Tracking in Silicon Detectors; Combination with TPC Tracking - Towards Full Tracking

A.Raspereza, A.Frey, X.Chen ***MPI-Munich***

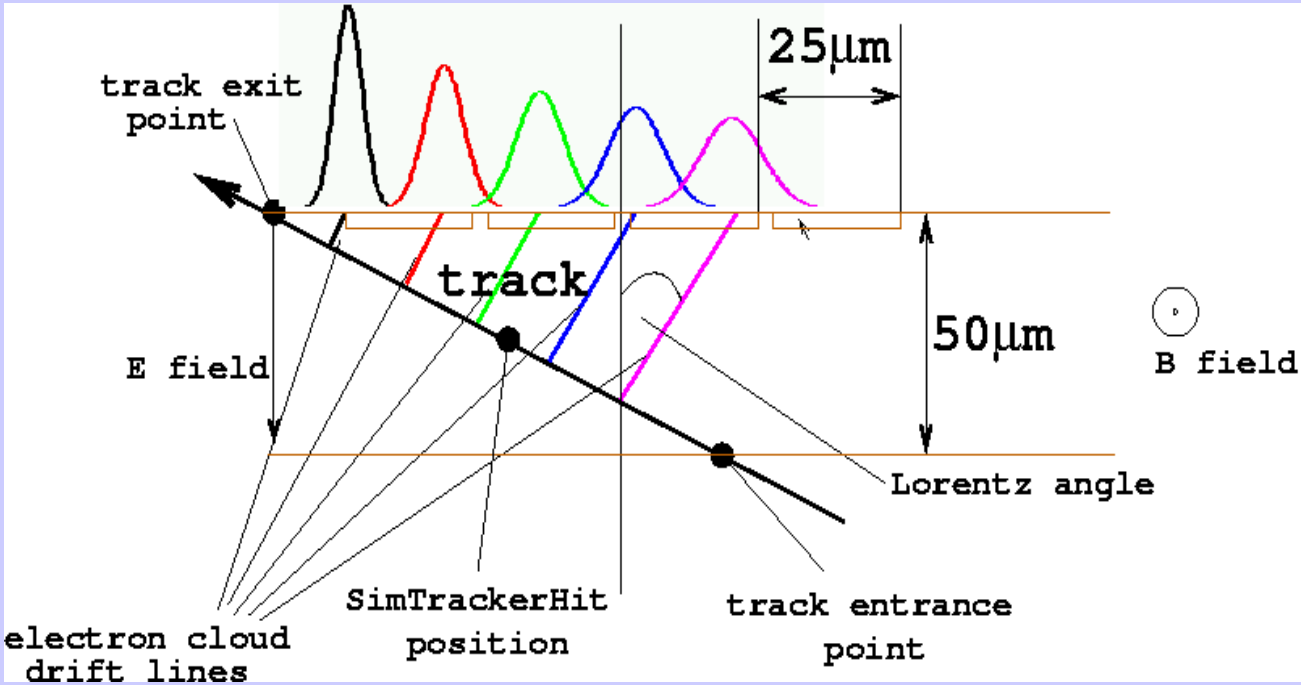
ILC Software video-meeting 17/07/06

Outline

- Si Detector Digitization
- Tracking in Si Detectors (VXD+FTD+SIT)
  - ✔ Algorithm
  - ✔ Structure of Code (Marlin processors)
  - ✔ Performance
- Track Fitting
  - ✔ Simple Helix Fit
  - ✔ Delphi Fit (Kalman fitter accounting for interactions)
  - ✔ RAM-wise Material Database
  - ✔ C++ interface to Delphi FORTRAN code
- Outlook : Combining Si Tracking and TPC Tracking

Si Detector Digitization

- Two approaches for Si Detector digitization
  - Straightforward gaussian smearing of SimTrackerHit position (VTX/FTD/SITDigiProcessor in Marlin)
  - Detailed digitization based on features of VXD technology and readout
- Example : DEPFET
  - VTXDigitizer ⇨



- Option to emulate/overlay beam induced background hits (integration time is specified for each detector layer as external Processor parameter)

Si Tracking (Strategy, Features)

- Tracking in Si Detectors is implemented via single Marlin Processor (SiTracking, obsolete name VertexTracking), which does
  - ✔ separate tracking in cyllindrical/laddered detectors (VXD-SIT, two SIT layers are treated as outermost layers of VXD)
  - ✔ separate tracking in planar discs (FTD)
  - ✔ combination of track segments found in VXD-SIT & FTD
  - ✔ track fitting, determination of track parameters @ PCA
- Inputs/Prerequisites
  - ✔ Collection of digitized TrackerHits (VXD/FTD/SIT)
  - ✔ Optionally, constructed RAM-wise material database needed for Delphi Fit (explained later)
- Output
  - ✔ Collection of tracks : track parameters + cov. matrix + list of hits contributing to a given track
- Code is accompanied by doxygen documentation (detailed tex/ps/pdf documentation is foreseen)

Algorithm (General Description)

- In each branch (VXD-SIT or FTD) hit triplets are searched for in different layers (starting from outermost layers), followed by inward search for additional hits in other layers
- Track fit χ^2 as the main criterion to accept track candidate in the branch (tracks are allowed to share only one hit if this is innermost hit in both tracks – handle of possible conversion in the Si layer)
- Track segments in two branches are merged on the basis of combined fit χ^2 criterion ➞ single track containing hits in VXD/SIT & FTD
- Track quality criteria (to suppress fake track rate due to beam induced background)
  - Total number of hits > 3
  - At least one hit in SIT if track is found to cross both layers of SIT
  - At least $N_{FTD}/2$ hits in FTD if track is found to cross N_{FTD} layers of Forward Tracking Discs
  - Mild cuts on IP D_0 (Z_0) < 2(4) cm (vanishing effect on secondary / tertiary vertex finding in decays of D- and B-mesons)

