Alma Mater Studiorum - Università di Bologna

Vertex Track Perfomance Studies

Giacomo Ubaldi

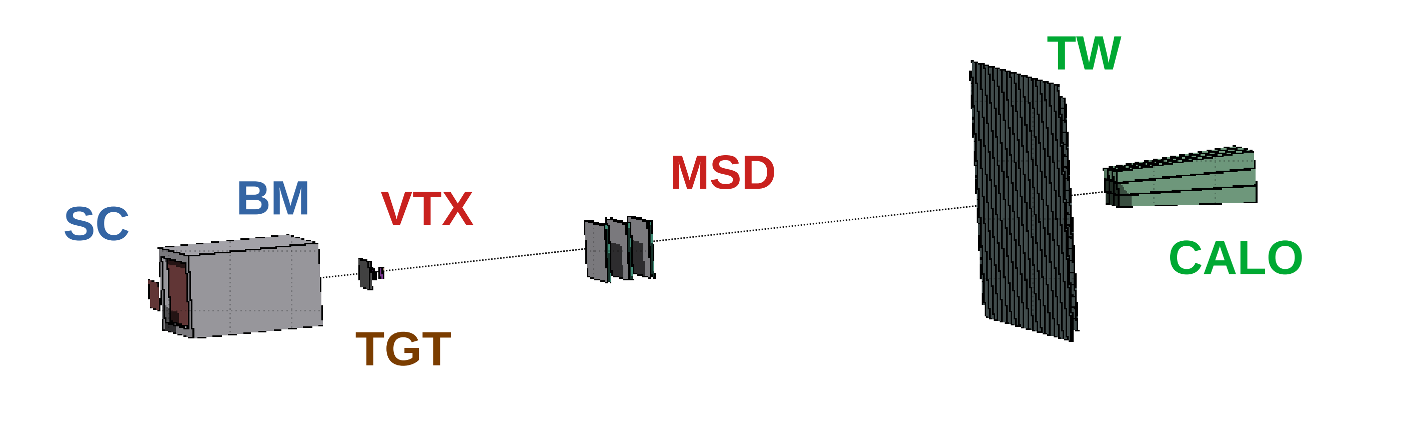
17/5/2023

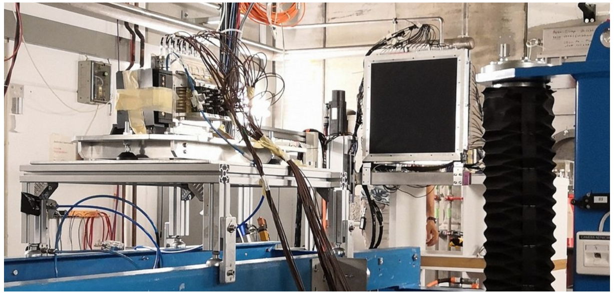
Cluster Reconstruction

- cluster position and its resolution
- fit of clusters in the different planes and its resolution
- residuals
- pulls
- Chi2 plots
- Studies on MC and first results on GSI2021 data

GSI 2021 Analysis

- Data-taking at GSI (Darmstadt, Germany) in 2021
- **^{16}O 400 MeV/u** on 5 mm **C target**
- Partial setup: no magnet, only one module of calorimeter





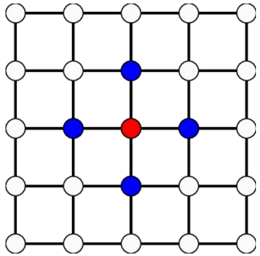
MC used Dataset:

- /gpfs_data/local/foot/Simulation/GSI2021_MC/16O_C_400_**2**_shoereg.root
  /gpfs_data/local/foot/Simulation/GSI2021_MC/16O_C_400_**3**_shoereg.root

Giacomo Ubaldi

Clusterization

- first neighbour search
- Two pixels are called first neighbours if they are contiguous in line or column
- search in an iterative way



- position of the cluster computing Center of Gravity
- sum in every pixel

$$x = \frac{\sum f_n x_n}{\sum f_n} \qquad y = \frac{\sum f_n y_n}{\sum f_n}$$

where f_n is the PulseHeight which is always 1

$$\sigma_x = \sqrt{\frac{\sum f_n (x_n - x)^2)}{\sum f_n}} \qquad \sigma_y = \sqrt{\frac{\sum f_n (y_n - y)^2)}{\sum f_n}}$$

“Performance of the reconstruction algorithms of the FIRST experiment pixel sensors vertex detector”
http://dx.doi.org/10.1016/j.nima.2014.08.024

Track reconstruction

- cluster positions on each sensor
- searching for all possible combinations between the clusters of the last plane and the previous plane
- 2) the micro-track is extrapolated to previous layers
- 3) a good cluster candidate is added to the micro-track
- repeat 2) and 3)

- At the end of the process, a final **linear fit** is done for the position of the track, composed of a fit in (x,z) and a fit in (y,z)

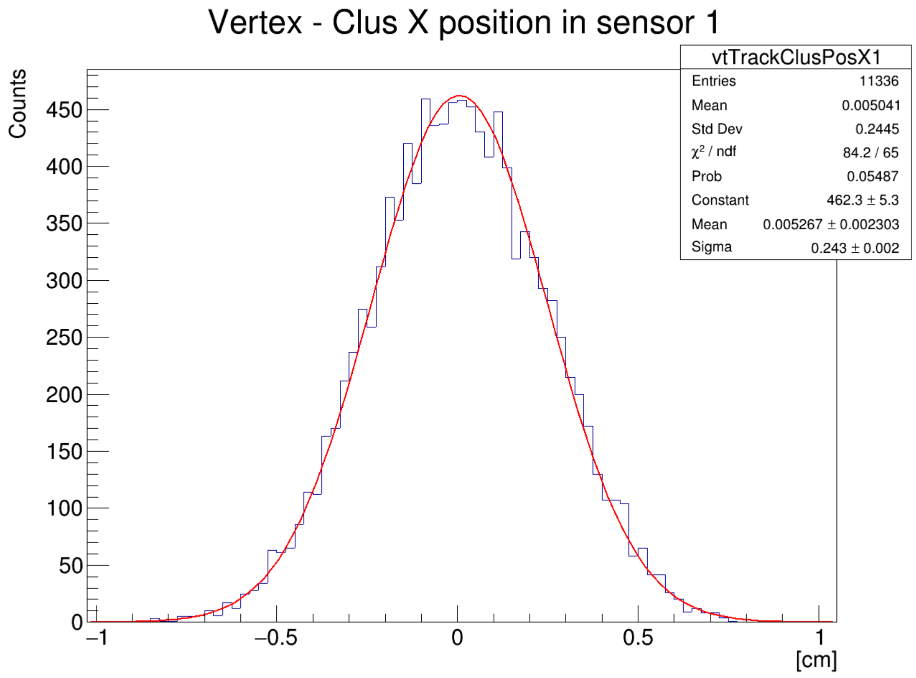
- I added the following function to study the performance of the fit in branch Ubaldi_studies TAGbaseTrack:: GetCovMatrix, GetChi2, GetPvalue

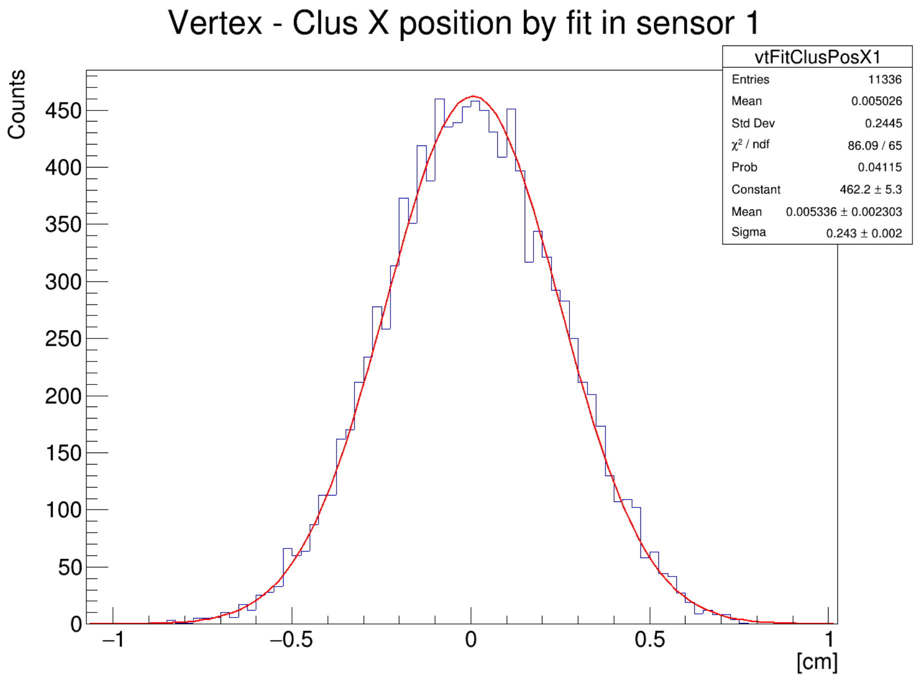
Position of the cluster

- Let’s start from the measurement of the position. As example, x coordinate of clusters for the sensor 1

From measurement: $x = \frac{\sum f_n x_n}{\sum f_n}$

From fit: $x = mz + q$



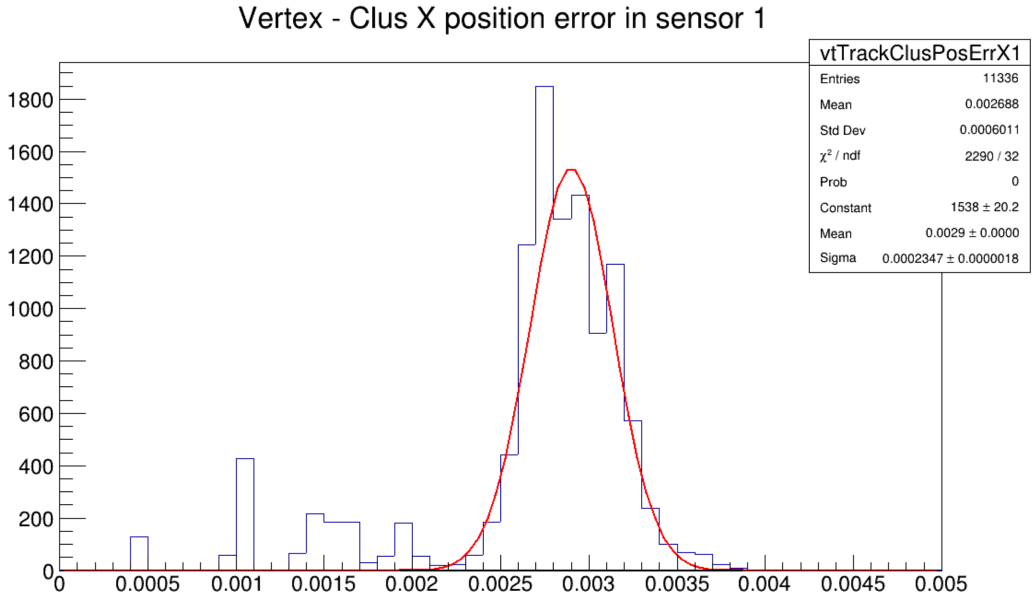


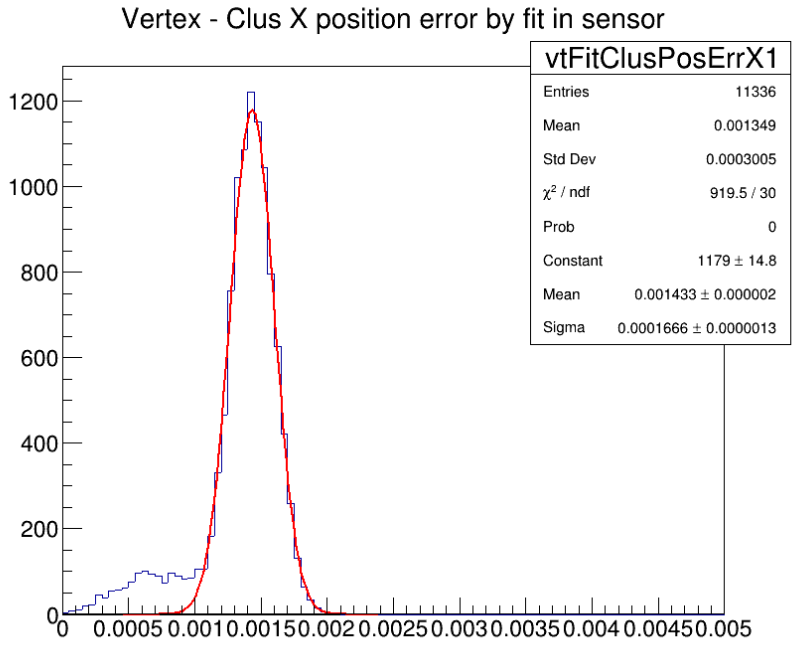
Resolution of the cluster

- The resolution associated to the position is shown. As example, x coordinate of clusters for the sensor 1

For the measurement: $\sigma_x = \sqrt{\frac{\sum f_n(x_n - x)^2)}{\sum f_n}}$

For the fit: $\sigma_x = \sqrt{(\frac{dx}{dm}\sigma_m)^2 + \frac{dx}{dq}\sigma_q)^2 + 2\frac{dx}{dm}\frac{dx}{dq}cov(\sigma_m, \sigma_q)}$





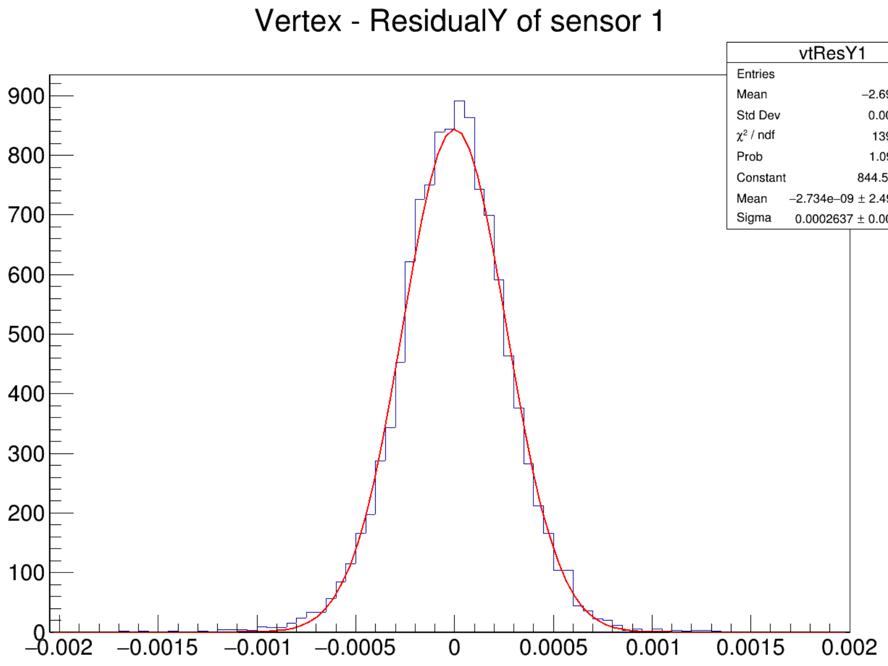
The mean of the error is about 30 μm. Is it overestiamated wrt what said in Frontiers Paper?

