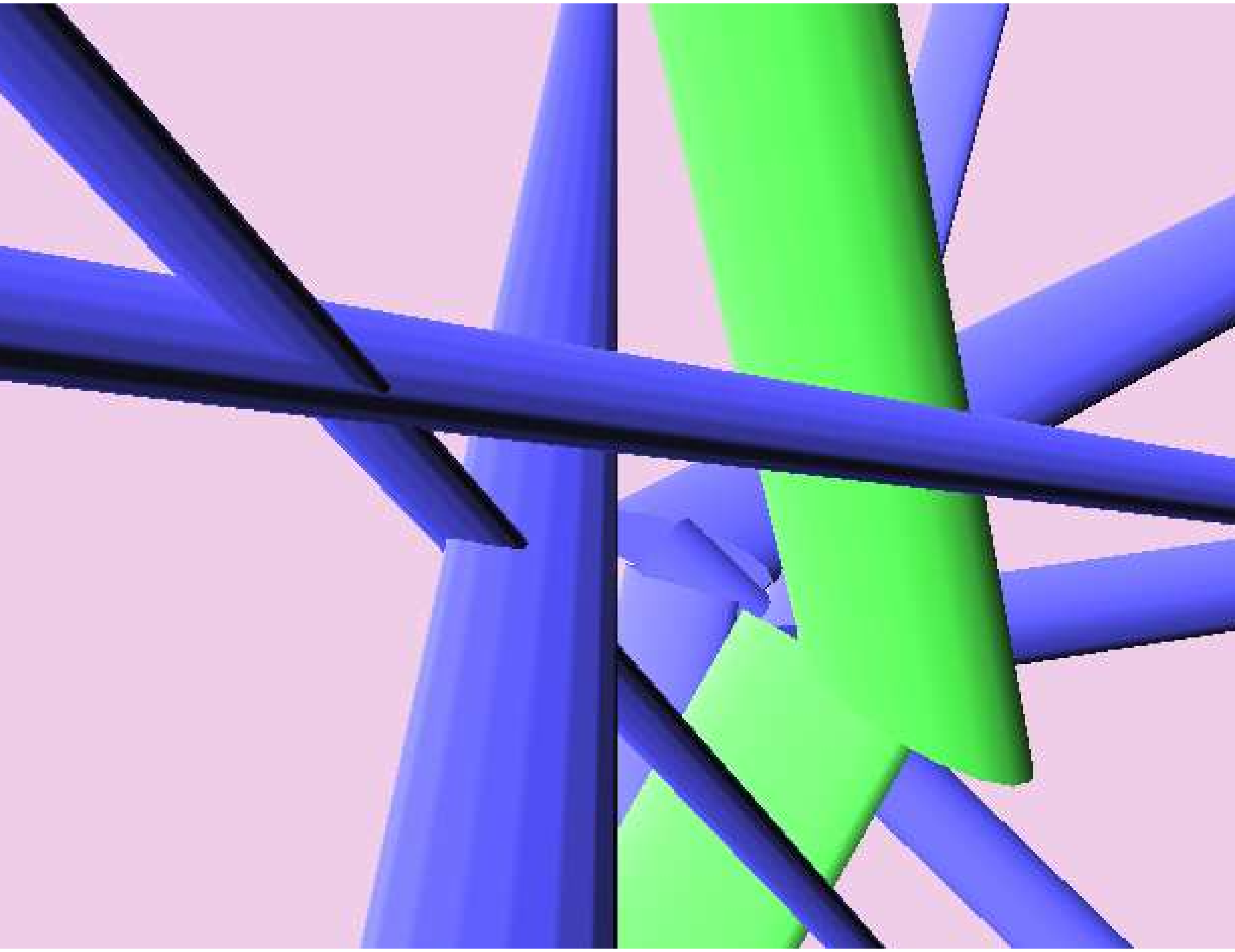
New Mexico model: much faster. Needs validation, comparison w/Pdv

Bottom Line: Factor of 2.5 degradation over geometrical + noise for multistrip clusters. (Matt Herndon)

