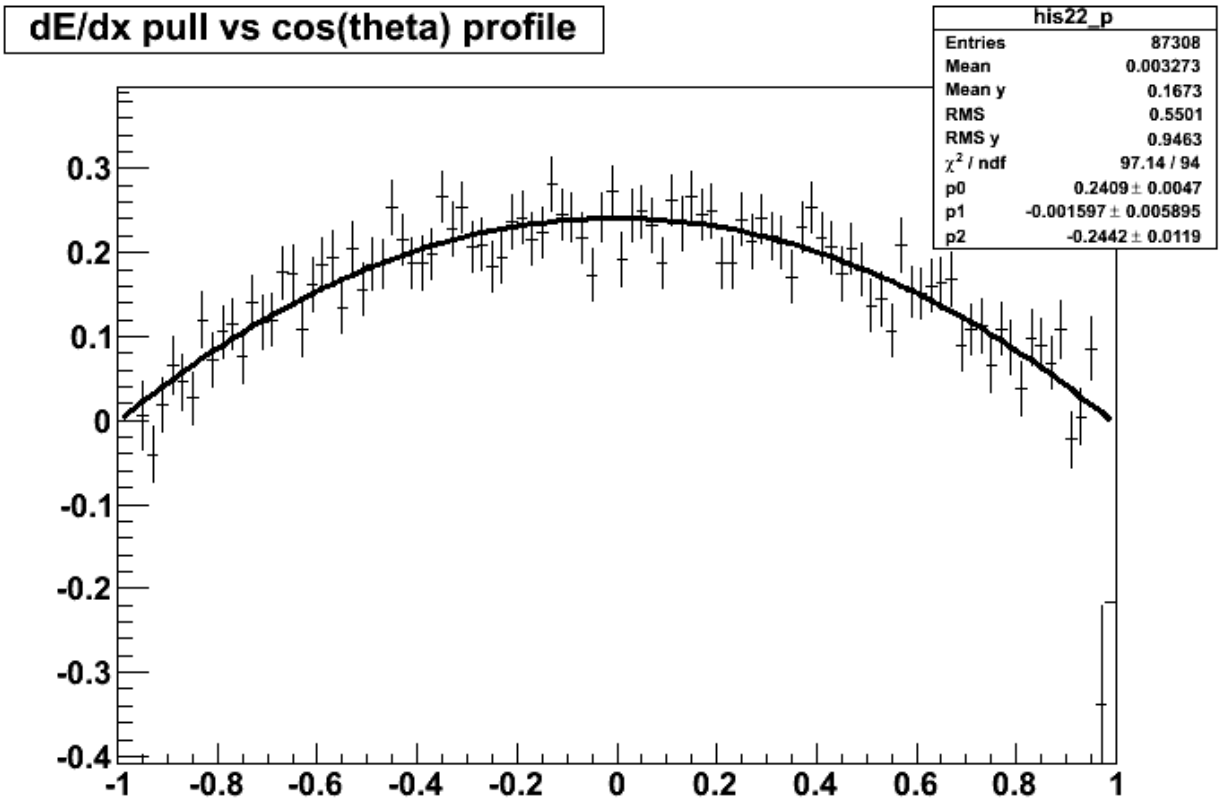
Problem at high momenta

Initially the simulation of dE/dx for the single hit was implemented in PacTrk/PacTrkHitViewDch.cc, and the spread was generated according to a Gaussian **without requiring dedx>0**.
Here is the relevant line in PacTrkHitViewDch::getHitInfo(...) in revision 1543:
dedx = RandGauss::shoot(rng,dedx_ave,ededx);
Note that the absence of a requirement "dedx>0" was a feature,not a bug. At some point the line above was changed to:
dedx = std::max(RandGauss::shoot(rng,dedx_ave,ededx),0.0);
See PacTrkdEdxMeas::get_dedx(...), revision 1714. I think that this cut has been introduced when the dE/dx of SVT has been implemented.

