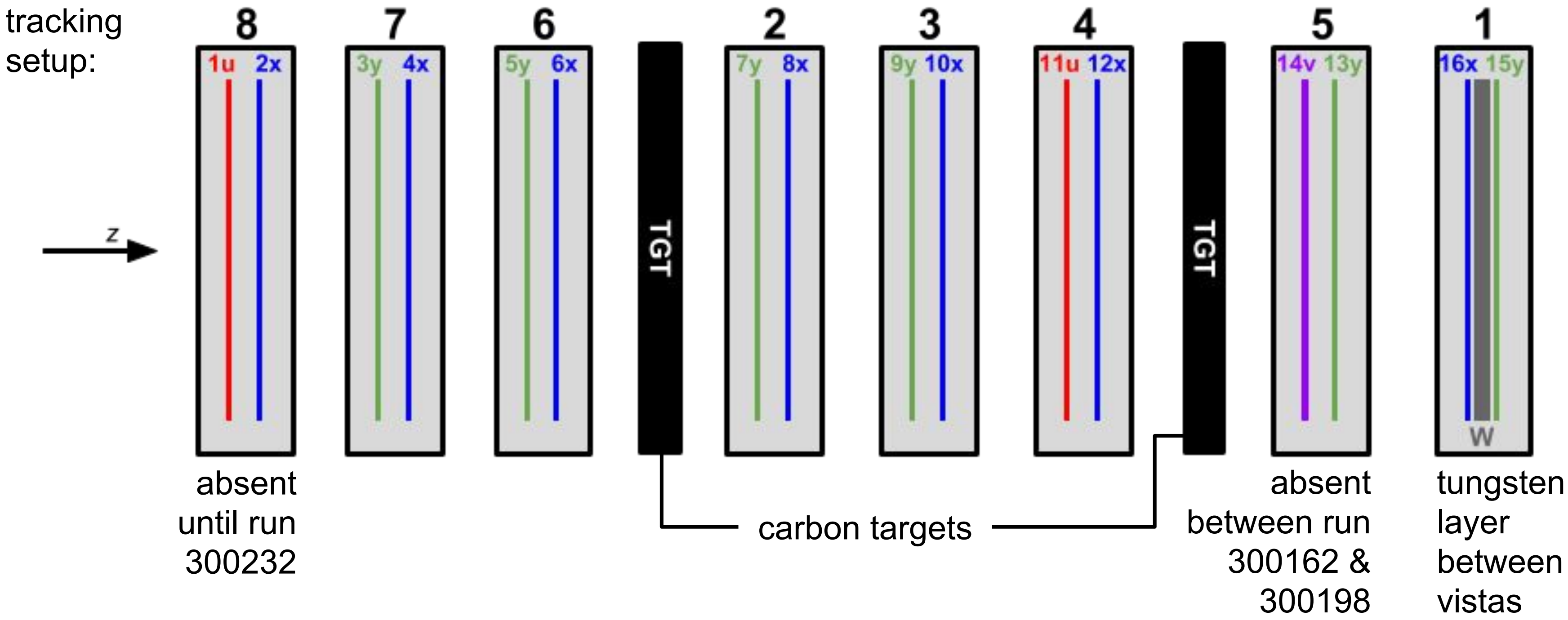
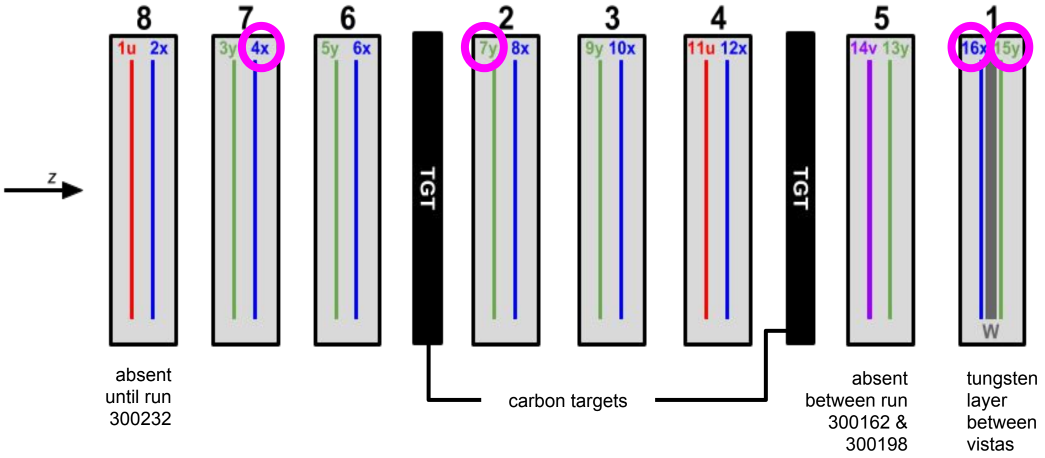
Si TRACKING SYSTEM PERFORMANCE



LOOKING FOR THE REFERENCE SYSTEM...



Some notes:

1) *W layer in the last module is just* 0.07X0 - *there isn't any other bulk layer along the beam line*
2) *ASCII files were created with these conditions:*
   - ➔ *single hit on 4x, 5y, 6x AND single OR zero hit on 3y for 'older' runs (until 300232)*
   - ➔ *single OR double hit on 2x, 4x, 5y, 6x (while 3y is free) from run 300232 on*

What's the best reference system for relative alignment? Let's try

4x,16x,7y,15y

- ➔ 4x because 2x was absent in older runs
- ➔ 7y in order to study 3y and 5y performance
- ➔ 16x & 15y in order to include most of the layers between the reference ones and because detector in box 1 is reliable

Using **run 300232**, which is the 1st one in the 8-box configuration - ~900k events (beginning of May)

...AND THEN RUNNING THE ALIGNMENT ALGORITHM

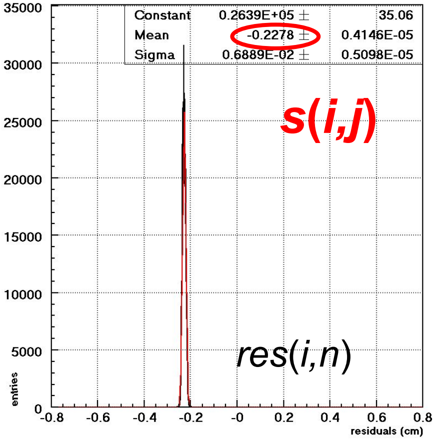
Multiple iterations on this algorithm, which corrects raw positions for transverse shifts and rotations about longitudinal axis, both relative to the direction identified by the reference system → output: relative-aligned hit points defined by raw hit points and alignment parameters, *SHIFT(i)* and *ROT(i)* with *i* layer index. @ *n*-th iteration:

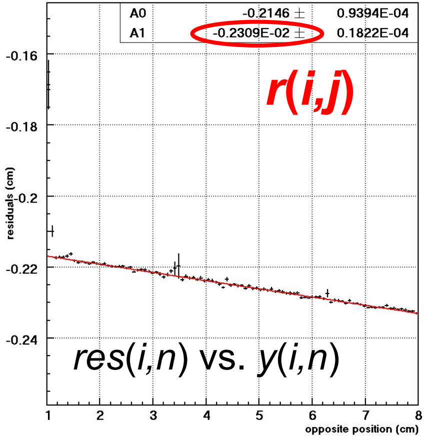
$$x(i,n) = x(i,0) - ROT(i)*y(i,0) - SHIFT(i)$$

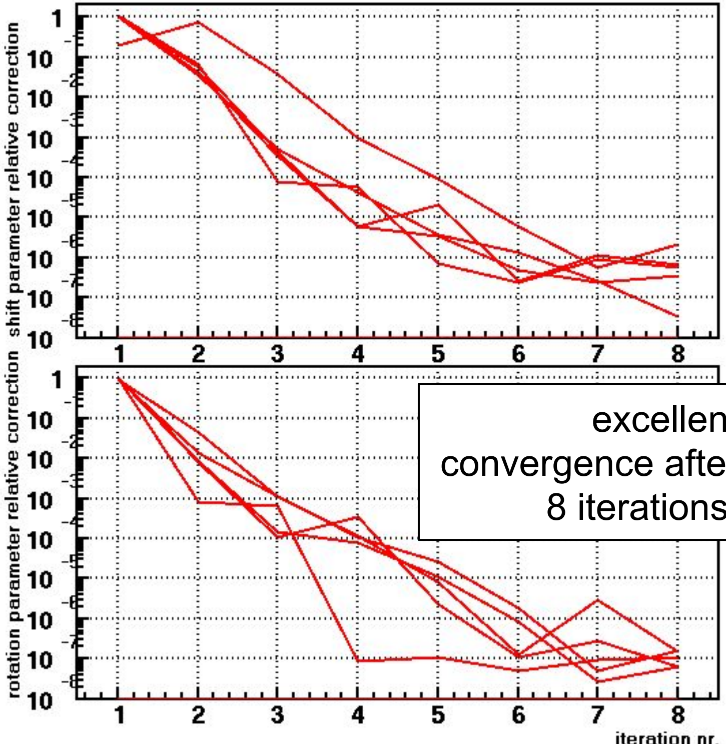
(with *y*(*i*,*0*) some layer correlated to *x*(*i*,*0*) in the opposite vista)

$$SHIFT(i) = s(i,n-1) + s(i,n-2) + \dots$$

$$ROT(i) = r(i,n-1) + r(i,n-2) + \dots$$



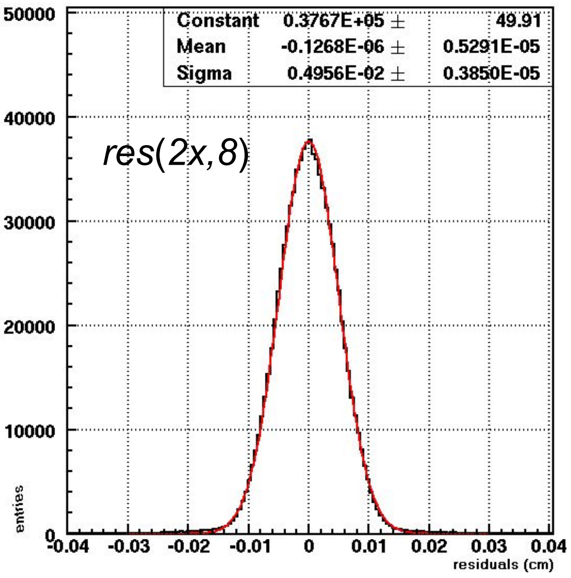


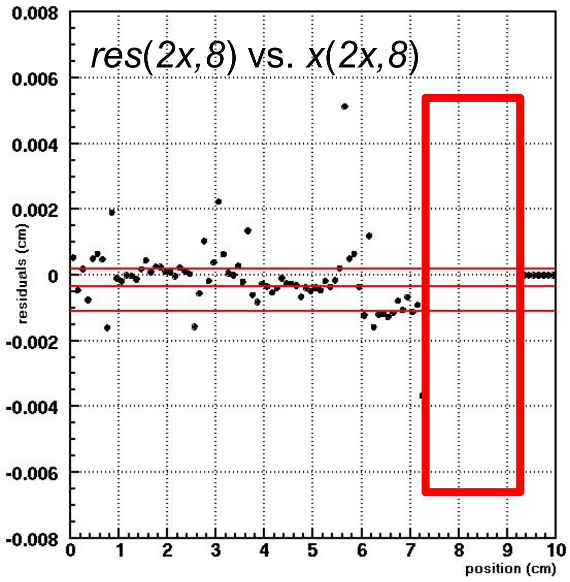


excellent convergence after 8 iterations!

Note: since the algorithm correlates x & y vistas, some single hit condition is needed on all the studied layers