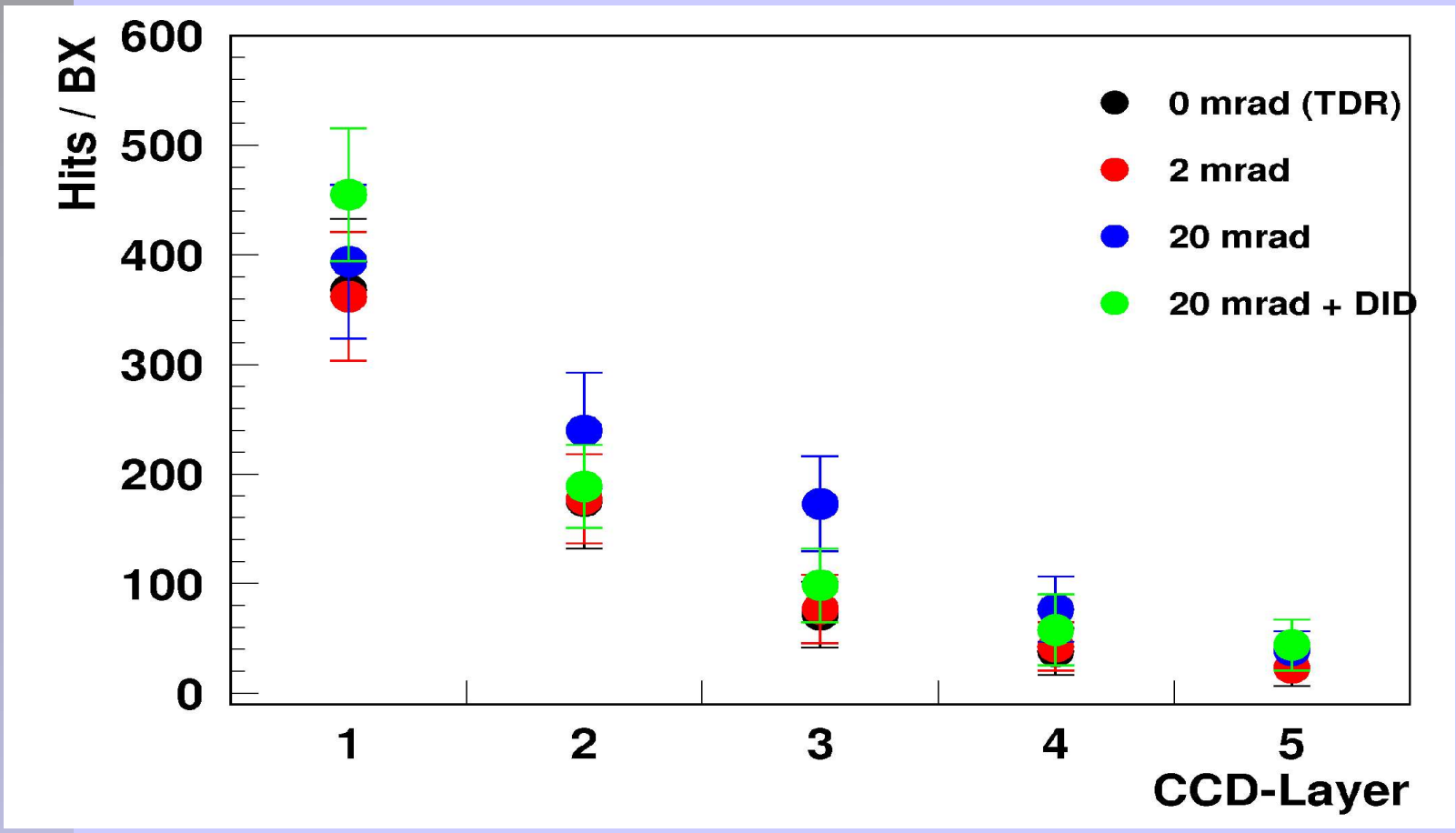
Structure of Code, Dependencies

- Patrec + Track fitting (Marlin processor “SiTracking”)
  - HelixClass (MarlinUtil package)
  - tfithl.F (simple helix fit from Brahms tracking package)
  - trkfit.F (interface to Delphi Kalman Fitter)
- An user has an option to chose between simple helix fit and Delphi Kalman Fitter, if second option is chosen an user should include processor “MaterialDB”
  - Reads GEAR and calculates detector material shapes
  - Fills FORTRAN common blocks (on-flight RAM-wise database) storing material shapes, properties
  - Database is accessed by Delphi fitting routines

Performance

- Performance is evaluated on the sample of $t\,t \Rightarrow$ 6-jet events in terms of
  - Track finding efficiency
  - Spoiled track rate
  - Fake track rate
  - Splitted track rate
- Distributions are binned in $\log_{10}(P_T)$ & $\cos\theta$
- Study is done for no-background scenario and for scenario assuming integration of 75 BX in the innermost layer of VXD and 150 BX in other layers (as suggested by M.Winter), FTD and SIT are assumed to be capable of tagging every single BX