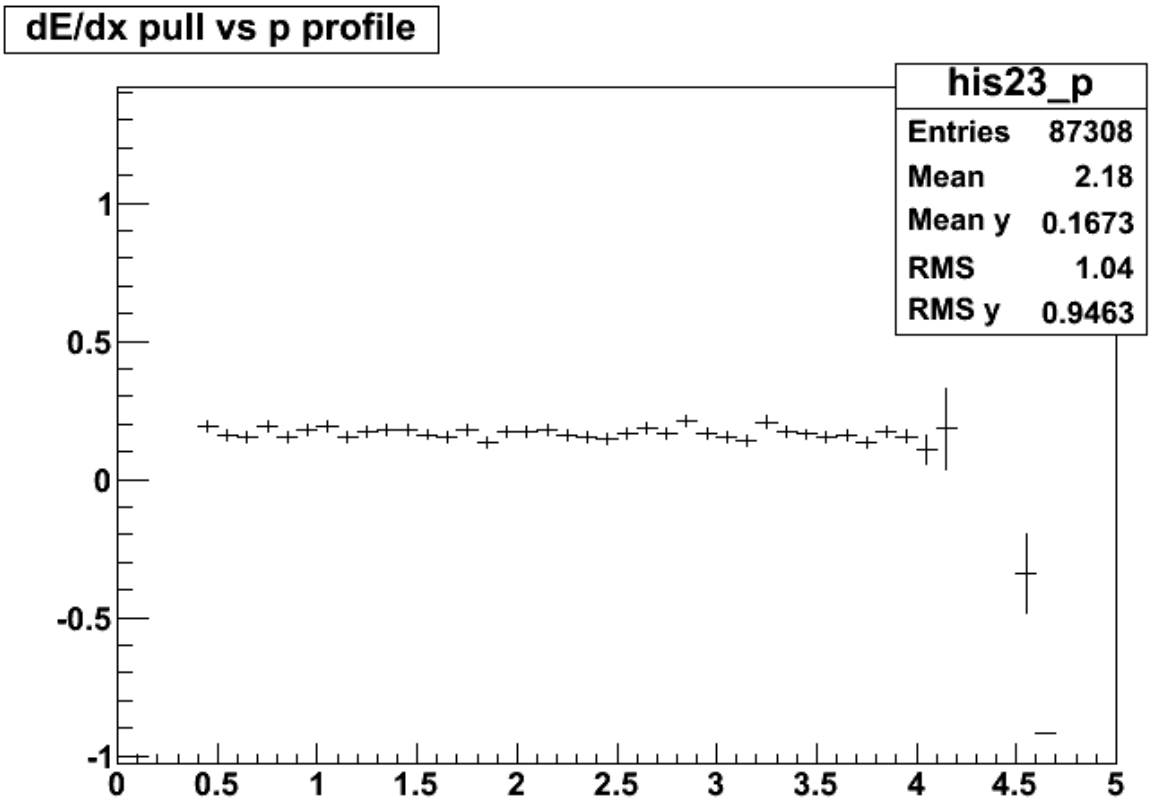
Why for DCH it's important that the requirement dedx>0 is NOT applied? Because the dE/dx error of the single hit is such that there is a non-negligible probability that the Gaussian-generated measurement takes a value<0, but at the same time the probability that the *average* over the hits of the track is negative is extremely small, practically zero.
In other words, it is fine if the |dE/dx| measurement of a single hit can be negative, provided that the |dE/dx| measurement of the *track* is *positive* defined.

With *dedx = std::max(RandGauss::shoot(rng,dedx_ave,ededx),0.0):*

Below: dE/dx pull vs. cos(theta) **before** the fix (p>0.4 GeV)