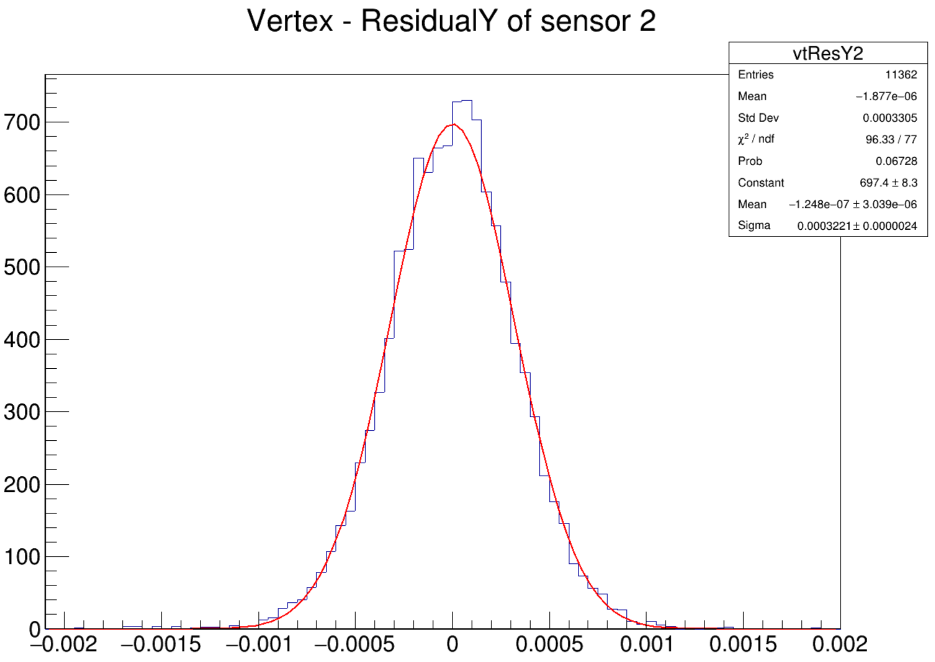
https://doi.org/10.3389/fphy.2020.568242

Residuals Y coordinate

$$r = y_{meas} - y_{fit}$$

- I measure the residual as in the formula. As example, y coordinate of clusters for the sensor 1





Is the mean of the residual the spatial resolution of the VT detector as reported in Frontier paper?

https://doi.org/10.3389/fphy.2020.568242 http://dx.doi.org/10.1016/j.nima.2014.08.024

In the following slide, the pull in x for all the sensors

Then the pull

$$g_c = \frac{\tau_f - \tau_c}{\sqrt{\sigma_c^2 - \sigma_f^2}} \tag{10}$$

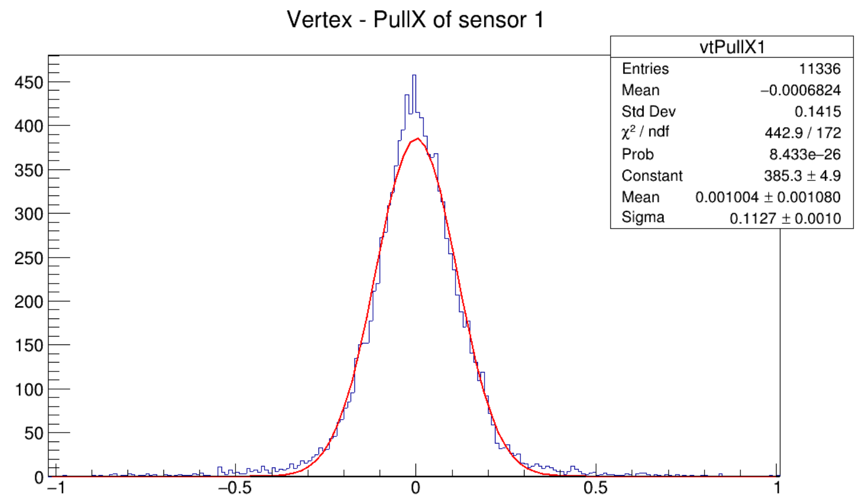
is usually a standard Gaussian. The denominator of the expression for g_c may at first sight look a bit surprising, but it is simply the error on the numerator, taking into account the correlation between the errors in the fit result τ_f and the measure τ_c.

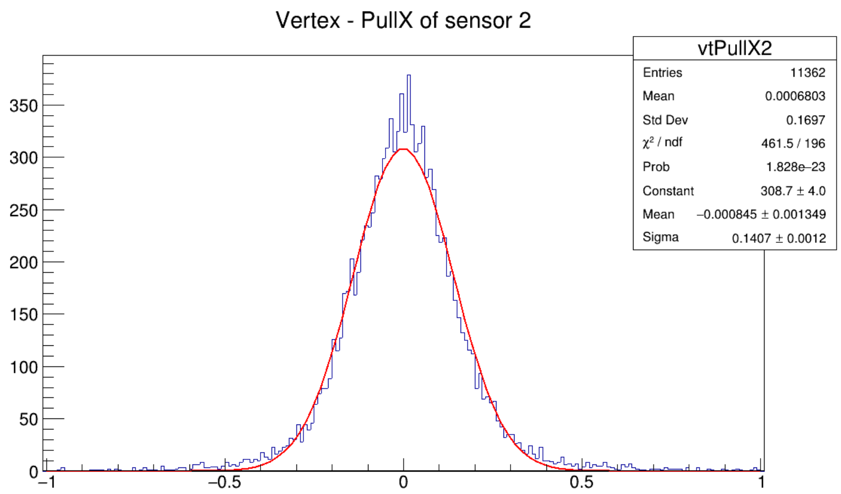
f stands for the fit
c stands for the measurement

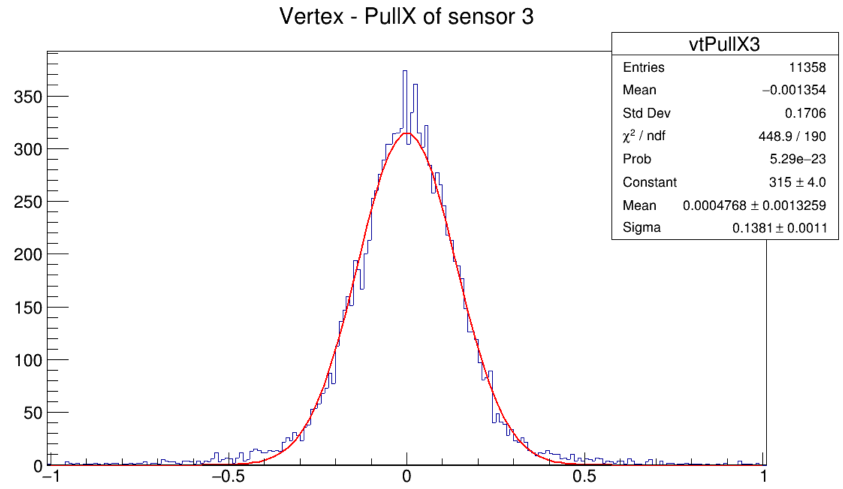
https://lucdemortier.github.io/assets/papers/cdf5776_pulls.pdf

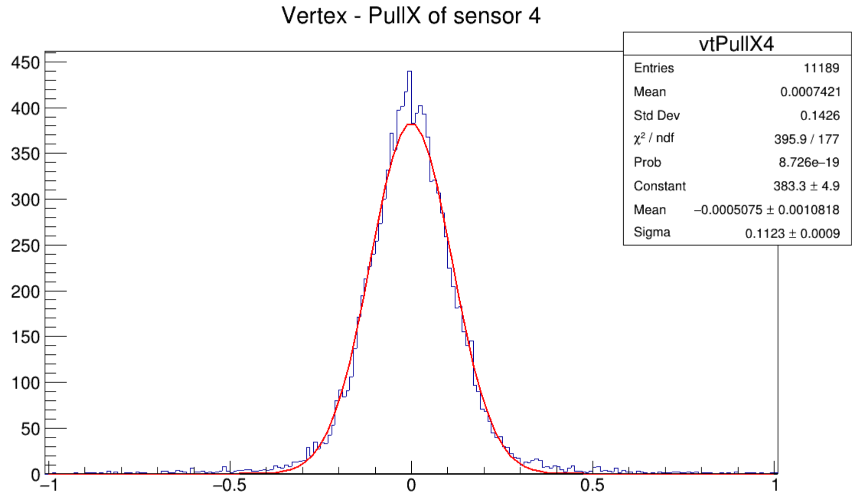
Pull X coordinate

$$g_c = \frac{\tau_f - \tau_c}{\sqrt{\sigma_c^2 - \sigma_f^2}}$$



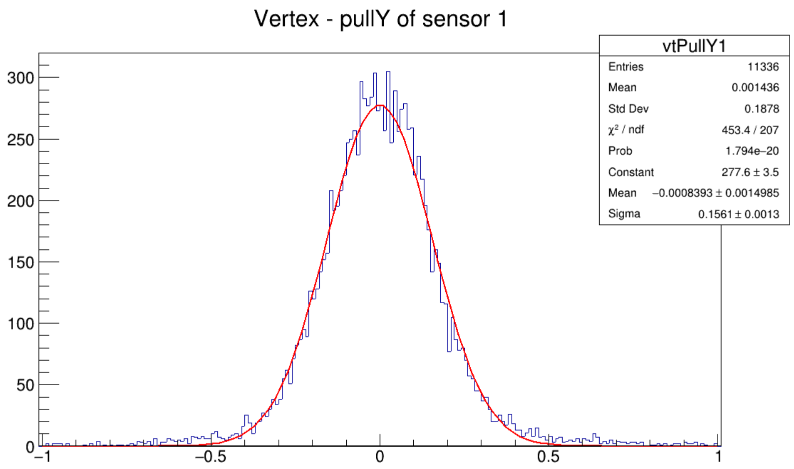


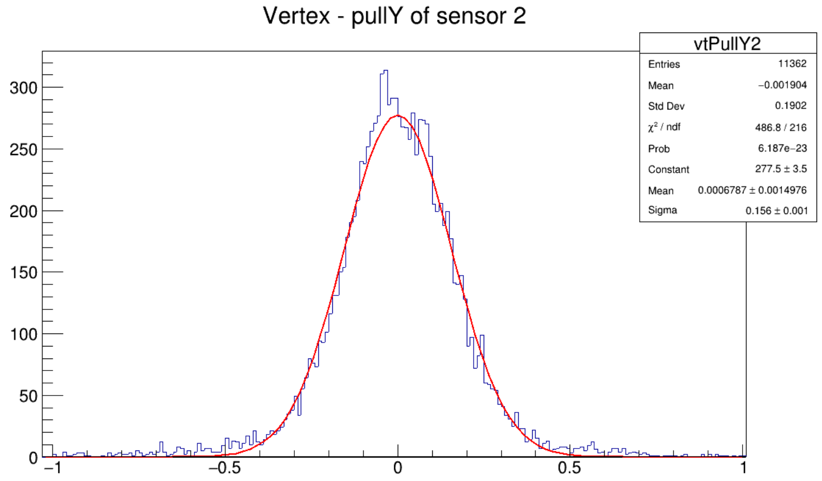


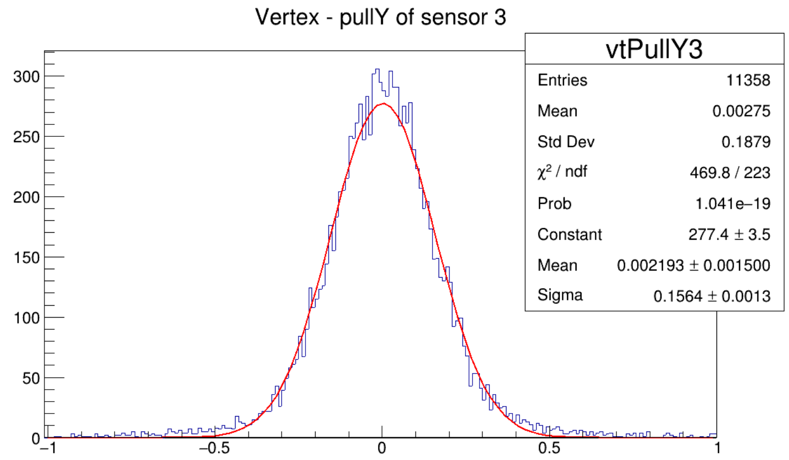


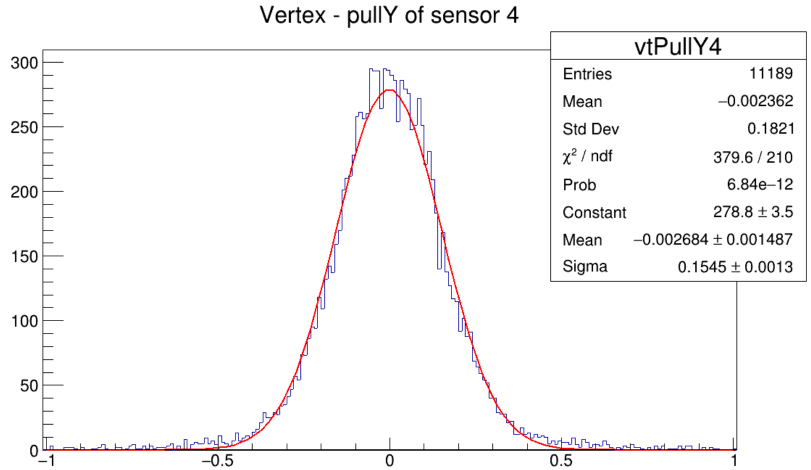
Pull Y coordinate

$$g_c = \frac{\tau_f - \tau_c}{\sqrt{\sigma_c^2 - \sigma_f^2}}$$





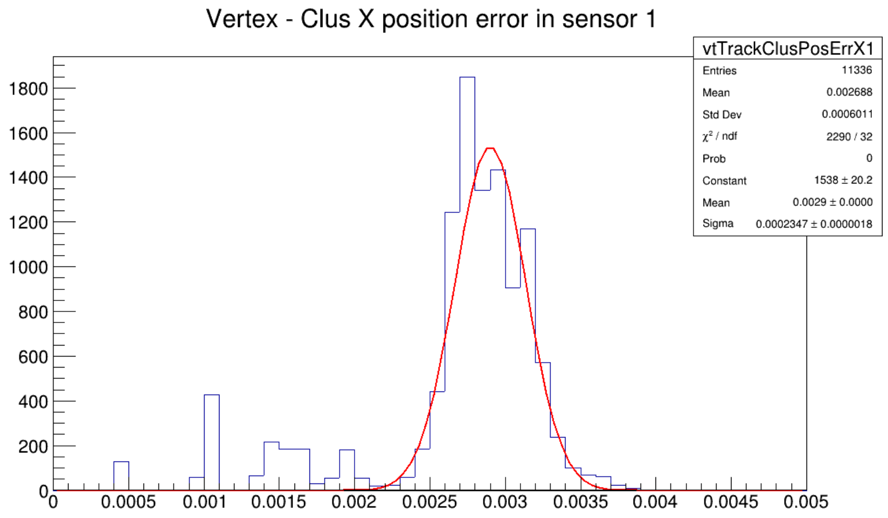




Observations

- In the best condition, a mean around 0 and a σ_{pull} around 1 are expected.
- In this case $\sigma_{pull} \sim 0.2 \rightarrow$ **$\sigma_{measurement}$ is overestimated**