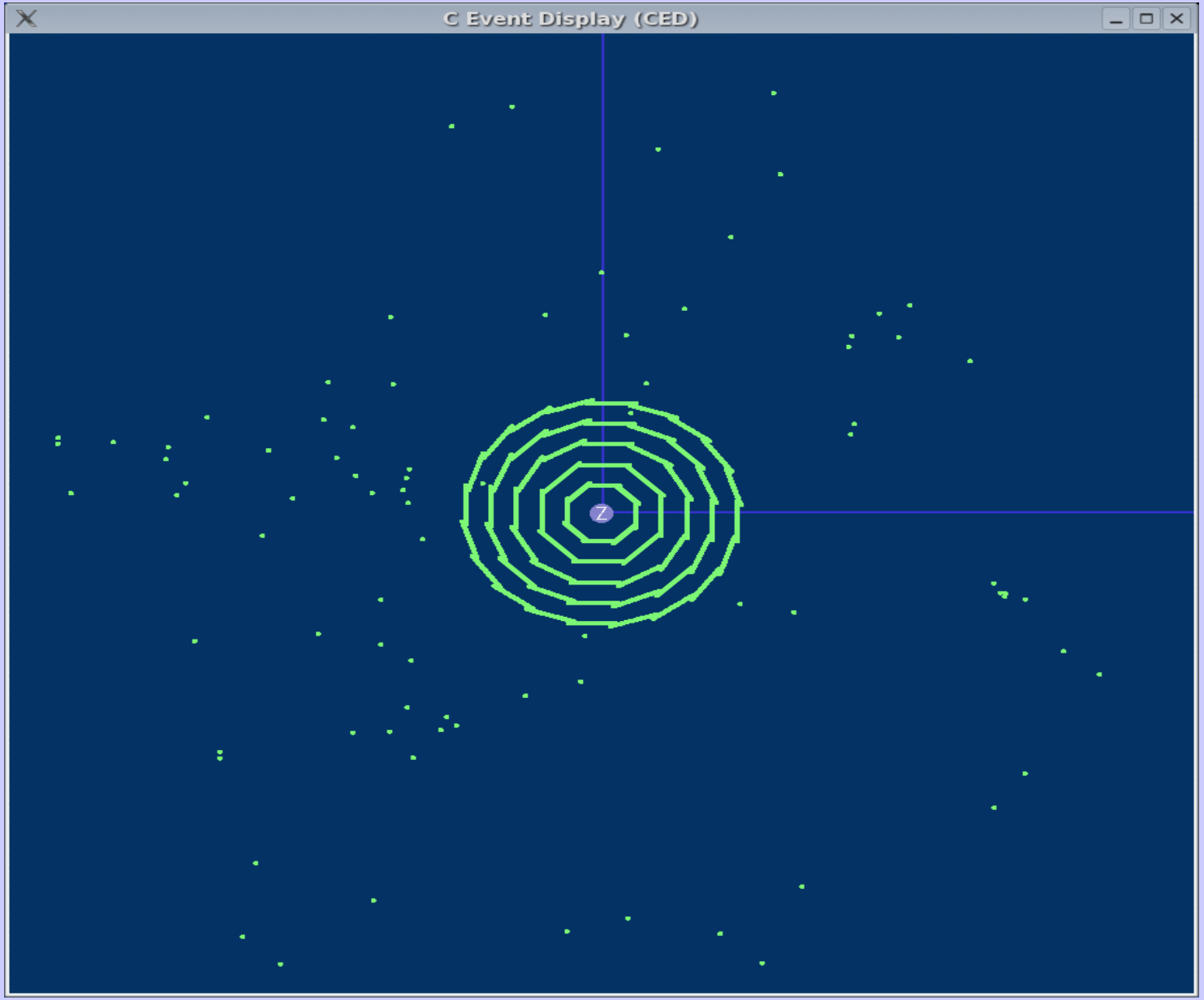
Background Rates



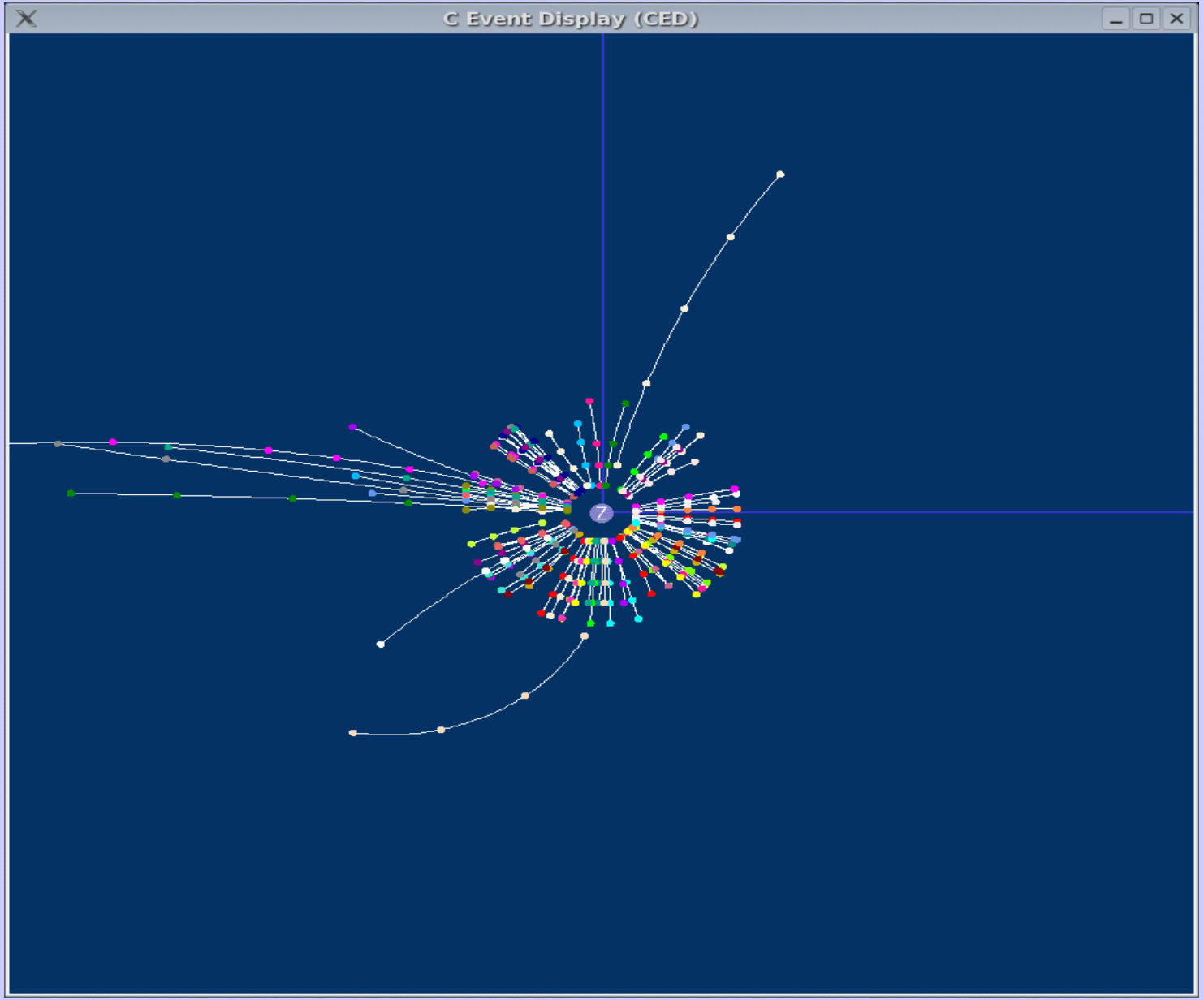
K. Buesser

- Assumed rates (0,2 mrad crossing angle)
  - ✔ Layer # 1 : 75 BX ⇒ 28k hits
  - ✔ Layer # 2 : 150 BX ⇒ 28k hits
  - ✔ Layer # 3 : 150 BX ⇒ 13k hits
  - ✔ Layer # 4 : 150 BX ⇒ 8k hits
  - ✔ Layer # 5 : 150 BX ⇒ 6k hits

Event Display with Background Hits (tt⇒6-jets)

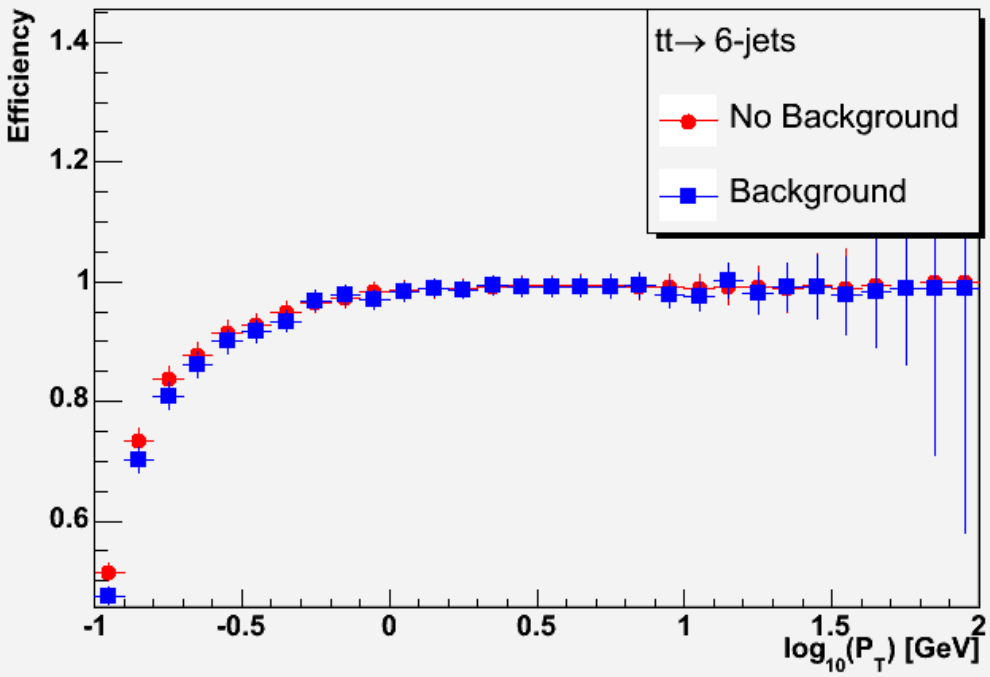


Event Display after Reconstruction (tt⇒6-jets)

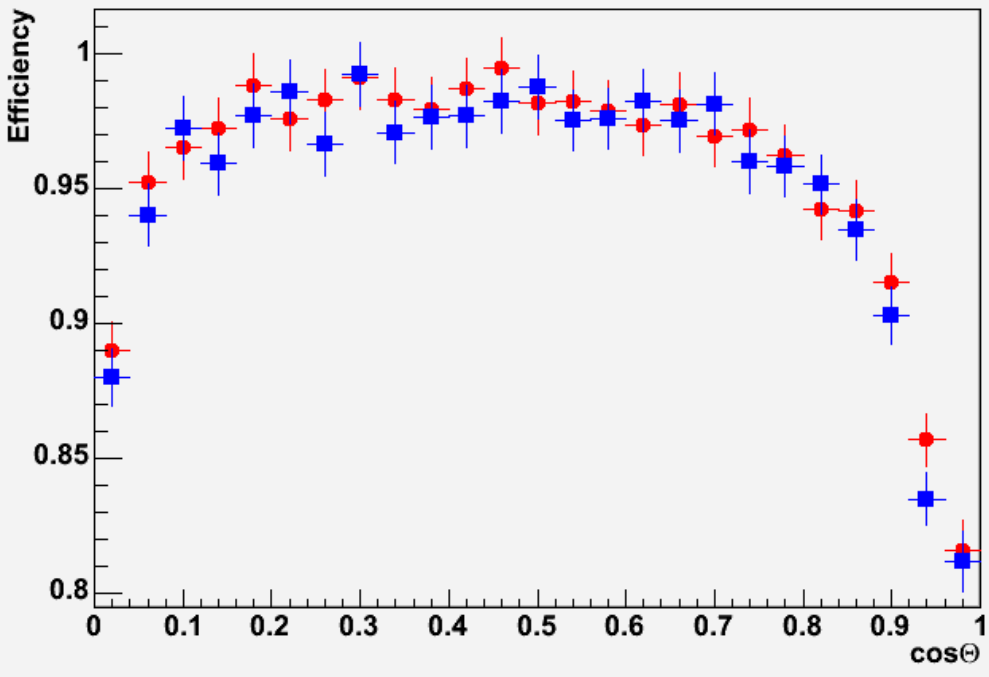


Performance

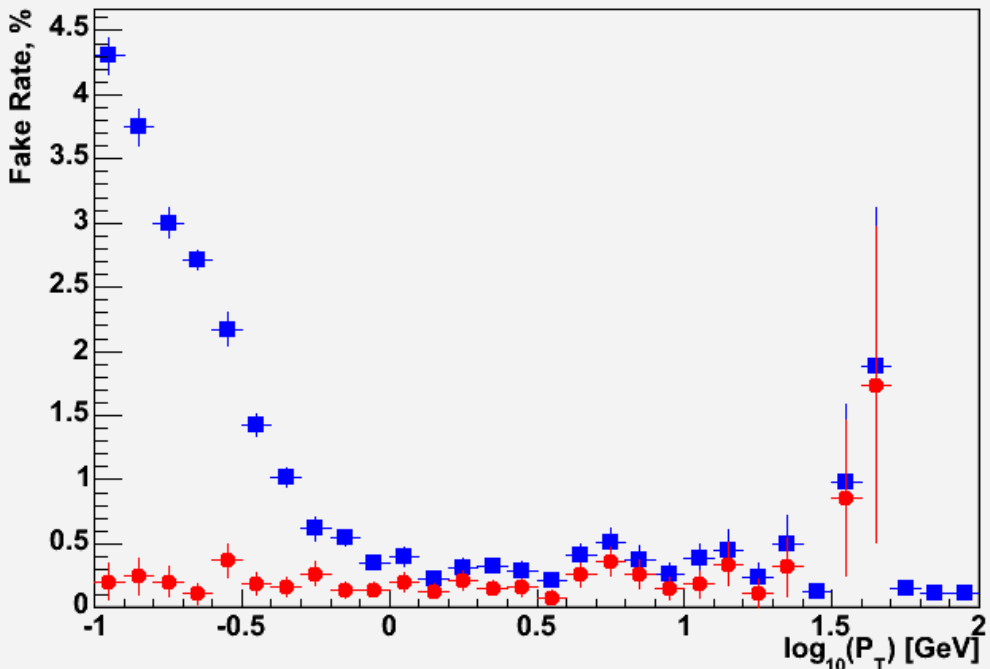
Track finding efficiency (VTX+FTD+SIT)