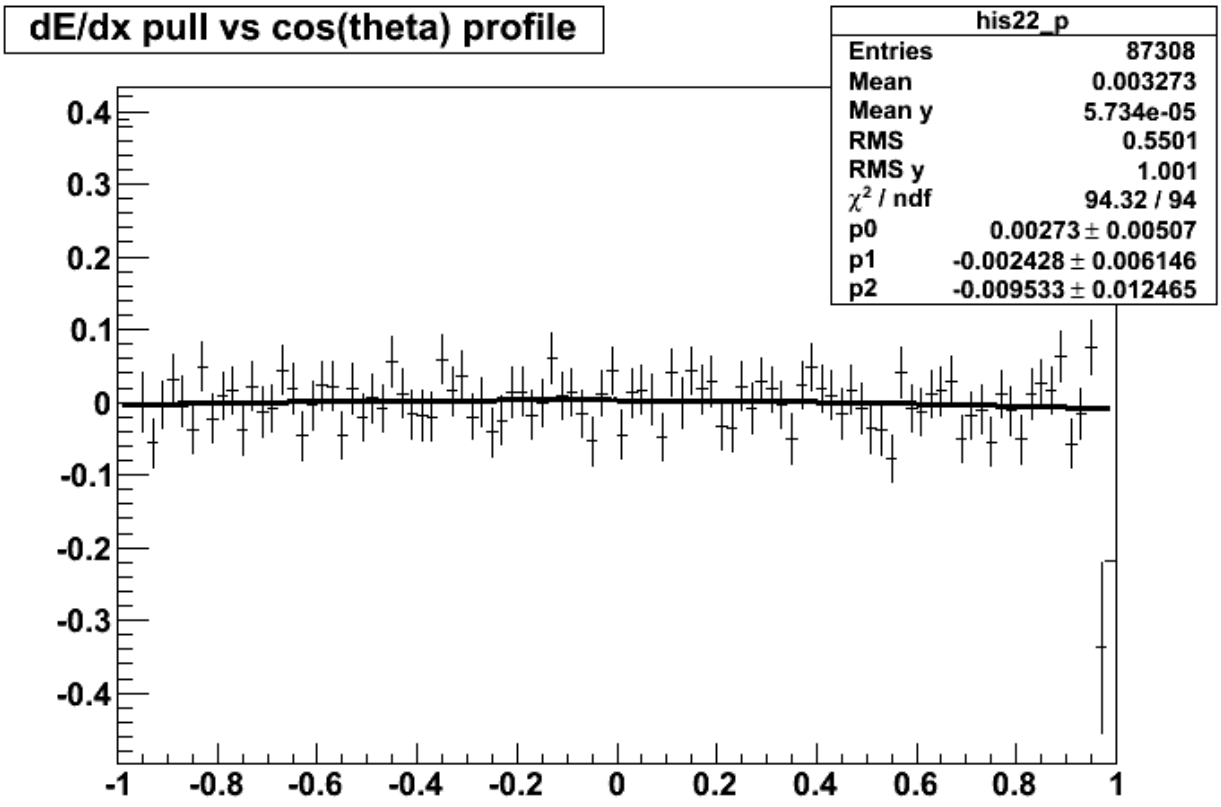


*Below: dE/dx pull vs. p **before** the fix (p>0.4 GeV)*

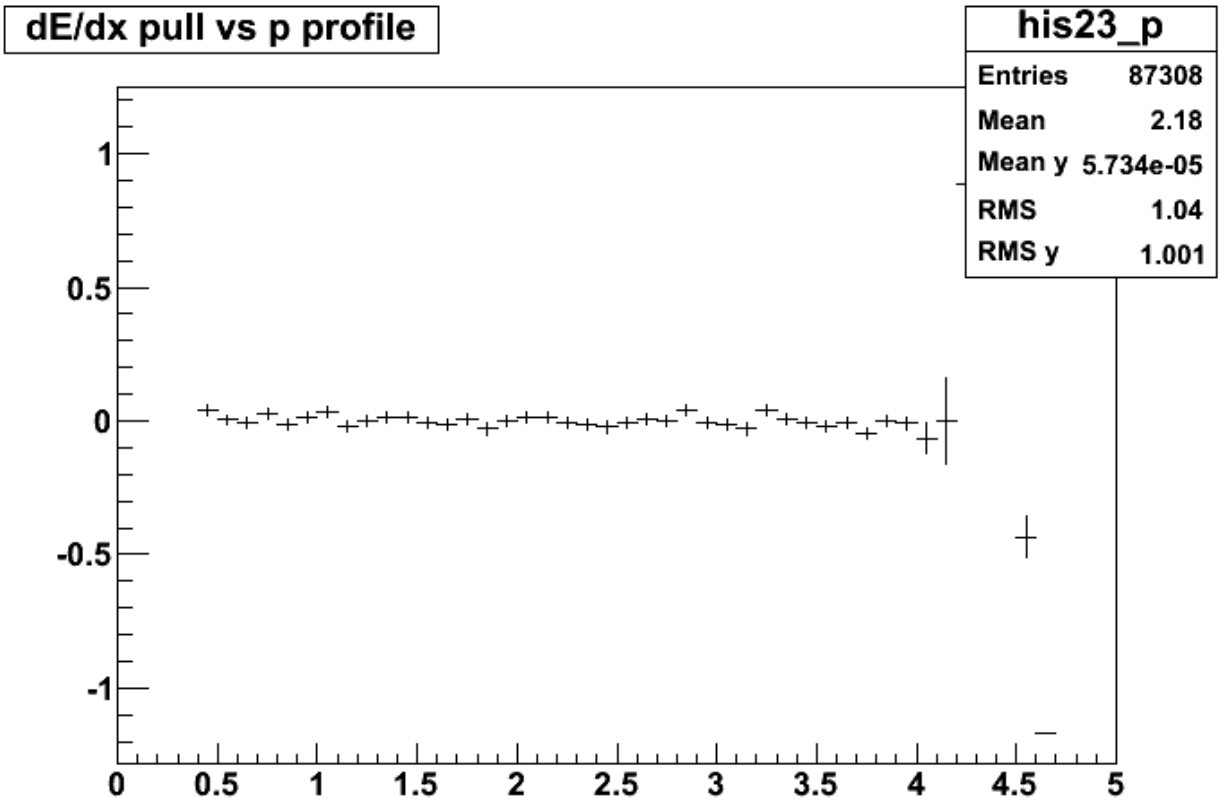


With *dedx = RandGauss::shoot(rng,dedx_ave,ededx);*

Below: dE/dx pull vs cos(theta) after the fix (p>0.4 GeV)