Resolution of the cluster



The $\sigma_{measurement}$ I am using now is the mean of this distribution:

$$\sigma_x = \sqrt{\frac{\sum f_n (x_n - x)^2)}{\sum f_n}}$$

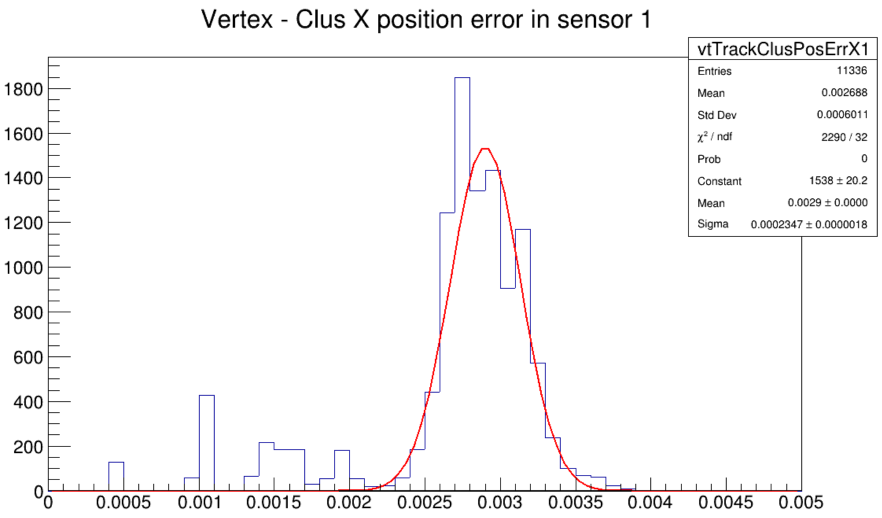
Since the estimator of the position is a mean,

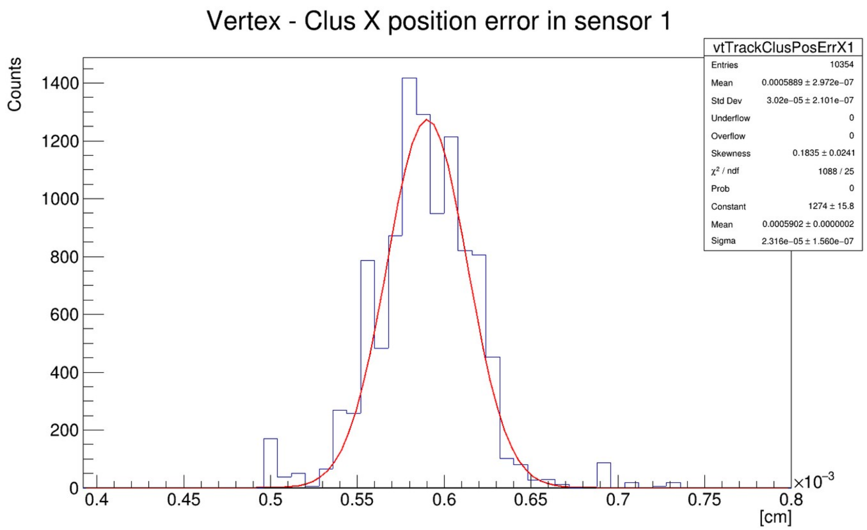
I want to use as $\sigma_{measurement}$ the one associated to the mean:

$$\boxed{\sigma_\mu = \frac{\sigma_x}{\sqrt{n}}}$$

Resolution of the cluster

- I run again my data with the new $\sigma_{measurement}$. Ex. resolution in x of sensor 1





$$\sigma_x = \sqrt{\frac{\sum f_n (x_n - x)^2)}{\sum f_n}}$$

~ 30 µm

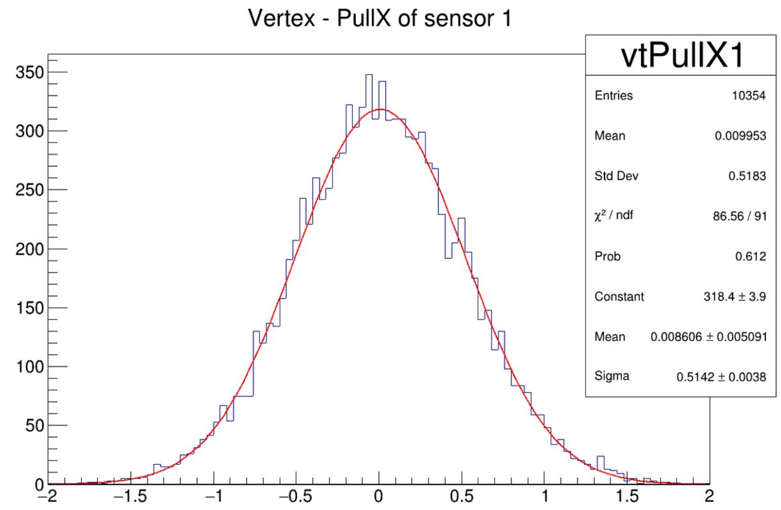
$$\sigma_\mu = \frac{\sigma_x}{\sqrt{n}}$$

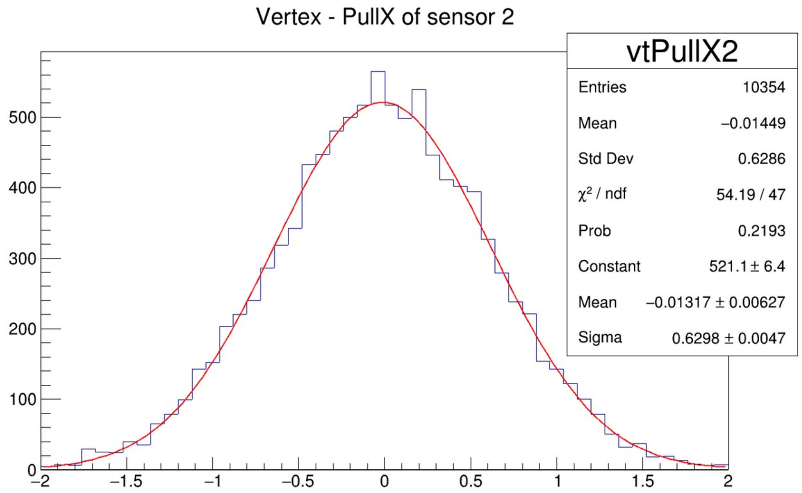
~ 6 µm

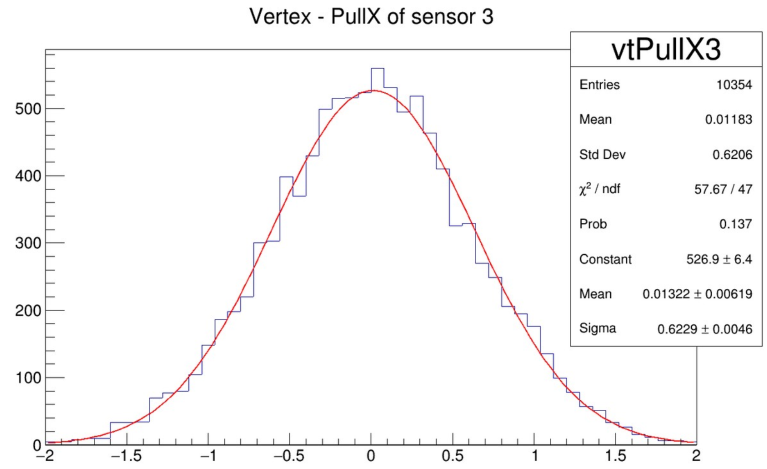
- Omitting all the steps already seen, I measure the new pulls in the next slide

