

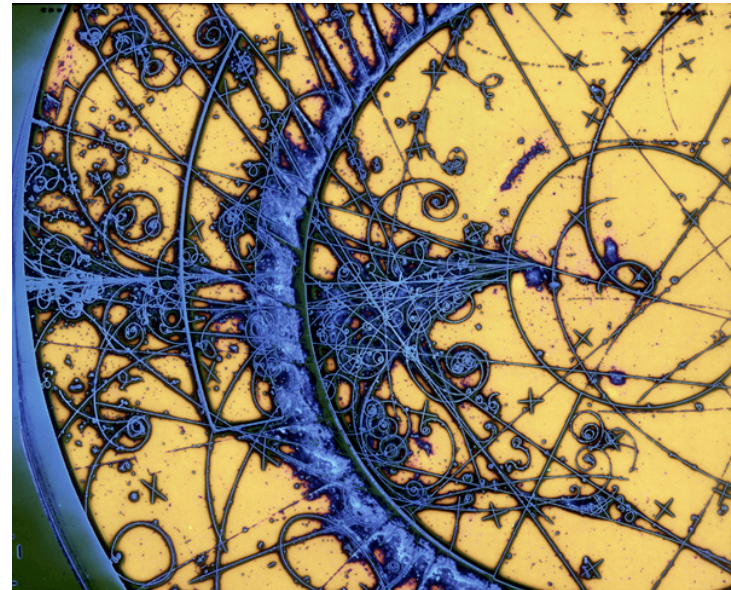
# Progress in ILD Tracking SW for the DBD

Steve Aplin  
DESY

ALCPG 2011 Univ. Orgeon  
20<sup>th</sup> March 2011



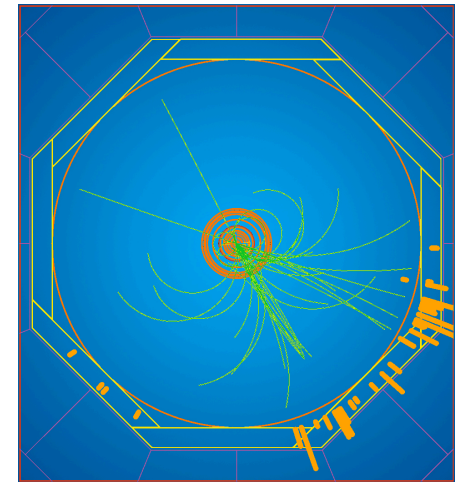
- Status
- Plans
- Summary



# Tracking @ ILC

- Tracking reconstruction goals:

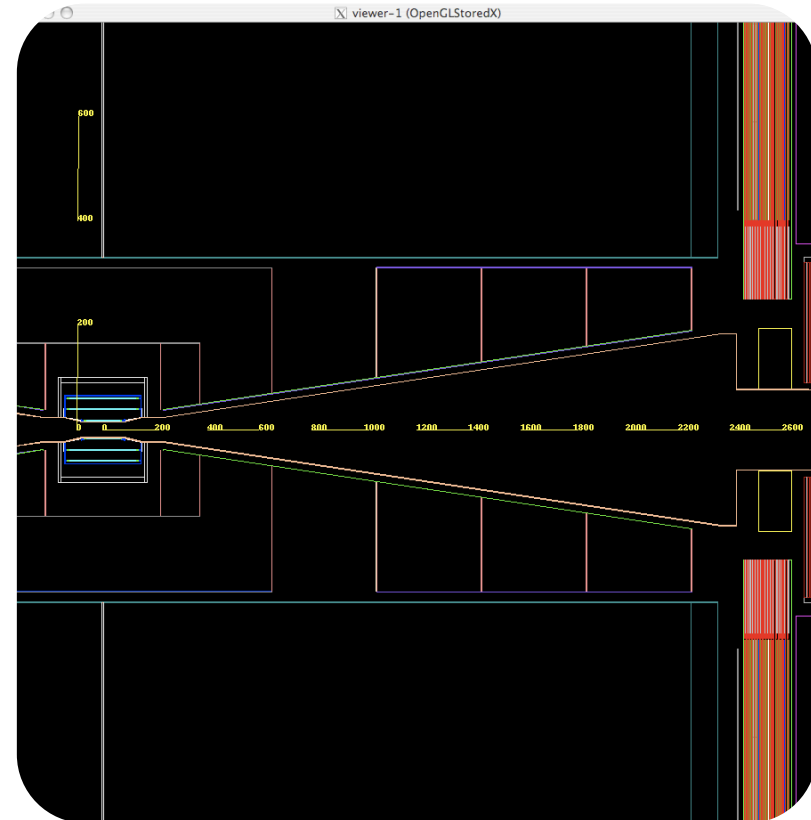
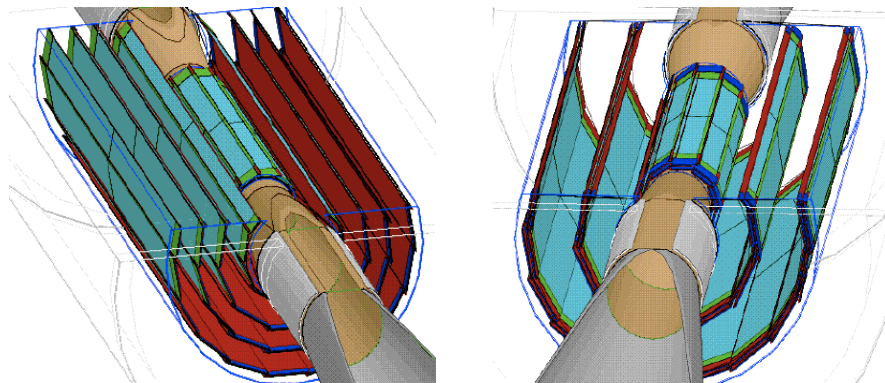
- Momentum resolution  $\frac{\Delta p}{p^2} < 5 \times 10^{-5} \text{ GeV}^{-1}$
- Impact parameter resolution  $\sigma_{r\phi} < 5 \mu\text{m} \oplus \frac{10}{p(\text{GeV}) \sin^{3/2} \theta} \mu\text{m}$
- Very high efficiency  $\epsilon > 99\%$
- Very low material budget



# ILD Track Simulation

**DBD** calls for more realistic detector descriptions

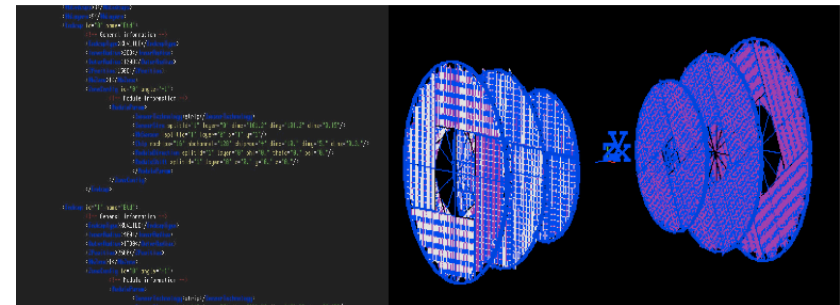
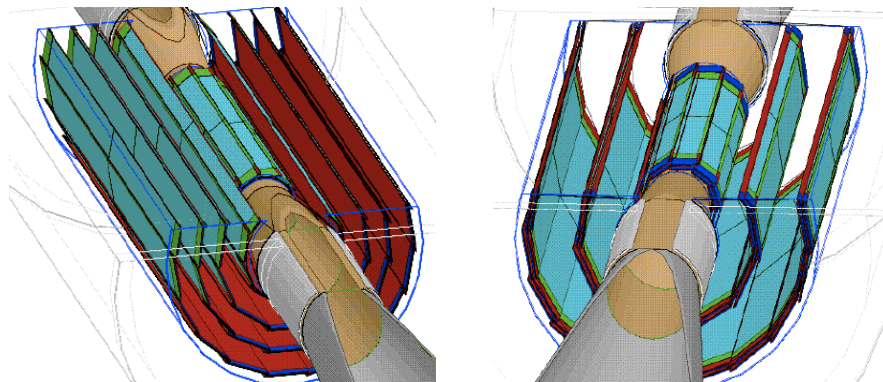
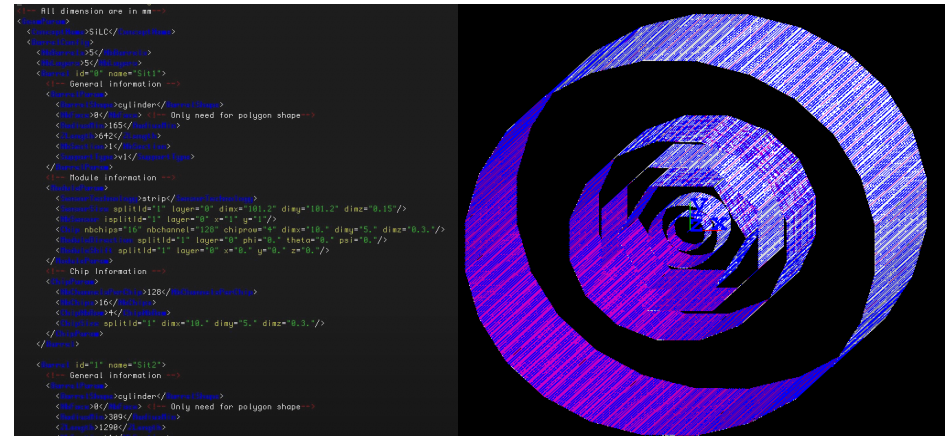
**LoI** Simulation uses a mixture of realistic and simplified detector descriptions