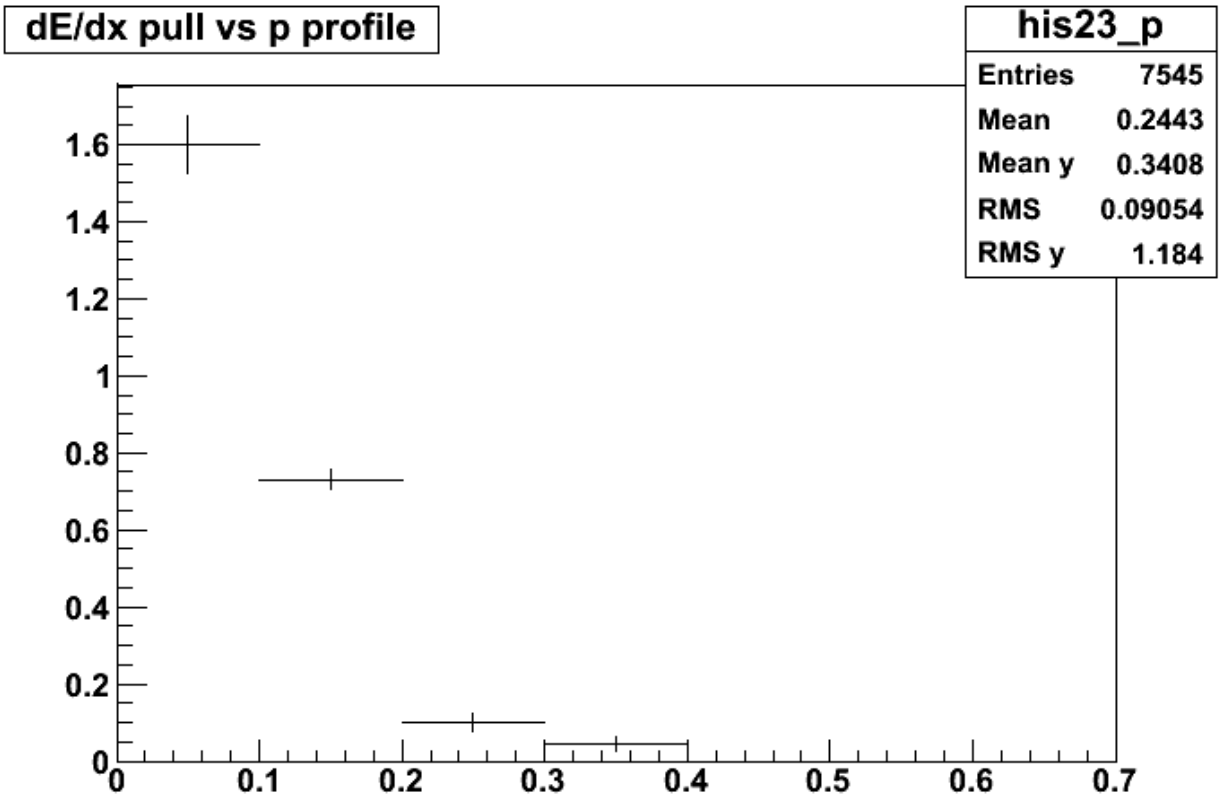


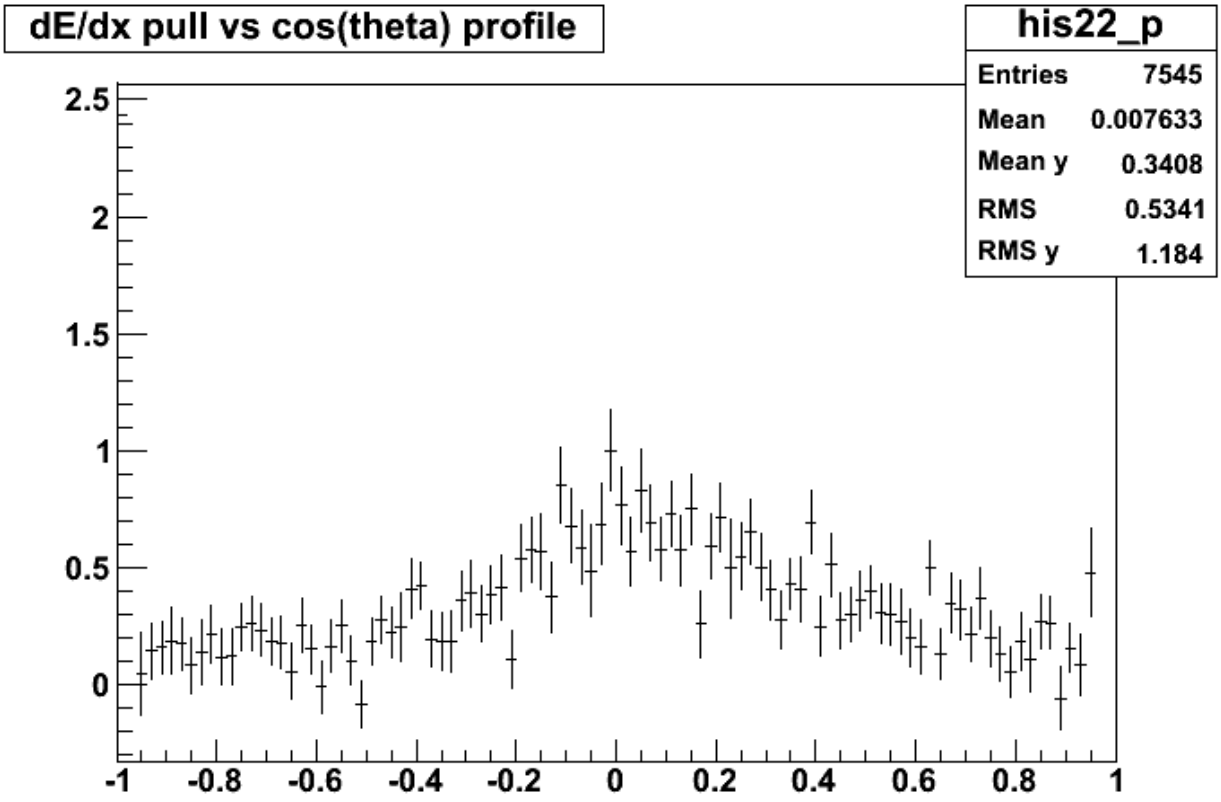
Below: dE/dx pull vs. p after the fix (p>0.4 GeV)