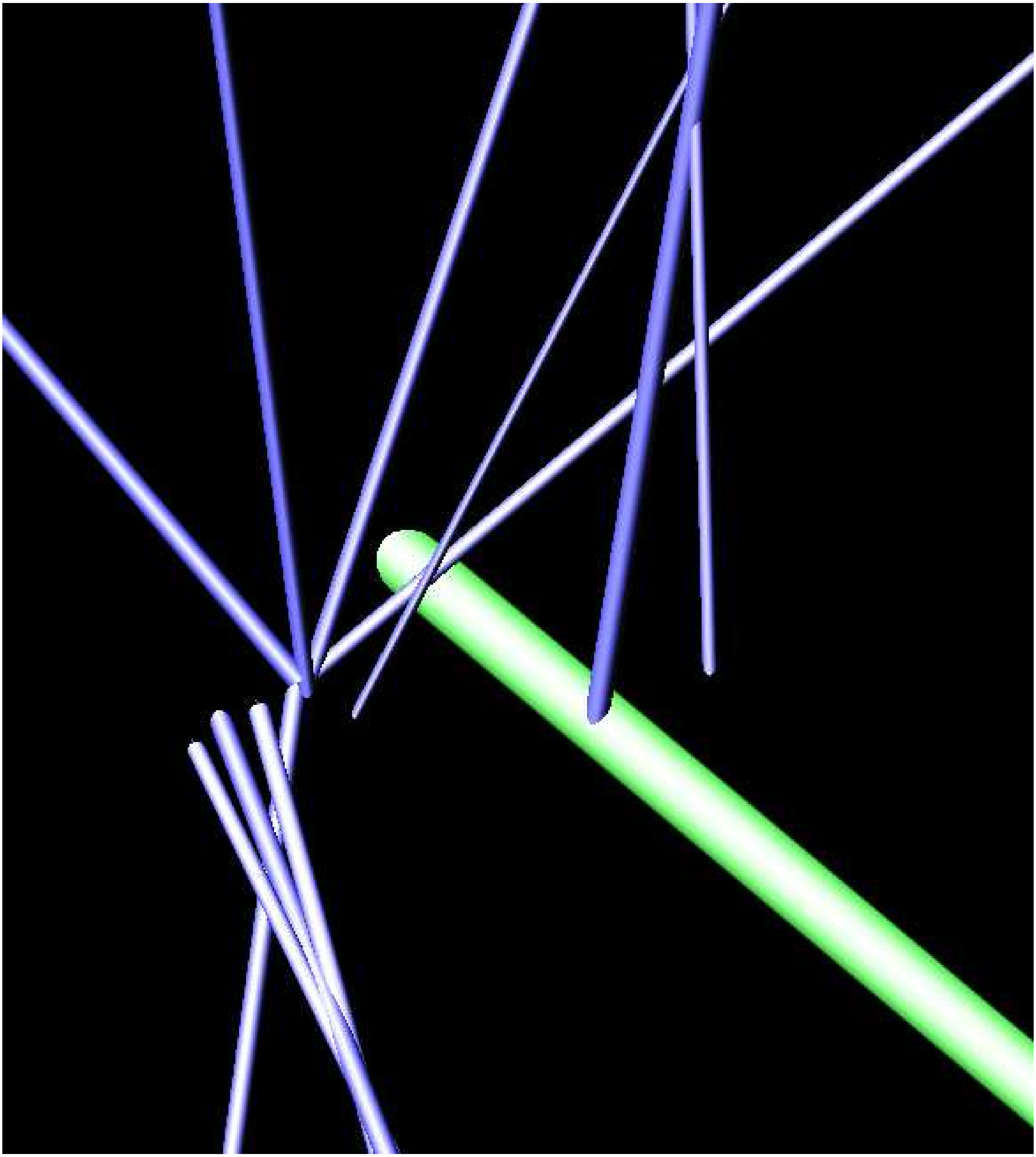
Pythia Bottom event, 3D tracks only.



Same Event. Secondary & tertiary vertices in foreground, primary vertex in background.



2 Bottom Vertices + Primary Vertex in one event in one Event. Pythia + Geant + Production.



Vertexing:

* VXPrim the primary vertex finder.

* VertexFit, the wrap of CVTMFT

* Progressive Vertex Fitters

Exclusive final state reconstruction:

* Universal Finders.

Caveats:

This does not include other objects nominally called "vertex" such as seed vertices for track reconstruction, Monte Carlo truth vertices.

We are cleaning up, getting rid of unused classes and rationalizing the whole system right now.

Vertex Fit:

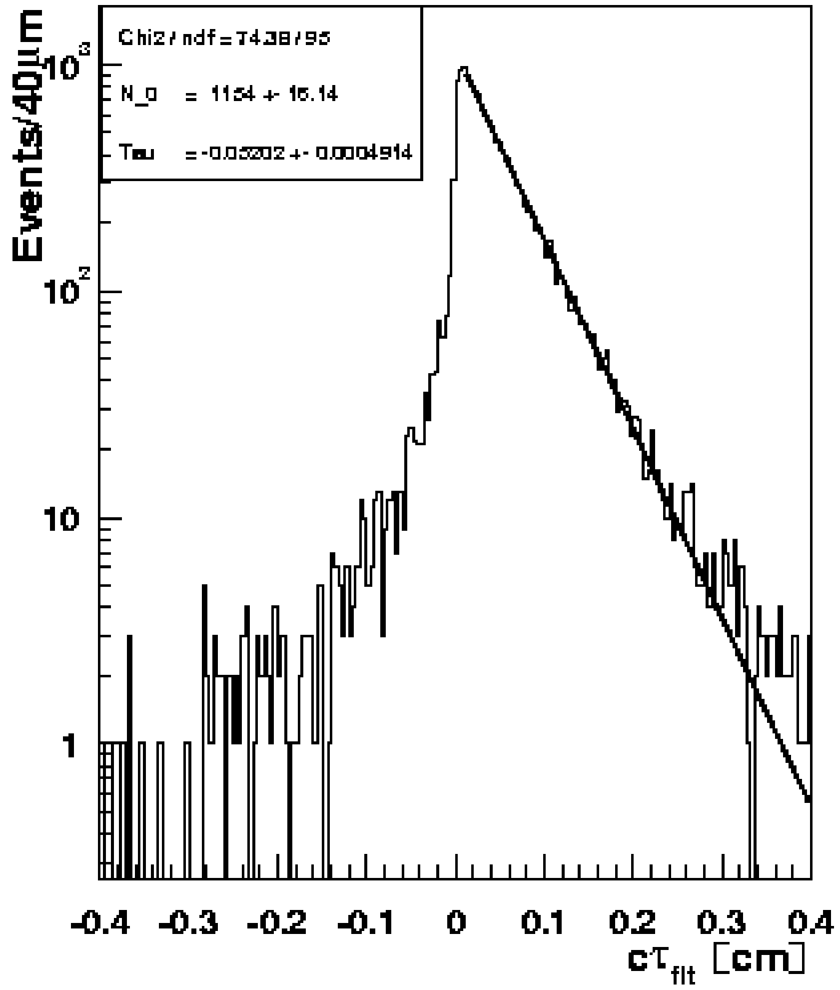
* Has been written by Jon Mariner, Peter Berge & wrapped by Craig Blocker.