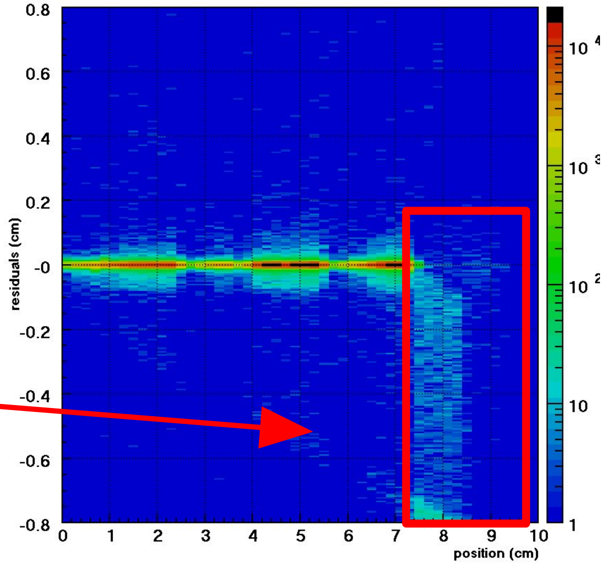
OUTPUT OVERVIEW





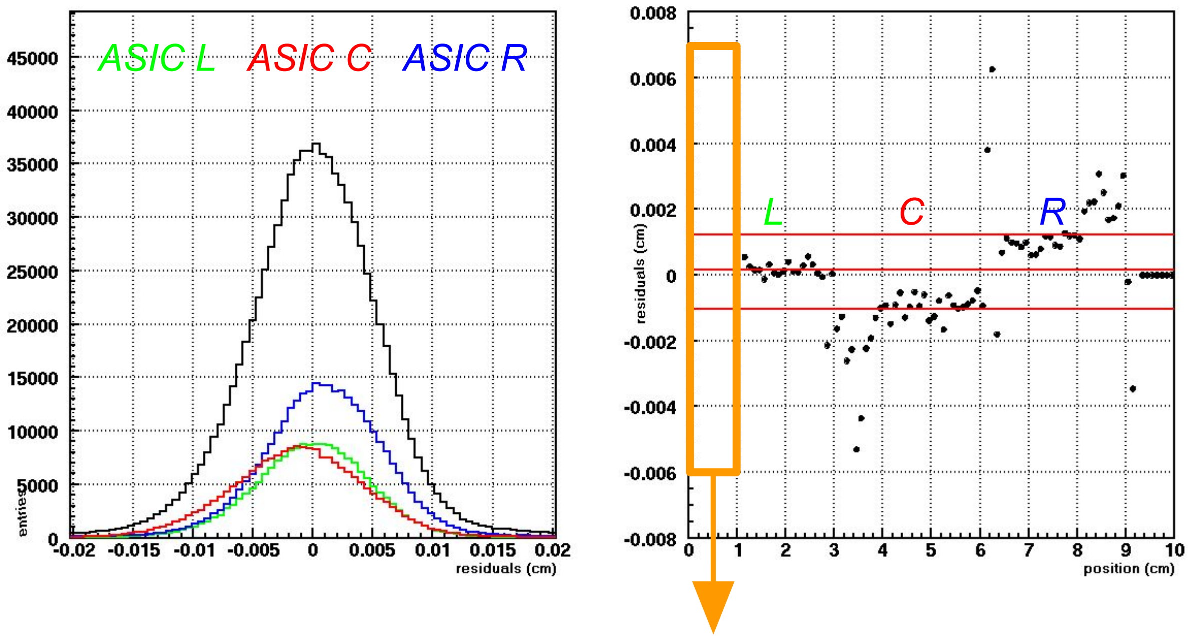
2x

- ➔ 1st layer in the *x* vista
- ➔ Shows no heavy ASIC dependence - it's hard to tell anything about the rightmost ASIC due to a lack of statistics in x>~7.5cm



What happens for x>~7.5cm? This pattern is shown in every x layer with no dependence on the chosen reference system → rightmost x region might be excluded due to misalignment between layers (in combination with global single track condition) and/or due to misalignment with trigger system

OUTPUT OVERVIEW



3y

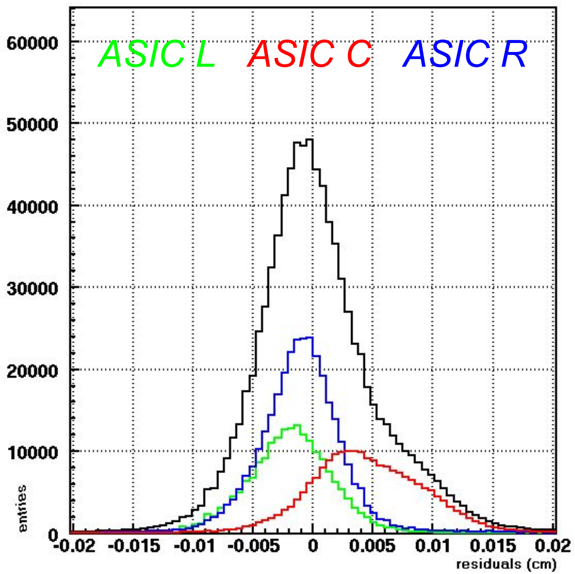
- 1st layer in the *y* vista
- Asymmetry! Indeed different ASICs show different shifts in residual distributions

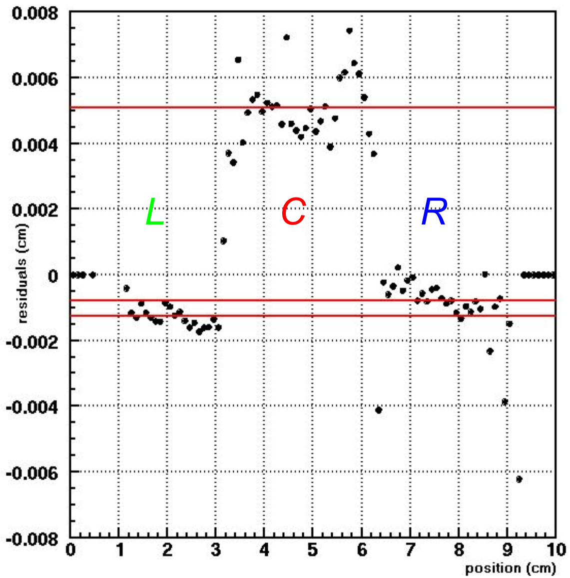
  ~12um between L & C
  ~10um between C & R

- What is more, there is some fluctuation pattern within ASICs C & R regions → bad behaviour of ASICs C & R (while L is ok, as can be seen from raw data), unpredictable and probably unsolvable (in zero suppression mode - see slide 16), but not so critical

Leftmost centimeter absent in all y vista layers. This is due to a malfuntioning in 5y first 20 strips, which are dead → nonzero hits condition on 5y (which is required @ the moment of ASCII files creation) makes this pattern propagate to all the y vista layers

OUTPUT OVERVIEW





5y

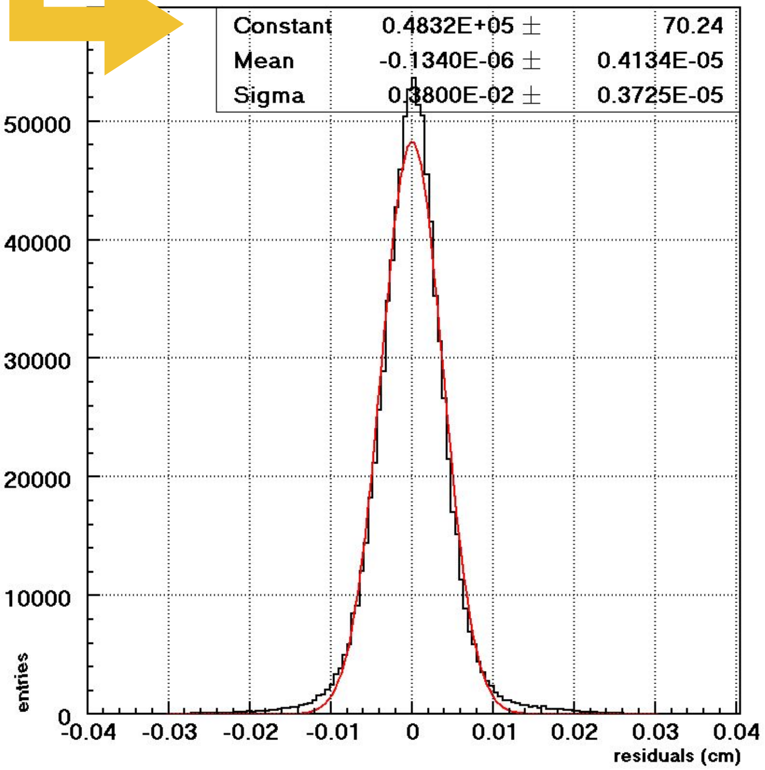
Huge asymmetry! Chip C shows a shift of ~61um from L & R levels. This is due to some ASIC C malfunctioning which has the net effect of biasing the reconstructed hit point ~61um to the right.