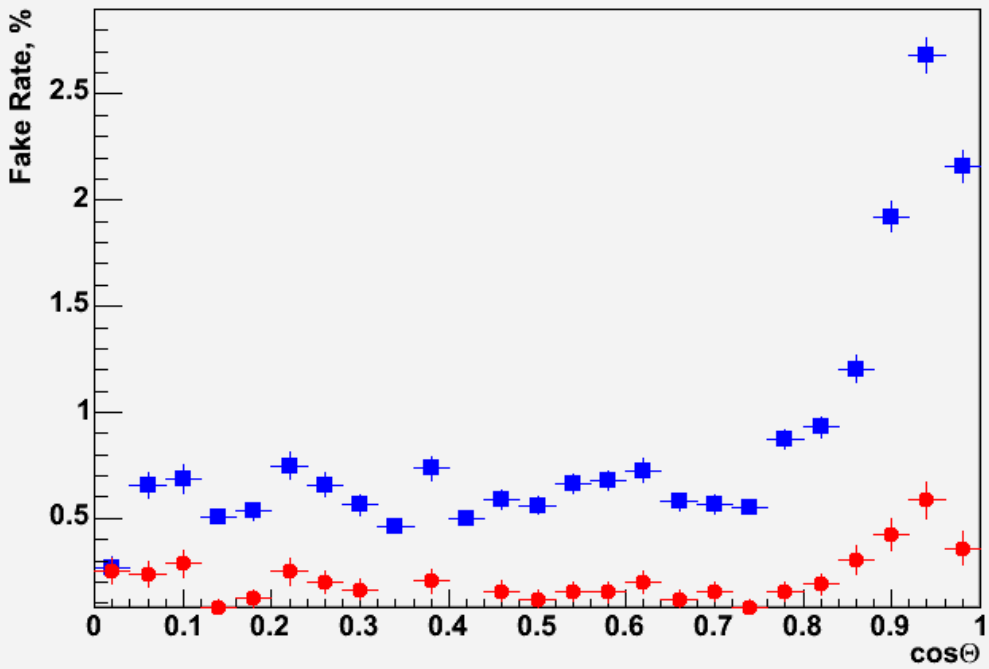
Track finding efficiency (VTX+FTD+SIT)



Fake Track Rate (VTX+FTD+SIT)



Fake track rate (VTX+FTD+SIT)



Track Fitting (Simple Helix Fit)

- Brahms tracking package provide a variety of methods to perform track fitting (adopted FORTRAN LEP code).
- Rewriting code in C++ is quite a challenge ⇒ use instead C++ interface to existing FORTRAN implementations (work is ongoing @ MPI to write track fitting code in C++)
- Simple helix fit is performed with routine tfithl.F
- Can be called within C++ code

```
extern "C" {
  void tfithl_(int & NPT, double * XF,double * YF, float * RF, float *PF,
               double * WF,float * ZF,
               float * WZF, int & IOPT, float * VVO,
               float * EEO, float & CH2PH, float & CH2Z);
}
```

- Fits in R-Φ & S-Z projections are disentengled
- No energy loss, MS are taken into account ➞ track parameters are universal / constant along track

Delphi Kalman Fitter

- A more sophisticated fit is provided by Delphi Kalman fitting routine (FORTRAN routine trkfit.F)
- A C++ interface is provided by C++ routine (part of MarlinUtil package)

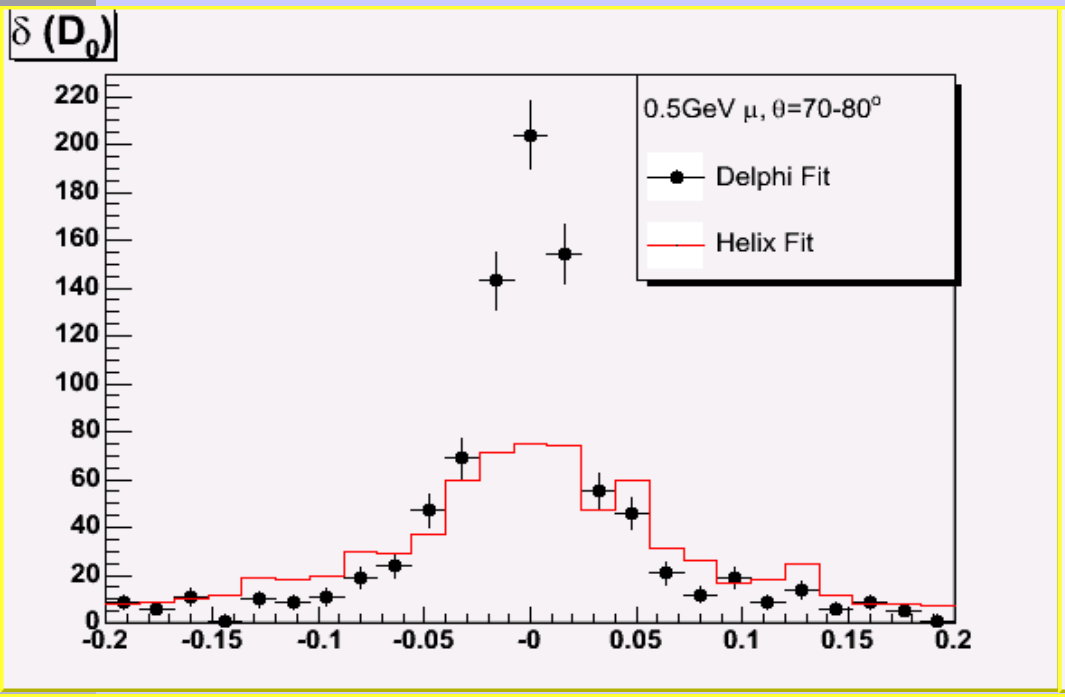
```
int TrackFitting(int & nhits, float bField, int * idet, int * itype,
                 float * x, float * y, float * z, float * RPReso,
                 float * ZReso, float xRef, float yRef, float zRef,
                 int iopt,
                 float * param, float * eparam, float & chi2, int & ndf)
```

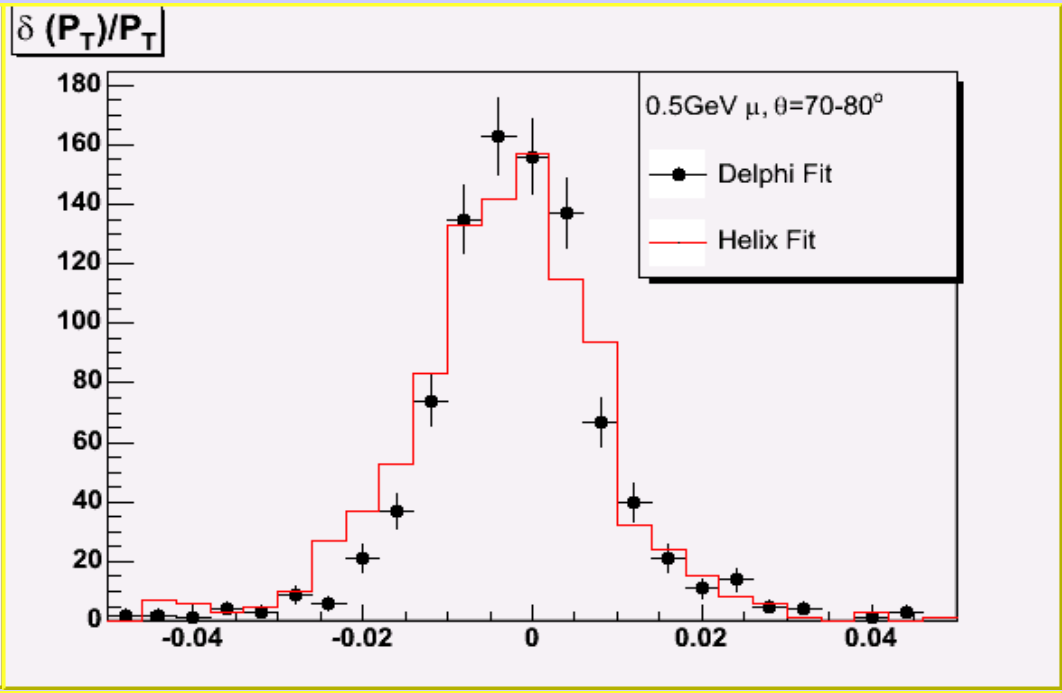
- Fit takes into account energy loss and MS ⇨ track parameters vary along track
- Track parameters @ a given point are accessed by specifying reference point (float xRef, float yRef, float zRef)
- Each hit must be assigned detector ID and type (hit in planar detector or in cyllindrical one : int * idet, int * itype)