* The wrap has been performed in the Sin2BetaMods area but will not remain there for much longer.

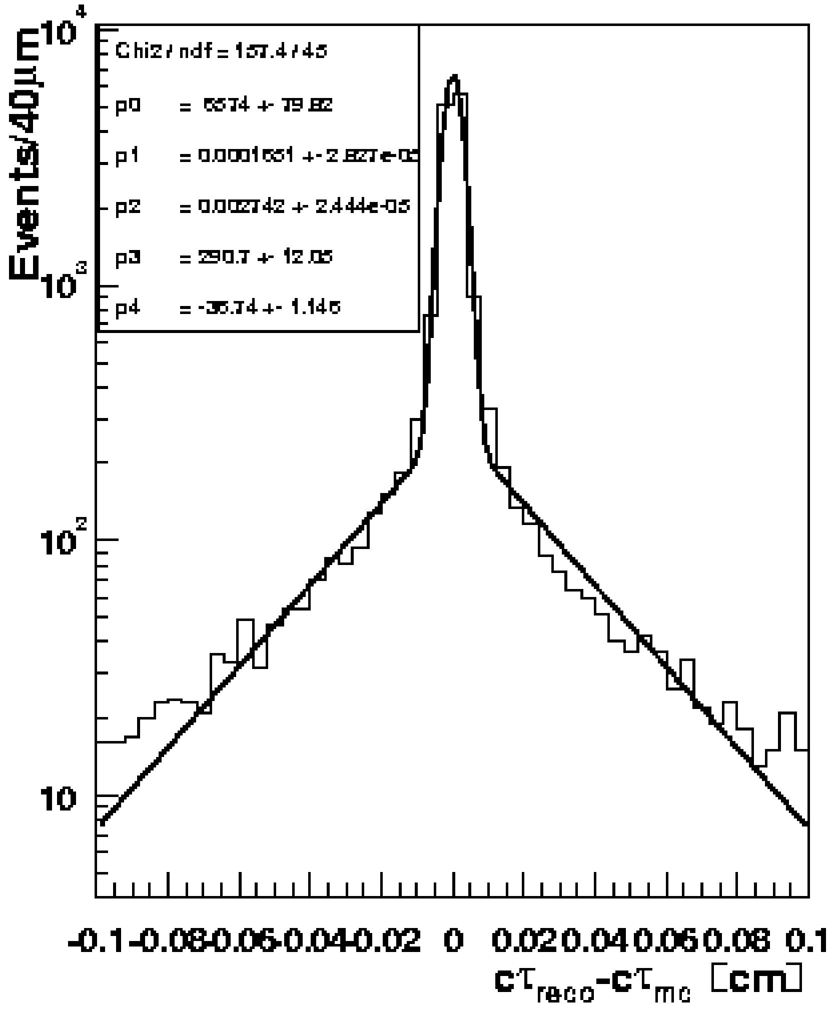
* Is a kinematic fit: all vertices in a decay chain are fit simultaneously. The output provides:

  * Estimates of secondary and tertiary vertices.
  * Fitted masses
  * Refit momentum.

  * Error estimates include correlations between vertex positions.