# ILD Track Simulation

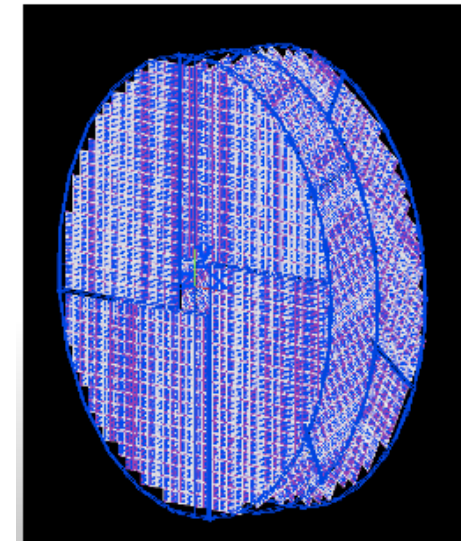
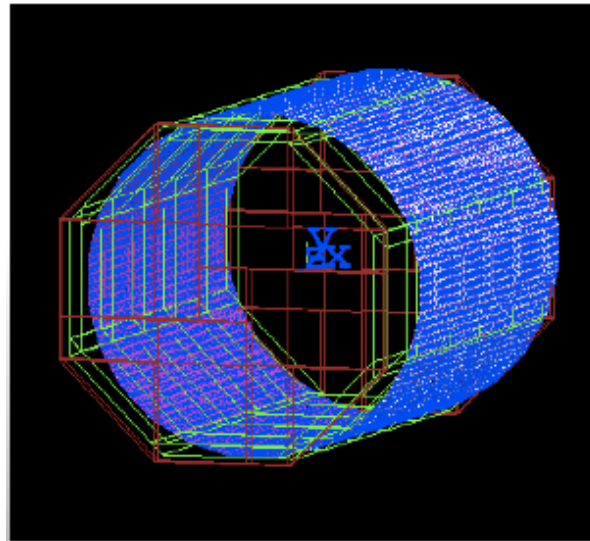
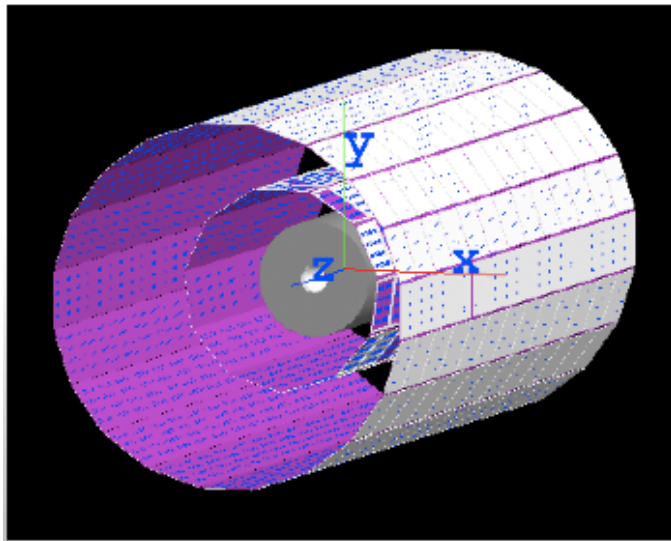
DBD calls for more realistic detector descriptions

Silicon Trackers have now been revised to bring them up to the same level of realism



# ILD Track Simulation

## SIT, SET & ETD

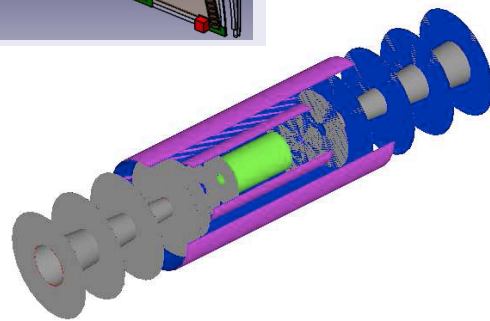
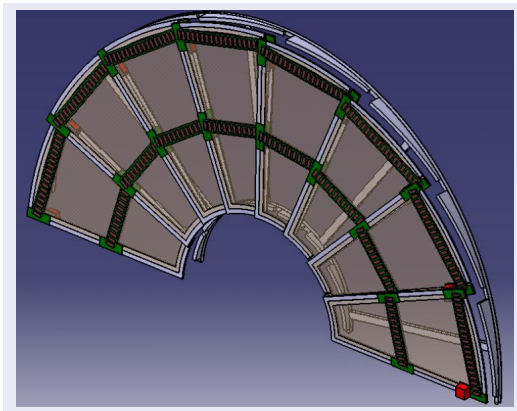


Alexandre Charpy & Konstantin Androsov(LPNHE)

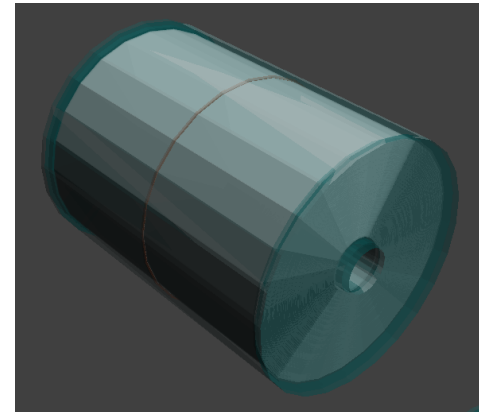
New Mokka Drivers in ILD\_01

# ILD Track Simulation

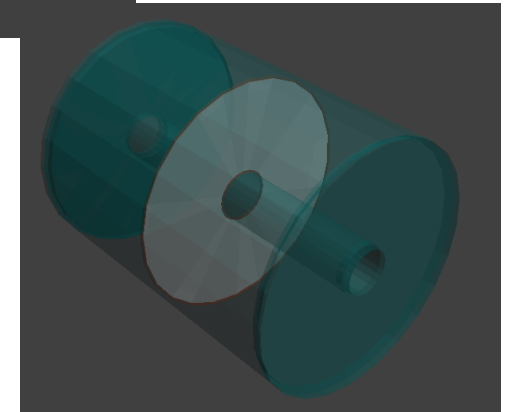
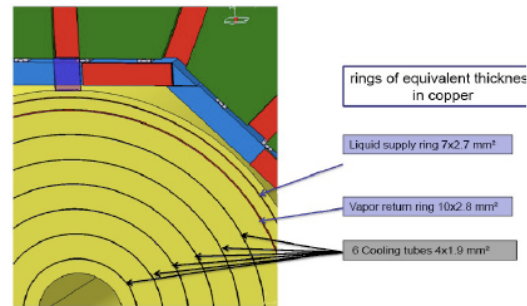
## FTD and TPC