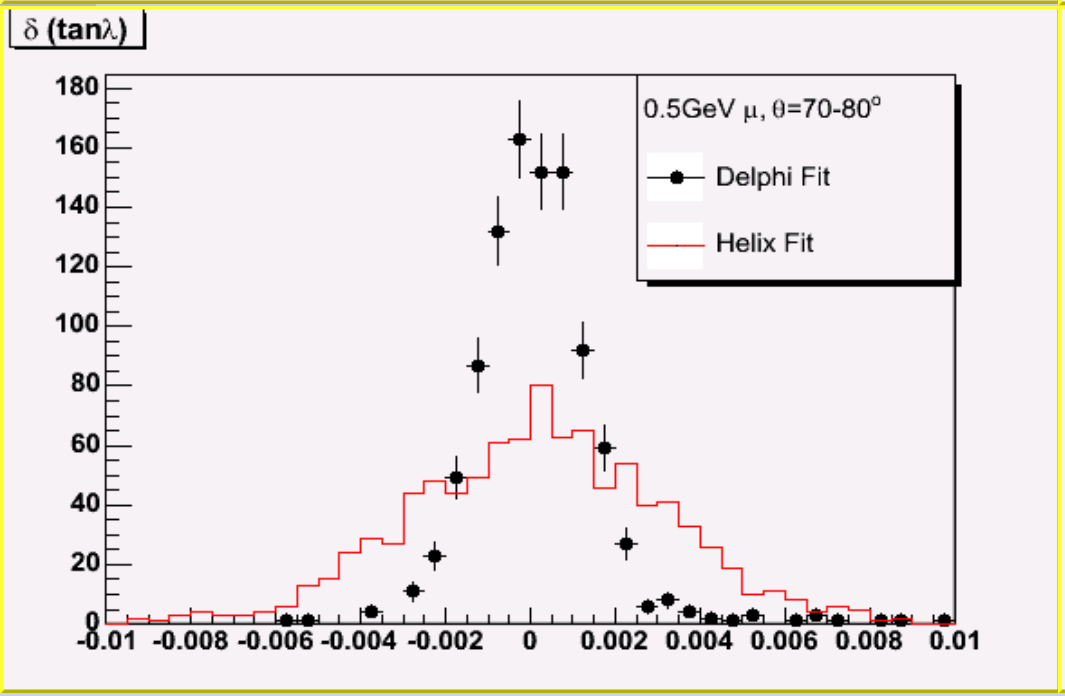
Material Database

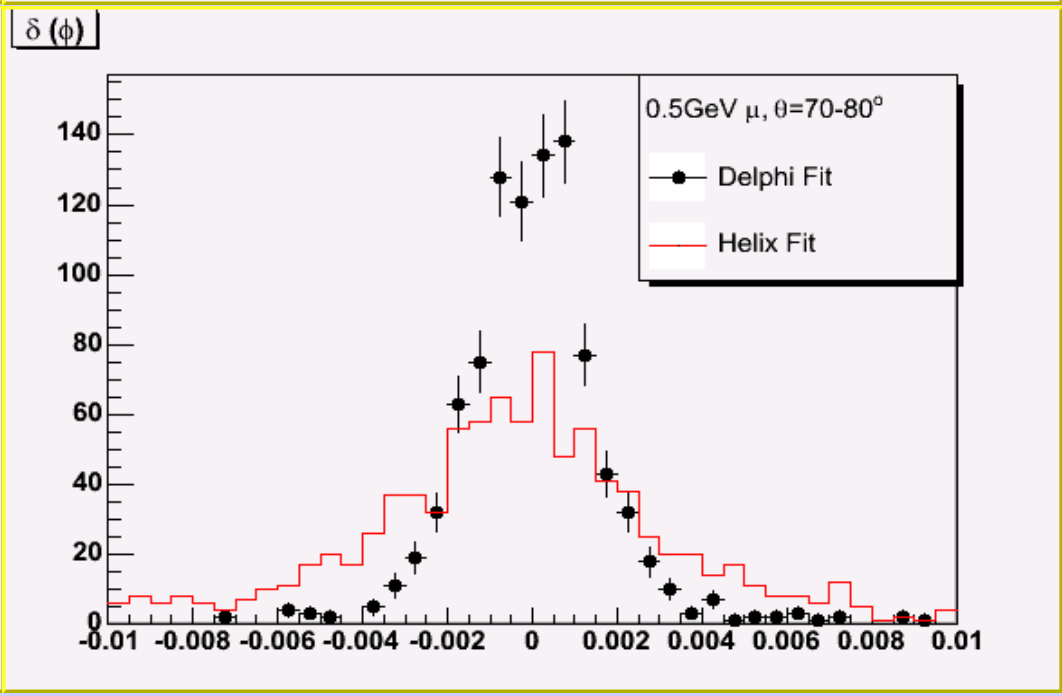
- Delphi fitting routine needs info on detector materials and geometry
- In original code these are represented in the form of infinitely thin geometrical shapes (cyllinders & discs orthogonal to beam axis)
- Original Delphi code is extended to handle also rectangular planar detector with arbitrary orientation and position in space and conical materials
- Information on materials is kept in FORTRAN common blocks
- Dedicated Marlin Processor has been developed to build material database / populate common blocks (Marlin Processor “MaterialDB”)
- Processor reads GEAR xml file with description of detector geometry & material properties and transfers this info in FORTAN common blocks

Track Fit Performance (VXD+SIT)