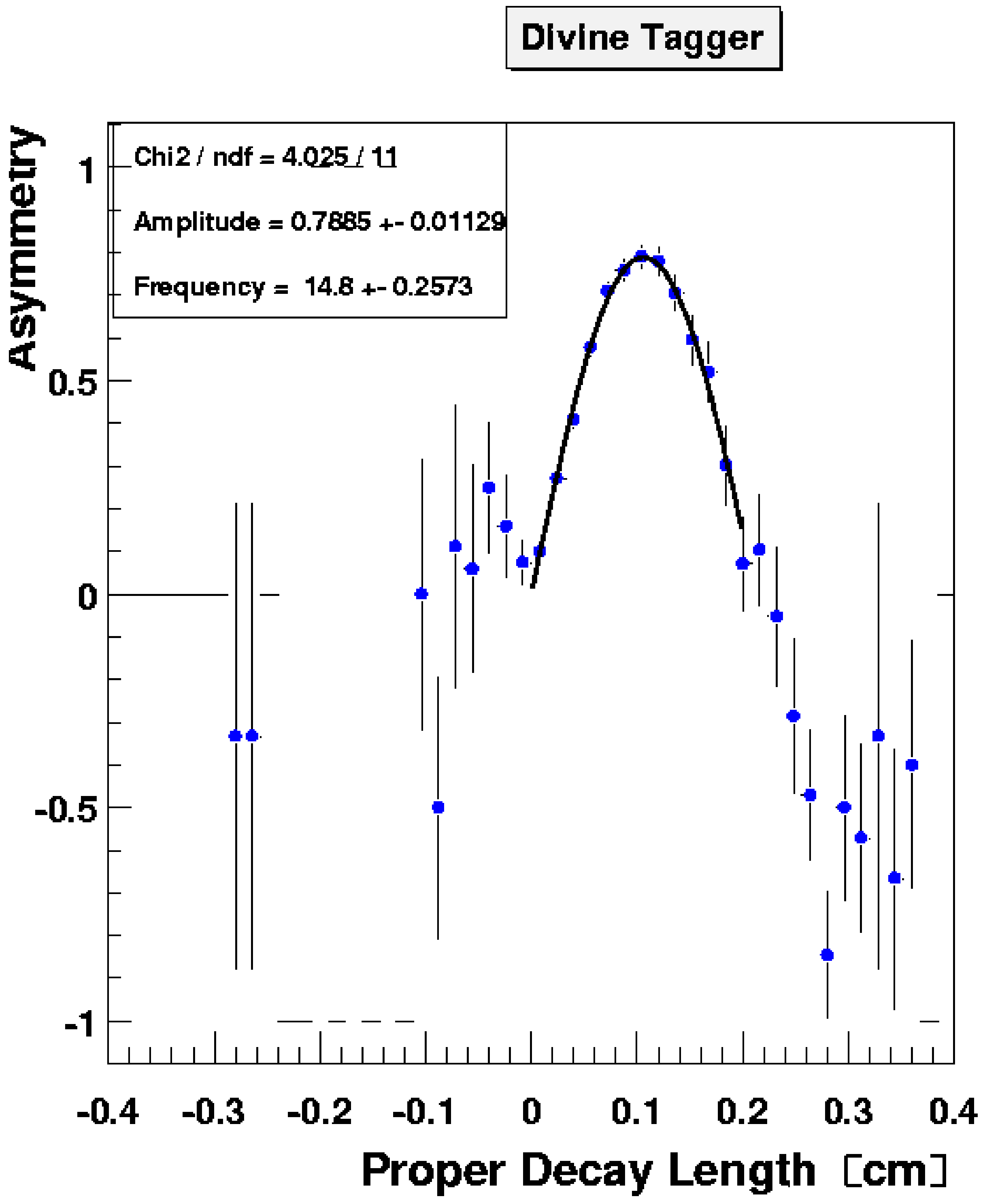


Proper time distribution for $B^0 \to J/\Psi\ K^0_s$



Proper time deviation distribution for $B^0 \to J/\Psi\ K^0_s$

Core: ~ 25-30 μ

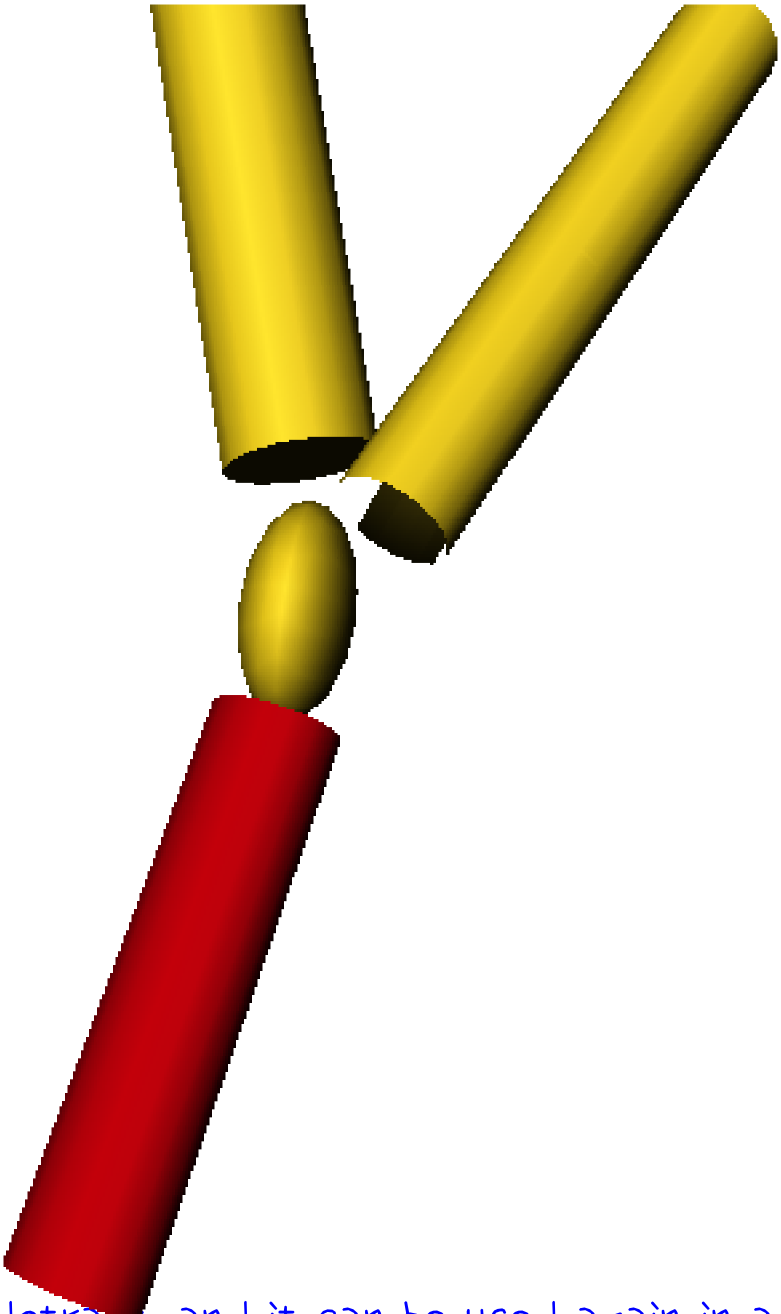


Divine Tagger
Chi2 / ndf = 4.025 / 11
Amplitude = 0.7885 +- 0.01129
Frequency = 14.8 +- 0.2573
Asymmetry
1
0.5
0
-0.5
-1
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
Proper Decay Length [cm]

```
//
// Do vertex fit using VertexFit
//

VertexFit bdFit;
bdFit.init();

// Set the primary vertex.  Until I figure how to properly
// do this, set vertex at (0,0,0) with 50 microns errors
// on x and y and 1 cm error on z.

bdFit.setPrimaryVertex(0.,0.,0.);
float pverr[3][3]={0.005,0.,0.,0.,0.005,0.,0.,0.,1.};
bdFit.setPrimaryVertexError(pverr);

// First vertex is psi -> mu+ mu-

bdFit.addTrack(muPlusTrack(),MU_MASS,VertexFit::VERTEX_1);