Jordi Durate Campderros  
ICFA



S. Aplin &  
Gabriel Musat



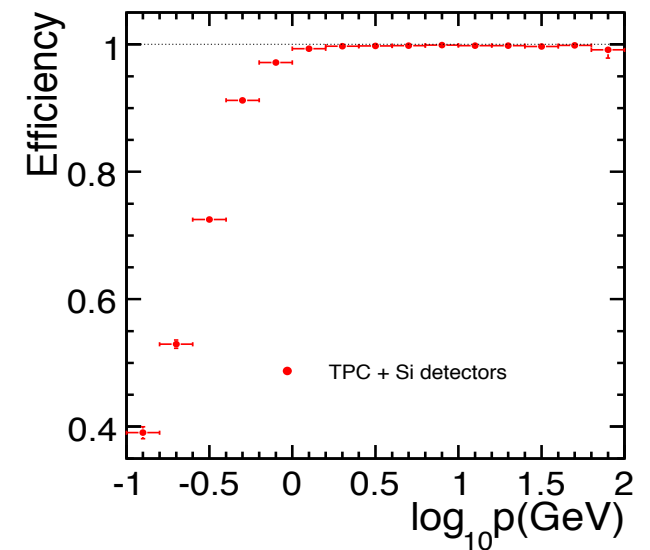
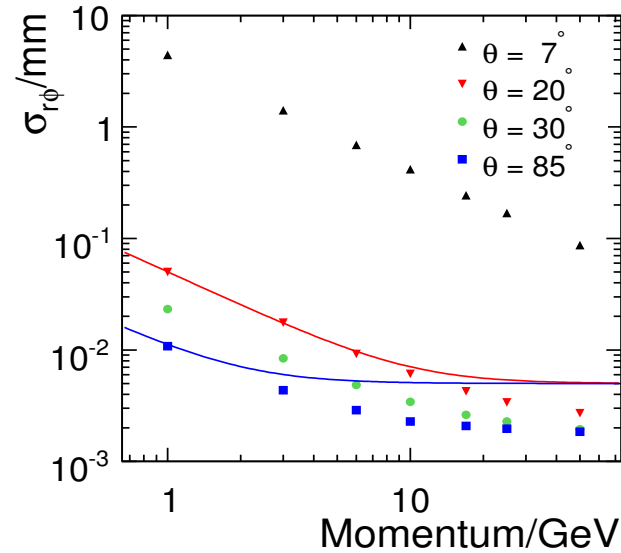
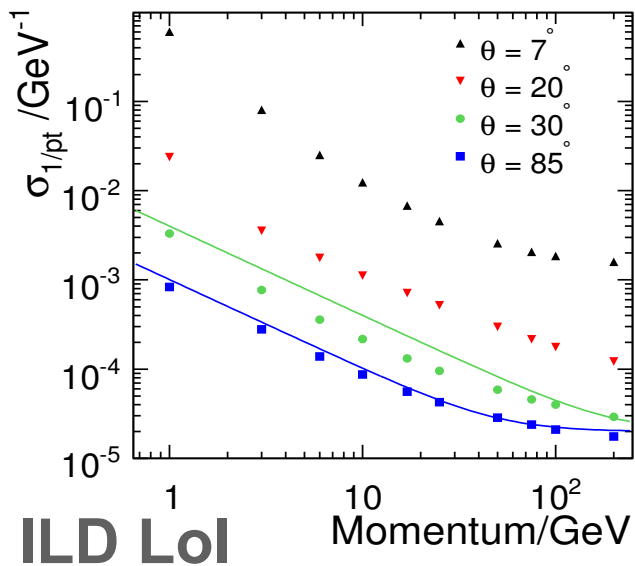
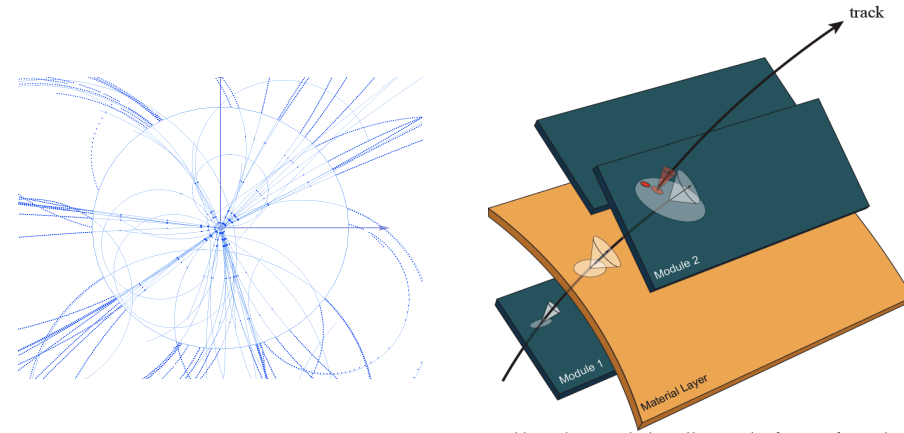
## New Mokka Drivers in ILD\_01

# ILD Track Reconstruction

## Full pattern recognition

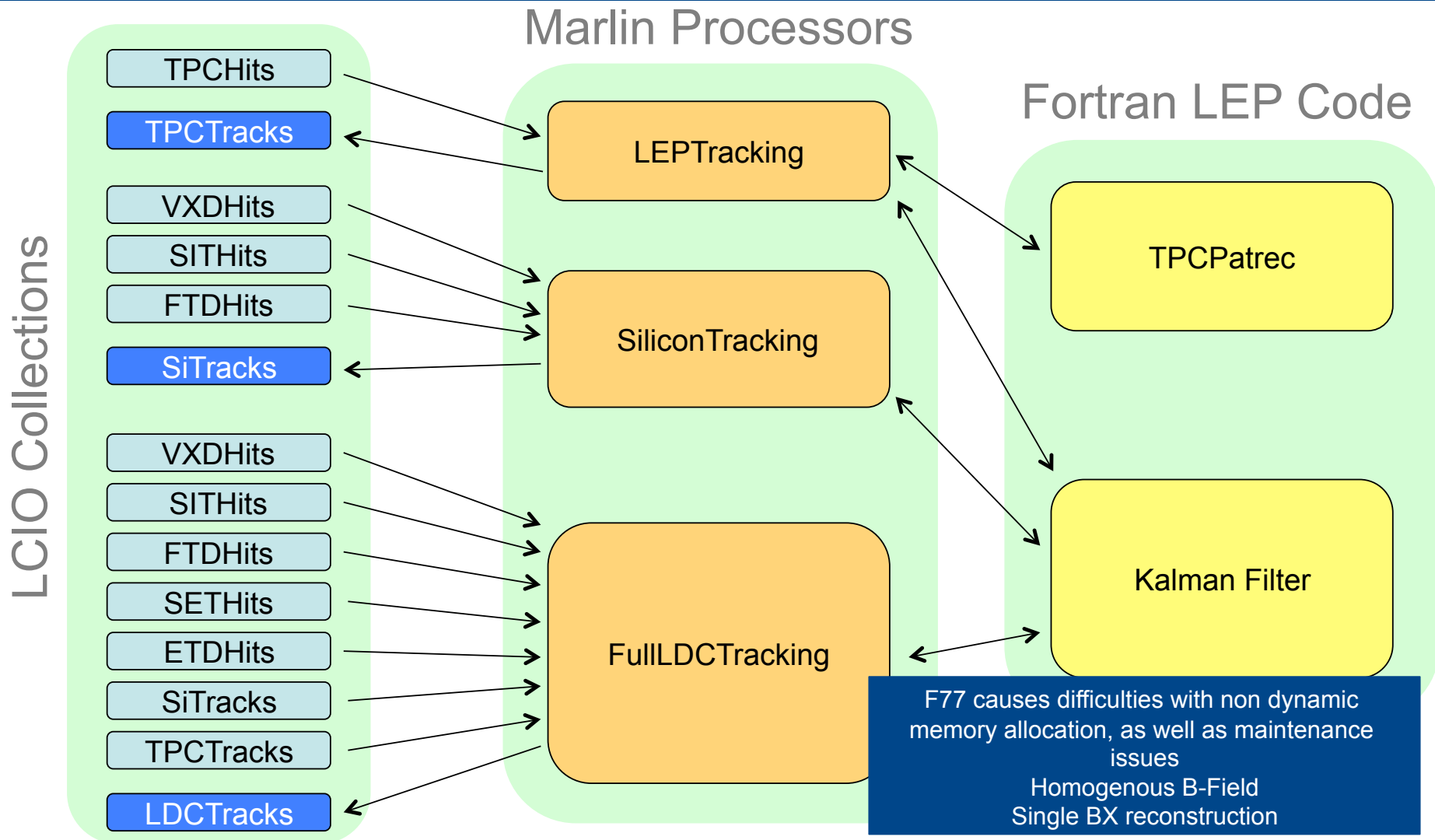
Stand-alone track finding in both Inner Silicon Trackers and TPC