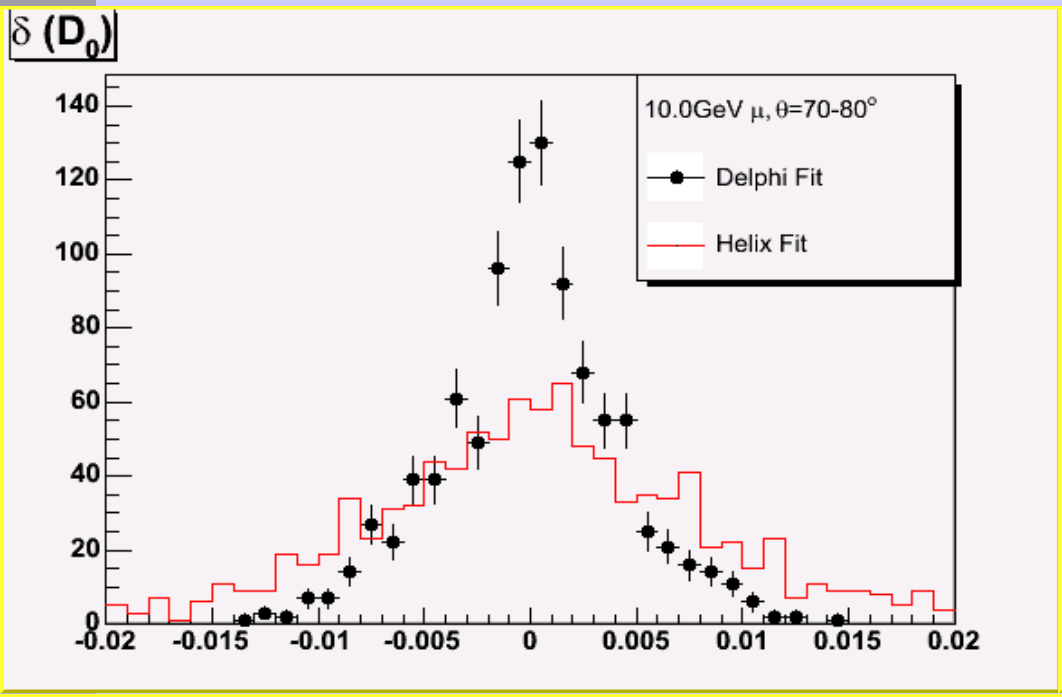


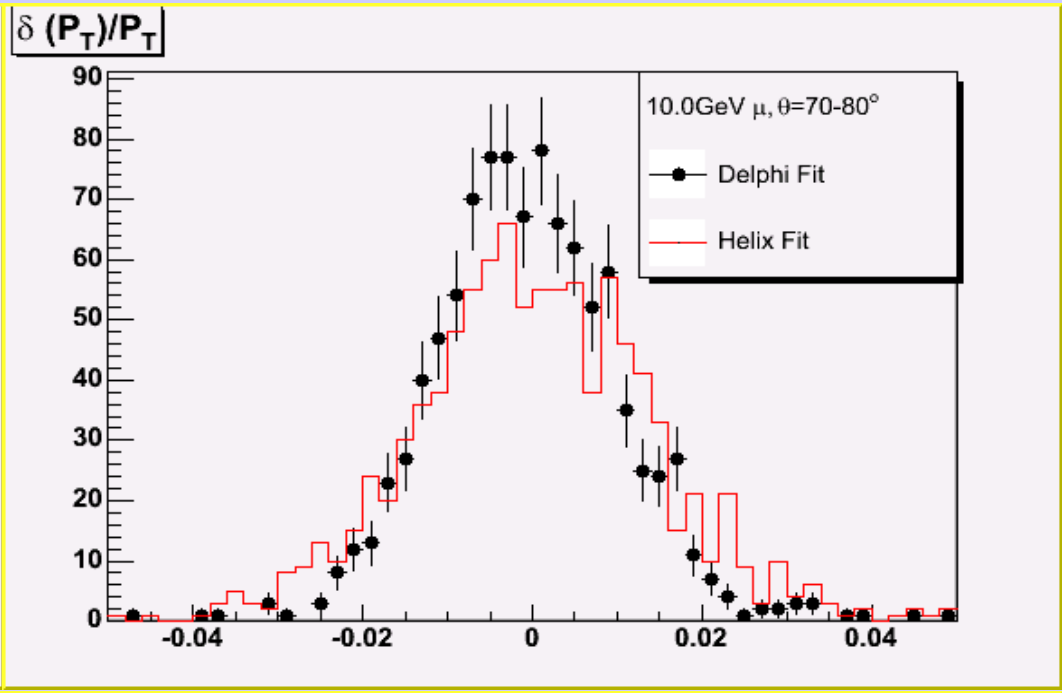


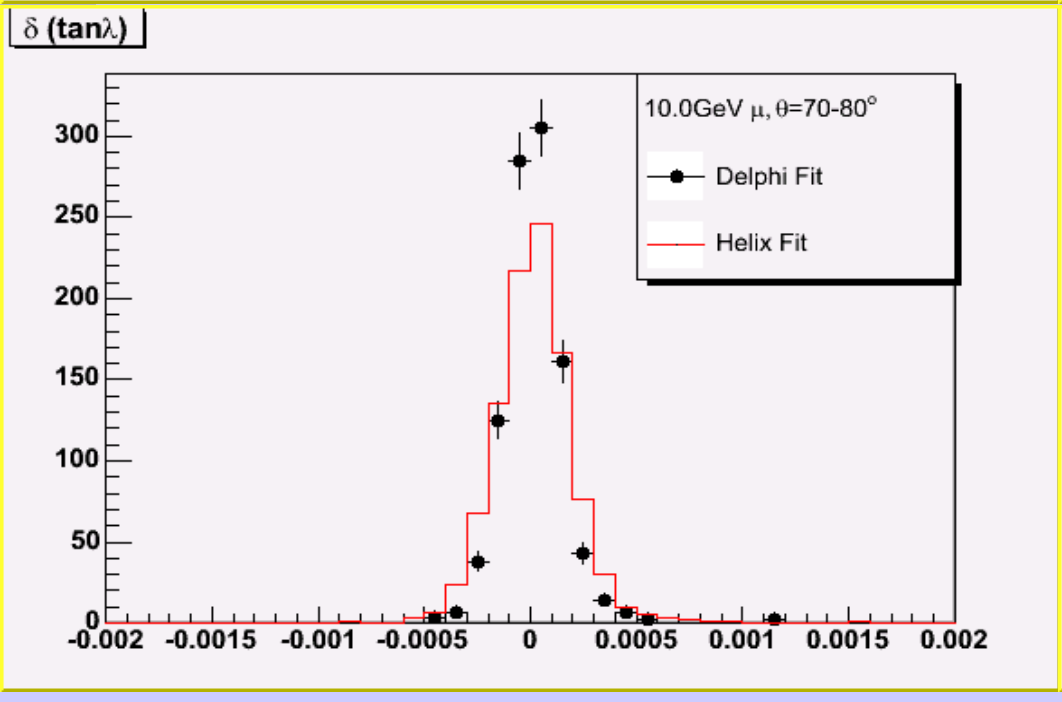


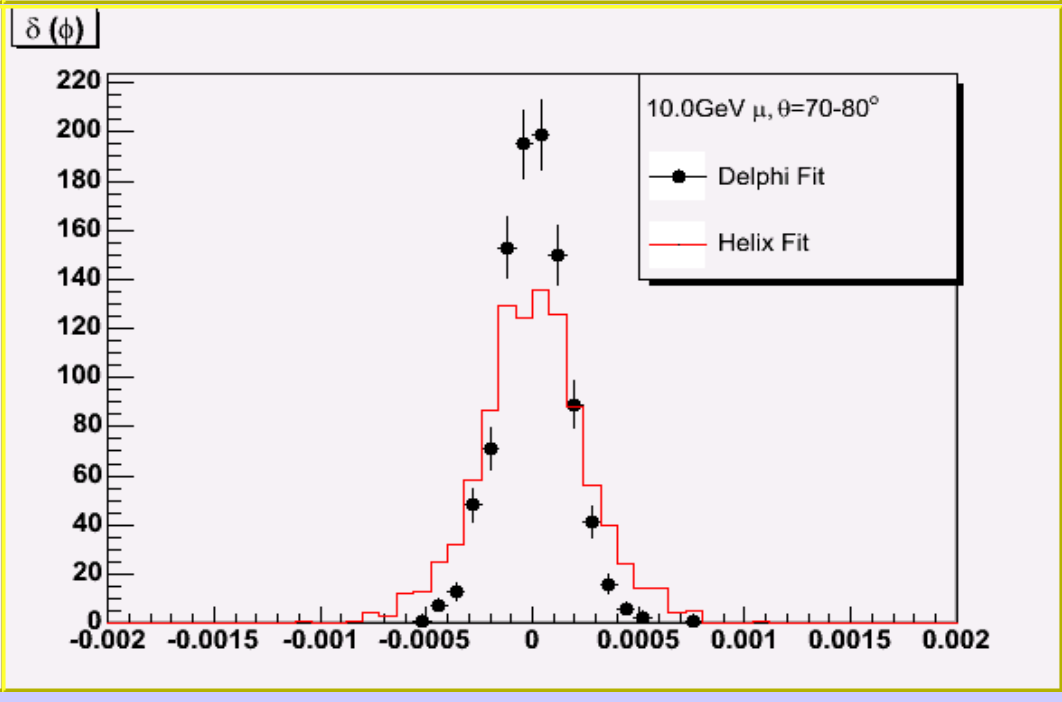


Track Fit Performance (VXD+SIT)