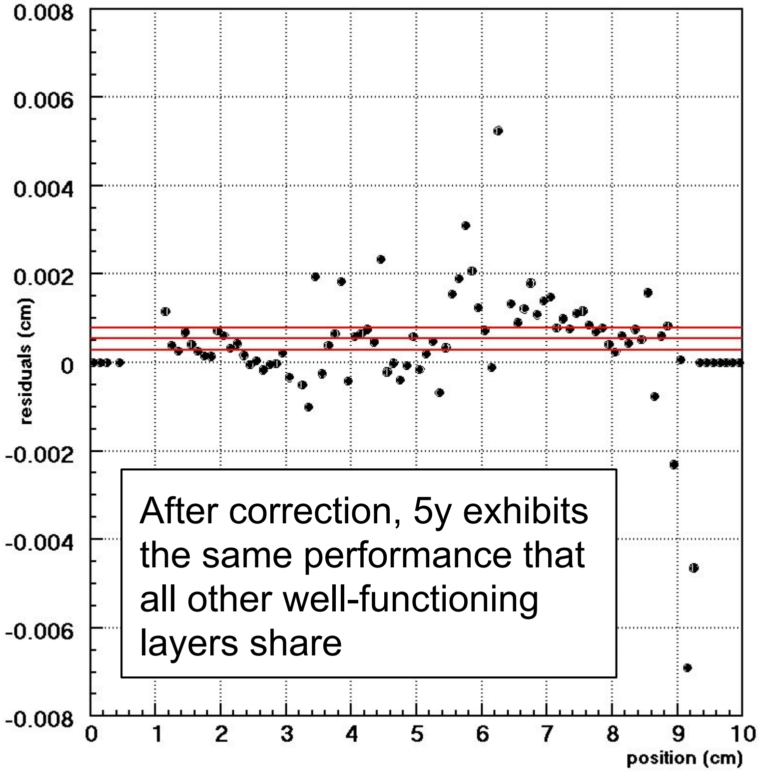
The behaviour of this ASIC seems to improve when running at lower clock frequency (1.25MHz instead of the typical value 2.5MHz) - see slide 17; at the moment there is poor statistics in these conditions though → might be interesting to study the problem with dedicated runs.

For the time being, a manual correction for the events passing through the central region of this layer should fix the problem…

$$x(4,0) \rightarrow x(4,0) - 6.1\text{E-}3 \text{ cm if } x(4,0) \in (3.1, 6.2) \text{ cm}$$

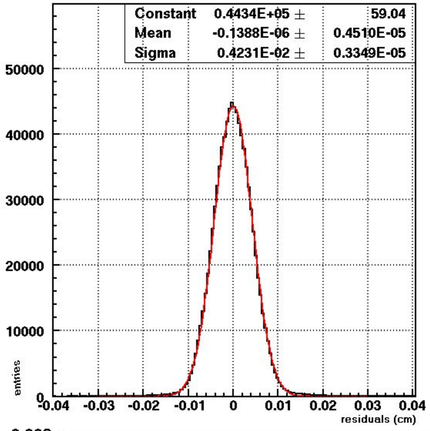
OUTPUT OVERVIEW

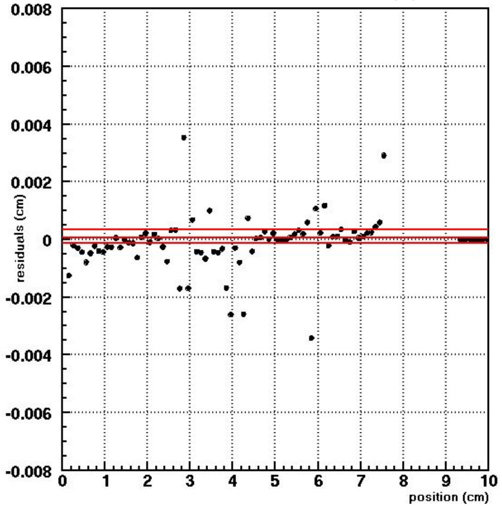




After correction, 5y exhibits the same performance that all other well-functioning layers share

OUTPUT OVERVIEW

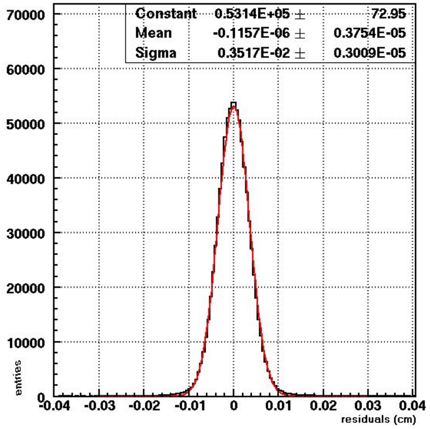


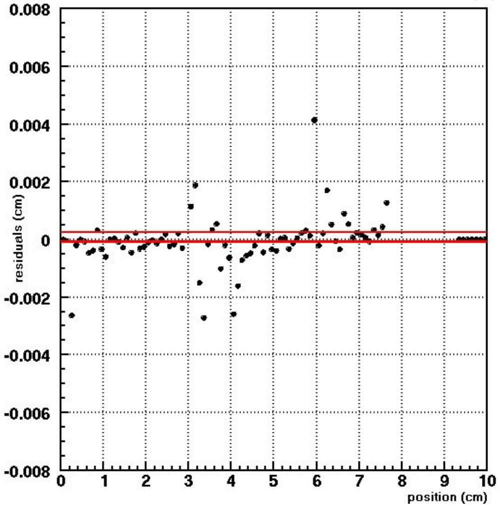


← 6x & 8x →

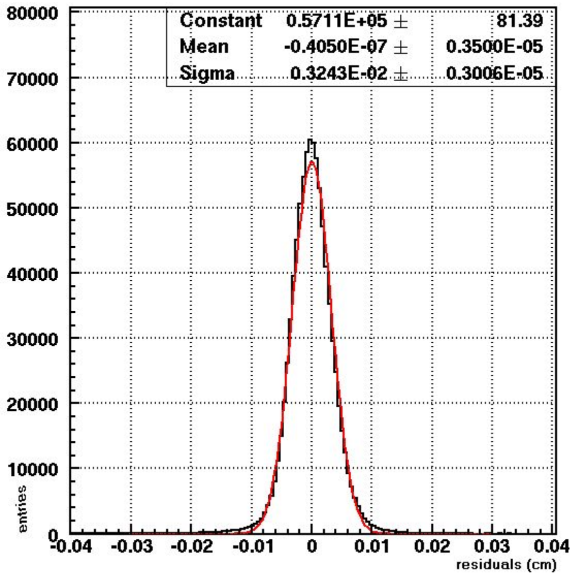
Everything seems fine.