## Kalman Filter Track Fitting





# Tracking Software used in LOI



# Tracking Software used in LOI

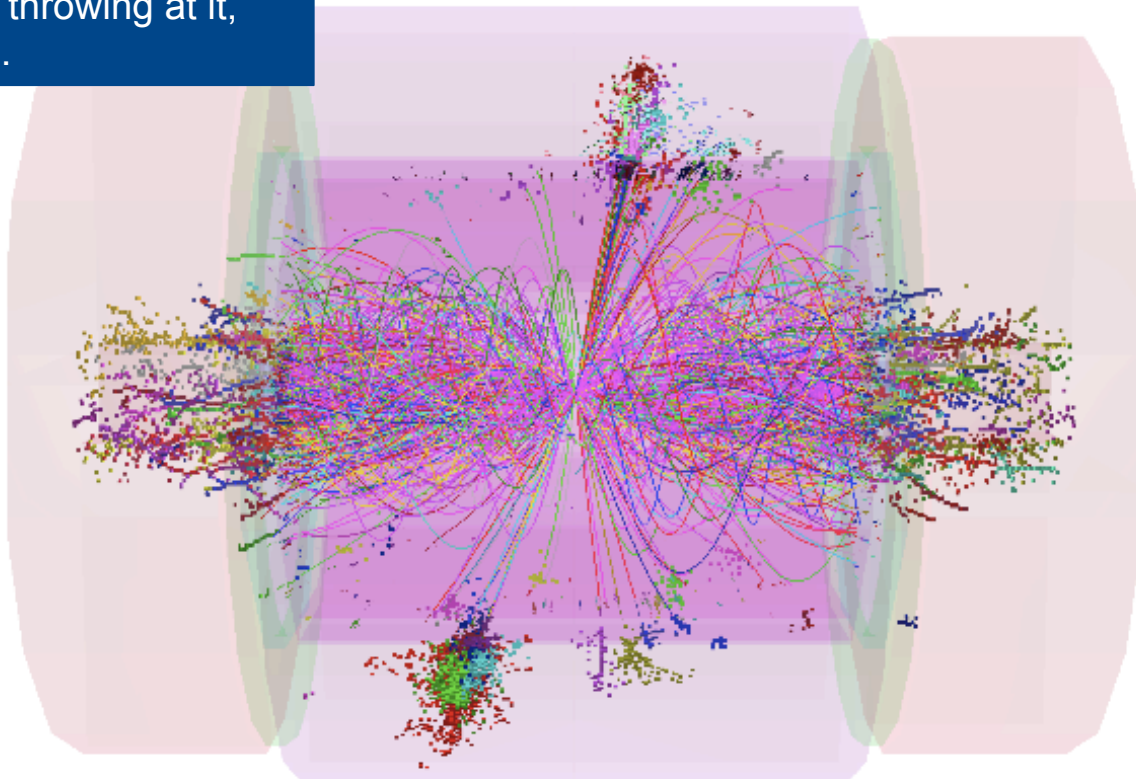
Nevertheless still surviving pretty much everything CLIC is throwing at it, just ...

## Marlin Processors

## LEP Code

LCIO Collections

- VXDHit
- SITHit
- FTDHit
- SiTrack**
- VXDHit
- SITHit
- FTDHit
- SETHit
- ETDHit
- SiTrack
- TPCTrack
- LDCTracks**



**1.4 TeV of background !**

PCPatrec

Iman Filter

with non dynamic  
cell as maintenance  
s  
Homogeneous B-Field  
Single BX reconstruction

# Tracking Code rewrite for the DBD

- Leave behind F77 LEP tracking code.
- Rewrite the TPC pattern recognition.
- Use KalTest Kalman Filter fitting library.
- Stand alone track reconstruction for the inner silicon trackers and forward region needs to be addressed so that it can cope with beam related background.

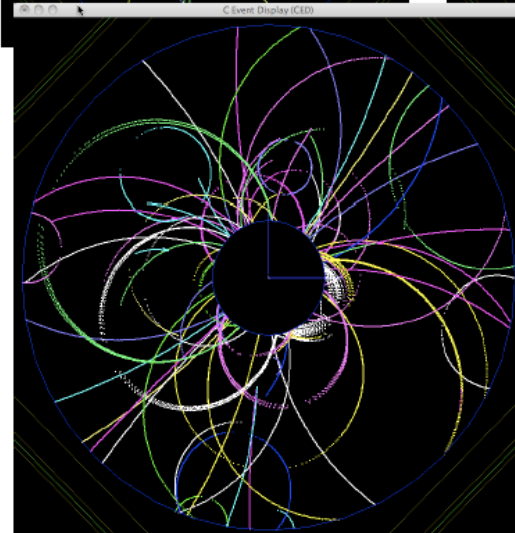
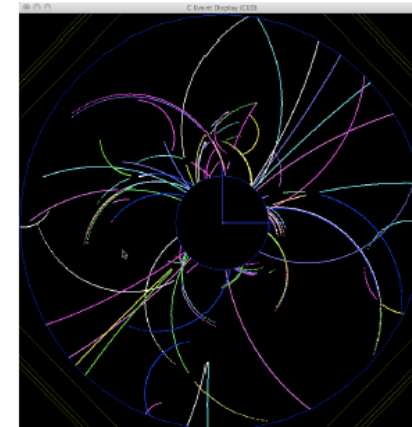
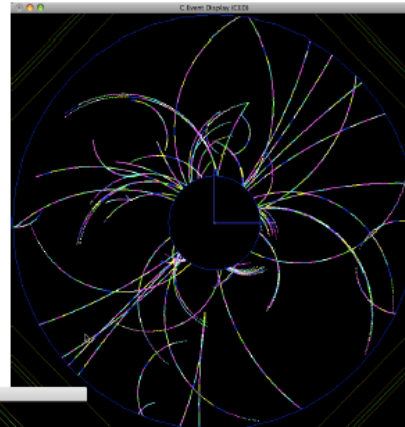
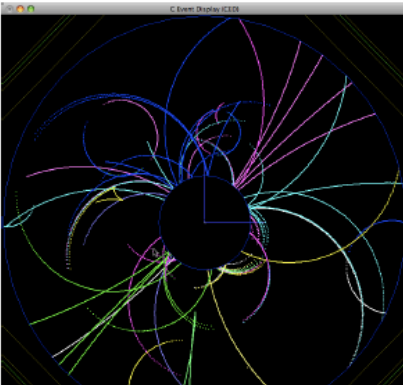
# TPC Pattern Recognition

```
template <class In, class Out, class Pred >
void cluster( In first, In last, Out result, Pred* pred ) {

    typedef typename In::value_type GenericHitPtr;
    typedef typename Pred::hit_type HitType;

    typedef std::vector< GenericCluster<HitType > * > Clu
    ClusterList tmp;
    tmp.reserve( 256 );

    while( first != last ) {
        for( In other = first+1; other != last; other
            if( pred->mergeHits( (*first), (*other) ) )
                if( (*first)->second == 0 && (*other)->se
                    GenericCluster<HitType >* cl = new Ge
                    cl->addHit( (*other) );
                    tmp.push_back( cl );
                }
            else if( (*first)->second != 0 && (*other
                (*first)->second->mergeClusters( (*ot
            } else { // one cluster exists
                if( (*first)->second != 0 ) {
                    (*first)->second->addHit( (*other
                } else {
                    (*other)->second->addHit( (*first
                }
            } // dOut
        } ++first;
    } // remove empty clusters
}
```



