MILLEPEDE II and experience with end user

Philipp Roloff (DESY)

JRA1 Meeting – Software Session
31/01/2008

Outline:

- Some more details on MILLEPEDE II
- Brief comments on user experience

MILLEPEDE II

- **EUTelAlign**: Only two planes are compared at a time.
- **Better solution**: Use full tracks in the alignment procedure.
- At the meeting in Paris it was announced to try the **MILLEPEDE II** package by V. Blobel for the alignment of the telescope.
- Used by LHC experiment: Should be maintained in the forseeable future.
- This is the first time MILLEPEDE II is used within the ILC software framework.

Basic principles

- Group parameters into two classes:

  local parameters q: only present in a subset of the data
  N = 4 • number of tracks
  (linear track model)
  global parameters p: shifts and rotations of sensors
  N = 3 • number of sensors
  (two shifts and a rotation)

- MILLEPEDE solves the **linear least sqares problem** with a simultanious fit of all global and local parameters, irrespectively of the number of local parameters.
- For mathematical details see www.desy.de/~blobel

History

- MILLEPEDE: Used since 1997 in H1, ZEUS, HERAb, PHENIX, STAR...

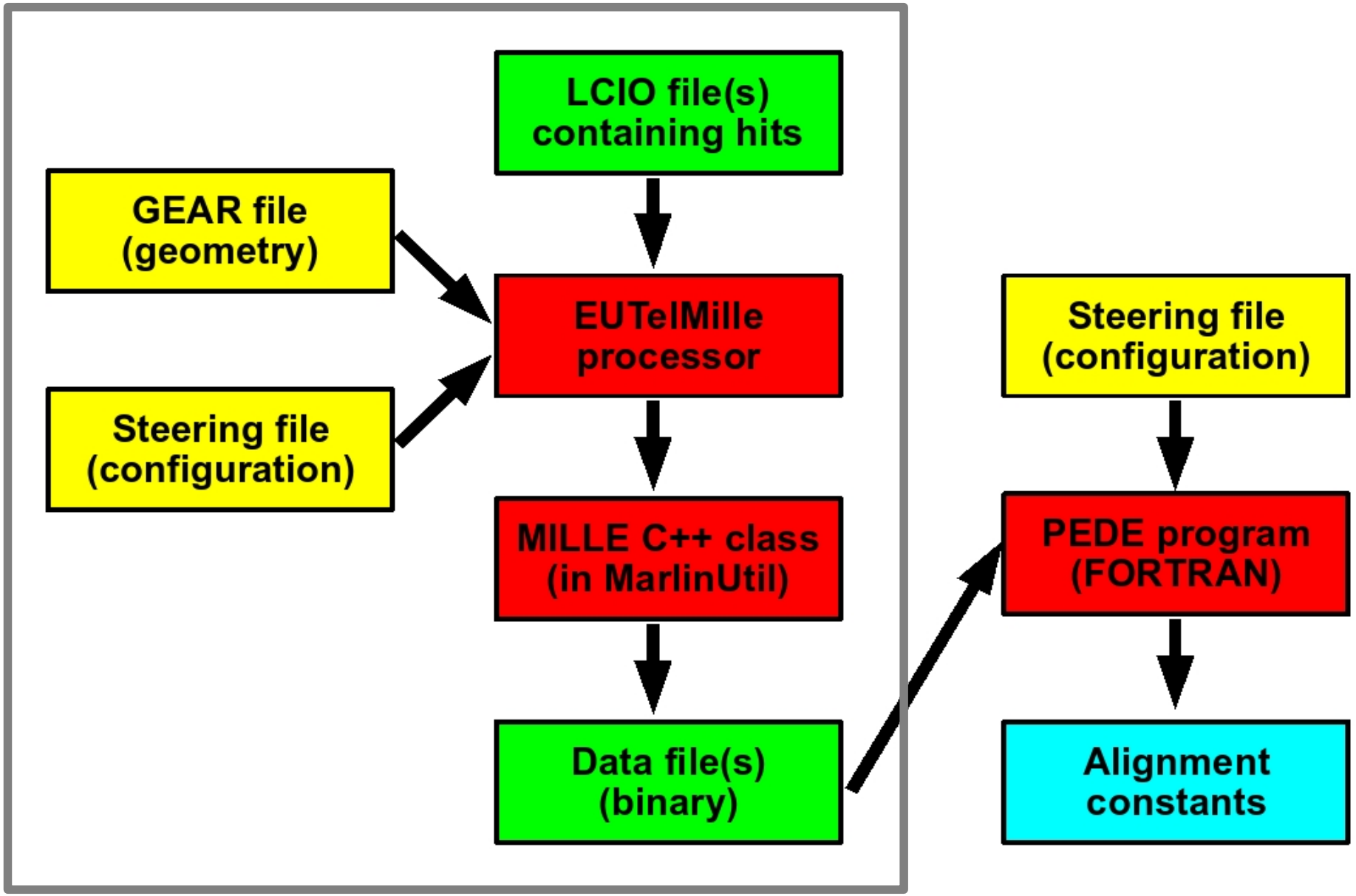
  Single FORTRAN sourcefile

- MILLEPEDE II: Available on the web since 2007

  Tested by H1 and CMS

  Main program still written in FORTRAN, but can be directly used in C++

MILLEPEDE II in EUTelescope



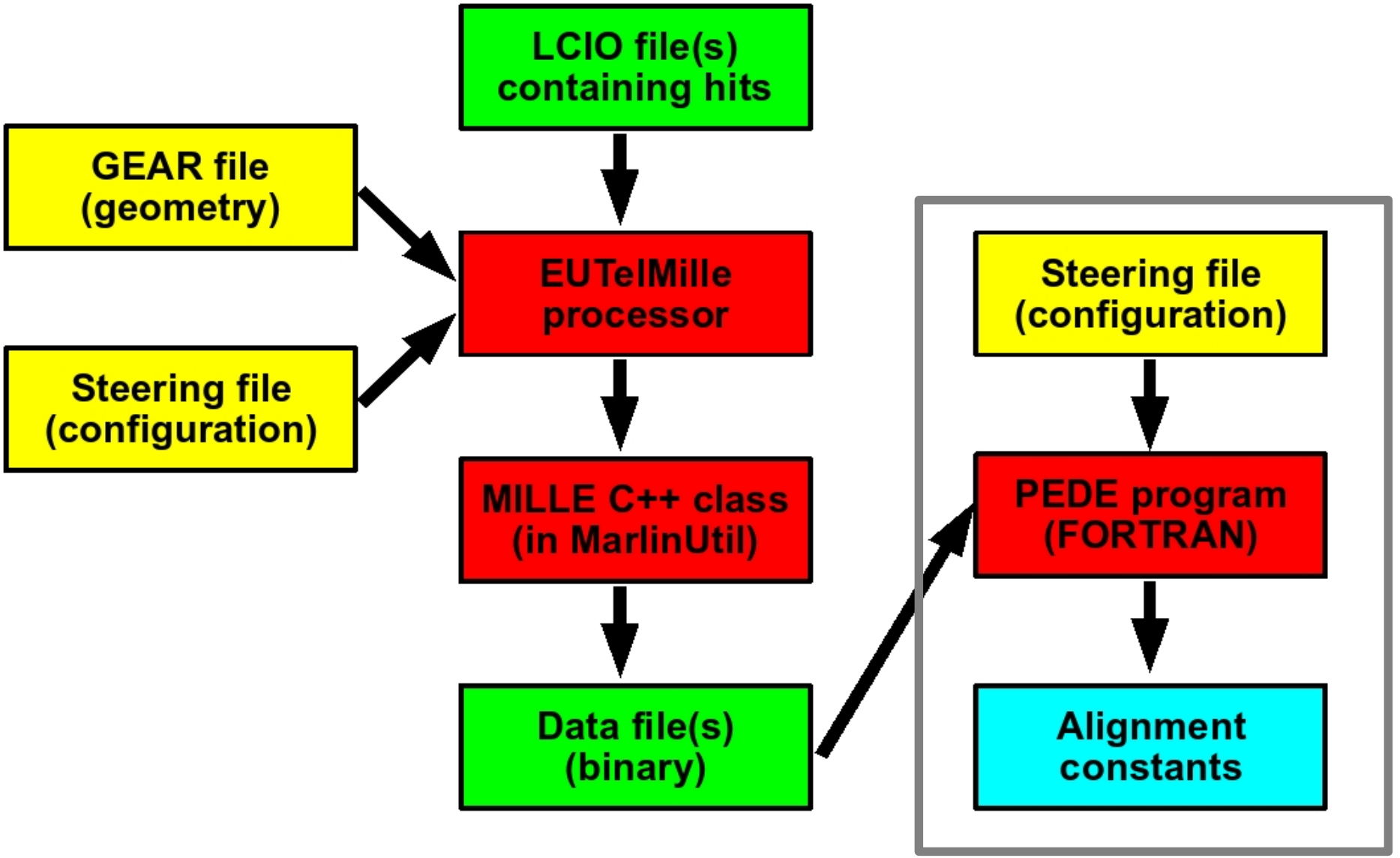
The EUTelMille processor