bdFit.addTrack(muMinusTrack(),MU_MASS,VertexFit::VERTEX_1);
bdFit.massConstrain(muPlusTrack(),muMinusTrack(), JPSI_MASS);

// Second vertex is Ks -> pi+ pi-

bdFit.addTrack(piPlusTrack(),PI_MASS,VertexFit::VERTEX_2);
bdFit.addTrack(piMinusTrack(),PI_MASS,VertexFit::VERTEX_2);
bdFit.massConstrain(piPlusTrack,piMinusTrack,KS_MASS);

// Point the Ks vertex (2) at the psi vertex (1) in
// 3 dimensions (type = 2).

bdFit.vertexPoint_3d(VertexFit::VERTEX_2,VertexFit::VERTEX_1);


// Do fit

 bdFit.fit();
 if(bdFit.status() != 0) bdFit.printErr();
```

Progressive Fitters

Principle: Propagation of errors.



Daughter tracks are first fitted to common vertex.

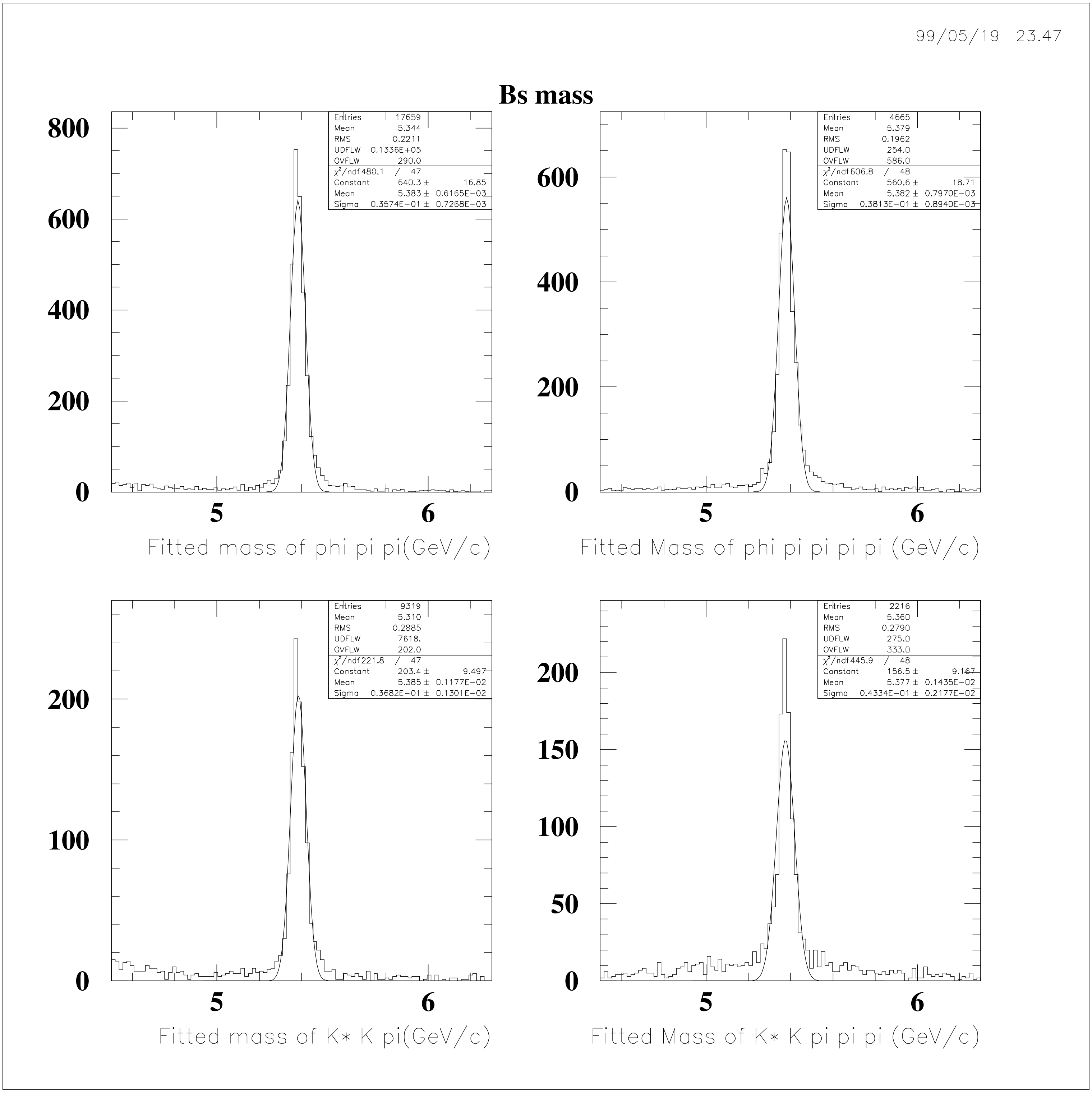
Then the momentum, position of these tracks plus the assumption of a common vertex are used to create the trajectory of the charged or neutral mother... with errors on its parameters.