- use NN-Clustering w/ euclidian distance
- cleanup merging regions with pad-row ranges
- assign leftover hits:
  - based on residual wrt. extrapolation x-ing
  - improved assignment since ILCWS 2010
- working on segment merging
  - based on simple 'circle criterion'
- implemented chi2 for track state comparison

Frank Gaede

# KalTest

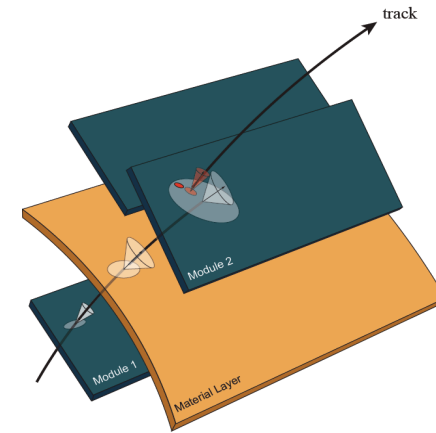
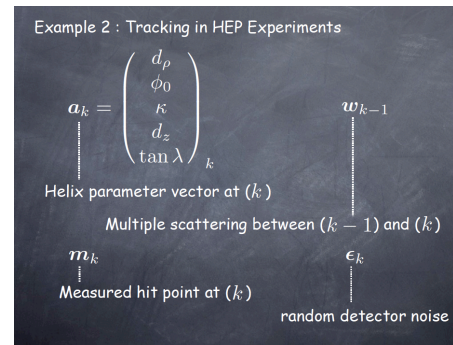
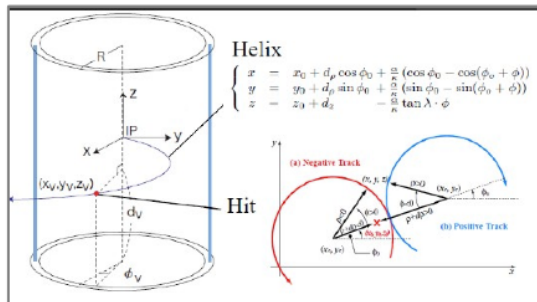
## Kalman Filter fitting library (Keisuke Fuji et al)

Based on Root

Structured in sub-libraries

- geomlib -- geometry
- kallib -- Kalman filter
- kaltracklib -- Kalman tracker
- utils -- utilities

Built into one libKalTest.so



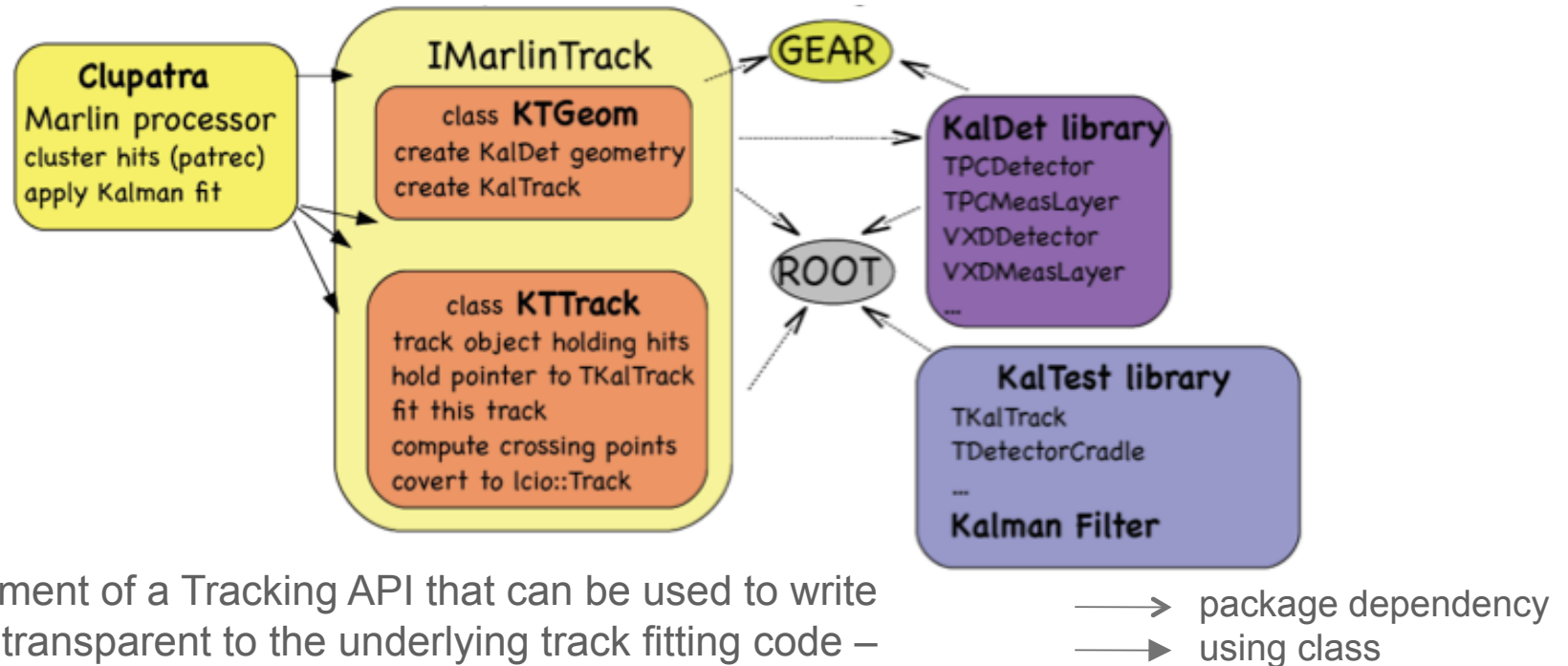
User needs to define their detector classes (KalDet)

- TVMeasLayer: meas. layer, coord. to track state transformation
- TVDetector: position of measurement layers and material properties

Included in iLCSoft as of v01-10

# Using KalTest within iLCSoft

- need to interface to KalTest Fitter
- would like to have loose coupling between patrec and fitting
- need several iterations between patrec and fitting
- LCIO::Track class not optimal for that (not designed to be)



Started development of a Tracking API that can be used to write patrec in Marlin transparent to the underlying track fitting code – partially implemented with Kaltest as well as F77 LEP Code