- Loops over all events and finds track candidates.
- Tracks described by: f(z,**q**,**p**).
- The following information is written to a binary file using the **MILLE** class in MarlinUtil for every data point (i.e. hit in a sensor assigned to a track):

  - **number and labels** of local and global parameters
  - **derivatives** with respect to the local and global parameters: $\partial f/\partial q_i$, $\partial f/\partial p_j$
  - **residual**
  - **standard deviation** of the measurement

- Dedicated processor for this enhances the modularity.

MILLEPEDE II in EUTelescope



The PEDE program

- Uses the binary file generated by EUTelMille to obtain the global parameters.
- Configured by a steering file (starting values of global parameters, constraints, number of iterations etc.).
- Allows to remove and / or downweight outliers.
- Typical execution time for telescope alignment on a 3 GHz Pentium 4: **≈ 10 s**. (Useful for online DQM?)

The global parameters

These scenarios were investigated so far:

1.) All planes can be shifted in the x and y direction: $x_0(1)...x_0(N)$, $y_0(1)...y_0(N)$.

2.) Additionally consider a rotation around the z-Axis: $\gamma(1)...\gamma(N)$.

$$R = \begin{pmatrix} 1 & \gamma & \beta \\ -\gamma & 1 & \alpha \\ -\beta & -\alpha & 1 \end{pmatrix}$$

NB: In scenario 1 and 2 the z coordinates of the hits are not affected by the alignment.

3.) Consider all three rotations (implemented in EUTelMille, needs to be tested).