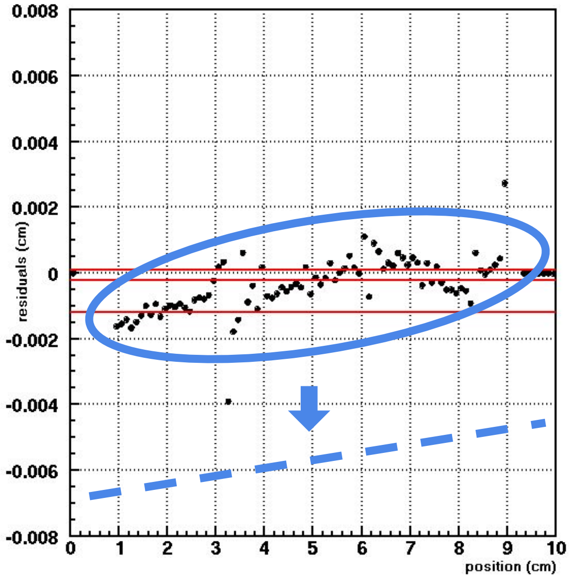
There are local fluctuations in the leftmost region of all the ASICs, probably due to the lack of statistics induced by the propagation of the 10x inefficiency pattern - see slide 10





OUTPUT OVERVIEW



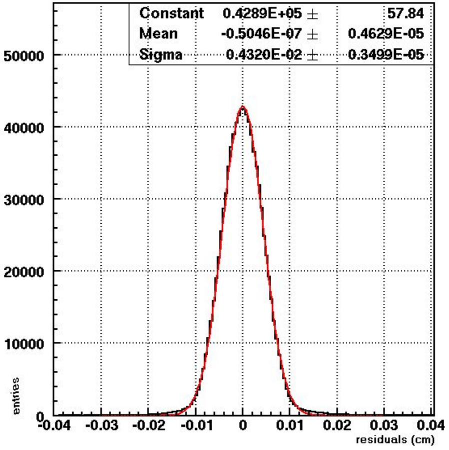


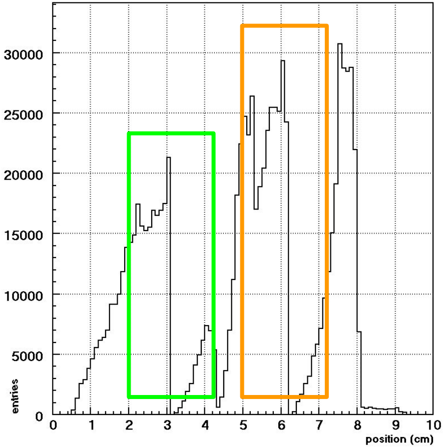
9y

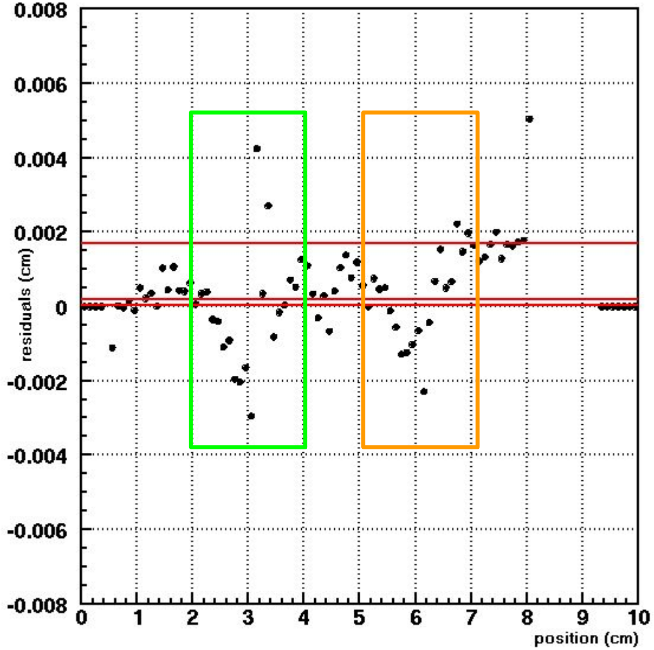
Everything ok. There isn't any local shift, though residual-over-position plot seems to exploit a linear dependence → rotation about the opposite (x) axis?

This would give us access to the relative tilts about transverse axes → interesting, must investigate

OUTPUT OVERVIEW







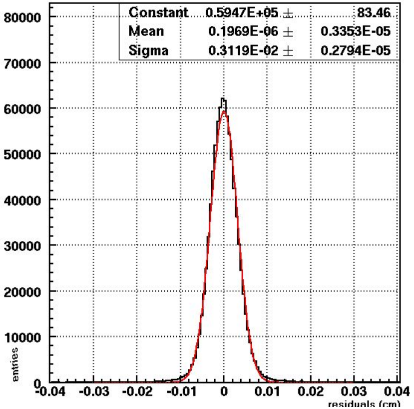
10x

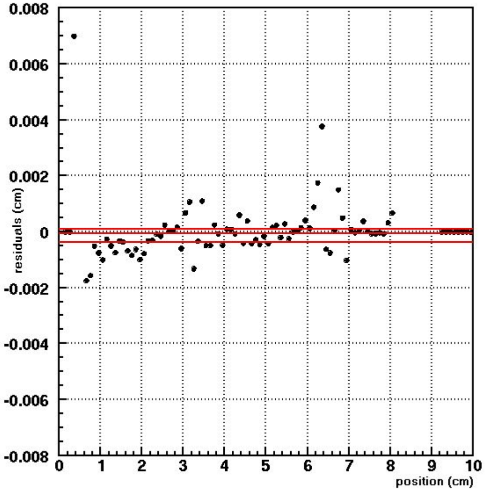
Many problems...