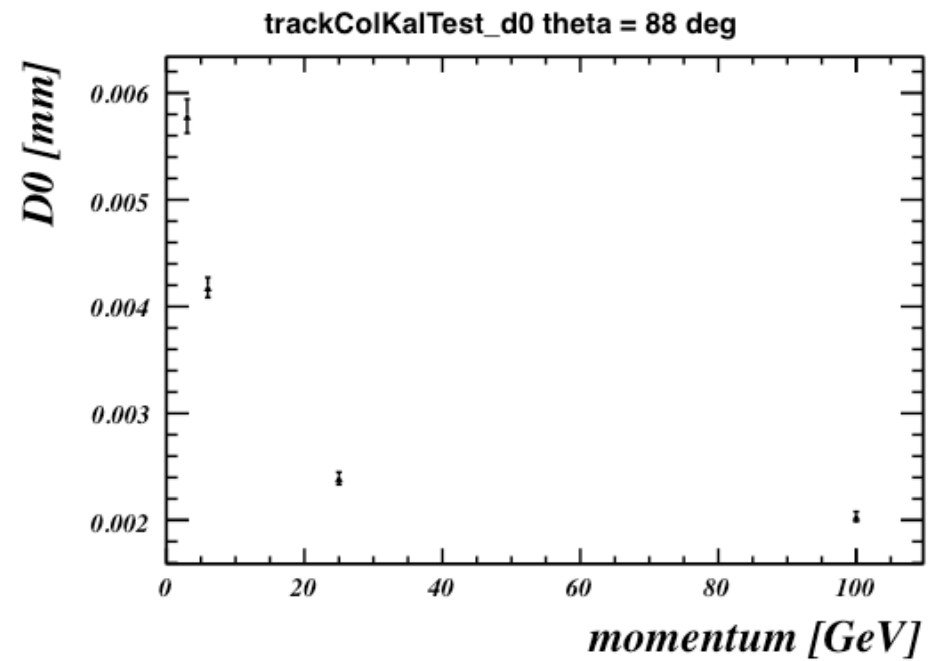
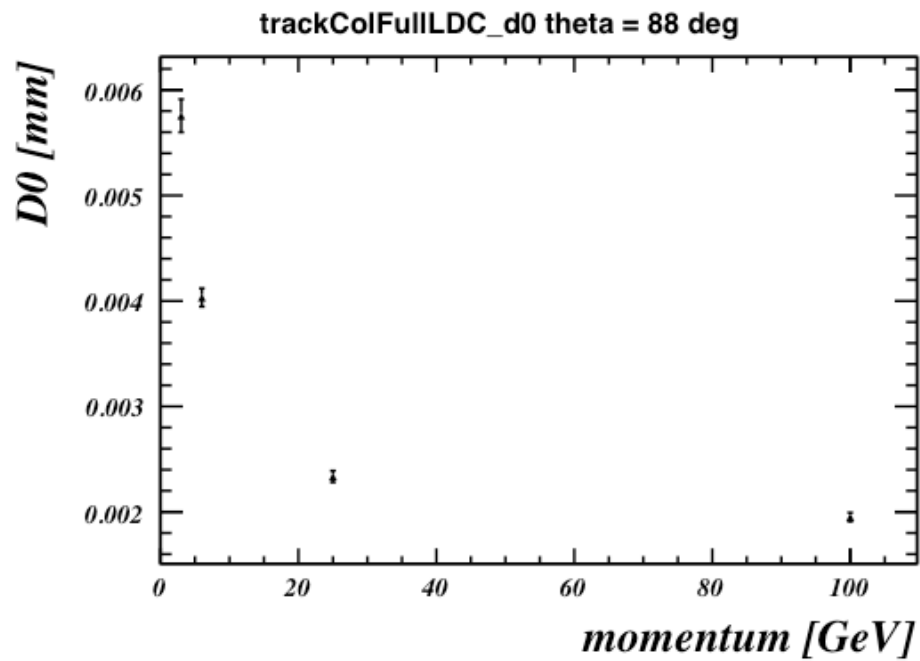
# Track Parameter Comparison

- Created a Refitting Processor to test development of the Tracking API as well as Track Parameter and error determination in the KalTest implementation
  - Takes Icio Tracks produced by LEPTracking and FullLDCTracking and refits the associated hits using the Kaltest Kalman Filter.
  - Presently fits are compared only at the IP
- Testing performed using a mock up of inner detectors in Mokka.
- Comparison made with Track Parameters and errors determined by F77 LEP fitting code using single muons at  $p = 3, 6, 40, 100$  GeV and  $\theta = 88, 40, 32$  degrees

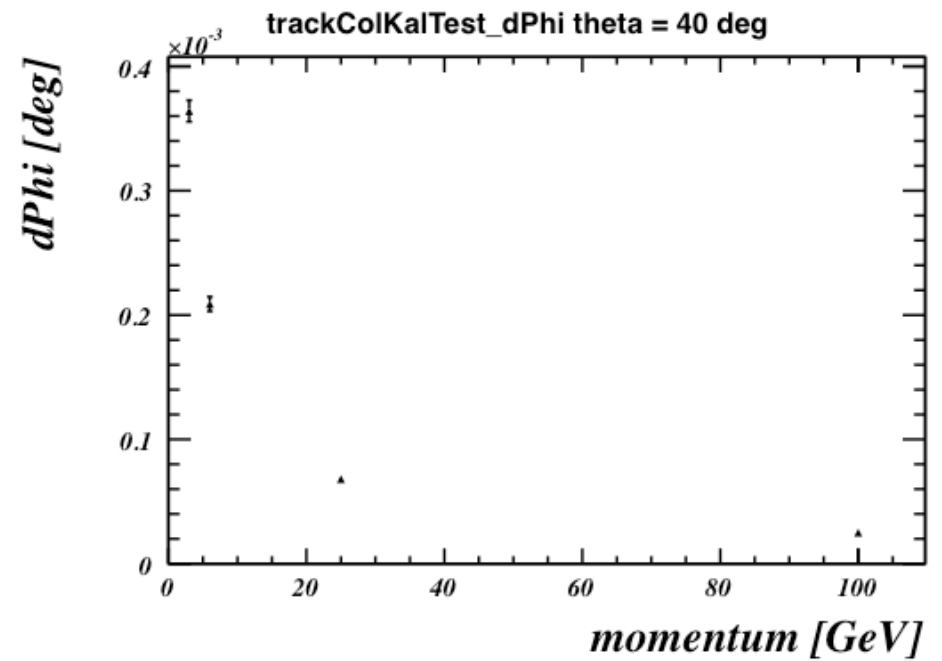
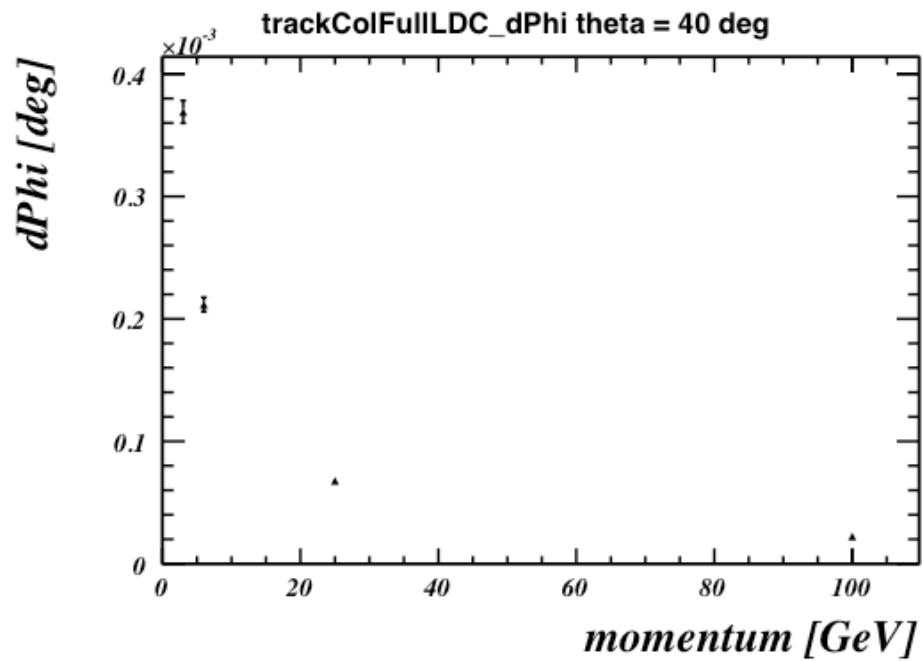