Constraints

Constraints are necessary to obtain stable results. Two possibilities were studied here:

1.) Fix all global parameters for the first and the last plane. *Maybe a good choice for large distances between the planes.*

2.) Fix all global parameters for the first plane and use these constraint equations:

$$x_0(1) / (z(0)-z(1)) + x_0(2) / (z(0)-z(2)) + x_0(3) / (z(0)-z(3)) + x_0(4) / (z(0)-z(4)) = 0$$

$$y_0(1) / (z(0)-z(1)) + y_0(2) / (z(0)-z(2)) + y_0(3) / (z(0)-z(3)) + y_0(4) / (z(0)-z(4)) = 0$$

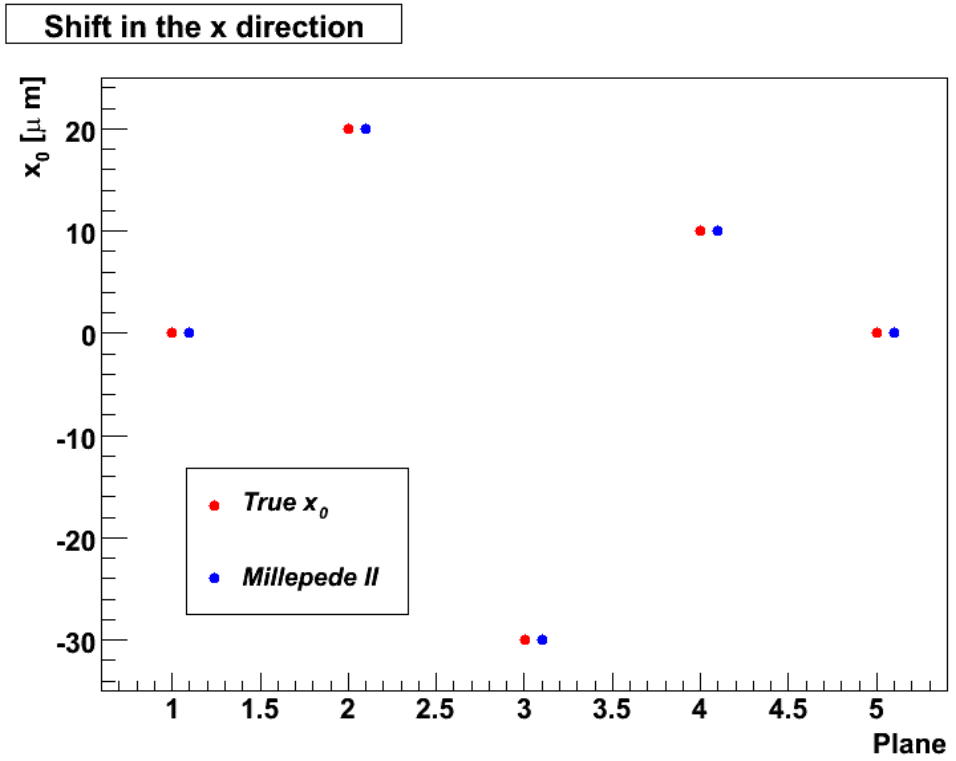
NB: The EUTelMille processor only needs to run once!

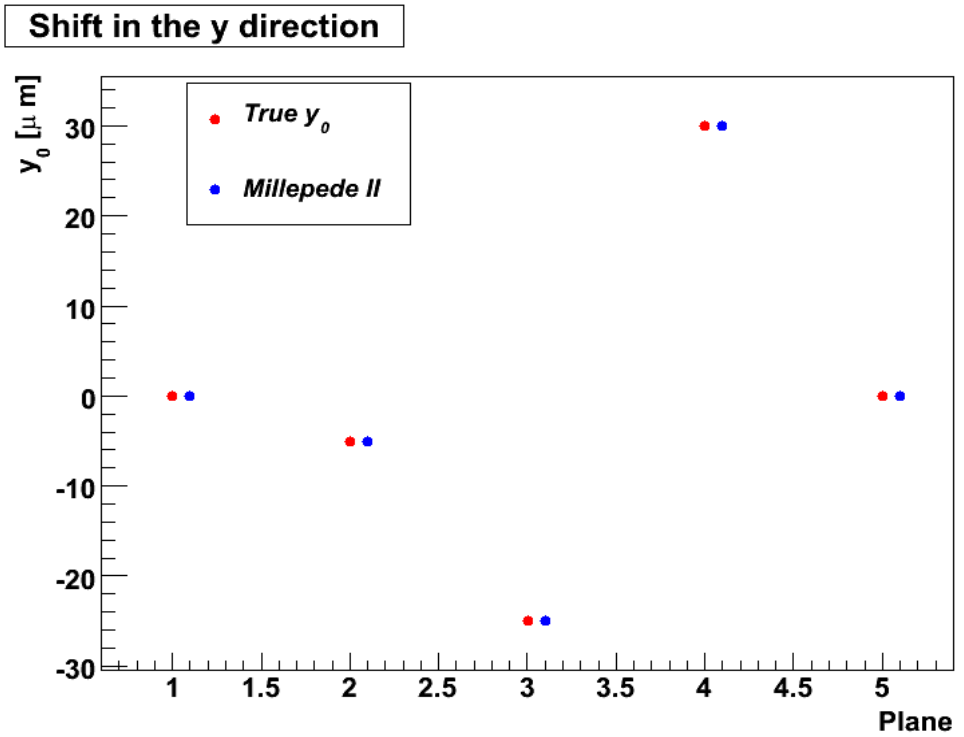
Test with “simulated” data

There is no simulation of the telescope available within the Marlin framework yet.
→ Use simple smearing approach:

- Geometry of the August testbeam at DESY.
- Generate about 80.000 tracks equally distributed in the middle sensor.
- Track slopes are distributed according to a gaussian function in x and y, **σ = 0,0005**.
- Smear hit positions in x and y with a gaussian function, **σ = 10 µm**