These take into account all the errors & correlations. Mathematics described in Lucshinger & Graab, Comp. Phys. Comm. 76 (1993)

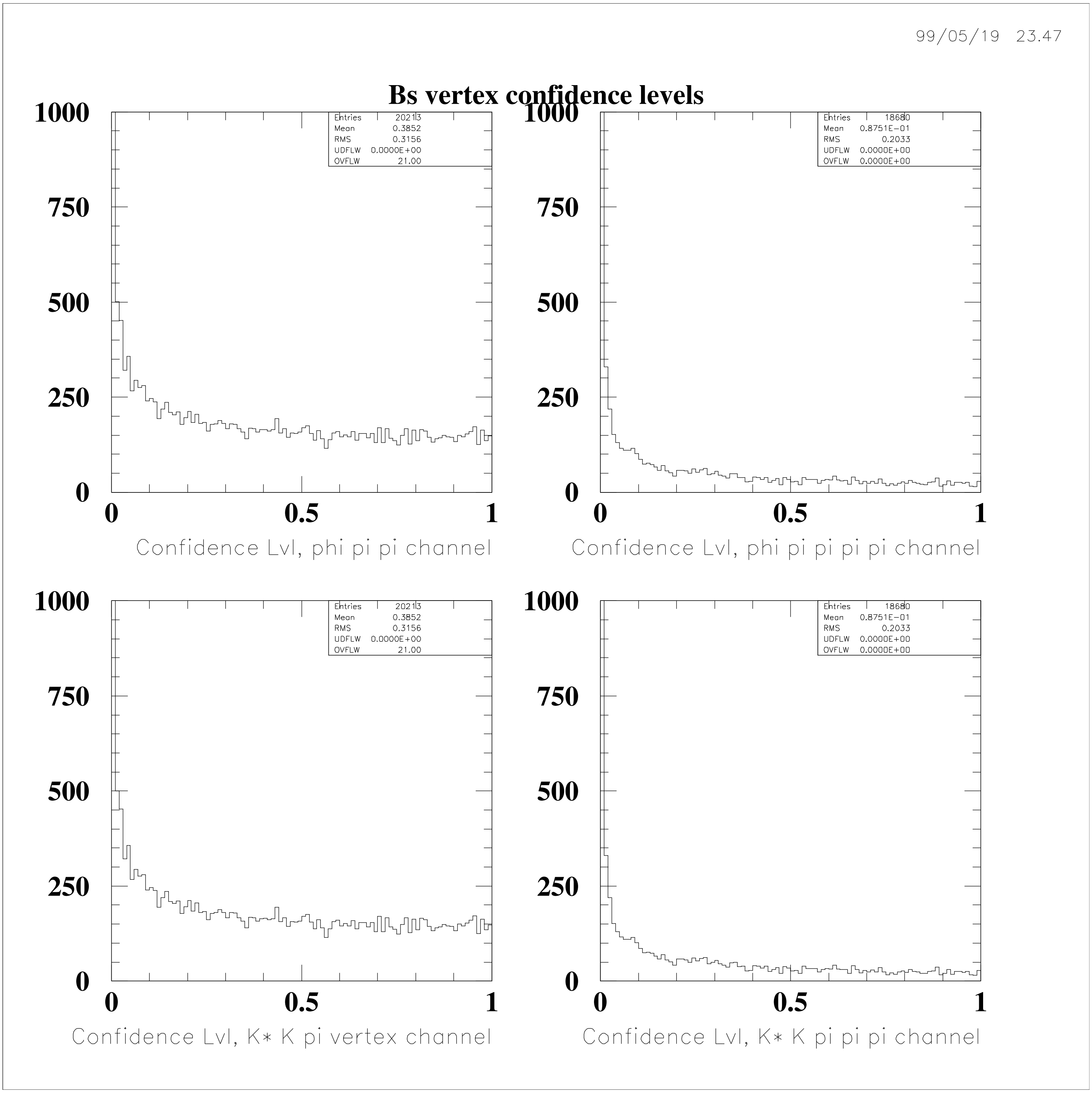
The new object is called pseudotrack, and it can be used again in a fit.

It shares part of its interface with CdfTrack.. That's the part that provides transverse displacement from a point & transverse error.

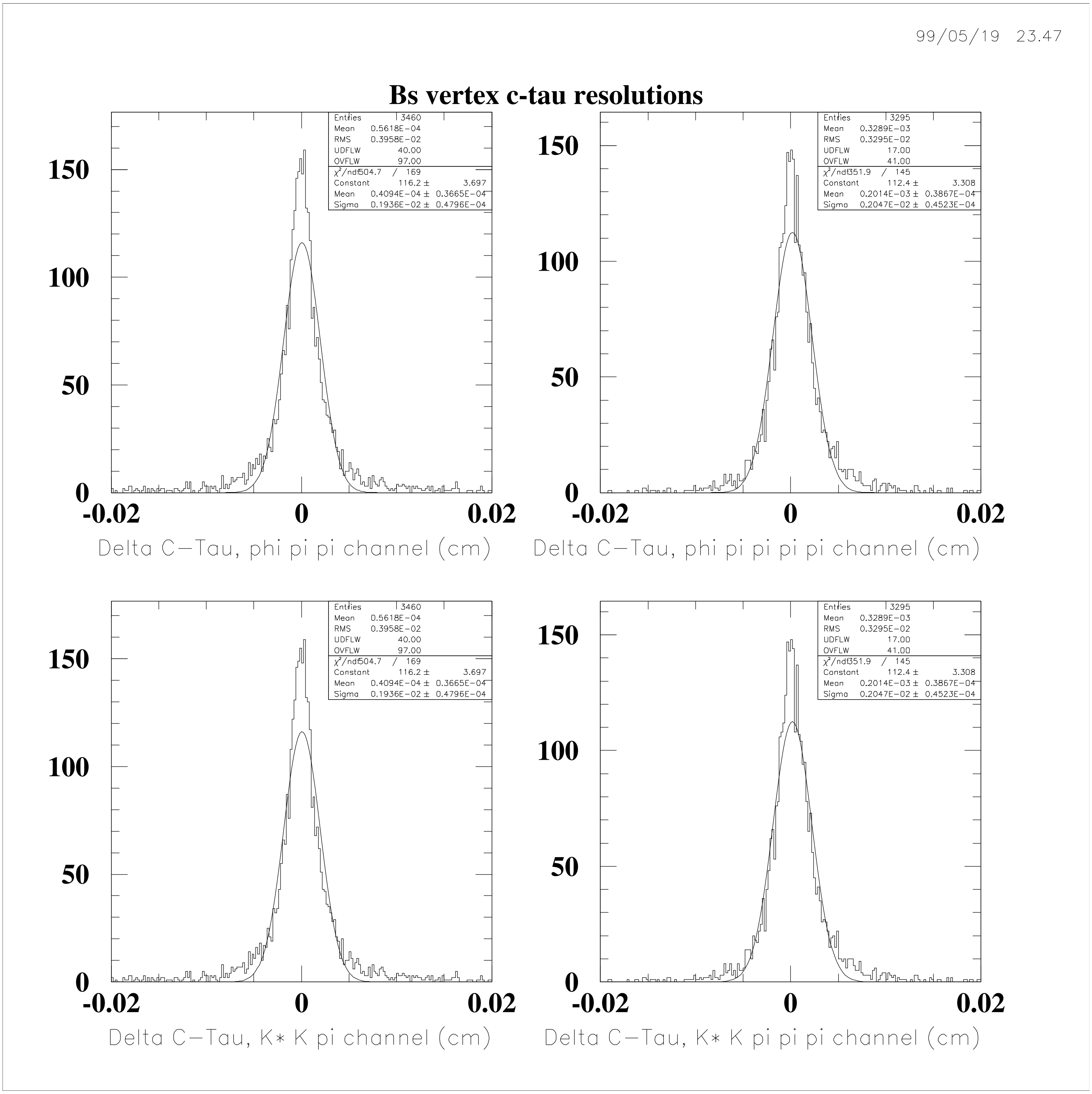
B_s mass reconstruction in all four channels.