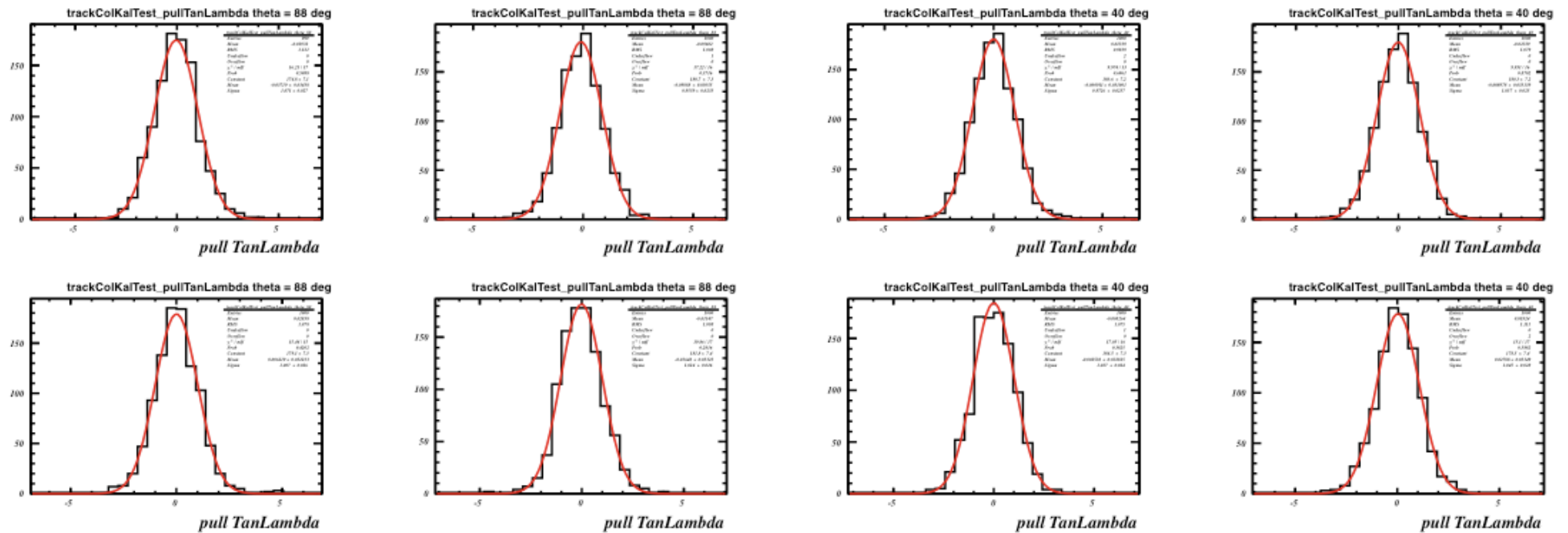
# Track Parameter Comparison



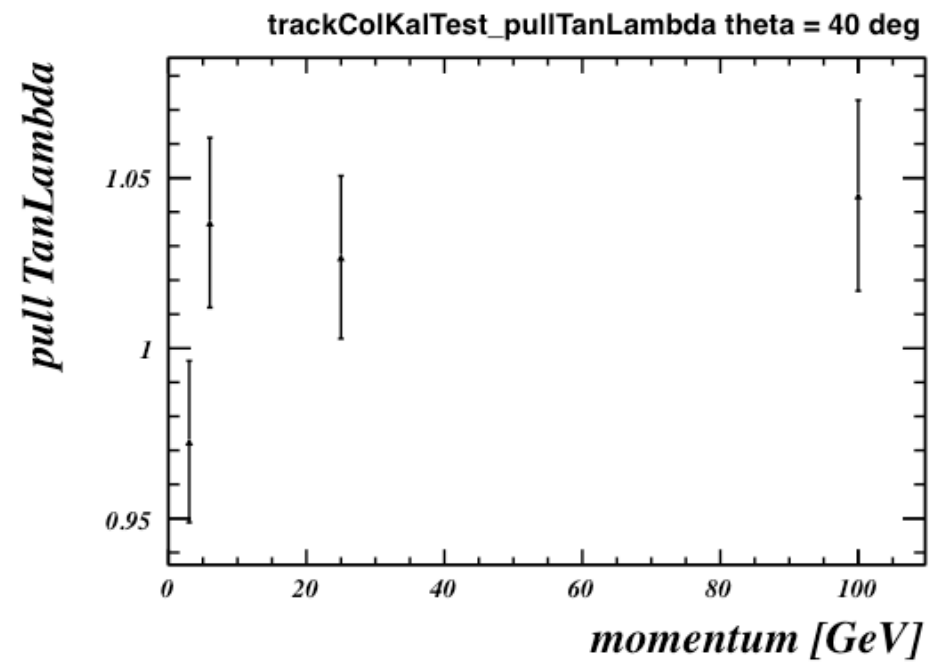
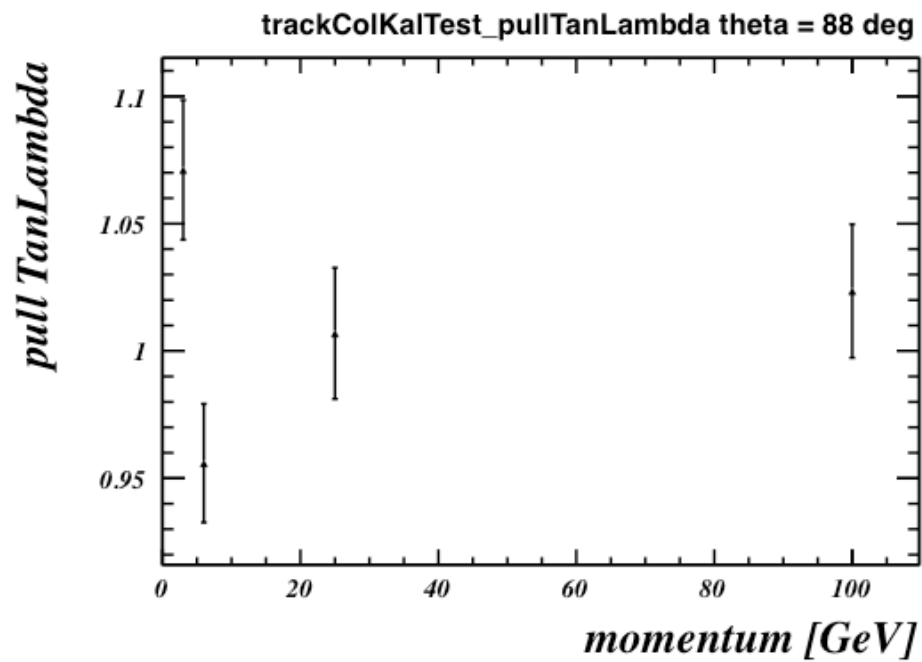
# Track Parameter Comparison