At low momenta there is a bias in the dE/dx pull. Under investigation:

Below: dE/dx pull vs. p ***after*** *the fix (p<0.4 GeV)*. Particles are pions.

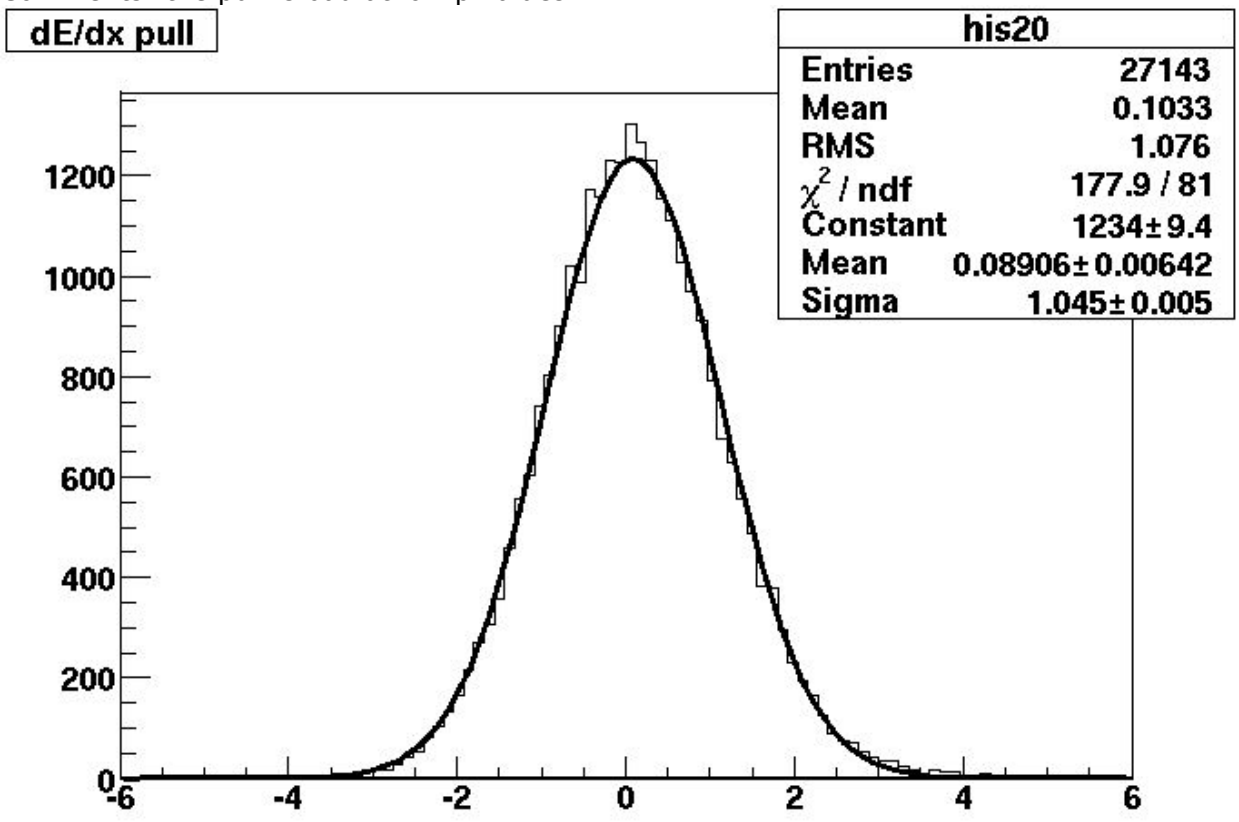

dE/dx pull vs p profile
his23_p
Entries 7545
Mean 0.2443
Mean y 0.3408
RMS 0.09054
RMS y 1.184
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7