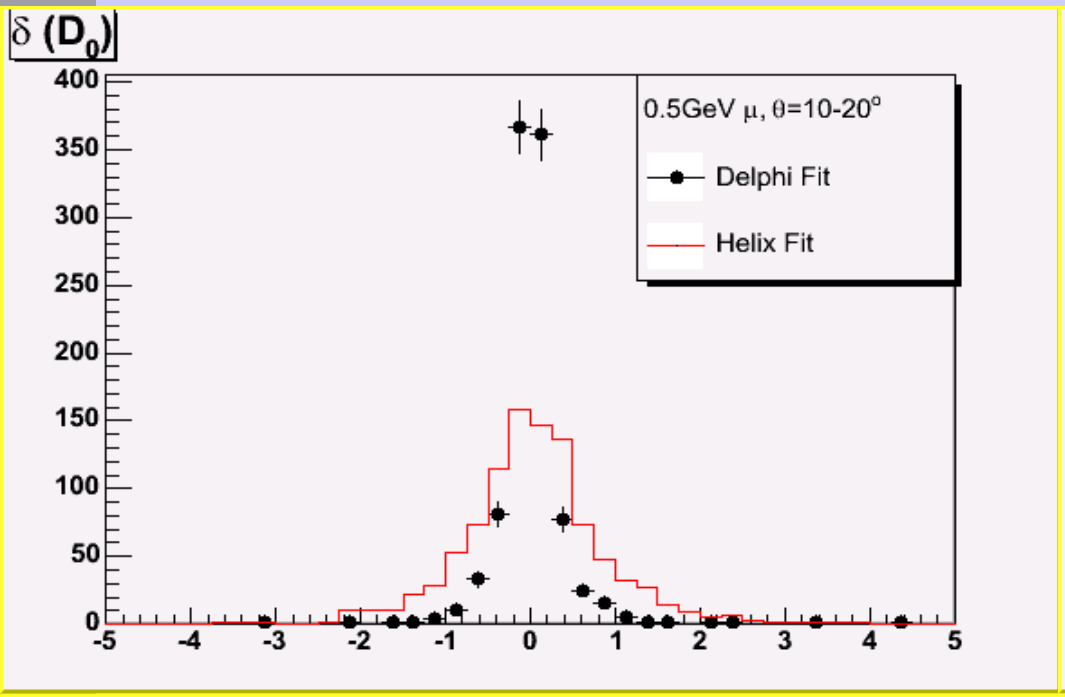


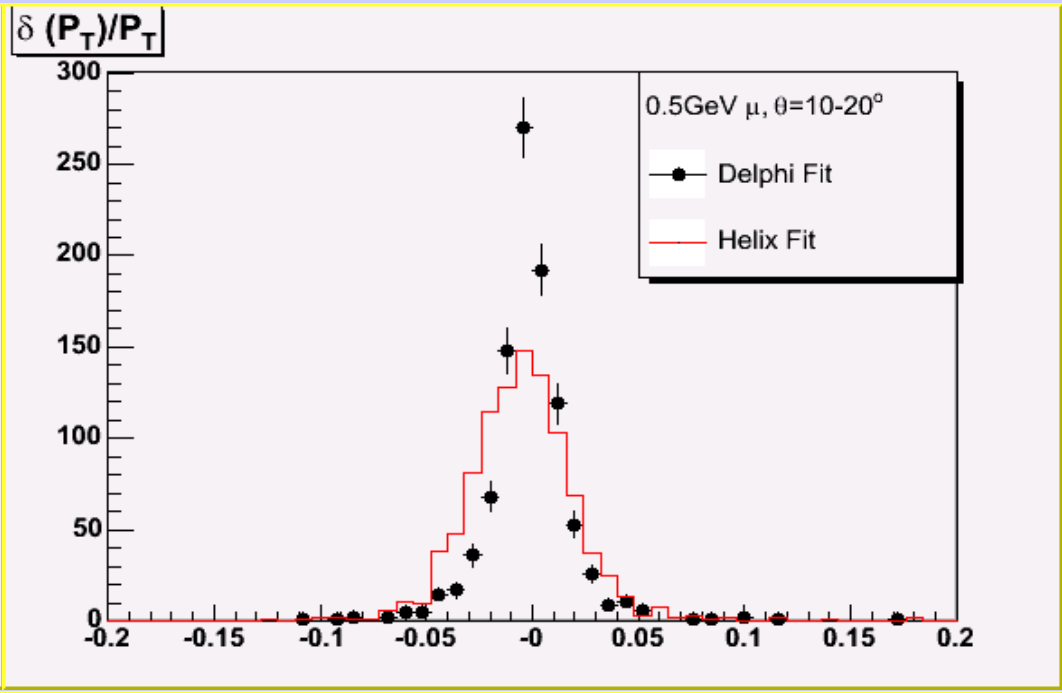


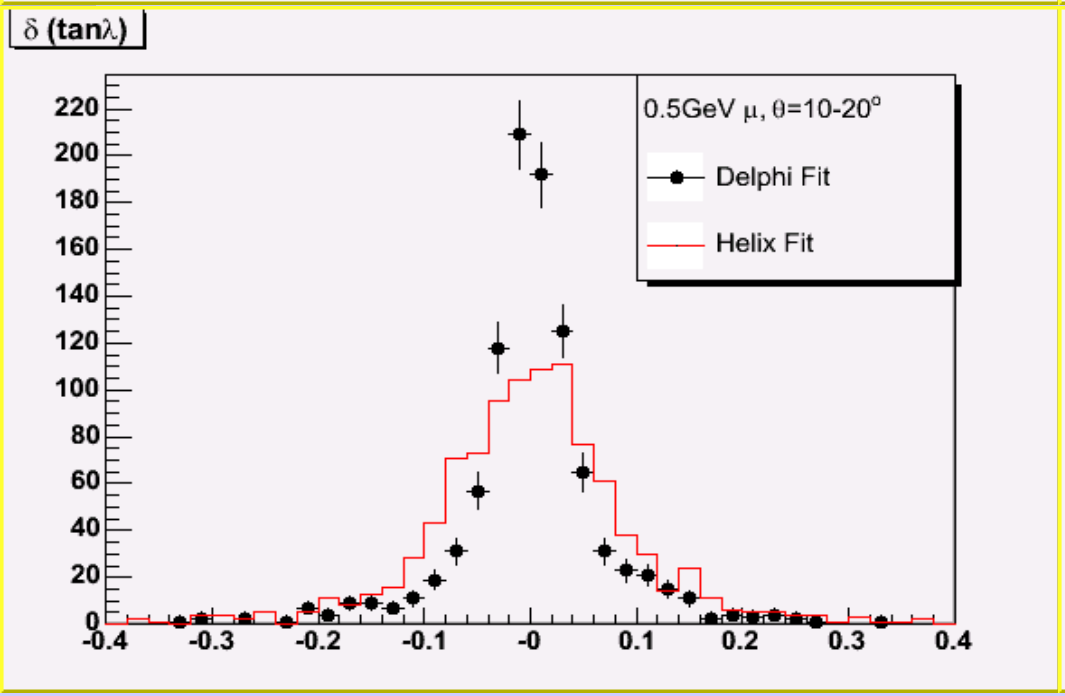


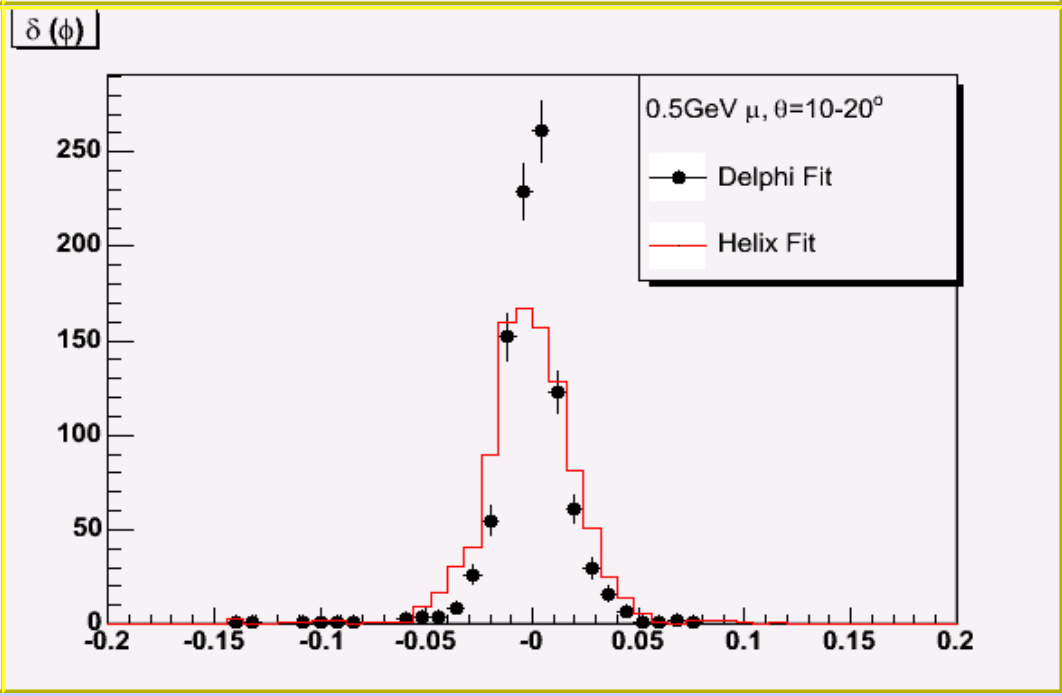


Track Fit Performance (VXD+FTD)