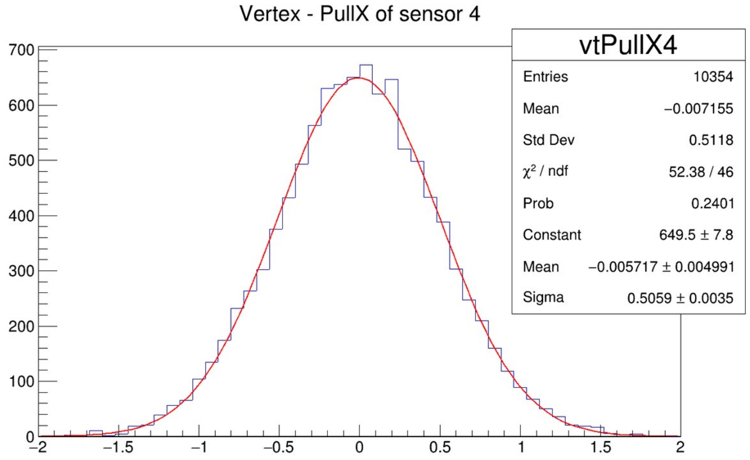
Pull X coordinate

$$g_c = \frac{\tau_f - \tau_c}{\sqrt{\sigma_c^2 - \sigma_f^2}}$$



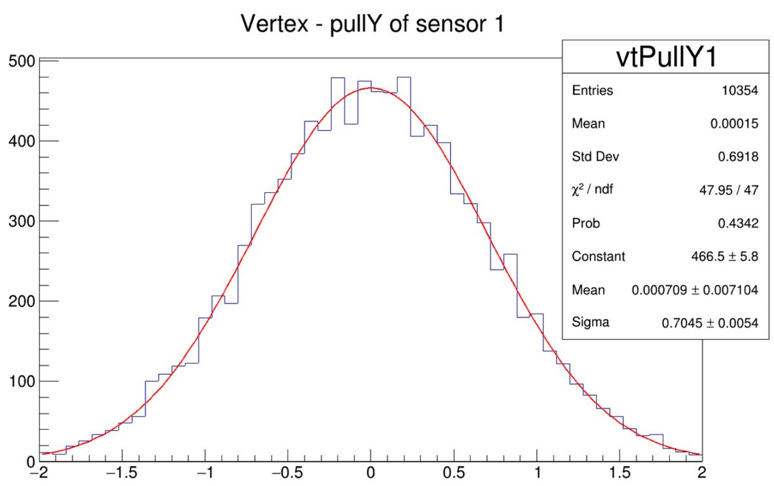


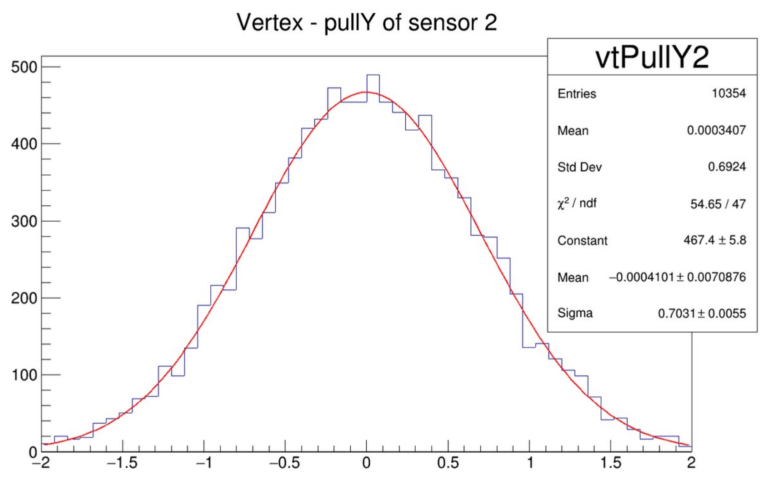


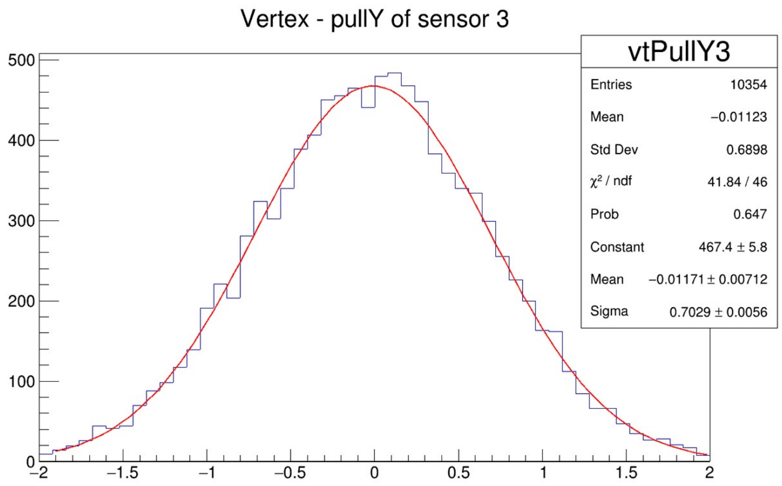


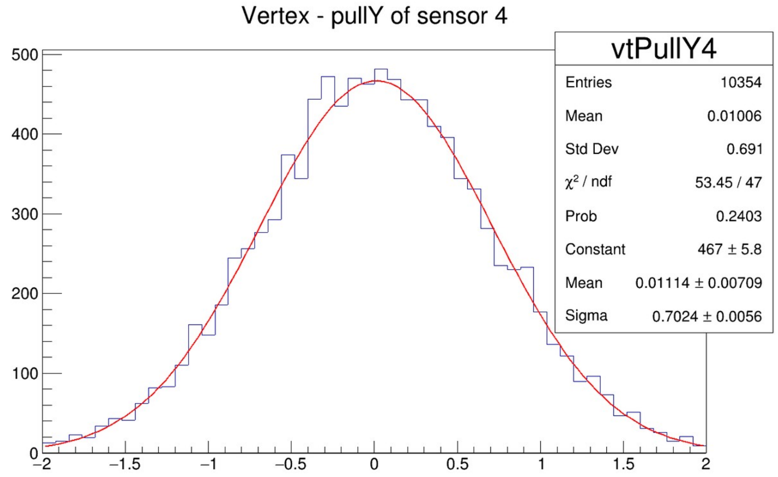
Pull Y coordinate

$$p = \frac{x_i - f(z_i)}{\sqrt{\sigma^2_{meas} - \sigma^2_{fit}}}$$



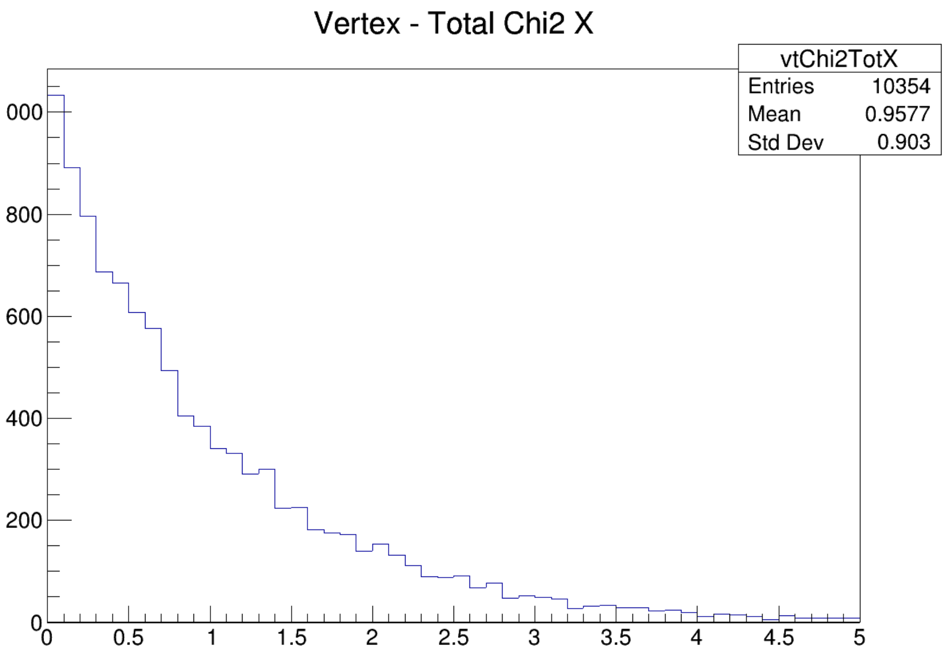


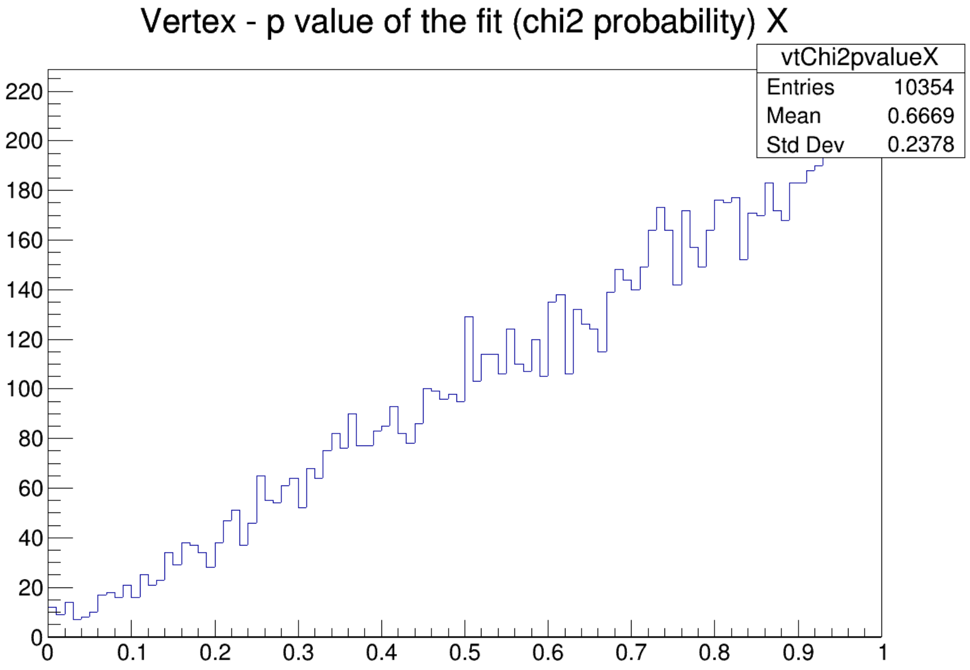




Chi2 Test X coordinate

The Chi2 distribution of every fitted track and its pvalue is shown.
The mean of the distribution should be ~ 2 (dof = 4 (number of points) / 2 (m,q parameters)
The p value distribution should be uniform





Observations

- went from 0.2 to 0.7 → error still overestimated
- Chi2 distribution show the fitting is still not perfect

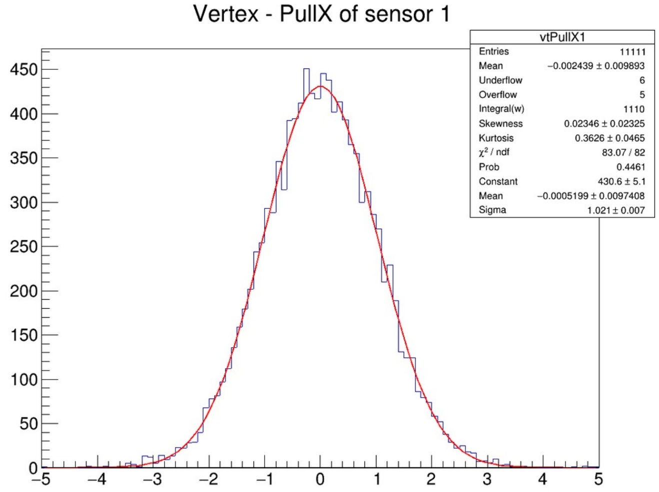
- let's multiply $\sigma_{measurement}$ by 1/ sqrt(2) (which is 0.7) in order to obtain σ_{pull} close to 1

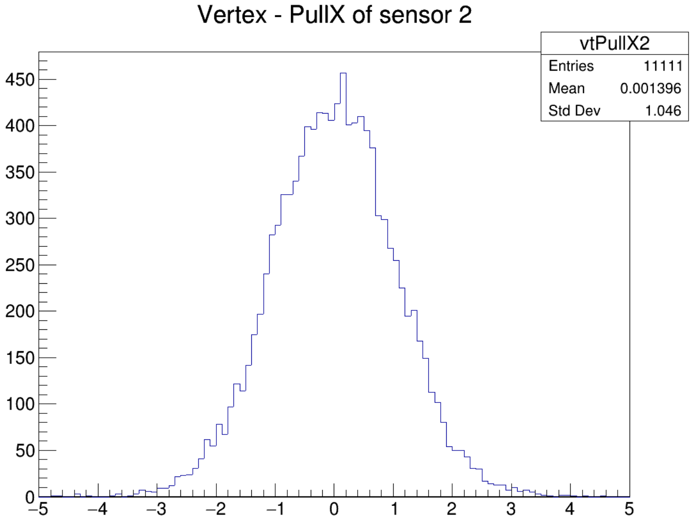
So I go from $\sigma_x = \sqrt{\frac{\sum f_n (x_n - x)^2)}{\sum f_n}}$ to $\boxed{\sigma_{\mu'} = \frac{\sigma_x}{\sqrt{2n}}}$

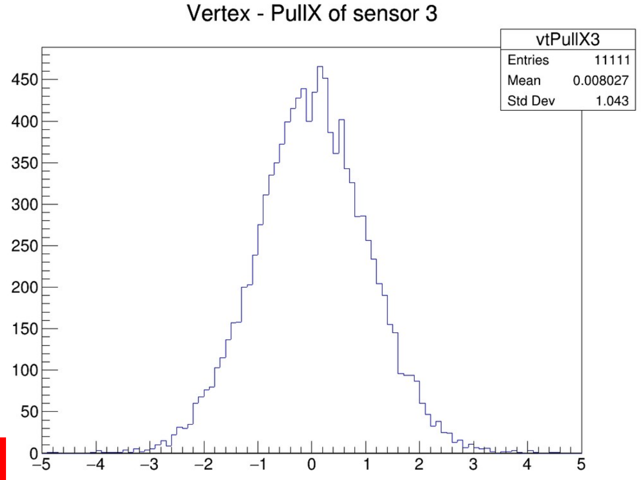
- In the following slide, I show the new pulls (omitting al the steps)