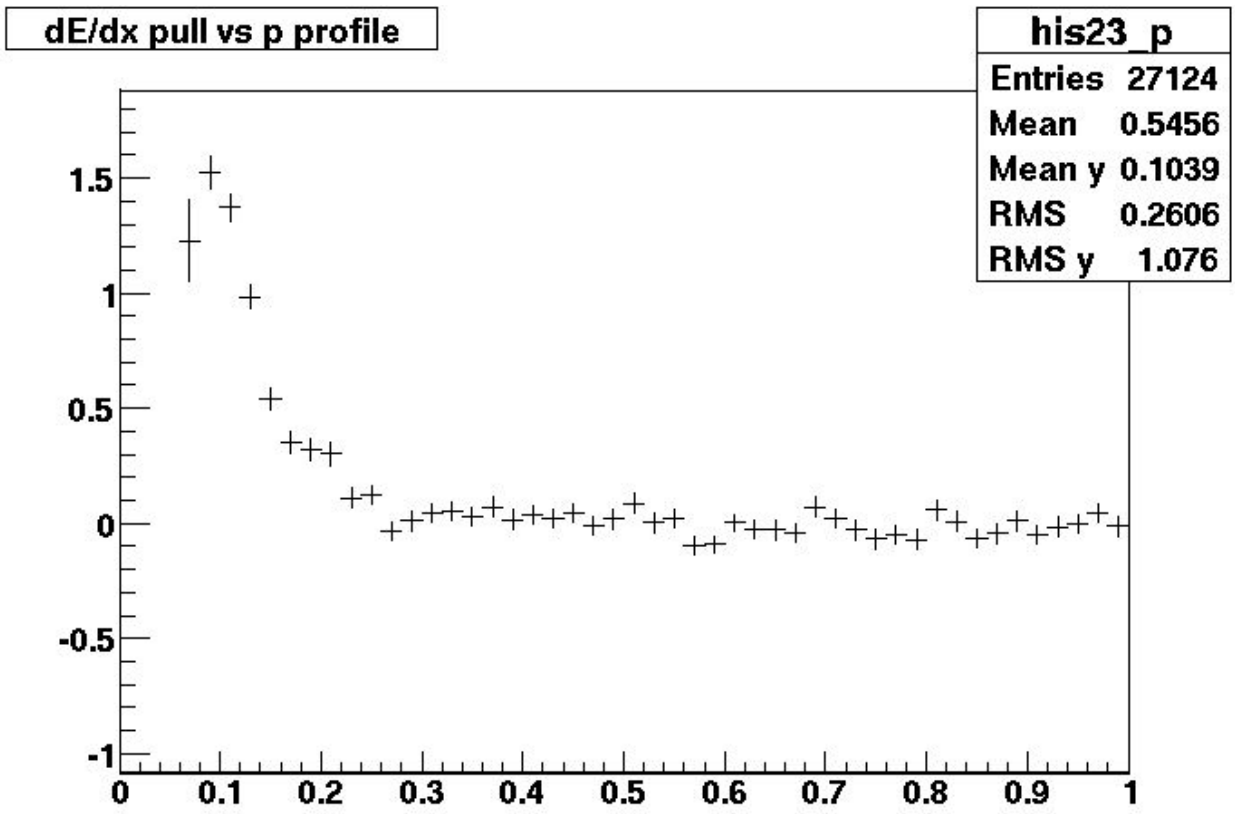

dE/dx pull vs cos(theta) profile
his22_p
Entries 7545
Mean 0.007633
Mean y 0.3408
RMS 0.5341
RMS y 1.184
2.5
2
1.5
1
0.5
0
-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1