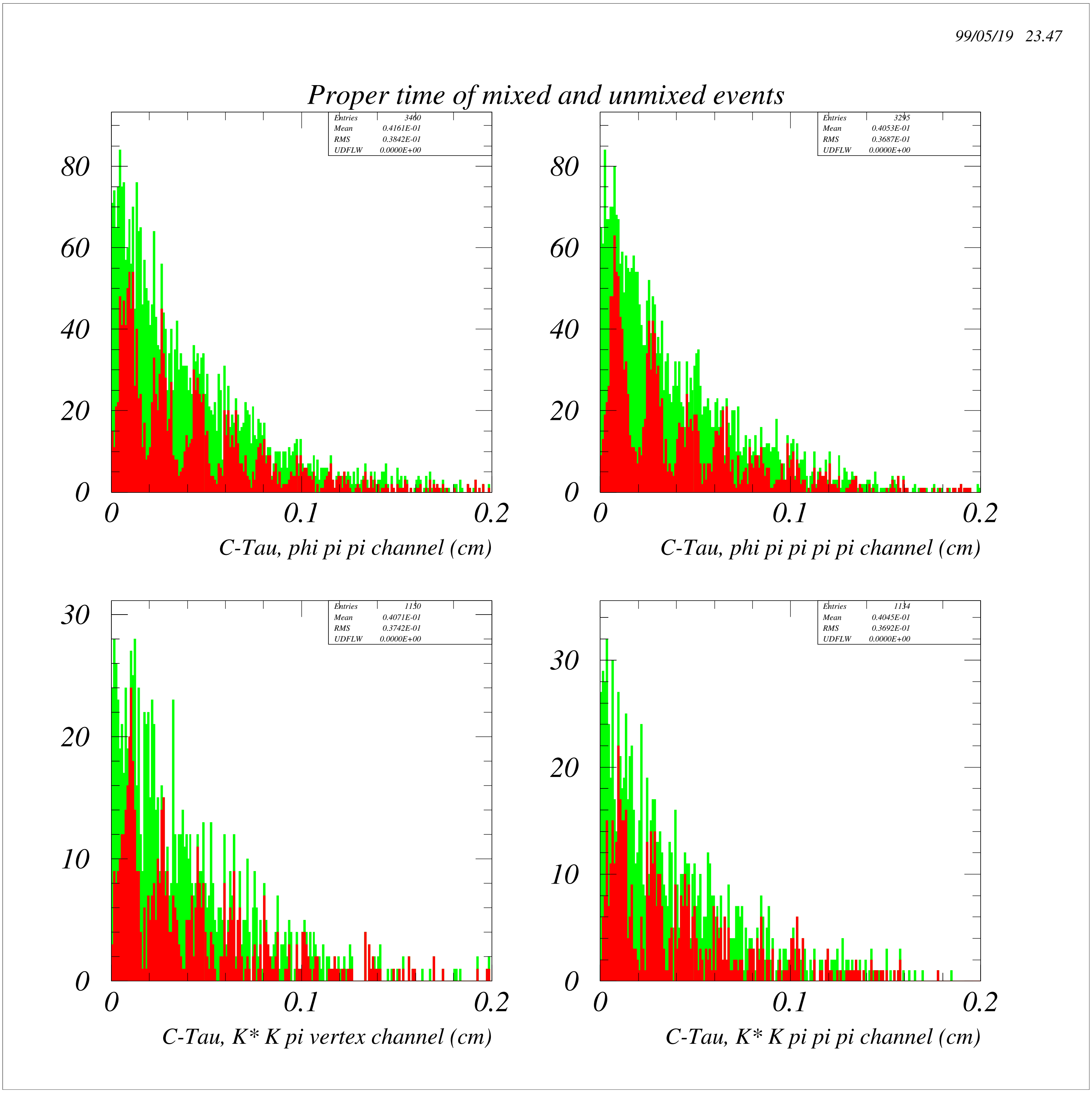
B_s vertex probability for all B_s Vertex Candidates.