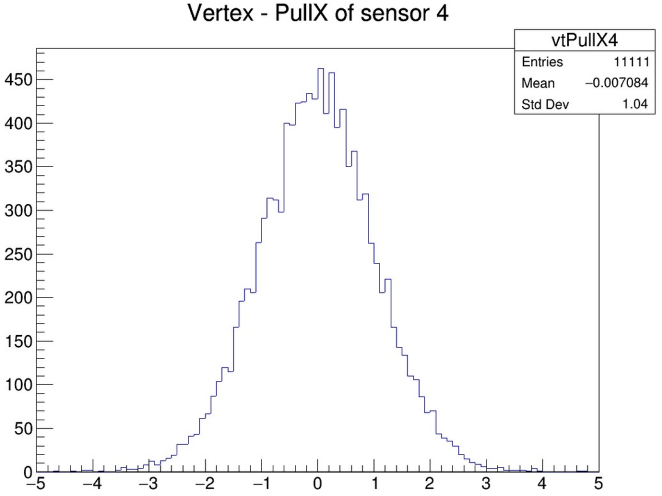
Pull X coordinate

$$g_c = \frac{\tau_f - \tau_c}{\sqrt{\sigma_c^2 - \sigma_f^2}}$$



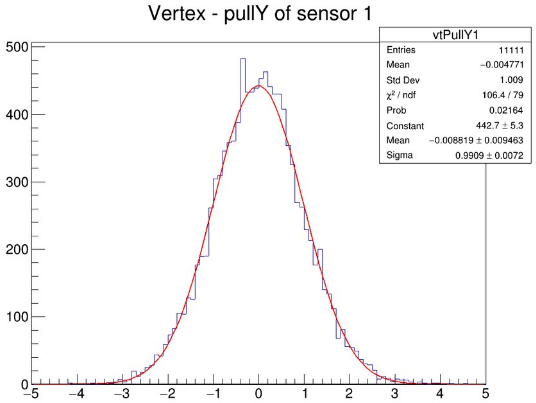


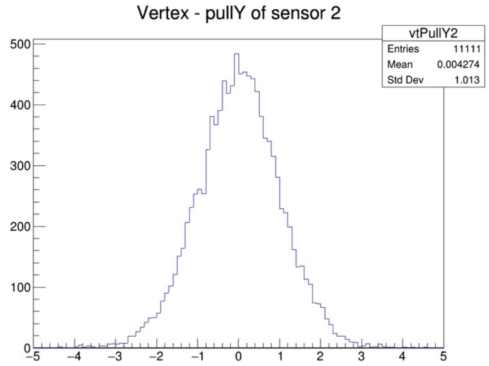


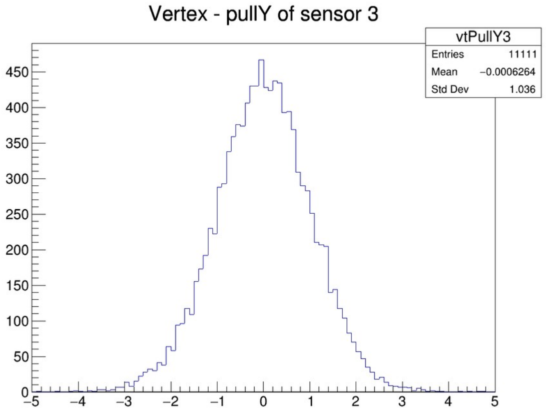


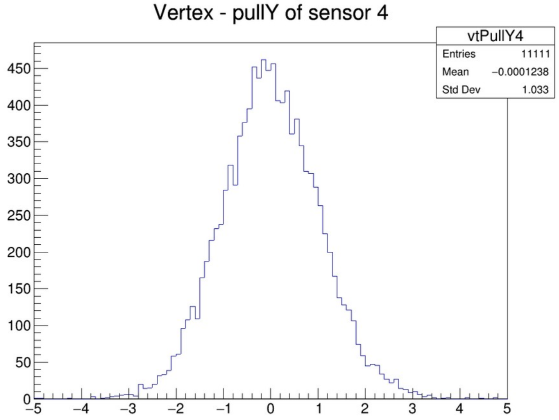
Pull Y coordinate

$$g_c = \frac{\tau_f - \tau_c}{\sqrt{\sigma_c^2 - \sigma_f^2}}$$

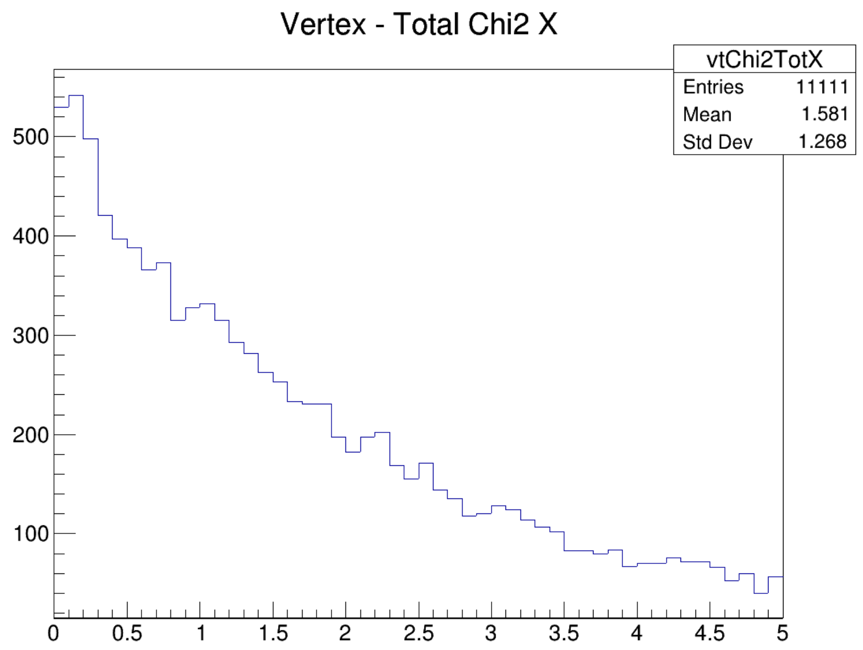


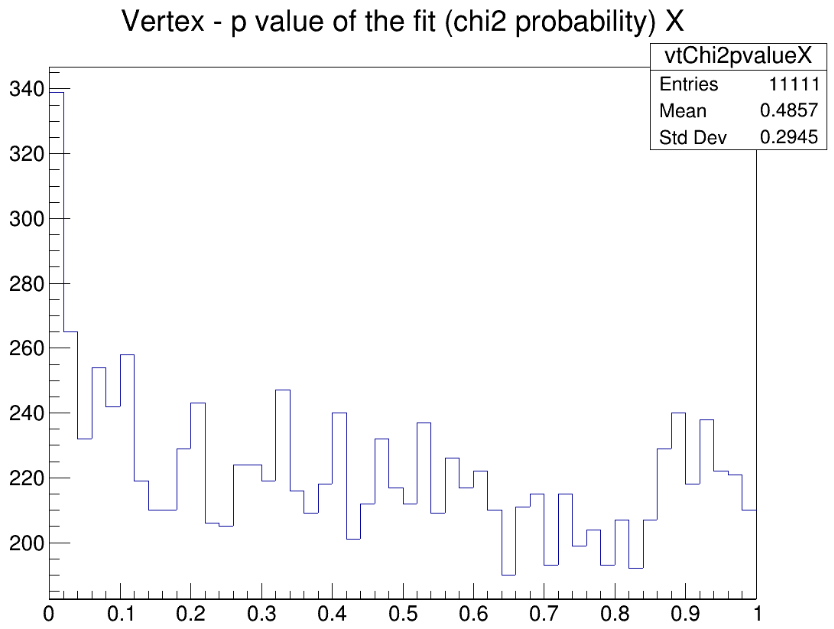






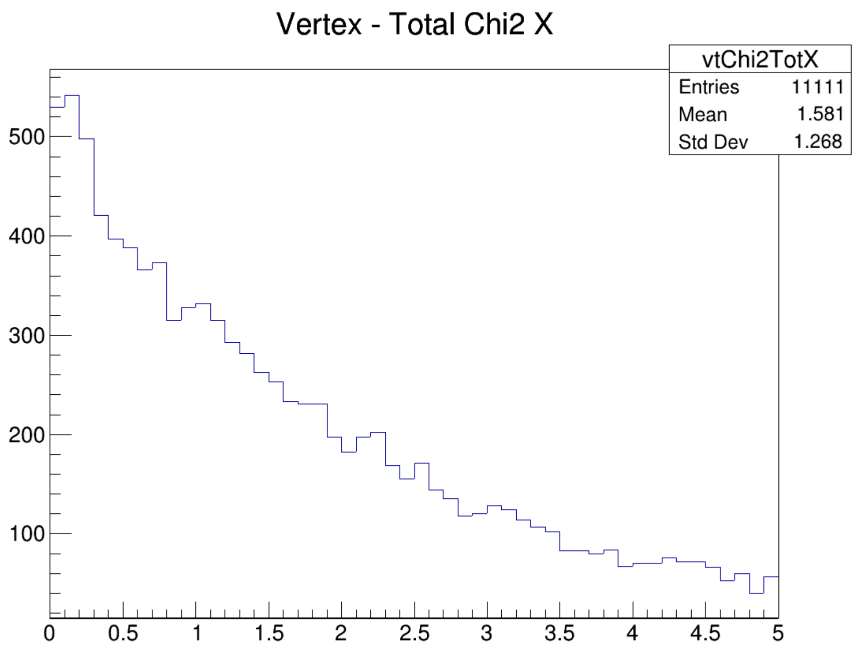
Chi2 Test X coordinate

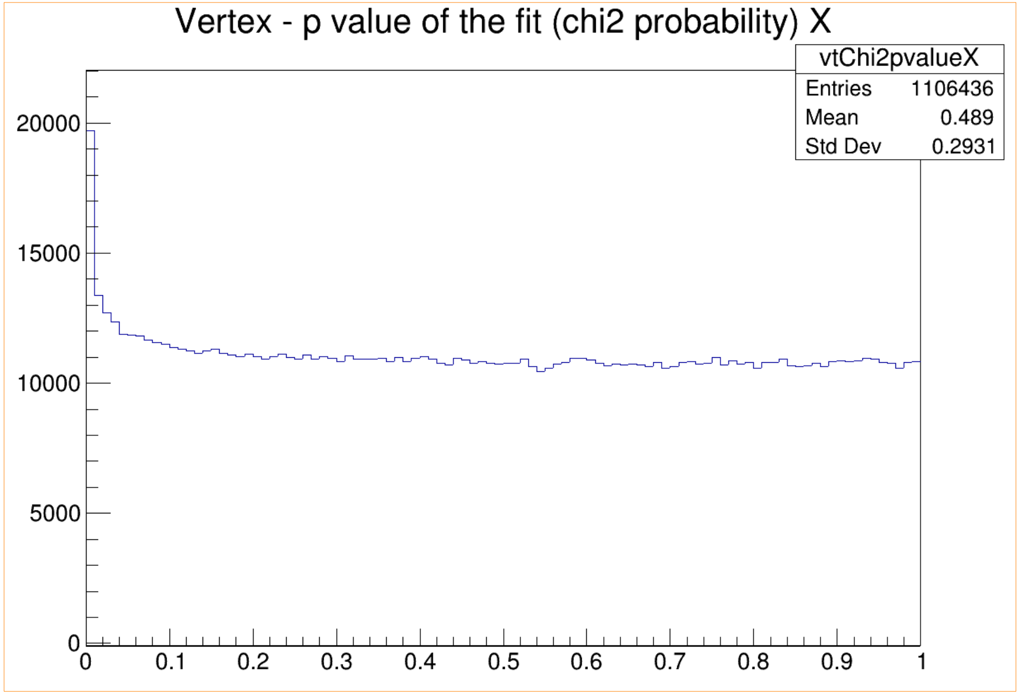




The p-value distribution is now much more uniform

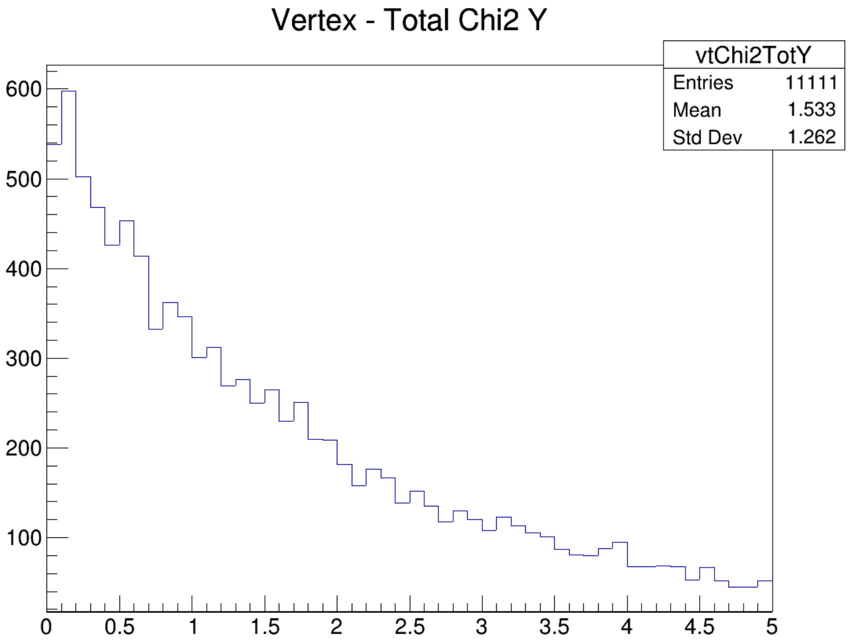
Chi2 Test X coordinate

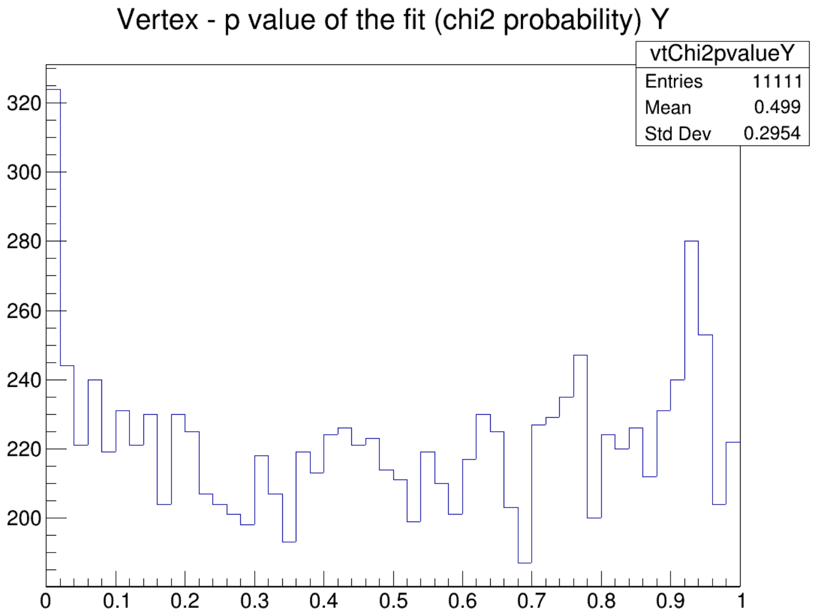




The p-value distribution is now much more uniform

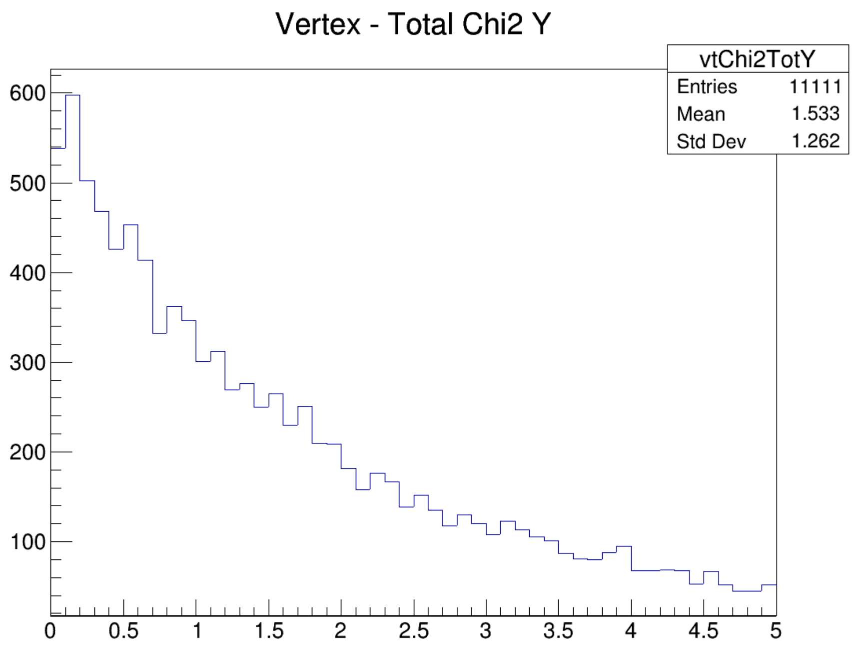
Chi2 Test Y coordinate

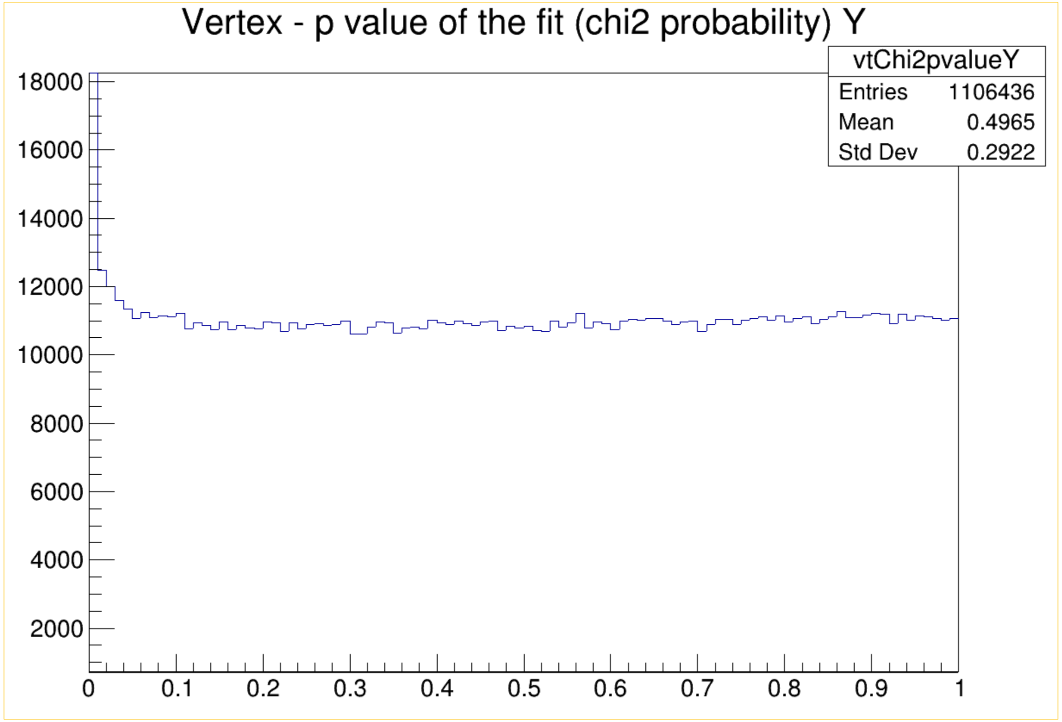




The p-value distribution is now much more uniform

Chi2 Test Y coordinate





The p-value distribution is now much more uniform

Observation

- According from pulls, the resolution associated to the cluster position is overestimated.
- Things become better if I use the following resolution associated to the position

$$x = \frac{\sum f_n x_n}{\sum f_n} \qquad \sigma_{\mu'} = \frac{\sigma_x}{\sqrt{2n}}$$

where $$\sigma_x = \sqrt{\frac{\sum f_n (x_n - x)^2)}{\sum f_n}}$$

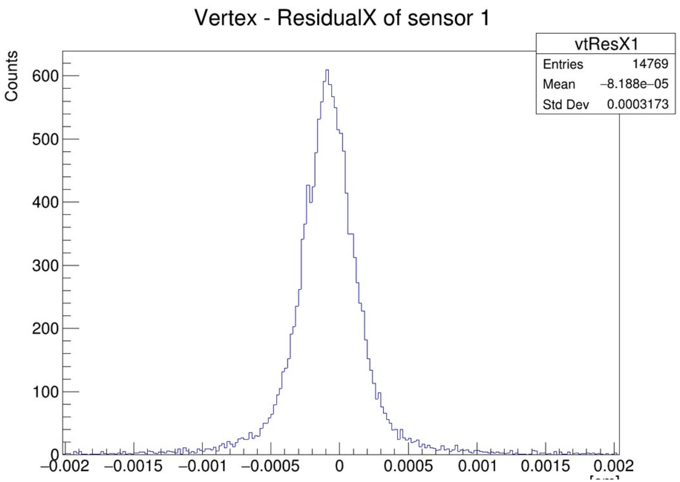
- How it can be explained physically?