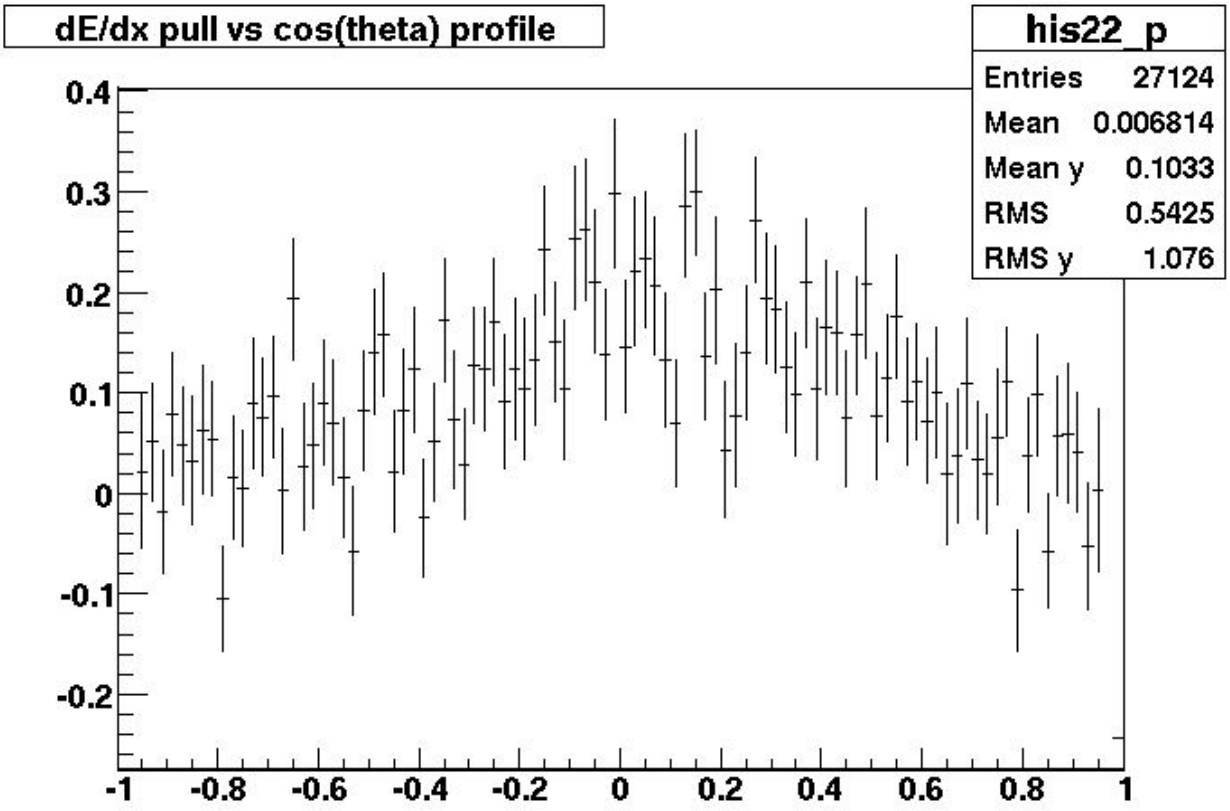
Problem at low momenta

Current code (using p (reco) at the origin)

dEdxExpPi computed with reco computed at the origin (current code in SVN)
dE/dx reco computed from (truncated) weighted mean in PacTrunc
Comments: the pull is bad at low p values.