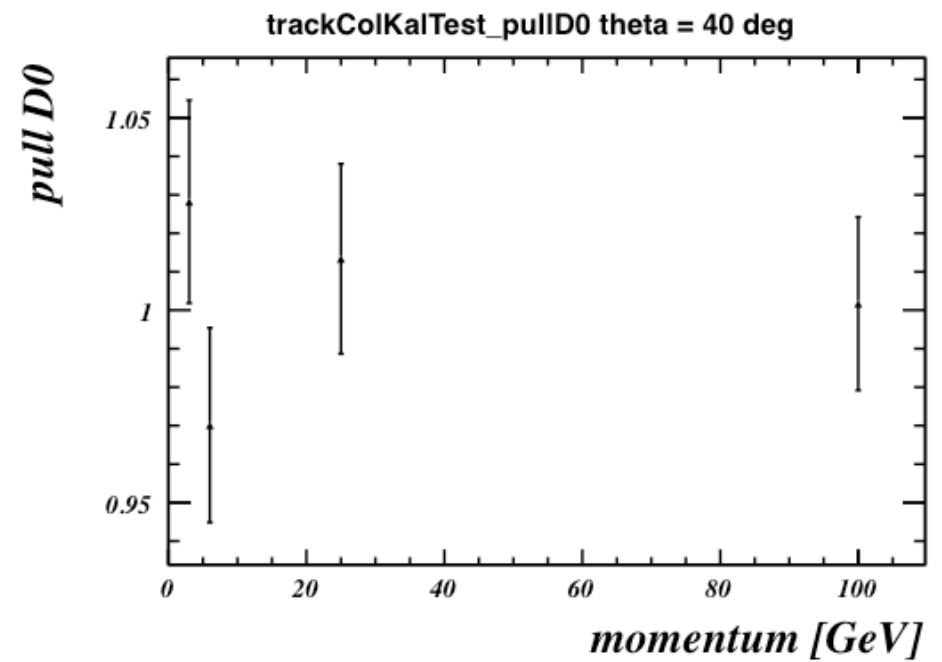
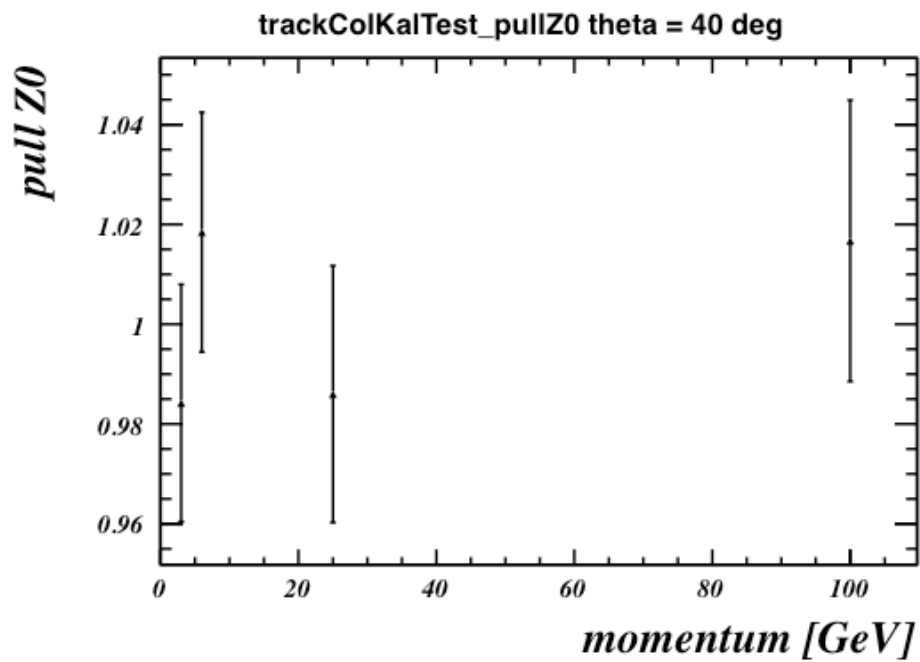
# KalTest Track Parameter Pull Distributions



# KalTest Track Parameter Pull Distributions

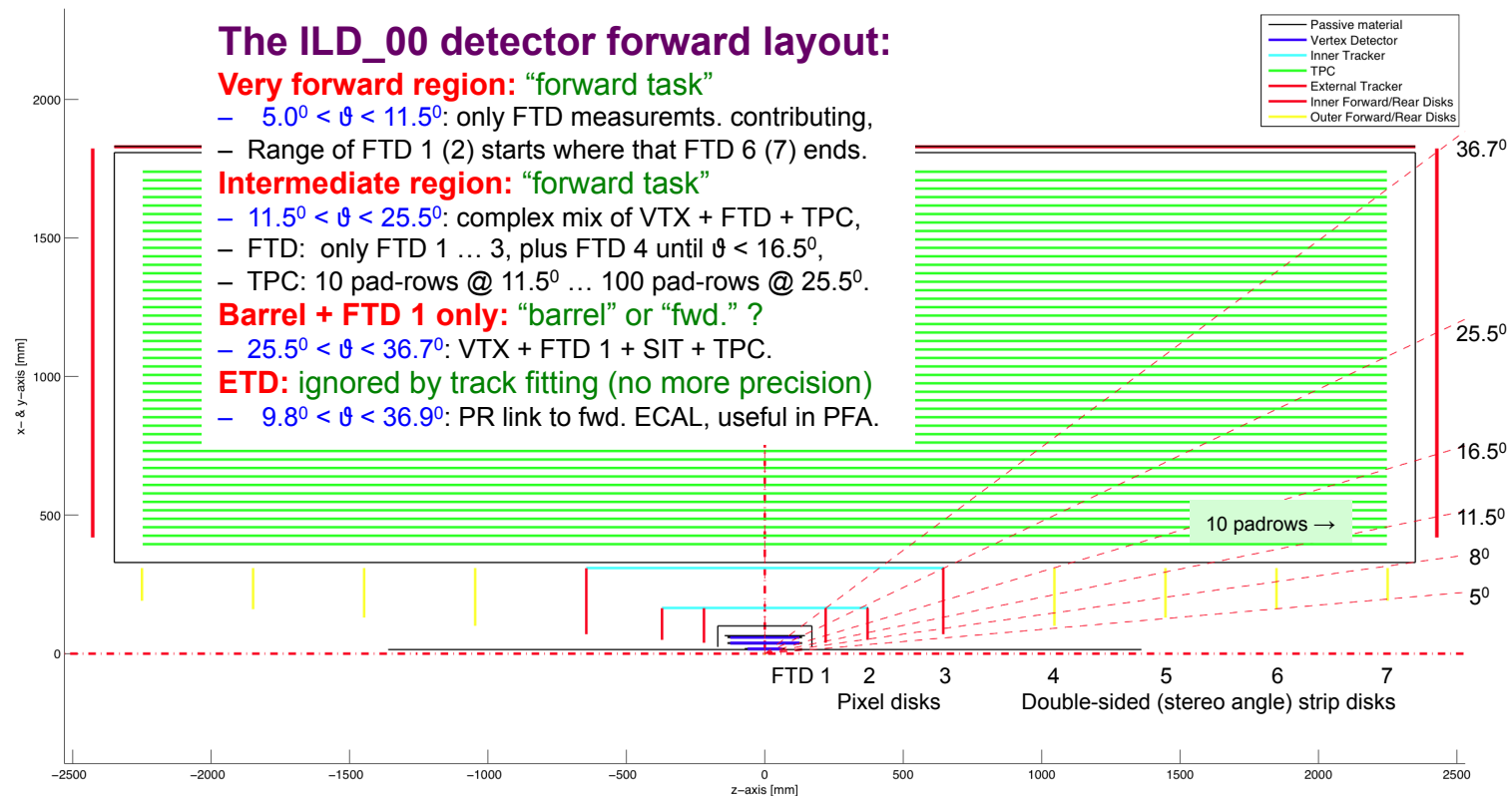


# KalTest Track Parameter Pull Distributions



# Track Reconstruction in the Fwd Region

## Example of a forward region



Winfried Mitaroff HEPHY

# Track Reconstruction in the Fwd Region

## Fwd. track search strategies

- **Stand-alone in FTD:**
  - This is the only possible strategy in the very forward region,
  - Various algorithms exist – which to choose needs careful study,
  - For small  $\theta$ , hits from **beamstrahlung-induced background** may cause further problems (we need a reliable estimate),
  - Layout for optimized track resolution (e.g. strip orientation and stereo angle) not necessarily optimal for track search.
- **Combined TPC–FTD:**
  - This may be an optional strategy for the intermediate region:
  - Inward extrapolation of tracks found by local PR in the TPC, FTD hits tested against and associated to them,
    - **Timing problems** hopefully solved by “time stamps”.
- **Soft hit association:**
  - Hits may be shared among tracks, and the final association relegated to track reconstruction based on the DAF.

Winfried Mitaroff HEPHY

# Summary

- Geant4/Mokka Simulation Models implemented.
- Started work on a Tracking API for use in Marlin.
- So far partially implemented for KalTest and F77 LEP Code.
- The addition of further tracking systems needs the implementation of bounded planar detectors in KalTest.
- Geometry and material budget needs tuning, although everything looks to working well with the new implementations.
- TPC Pat-Rec currently working on merging of track-segment found.
- Working on improving the diagnostics.