- ➔ The leftmost region of every ASIC is messed up → inefficiency pattern which propagates to all the x vistas in the global single hit condition
- ➔ The lack of statistics in these ‘holes’ makes the residuals distribution poor of events in these areas → local big uncertainty
- ➔ Rightmost ASIC?

Fortunately, this does not seem to be a problem for the alignment process

OUTPUT OVERVIEW

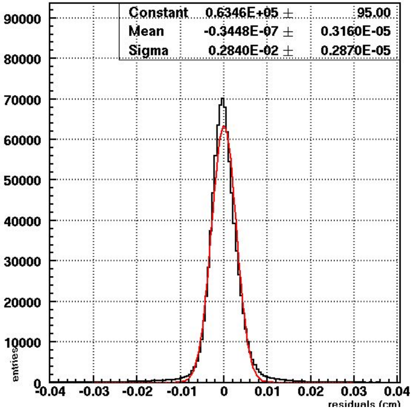


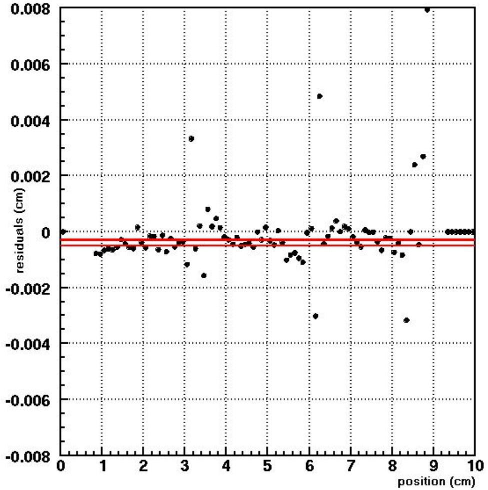


← 12x & 13y →

Everything seems fine.
(13y is the highest resolution layer in this configuration)

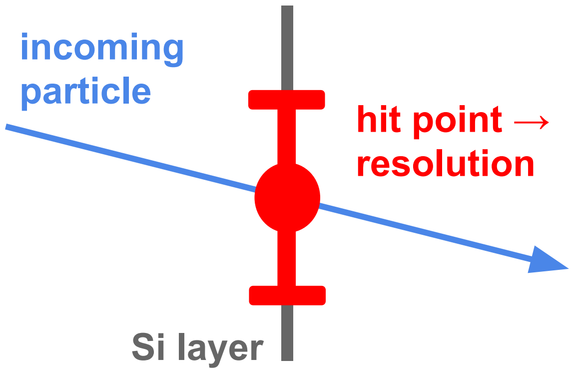
12x shows the 'typical' inefficiency pattern from single hit condition on 10x





SPATIAL RESOLUTION & LOCAL SHIFTS

Every fluctuation smaller than spatial resolution falls within hit point sigma & is way lower than minimum distance which allows multiple track discrimination

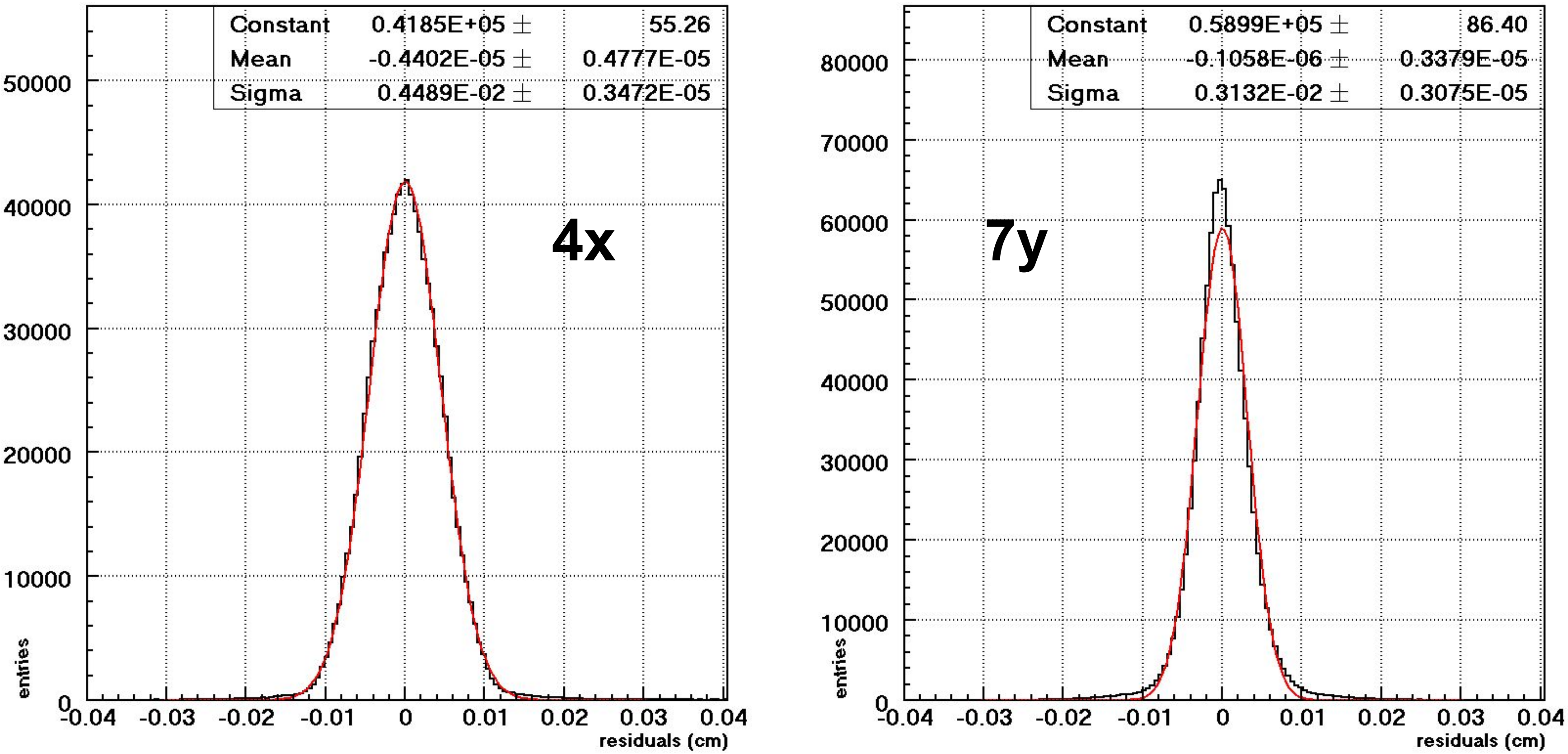


	measured spatial resolution (um)
2x	49.56 ± 0.04
5y	38.00 ± 0.04 (after correction)
6x	42.31 ± 0.03
8x	35.17 ± 0.03
9y	32.43 ± 0.03
10x	43.20 ± 0.03
12x	31.19 ± 0.03
13y	28.40 ± 0.03