Planes are shifted by x_0 **and** y_0 **(and γ)**

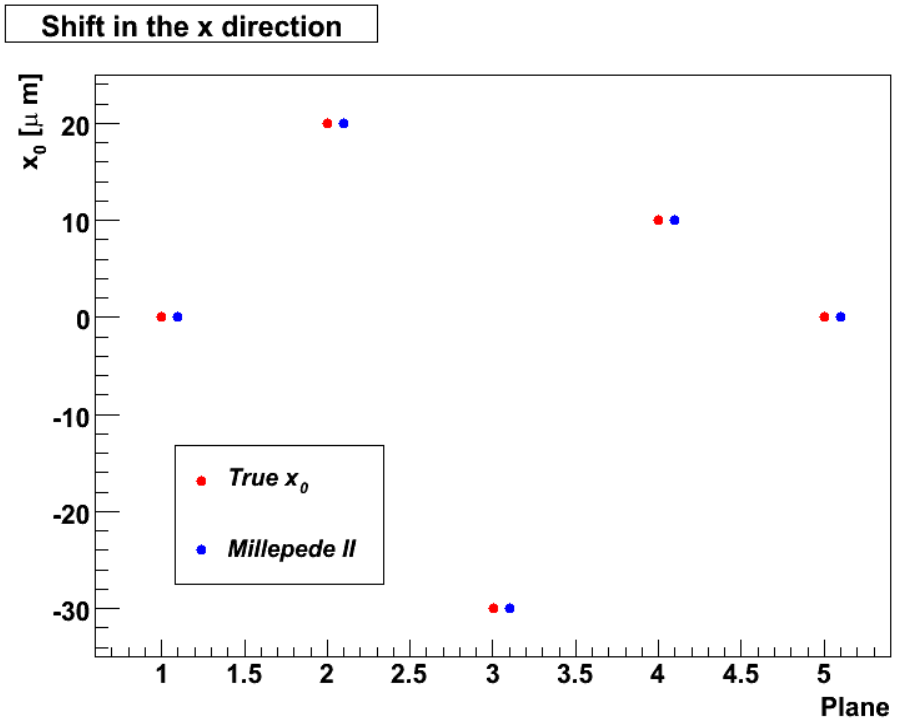




Constraints 1:
Fix first and last plane

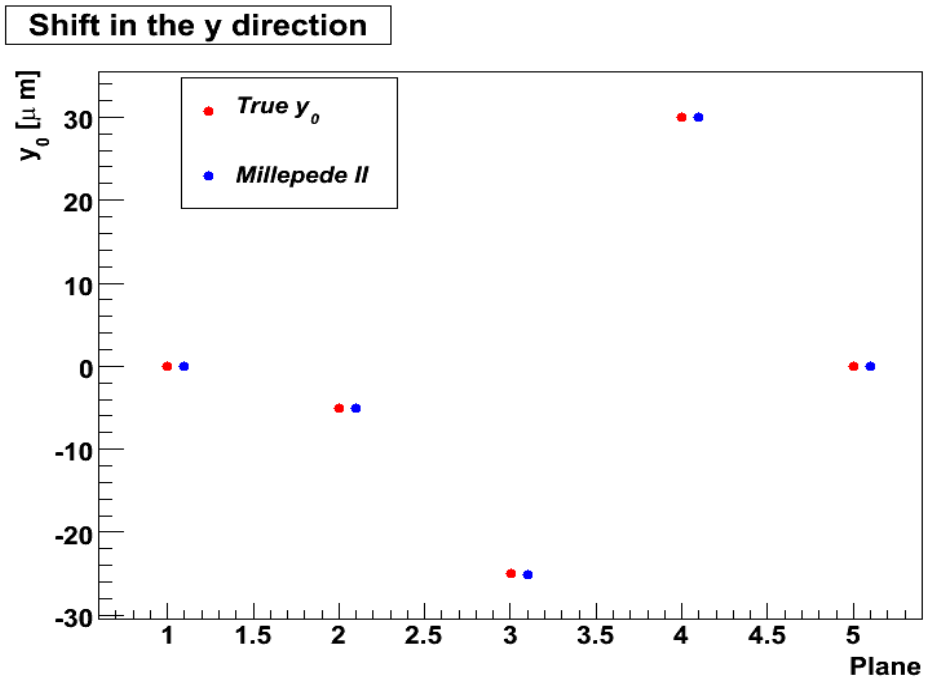
Only shifts

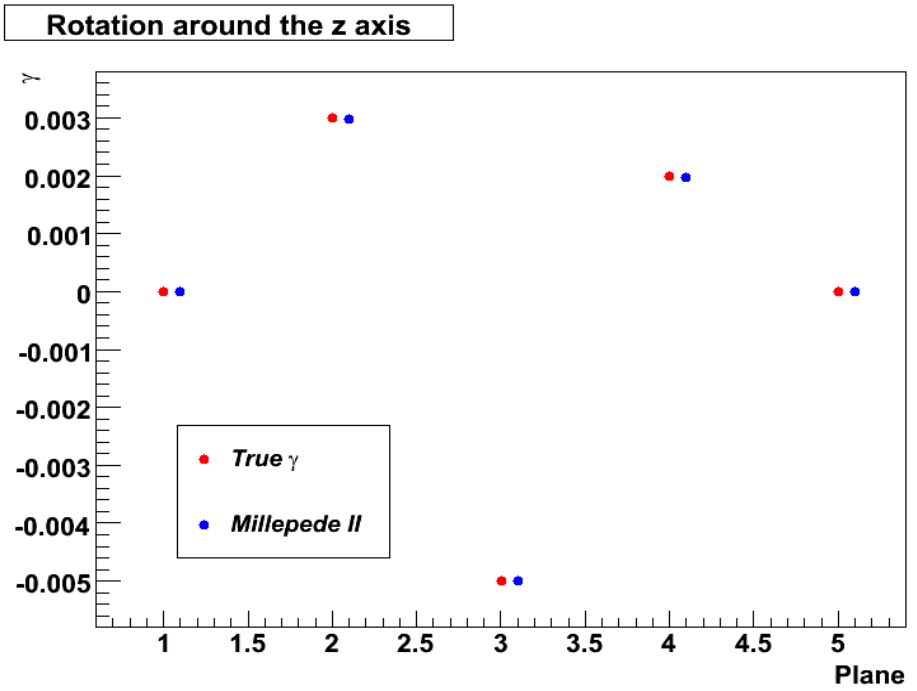
Alignment works perfectly.