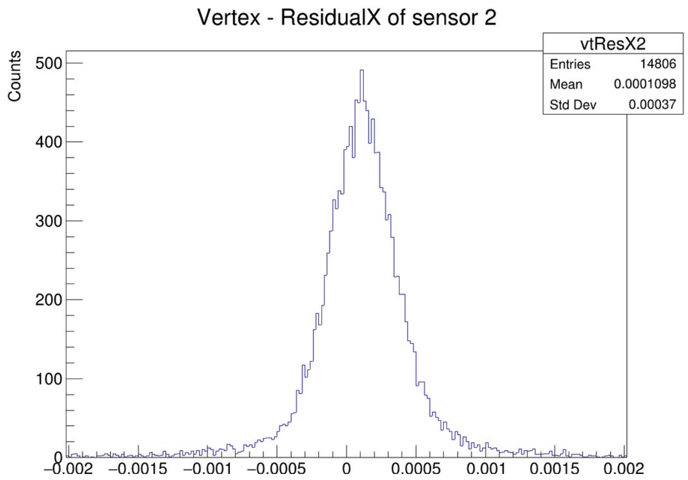
- Is actually the mean the best estimator of the position of the cluster?
- How does it changes the performance of the reconstructed track?
  next time: track efficiency, purity etc

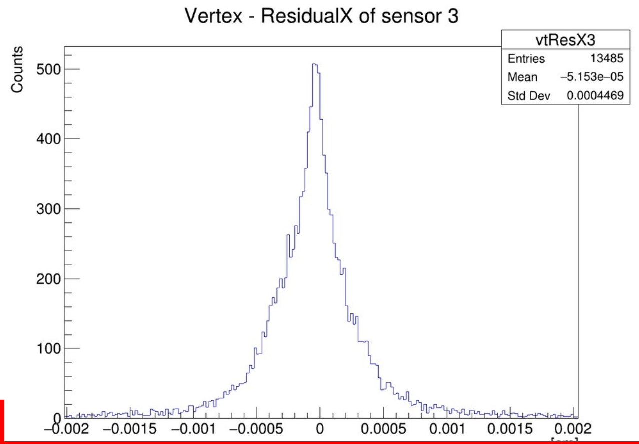
Observations

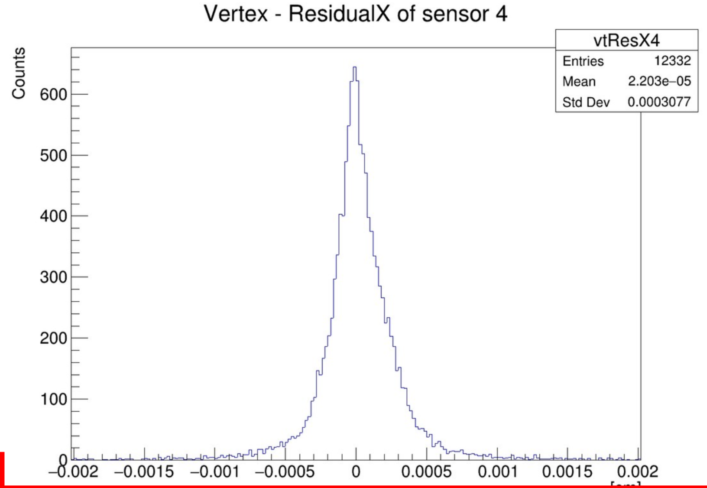
back up slides

run 4306

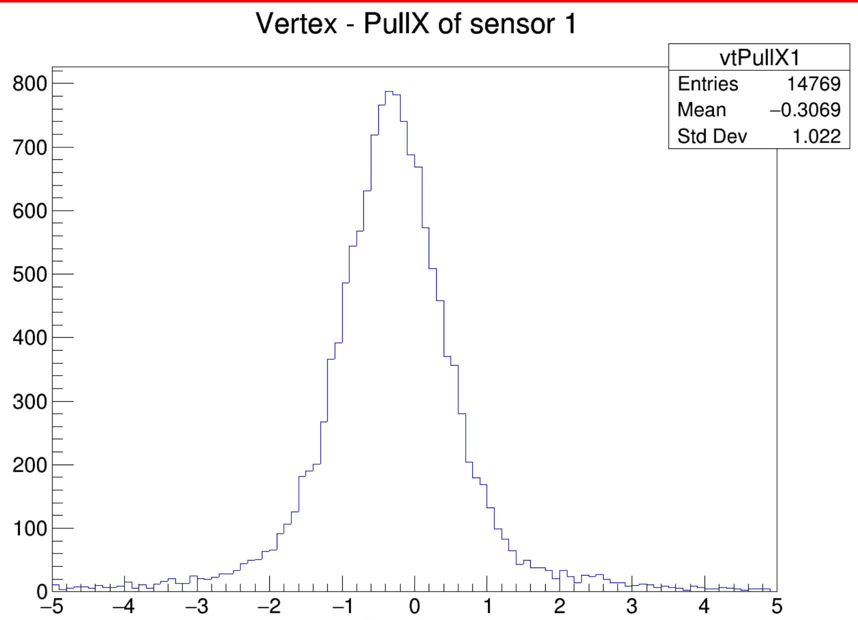

Vertex - ResidualX of sensor 1
vtResX1
Entries 14769
Mean −8.188e−05
Std Dev 0.0003173
Counts


Vertex - ResidualX of sensor 2
vtResX2
Entries 14806
Mean 0.0001098
Std Dev 0.00037
Counts

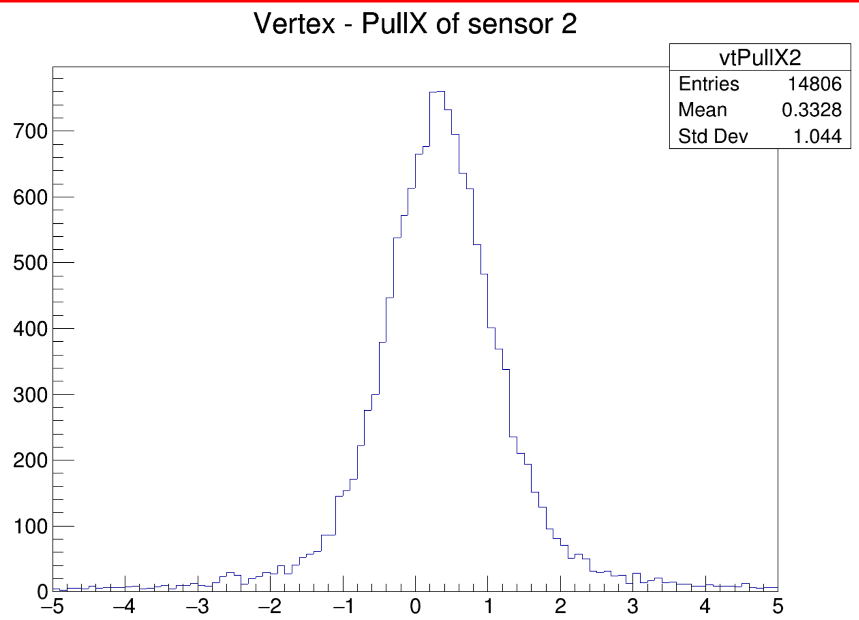

Vertex - ResidualX of sensor 3
vtResX3
Entries 13485
Mean −5.153e−05
Std Dev 0.0004469
Counts


Vertex - ResidualX of sensor 4
vtResX4
Entries 12332
Mean 2.203e−05
Std Dev 0.0003077
Counts

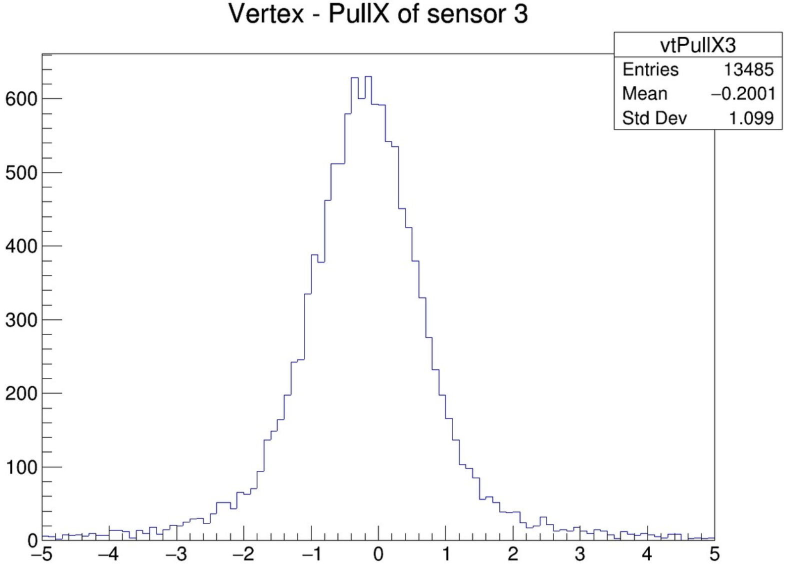
run 4306



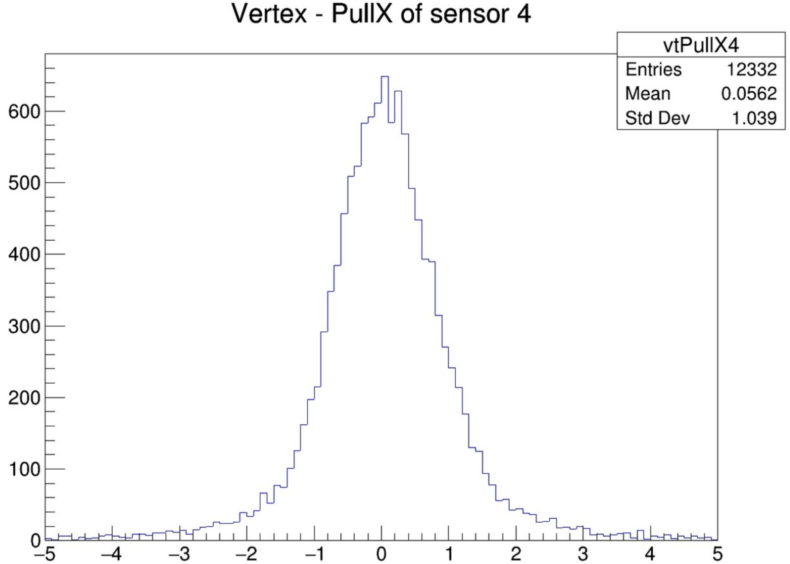
Vertex - PullX of sensor 1
vtPullX1
Entries 14769
Mean −0.3069
Std Dev 1.022