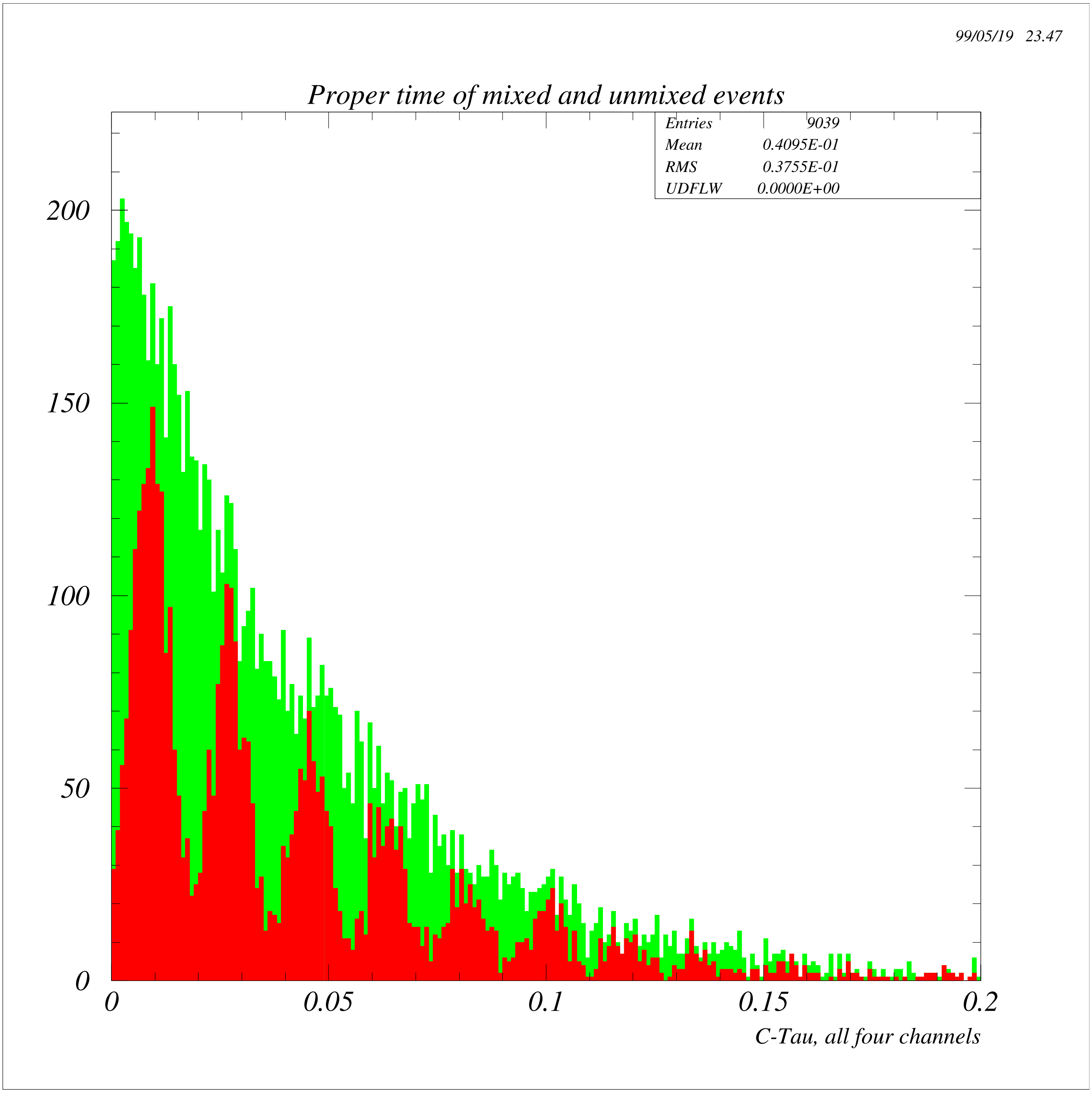
B_s $c\tau$ resolutions



B_s oscillation in all four channels.



All four channels combined....



Plans:

* We are now working to define a common EDM2-compliant vertex class.

* This vertex should be storable (want to be able to save it in the event,for use by later by other modules, even other jobs).

* We'd then like an abstract base class for vertex fitters.

* Should be possible to run a whole analysis with different vertex fitting algorithms... may be good for comparisons.

Universal Finders (Petar Maksimovic) for fully reconstructed final states.

Should be possible to describe the reconstruction of final states thatone wants to do using a few classes to create the topology that one is trying to reconstruct, and the cuts to be applied.

E.G. $B^0 \to J/\Psi\ \phi$

$J/\Psi \to \mu^+ \mu^-$

$\phi \to K^+K^-$

A universal finder may ultimately consult a particle database and figure out for itself that it needs to:

* make J/Ψ, ϕ vertex copunctual with B^0 vertex.

Ultimately these classes may make it uneccessary to know anything about the vertex fitters, at all (for certain users).

Summary:

* The full silicon system including layer 00 is present in both simulation and reconstruction. Simple benchmarks have been done, more stringent tests to come.

* The Padova charge deposition model is operational. It should be used for serious physics studies.

* Physics groups should begin an effort to characterize the tracks that come out of default tracking and to try to understand their impact parameter properties and develop basic track quality cuts.

* For now we have GEANT to do this.

* Algorithms for vertex fitting are in good shape.

* Their interfaces are another story.

* Procedure to rationalize this is beginning now and should be usable in a few more weeks.