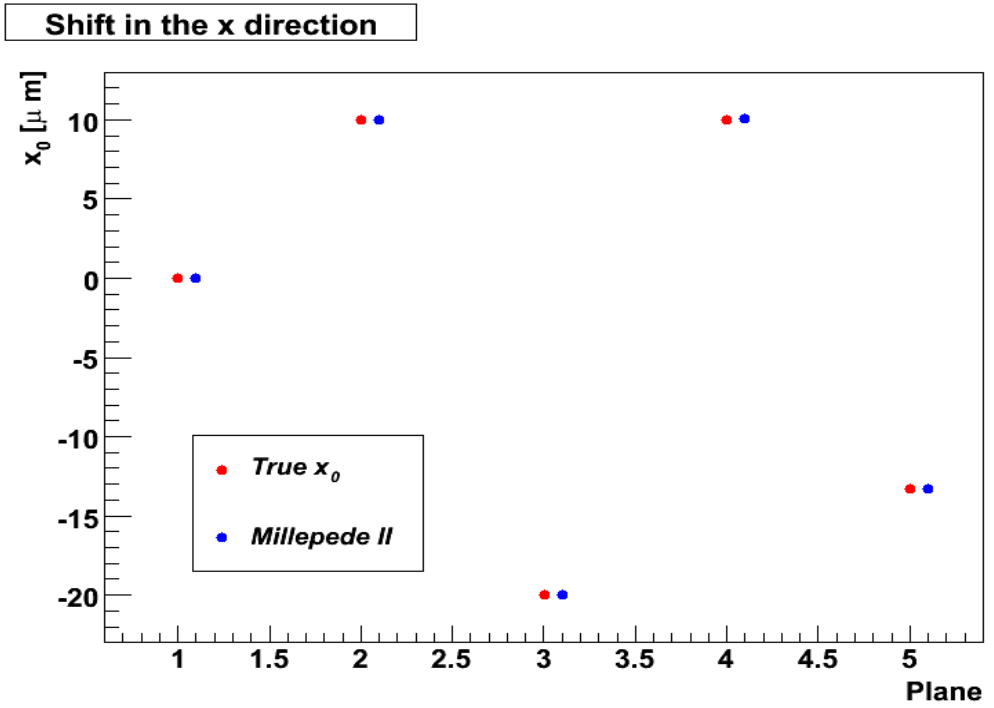


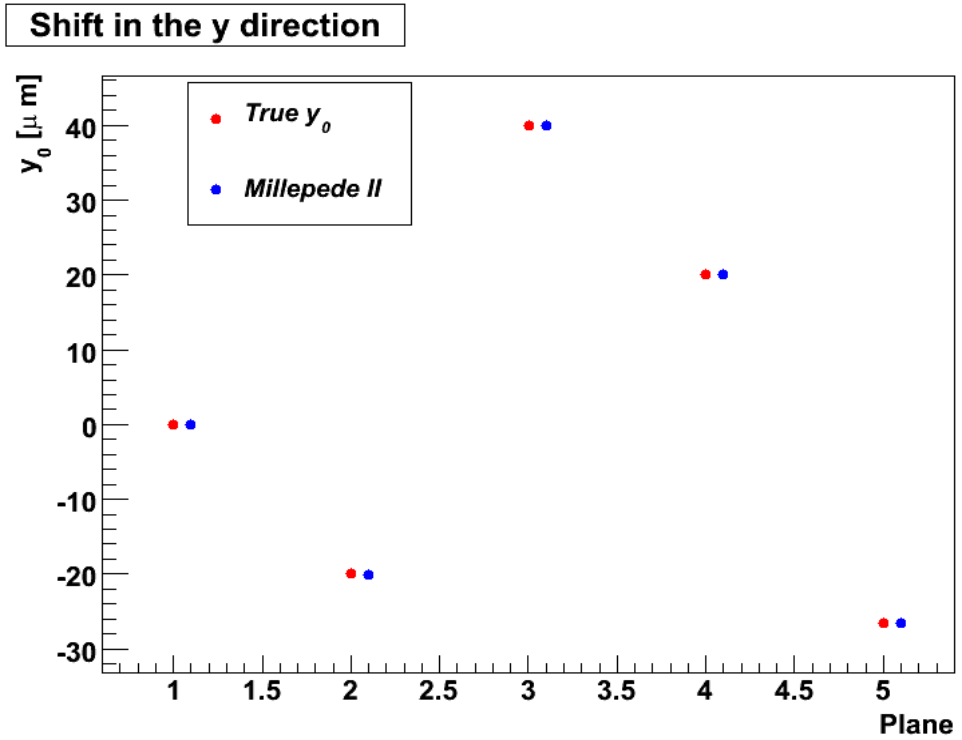
Now add rotations around the z axis...