Vertex - PullX of sensor 2
vtPullX2
Entries 14806
Mean 0.3328
Std Dev 1.044

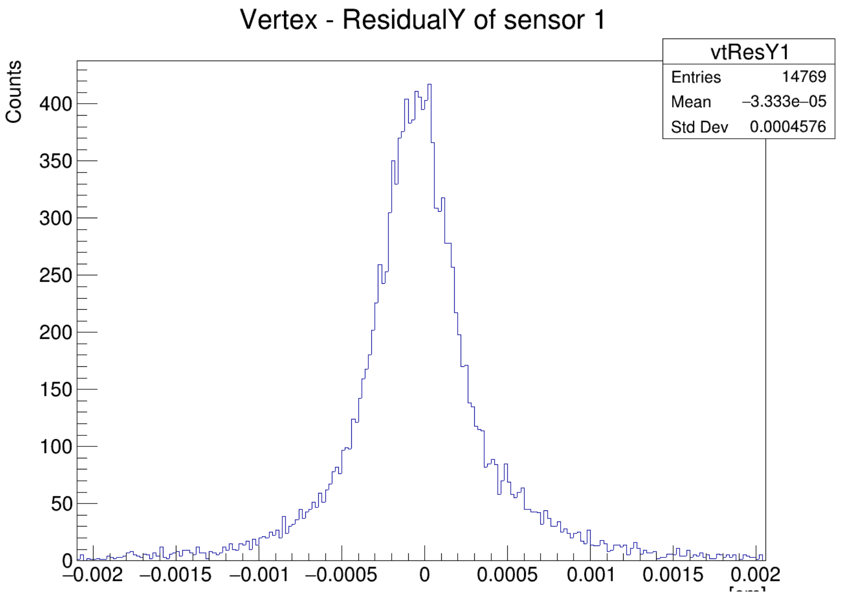


Vertex - PullX of sensor 3
vtPullX3
Entries 13485
Mean −0.2001
Std Dev 1.099

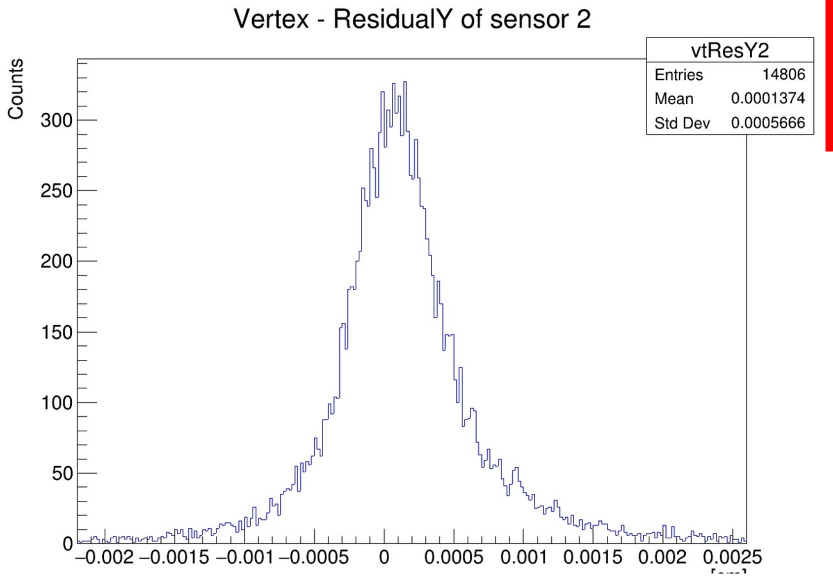


Vertex - PullX of sensor 4
vtPullX4
Entries 12332
Mean 0.0562
Std Dev 1.039

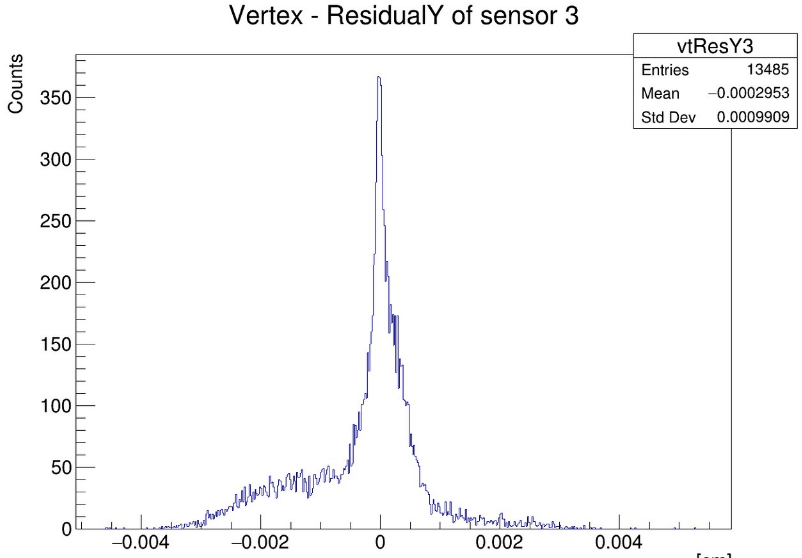
run 4306



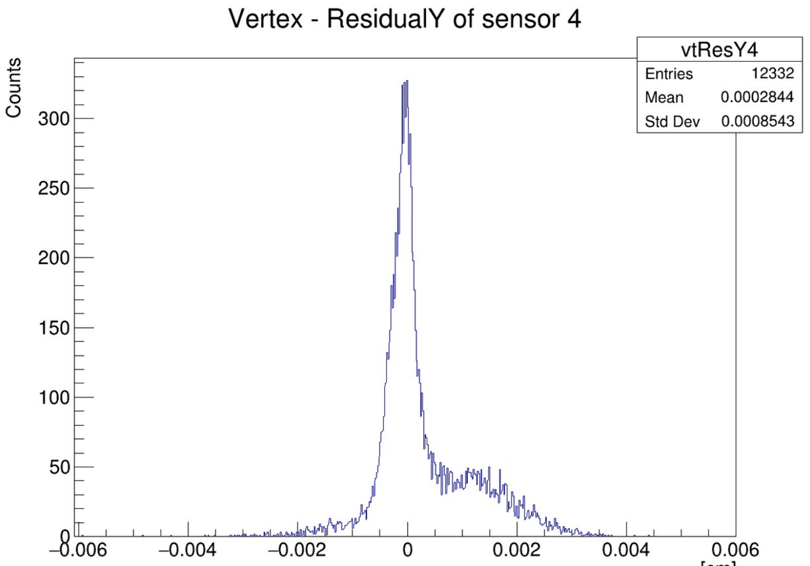
Vertex - ResidualY of sensor 1
vtResY1
Entries 14769
Mean −3.333e−05
Std Dev 0.0004576
Counts



Vertex - ResidualY of sensor 2
vtResY2
Entries 14806
Mean 0.0001374
Std Dev 0.0005666
Counts

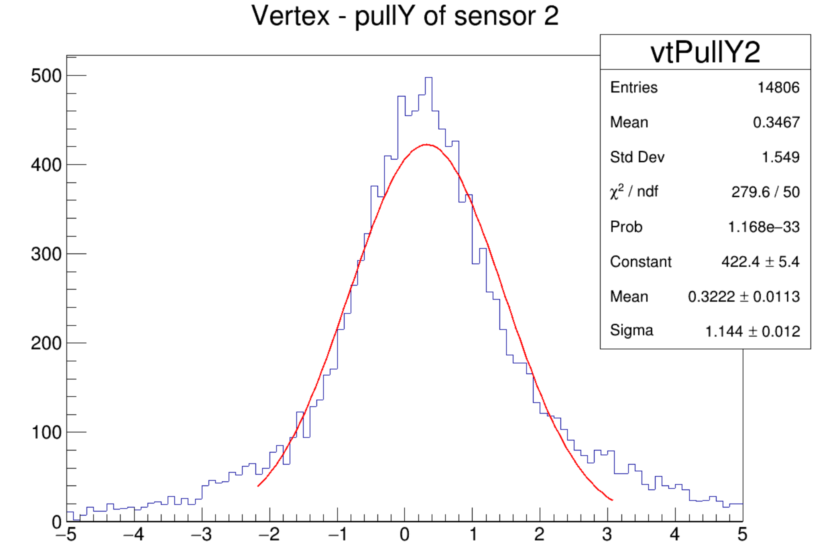


Vertex - ResidualY of sensor 3
vtResY3
Entries 13485
Mean −0.0002953
Std Dev 0.0009909
Counts

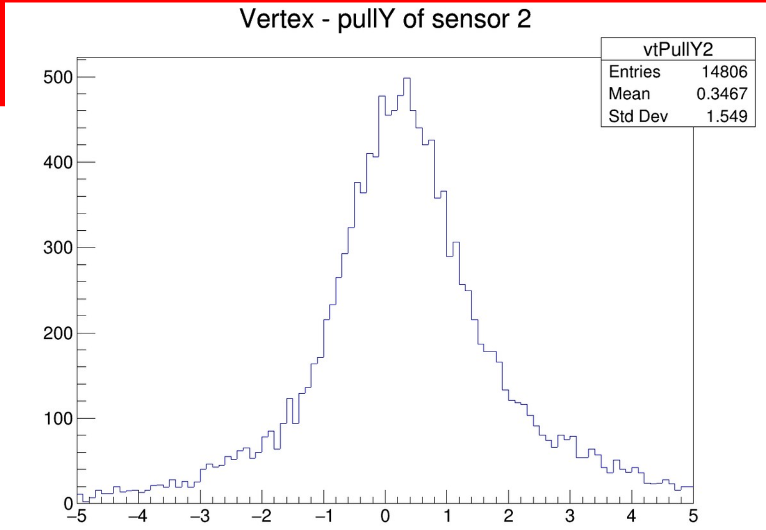


Vertex - ResidualY of sensor 4
vtResY4
Entries 12332
Mean 0.0002844
Std Dev 0.0008543
Counts

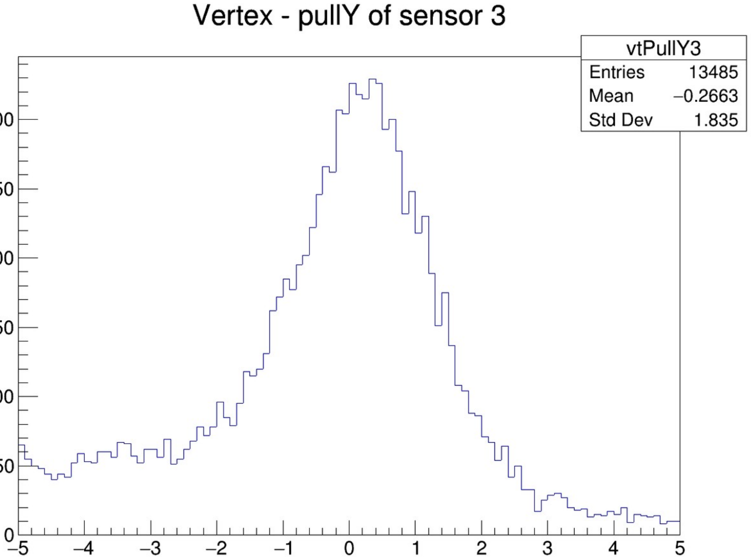
run 4306



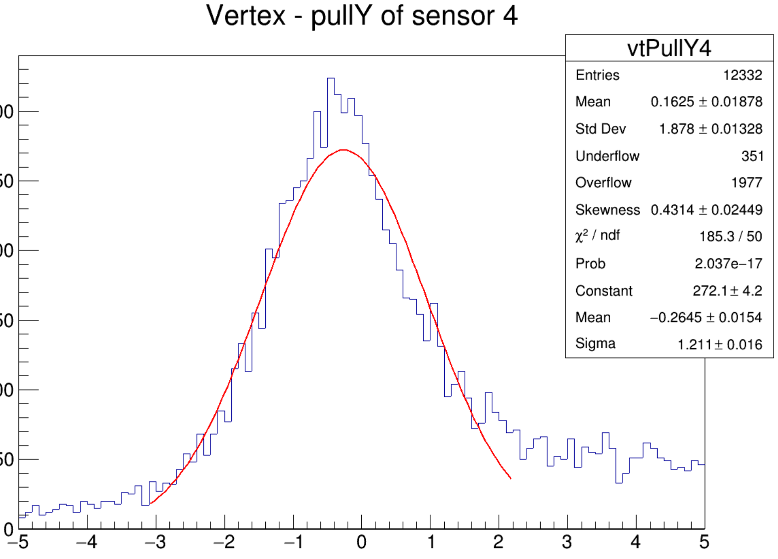
Vertex - pullY of sensor 2
vtPullY2
Entries 14806
Mean 0.3467
Std Dev 1.549
χ² / ndf 279.6 / 50
Prob 1.168e−33
Constant 422.4 ± 5.4
Mean 0.3222 ± 0.0113
Sigma 1.144 ± 0.012



Vertex - pullY of sensor 2
vtPullY2
Entries 14806
Mean 0.3467
Std Dev 1.549

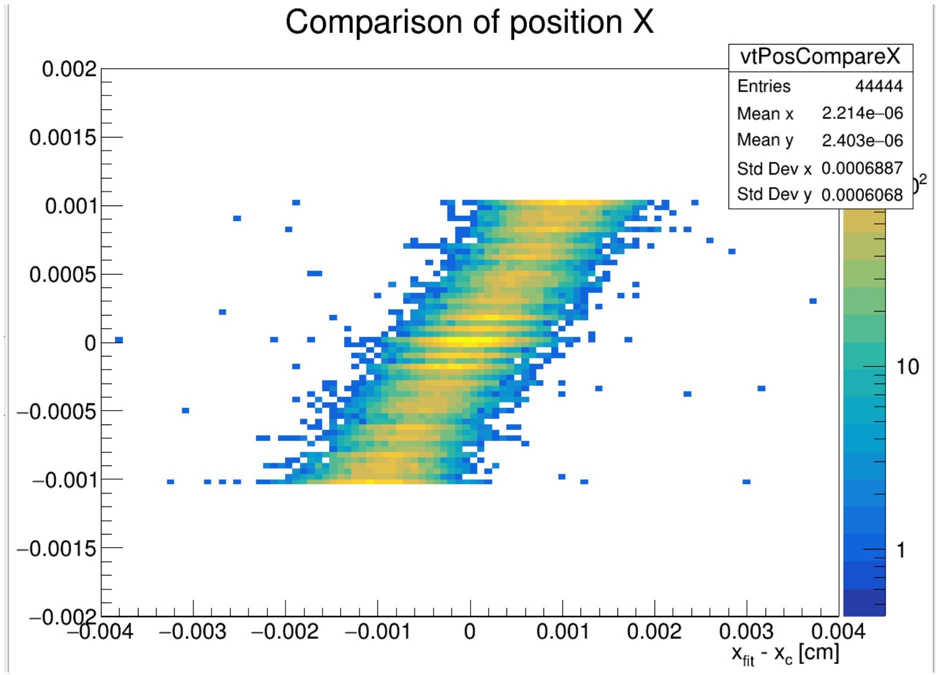


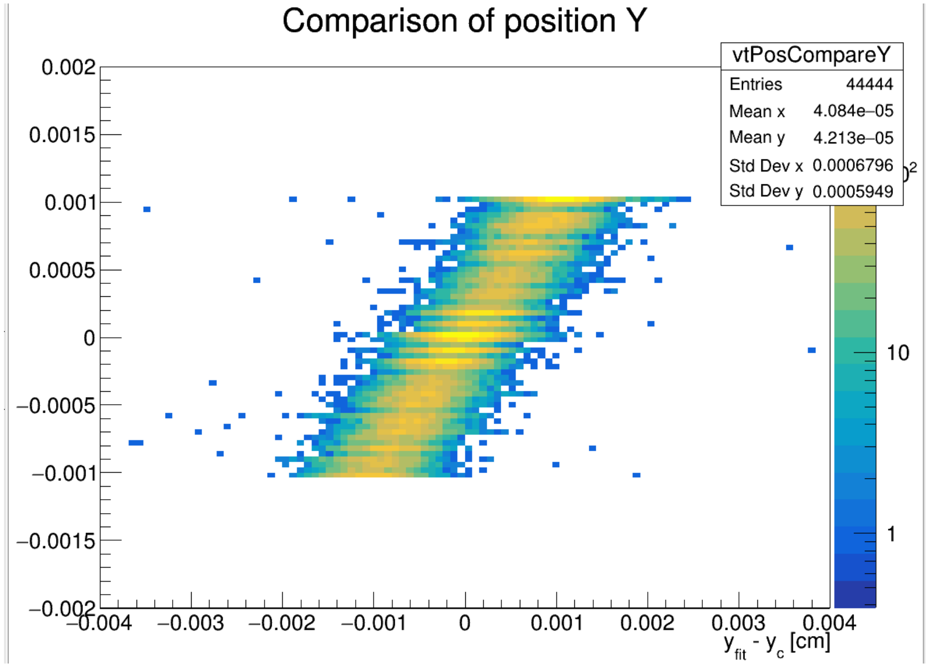
Vertex - pullY of sensor 3
vtPullY3
Entries 13485
Mean −0.2663
Std Dev 1.835