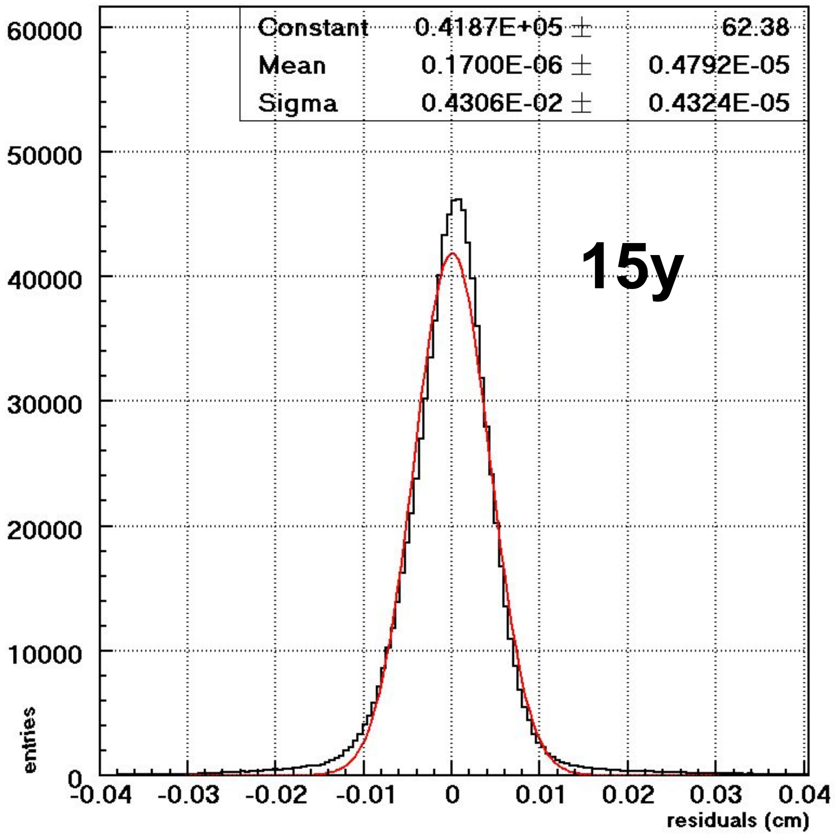
Do we need to go further? Would it make sense?

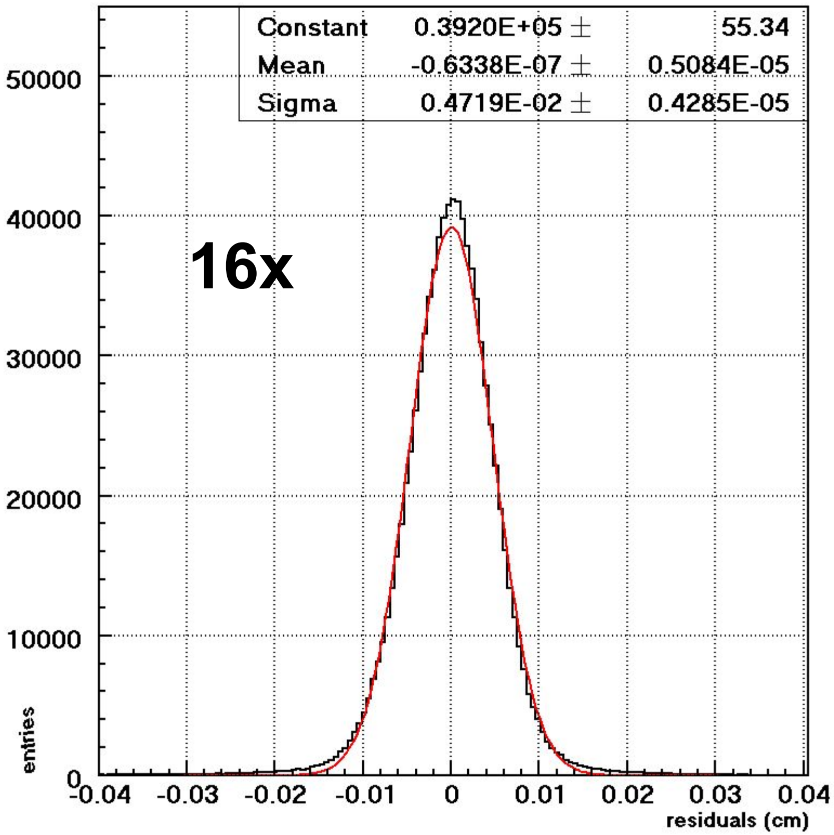
	measured spatial resolution (um) (obtained with reference **2x,5y,12x,13y** - see slides 13-14)
4x	44.89 ± 0.03
7y	31.32 ± 0.03
15x	43.06 ± 0.04
16y	47.19 ± 0.04