The alignment still works very well.









Constraints 2:

Fix first plane and two constraint equations

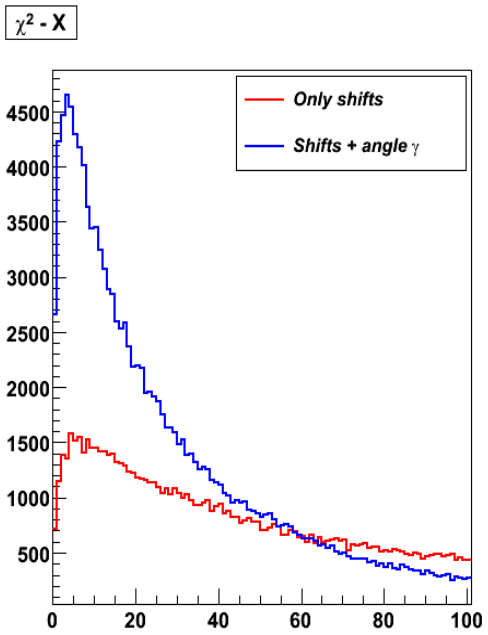
NB: true values were chosen to fulfill the constraints

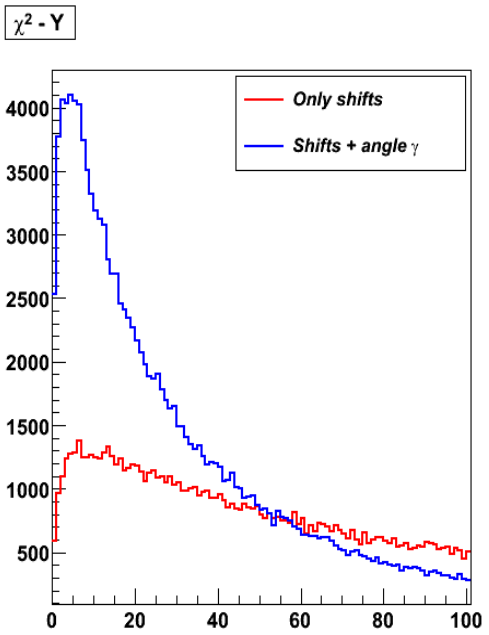
Again, the result looks very good.

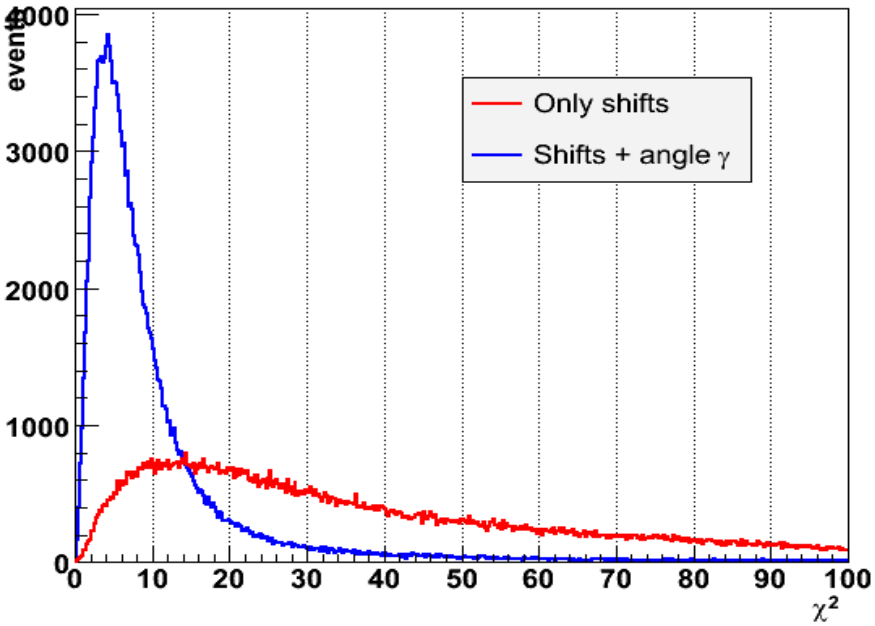
First look at real data

- Implementation done (last bug found about a week ago). Now starting to look at testbeam data, finding the optimal constraints etc.
- Perform the alignment for the 3 GeV electron data from the testbeam at DESY in August 2007.
- Use only single track events in the alignment procedure (about 28.000 events).
- Then test the derived constants in the track fitters.