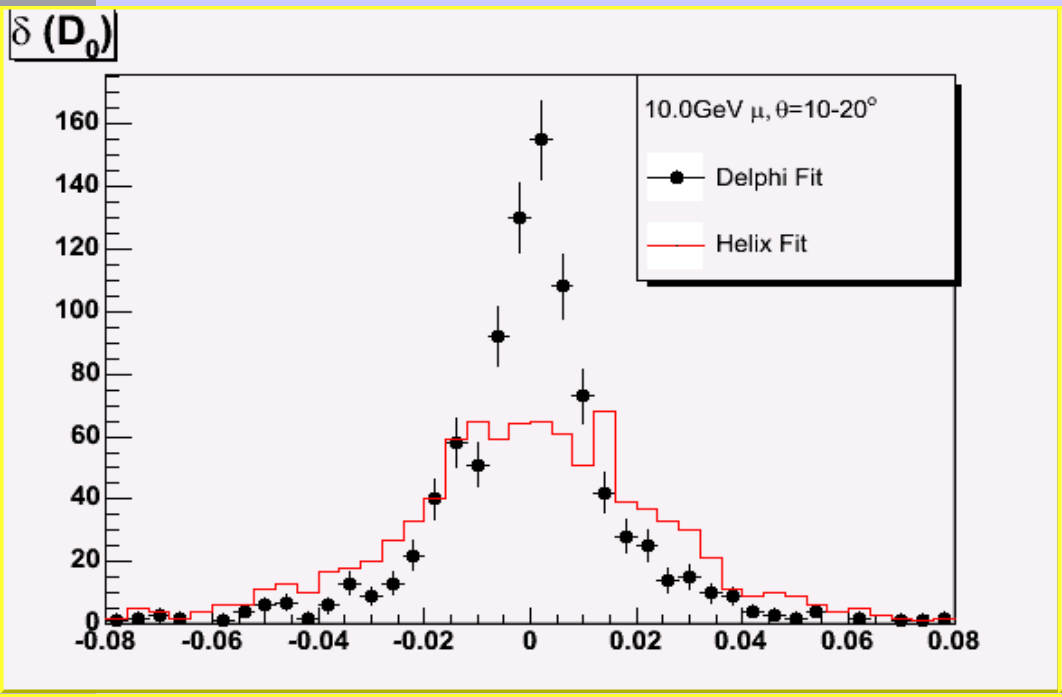


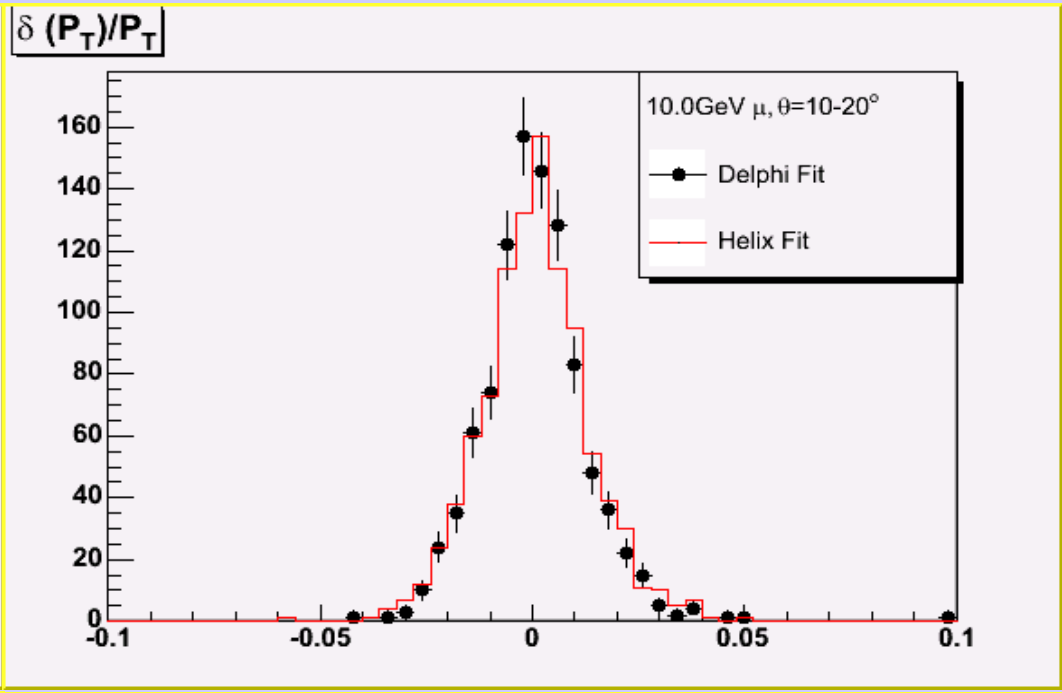


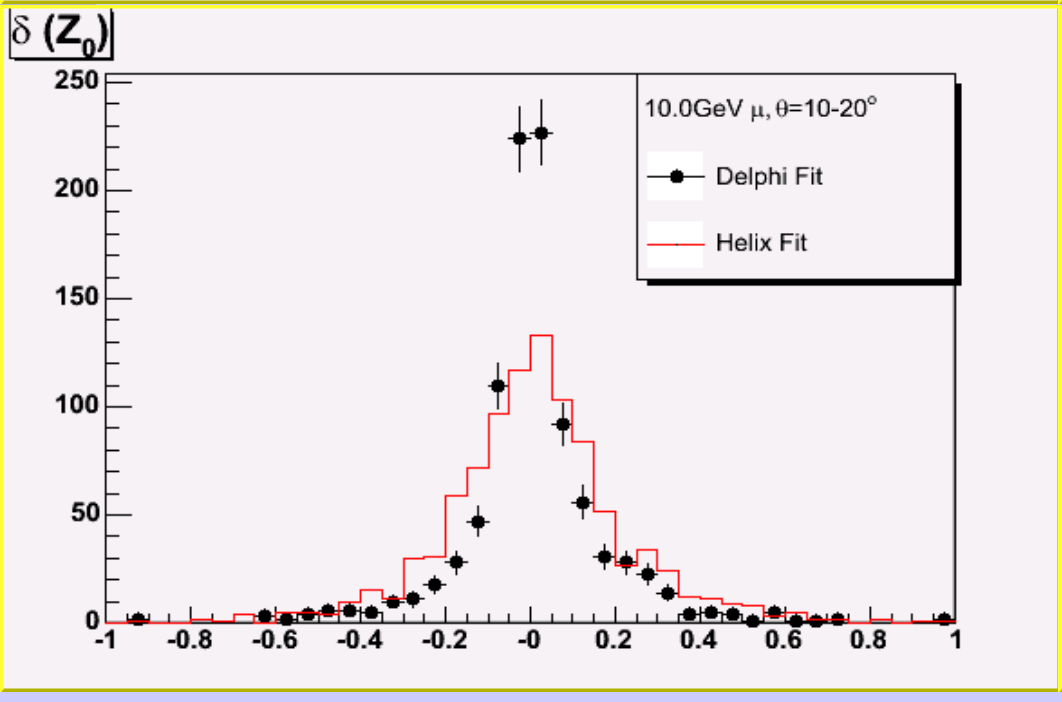


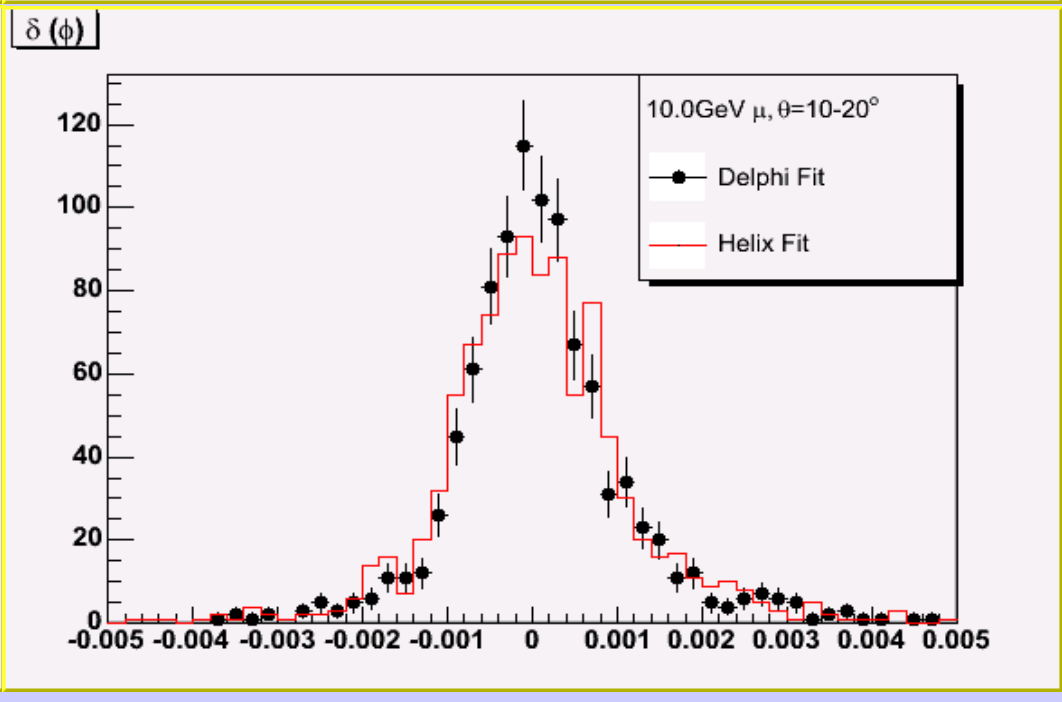


Track Fit Performance (VXD+FTD)









Outlook :Combining TPC & Si Tracking

- Two codes exist which do stand-alone tracking in TPC
  - ➔ BrahmsTracking (Patrec + track fitting with DELPHI code)
  - ➔ BoojumTracking (Patrec only, must be supplemented by track fitting) : **capable of reconstructing spiralling tracks!**
- Two approaches for combining TPC & Si Tracking
  - ➔ Do first stand-alone tracking in Si detectors & TPC separately, then combine matching track segments
  - ➔ Use TPC tracks to perform outward-wise seeding of tracks in Si detectors, reconstruct these tracks, then do stand-alone tracking in Si detectors on remaining hits
  - ➔ We plan to implement both approaches, a switch will be introduced to choose between two.

Plans

- Submission of SiTracking processor (stand-alone tracking in Si detectors) with all accompanying utilities to Zeuthen CVS before 1/08/2006

- Implementation of full tracking : September/October 2006