REFERENCE LAYERS

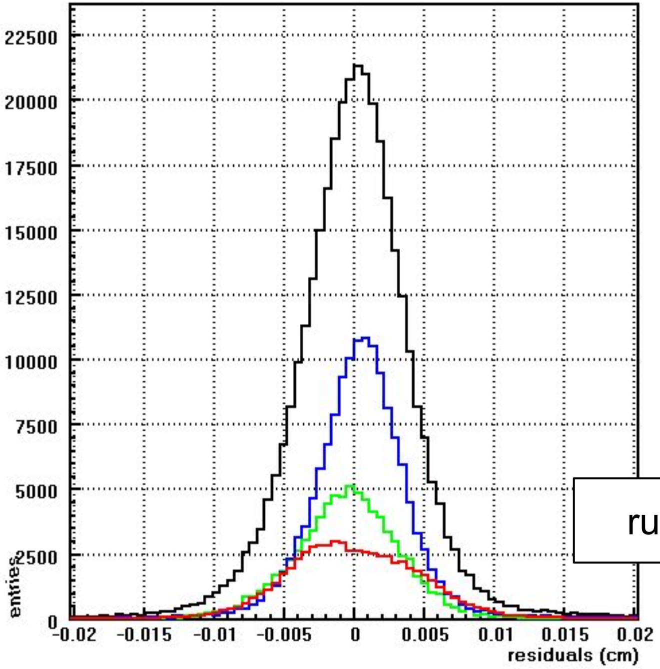


REFERENCE LAYERS





CROSS CHECK



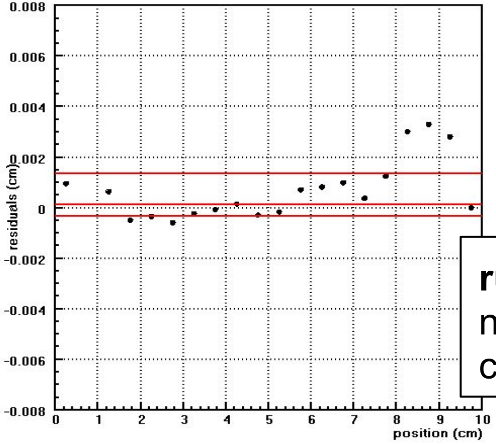
run 300269

The results presented above were measured with other runs

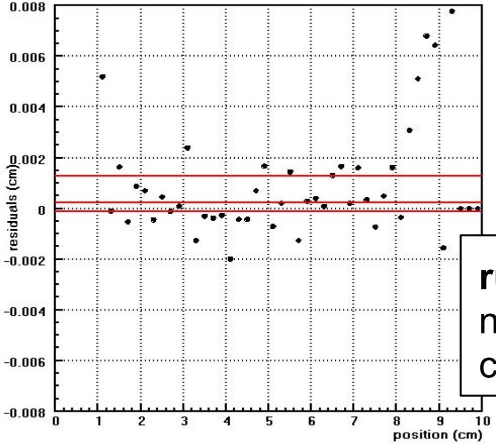
- ➔ **run 300304** - the last data run taken before the June setup upgrade → poor statistics (~40k good events only)
- ➔ **run 300269** - taken in the middle of May → richer statistics (~360k events)

In all the cases the algorithm gives the same results in terms of residuals distributions centering and spatial resolutions. In particular, the 5y central ASIC shift correction (slide 7) works well for all the runs

WHAT ABOUT NON ZERO SUPPRESSION MODE?



run 300066
no zero suppression
clock @ 1.25MHz



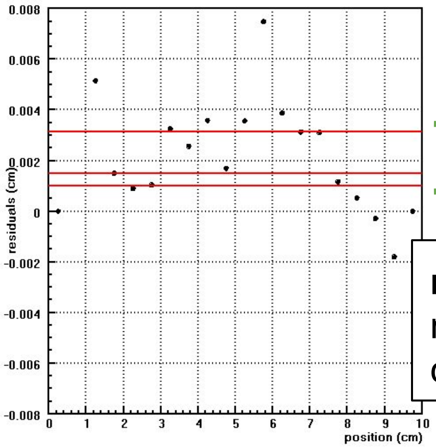
run 300112
no zero suppression
clock @ 2.50MHz

3y

At least part of the problem seems to be solved when running without zero suppression (and hence with common mode subtraction)

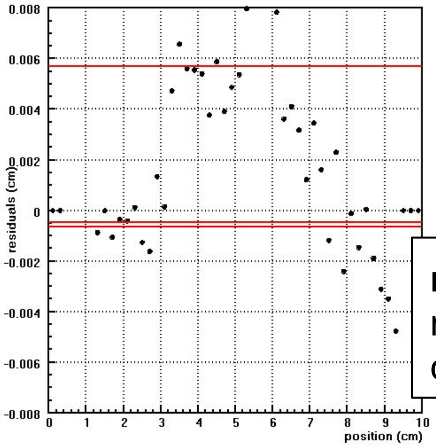
Statistics is these configurations is a bit poor for the purpose of this kind of investigation (~80k events in run 300112 and ~180k events in run 300066) → if needed (?) dedicated runs in this mode can be performed

WHAT ABOUT NON ZERO SUPPRESSION MODE?



~20um - with this statistics it's hard to perform a precise estimate, but the shift reduction is well evident

run 300066
no zero suppression
clock @ 1.25MHz



run 300112
no zero suppression
clock @ 2.50MHz

5y

Seems that this problem does not affect common mode → indeed running in non zero suppression mode while keeping the same clock (run 300112) hasn't any effect on the central ASIC behaviour

On the other hand, this disease seems to be affected by the clock frequency, which in our normal running condition (2.50MHz) is fine for side ASICs but too high for the central one → running at lower clock frequency (run 300066) seems to improve the situation

Nevertheless, we noticed that the net effect of this disease is a well known shift which can simply be corrected via software (slide 6) → it might be better to leave it be

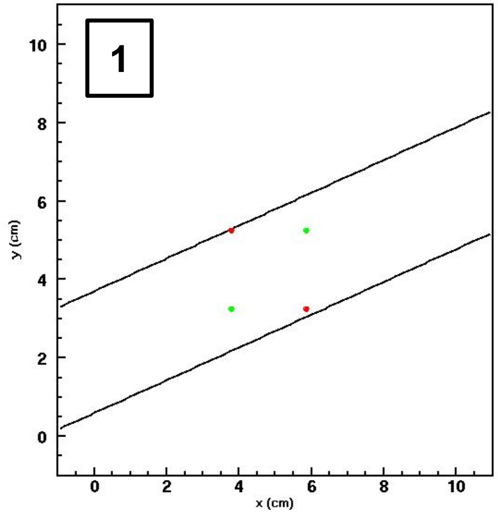
OUTLOOK

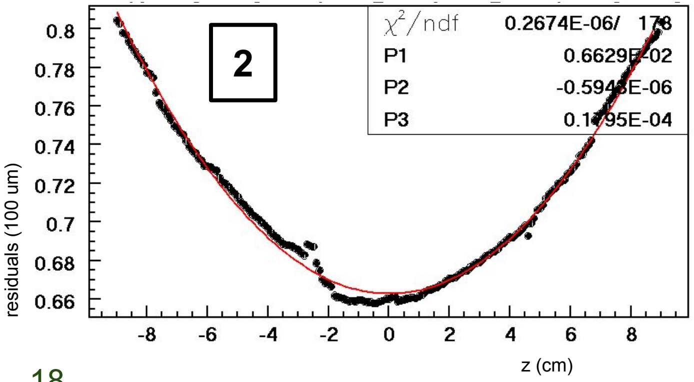
As soon as we decide some universal guidelines for performing the alignment, we will get to the right environment for the next steps of the tracking system tuning. In particular, in random order:

1) Inclusion of stereo layers as tracking layers *in toto*.
2) Implementation of some algorithm which will allow us to resolve double track events with stereo layers
3) If needed (?) implementation of corrections for tilts about *x* axis
4) Implementation of some algorithm for *z* positions check

I used to work a bit on software for points 2 (figure 1) and 4 (figure 2) → I think that these primitive algorithms can be upgraded and recycled for the present needs

Calorimeter(s) data are waiting to be studied too!





Thank you!