Constraints 1

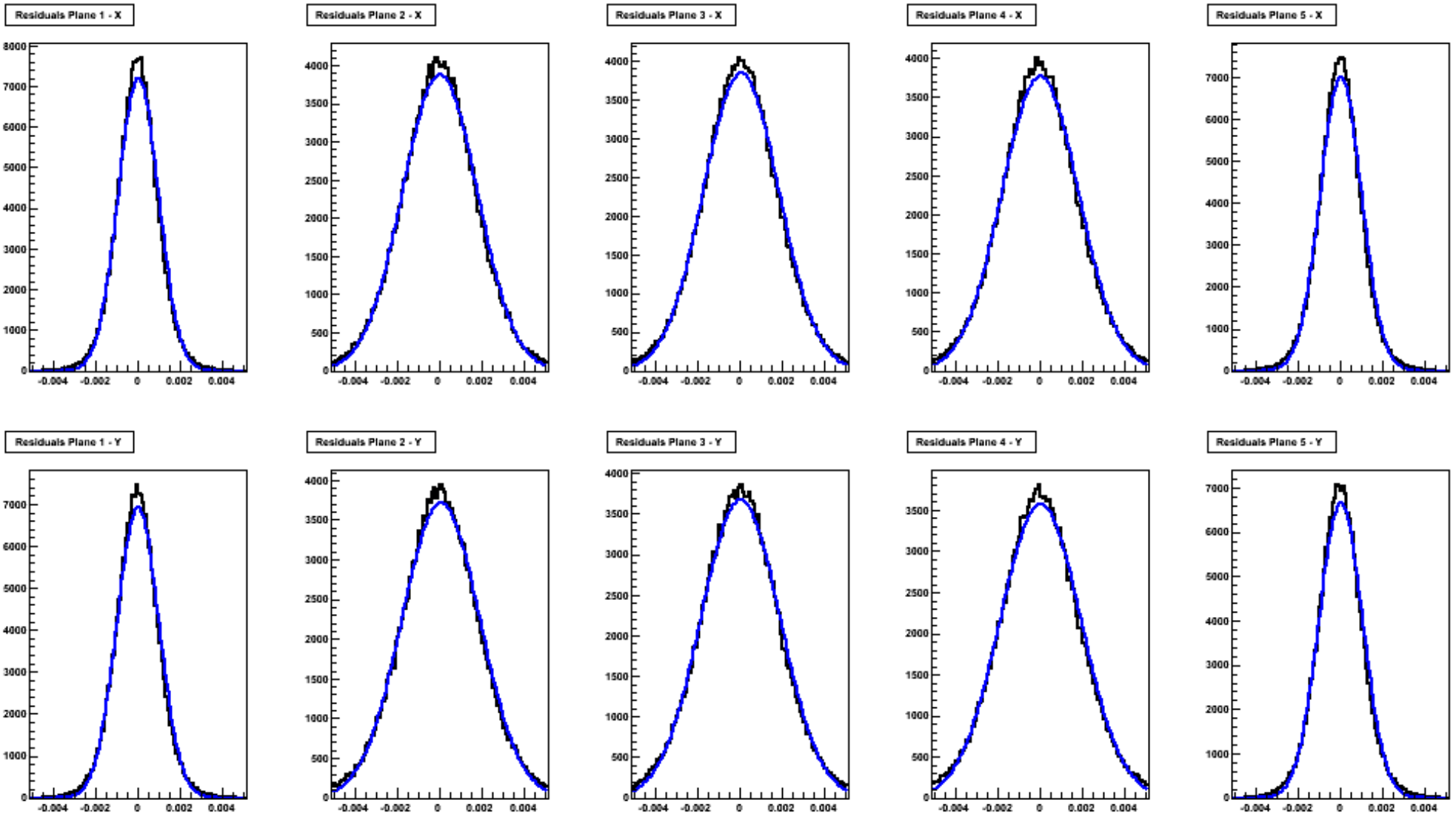






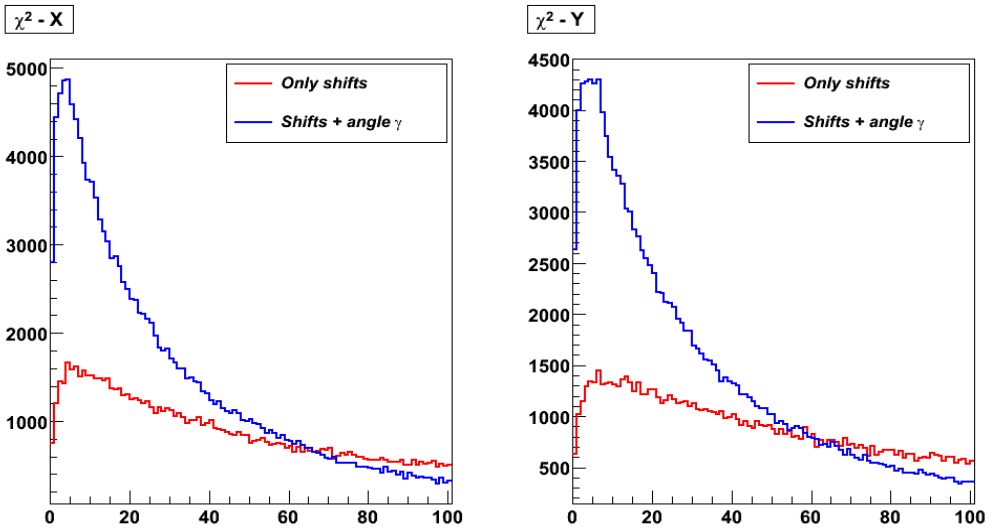
EUTelMultiLineFitter **EUTelTestFitter**

Fits much better using the angle γ in the alignment.

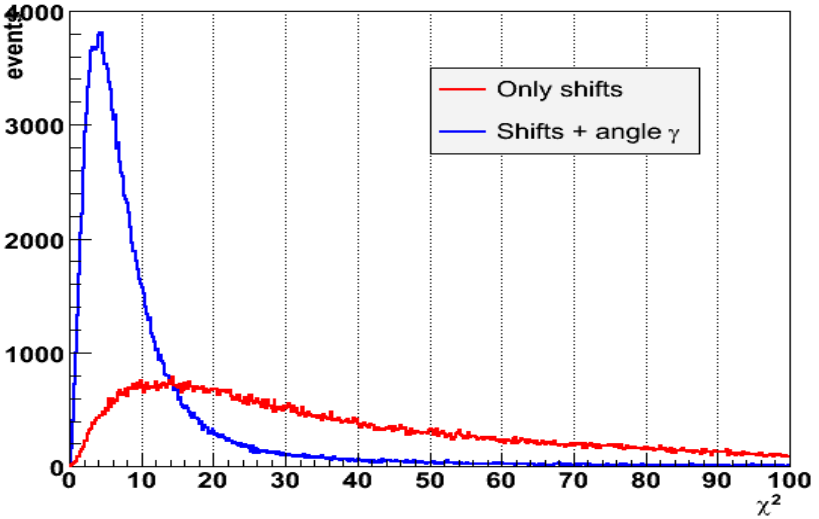


Alignment precision better than 0.05 μm. (From the mean values of the gaussian fits)