Vertex - pullY of sensor 4
vtPullY4
Entries 12332
Mean 0.1625 ± 0.01878
Std Dev 1.878 ± 0.01328
Underflow 351
Overflow 1977
Skewness 0.4314 ± 0.02449
χ² / ndf 185.3 / 50
Prob 2.037e−17
Constant 272.1 ± 4.2
Mean −0.2645 ± 0.0154
Sigma 1.211 ± 0.016

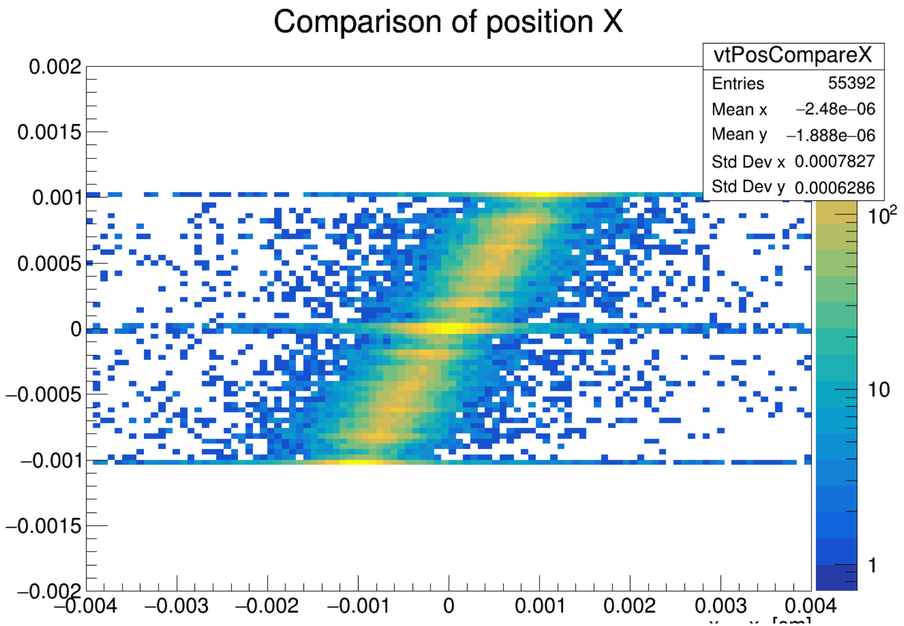
There is dependence between the measurement x_i and the center of the pixel x_c?

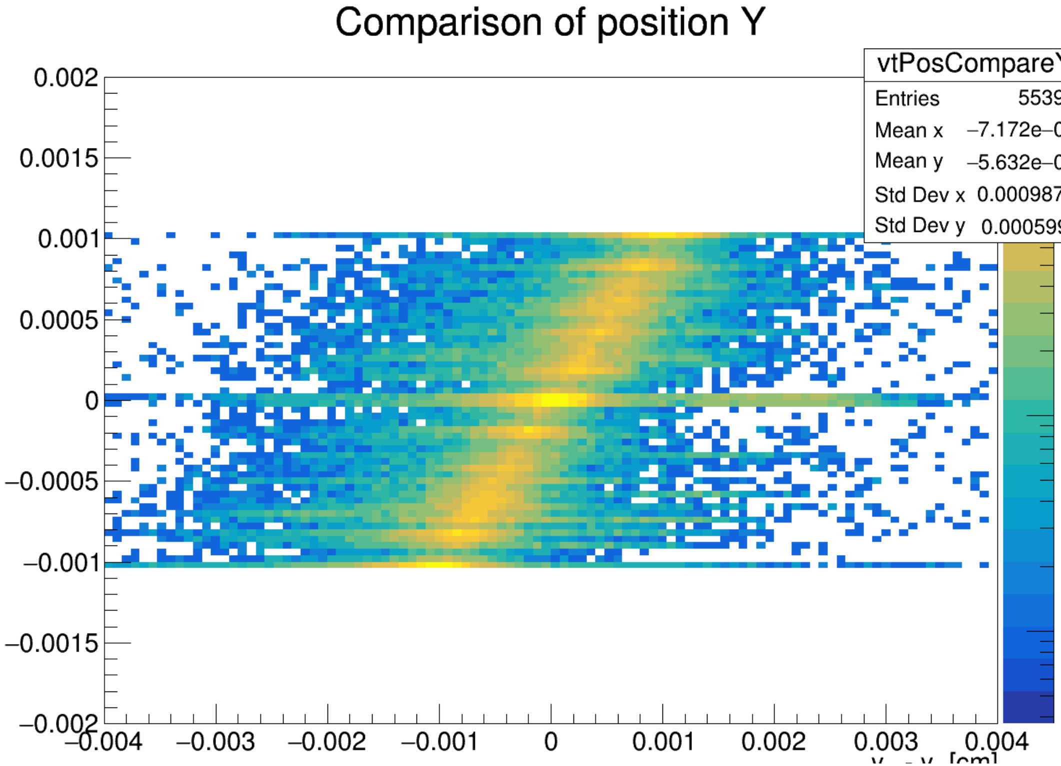




The relation seems linear. The fit has a shift of 1 pixel

There is dependence between the measurement x_i and the center of the pixel x_c?





The relation seems linear. The fit has a shift of 1 pixel

Observation

It is right to use the mean and its standard deviation as best estimator of the position of a cluster?
We are supposing that every fired pixel is a independent and identically distributed (IID) random variable, distributed as a gaussian.
We know actually that they are not independent, but fired according to the energy loss of the particle

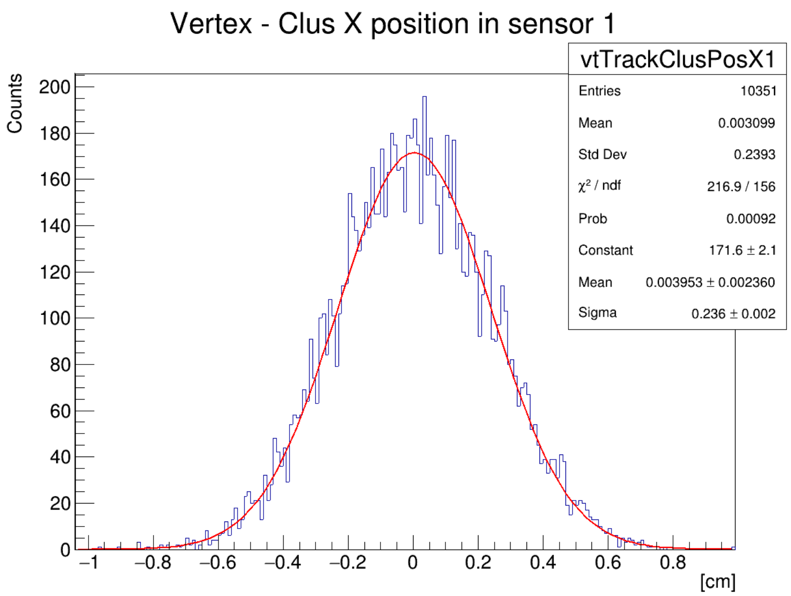
$$x = \frac{\sum f_n x_n}{\sum f_n} \qquad \sigma_x = \sqrt{\frac{\sum f_n (x_n - x)^2)}{\sum f_n}}$$

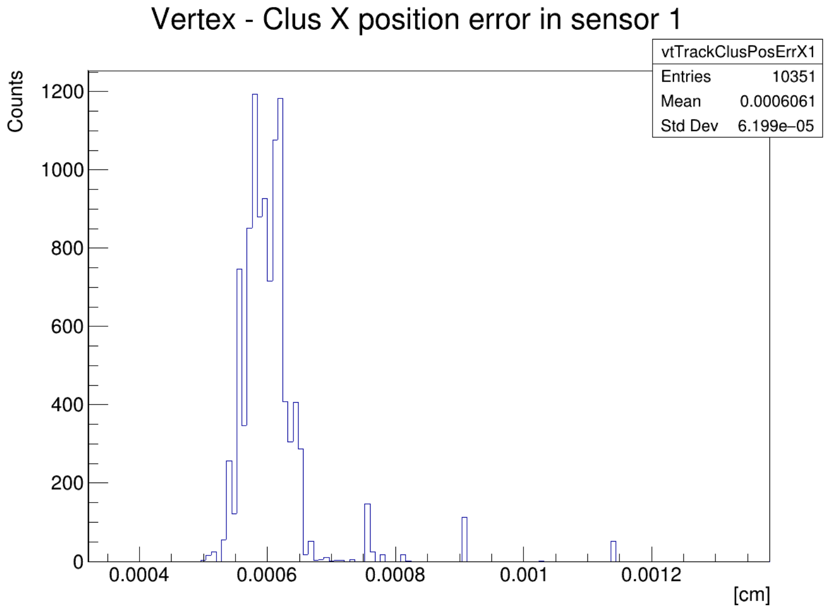
Observations

another estimator of position is needed? → Let’s try median

Position of the cluster

measure using **median** for cluster position





if n is odd, $\mathrm{median}(x) = x_{(n+1)/2}$

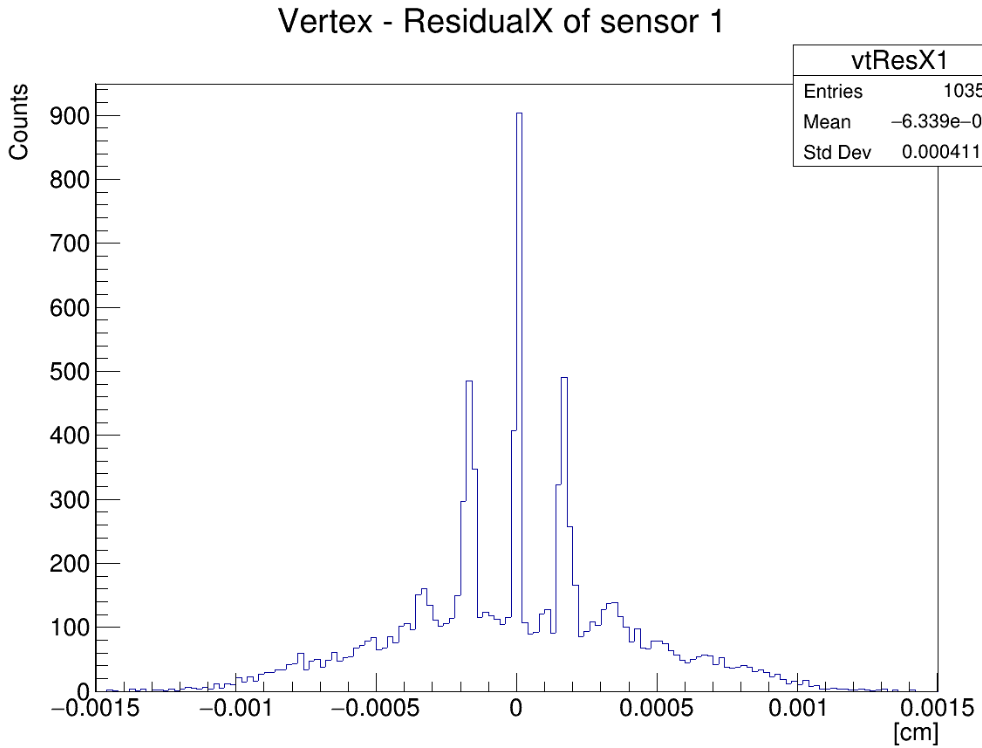
if n is even, $\mathrm{median}(x) = \dfrac{x_{(n/2)} + x_{((n/2)+1)}}{2}$

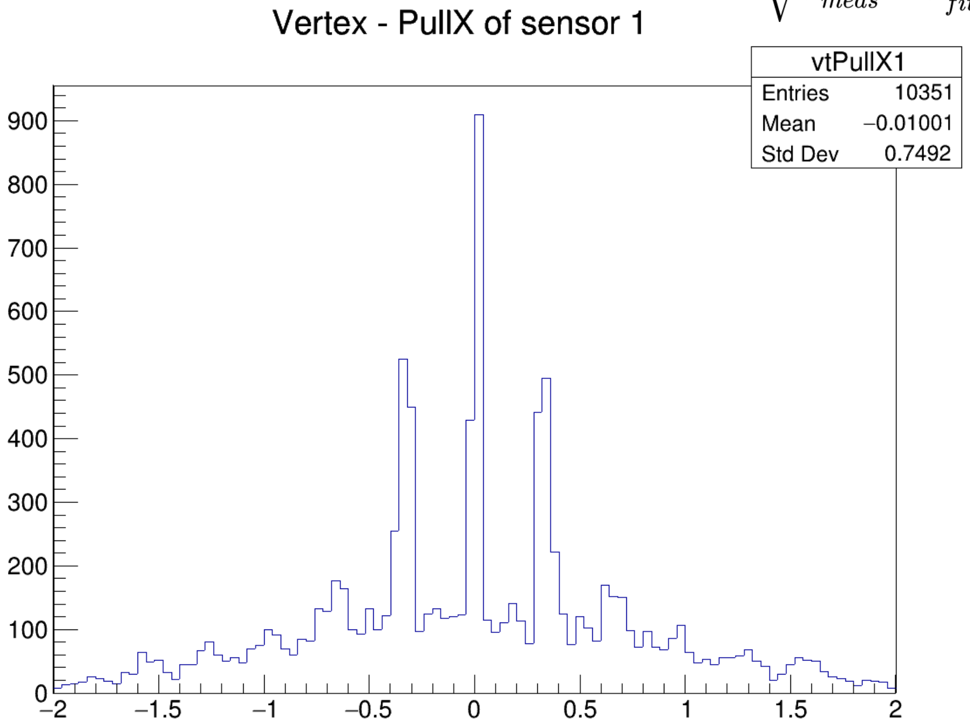
$$\sigma_\mu = \frac{\sigma_x}{\sqrt{n}} \qquad \sigma_x = \sqrt{\frac{\sum f_n (x_n - x)^2)}{\sum f_n}}$$

Residuals and pull X coordinate

$$r = x_{meas} - x_{fit}$$

$$p = \frac{x_i - f(z_i)}{\sqrt{\sigma^2_{meas} - \sigma^2_{fit}}}$$





peaks?