Constraints 2

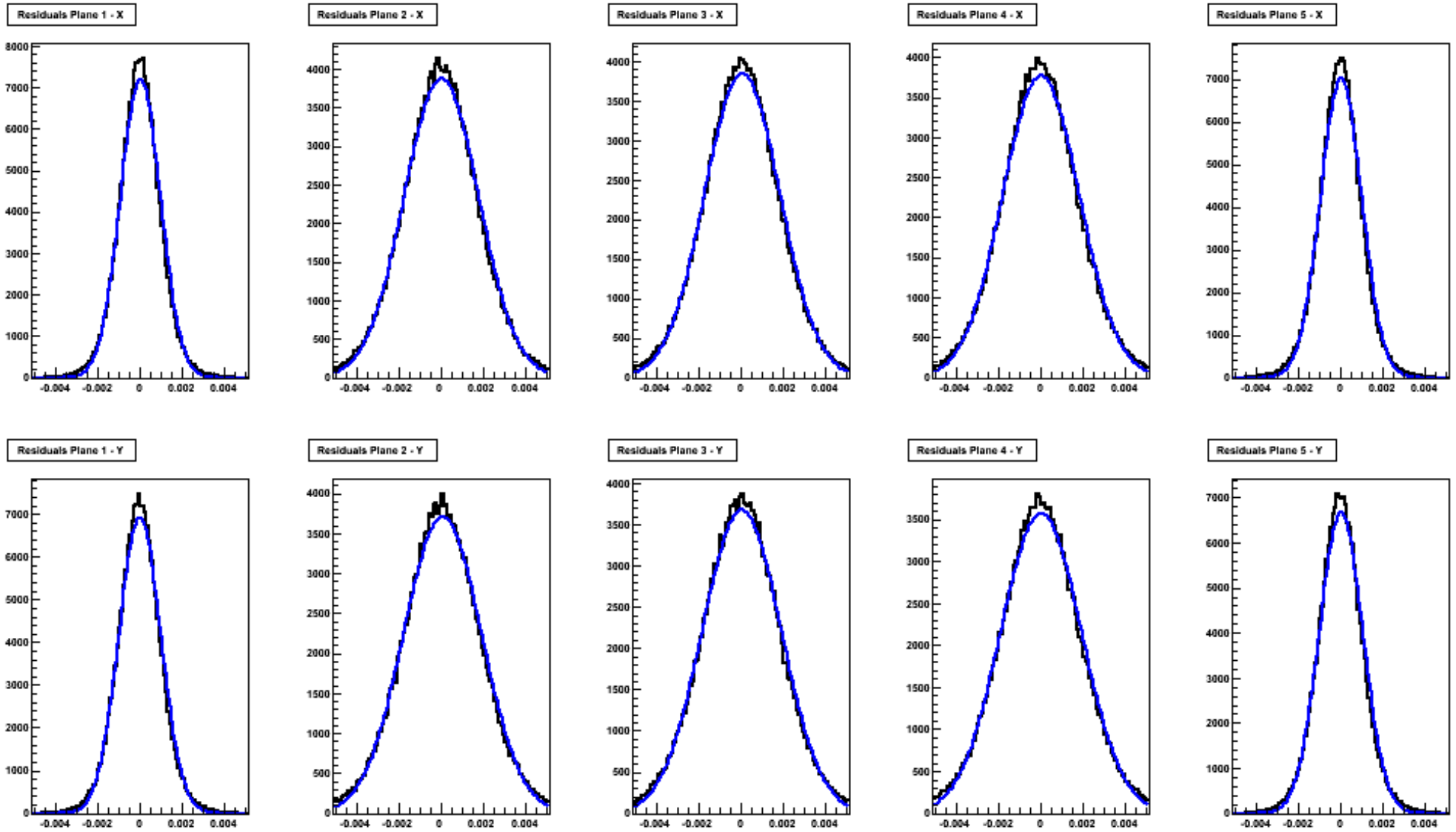


EUTelMultiLineFitter



EUTelTestFitter

Very similar as before.



Alignment precision better than 0.05 μm.

Summary on MILLEPEDE II

- The integration of MILLEPEDE II into the EUTelescope software package was sucessful.
- The alignment procedure is fast and flexible.
- First tests look promising.
- Next steps:
  - Try 3 rotations
  - Submit everything to CVS
  - Store constants in LCCD
  - Write memo
- The deadline (summer 2008) will be easily matched ☺

Experience with BeamCal: users wishes

- ROOT-File with hit positions in DUT layer.
- **Solution**: Conversion processor running after tracking.
- As fast as possible.
- As little work as possible.
- Typical for group not developing pixels?

A user might not know anything about...

- **The GRID** (→ PERL scripts, lots of documentation available online)
- **Marlin etc.** (→ ilcinstall, example xml-Files)
- **Pixel sensors** (→ example DQM plots, automatise DQM?)
- **Our software package** (→ as good documentation as possible, tutorial?)