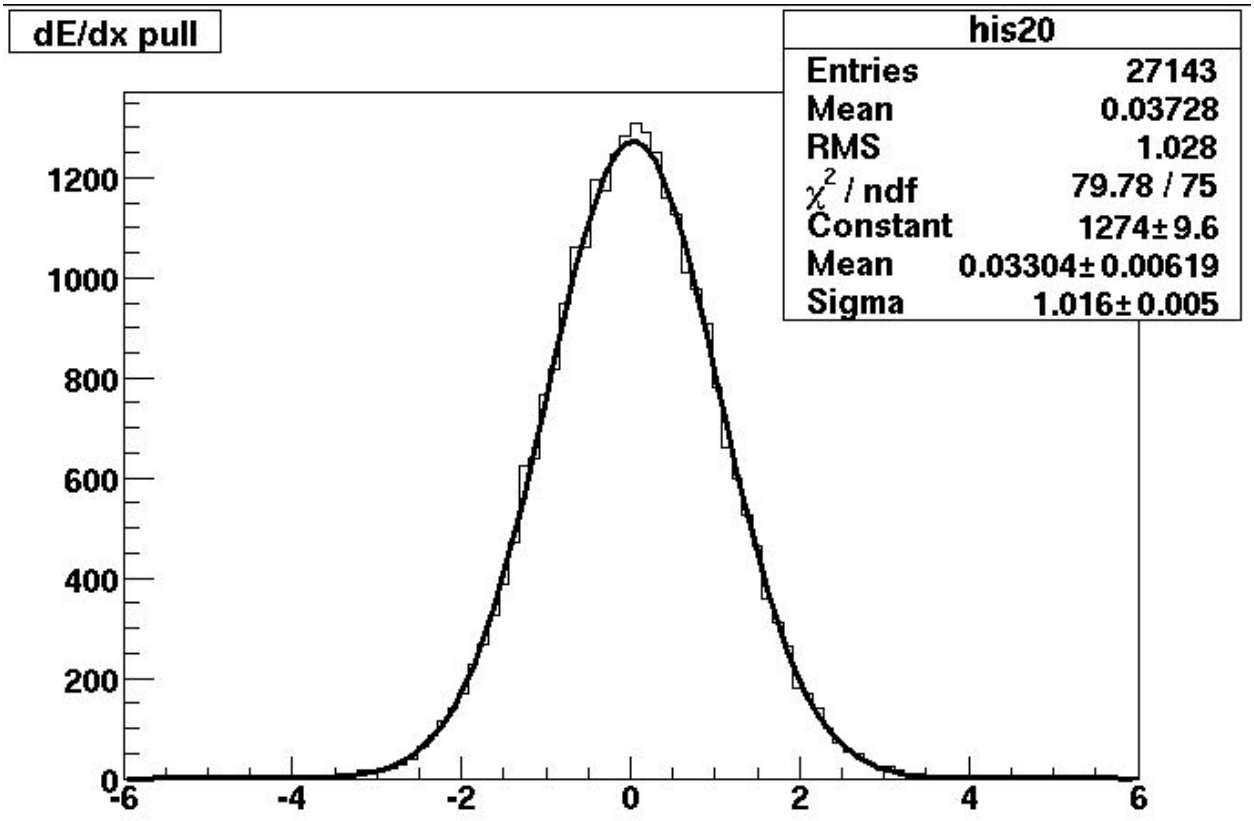


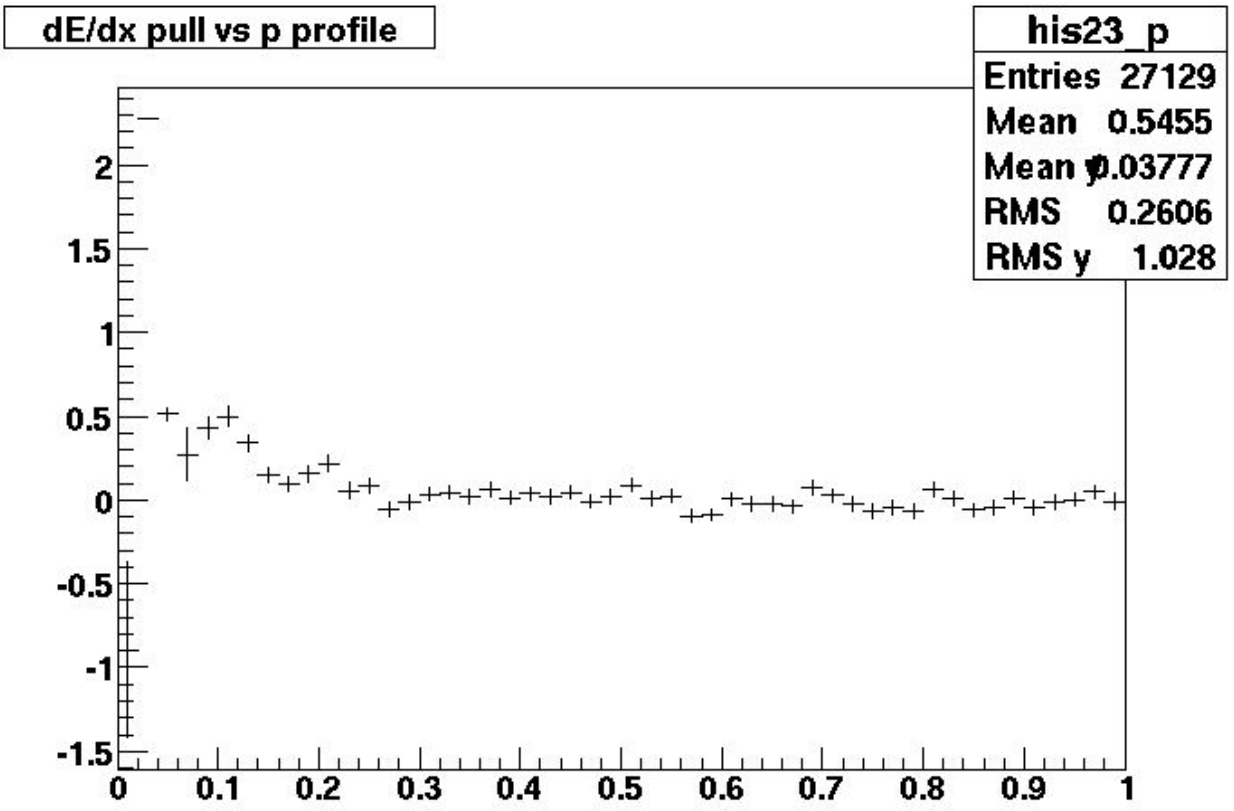

dE/dx pull vs p profile
his23_p
Entries 27124
Mean 0.5456
Mean y 0.1039
RMS 0.2606
RMS y 1.076
1.5
1
0.5
0
-0.5
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

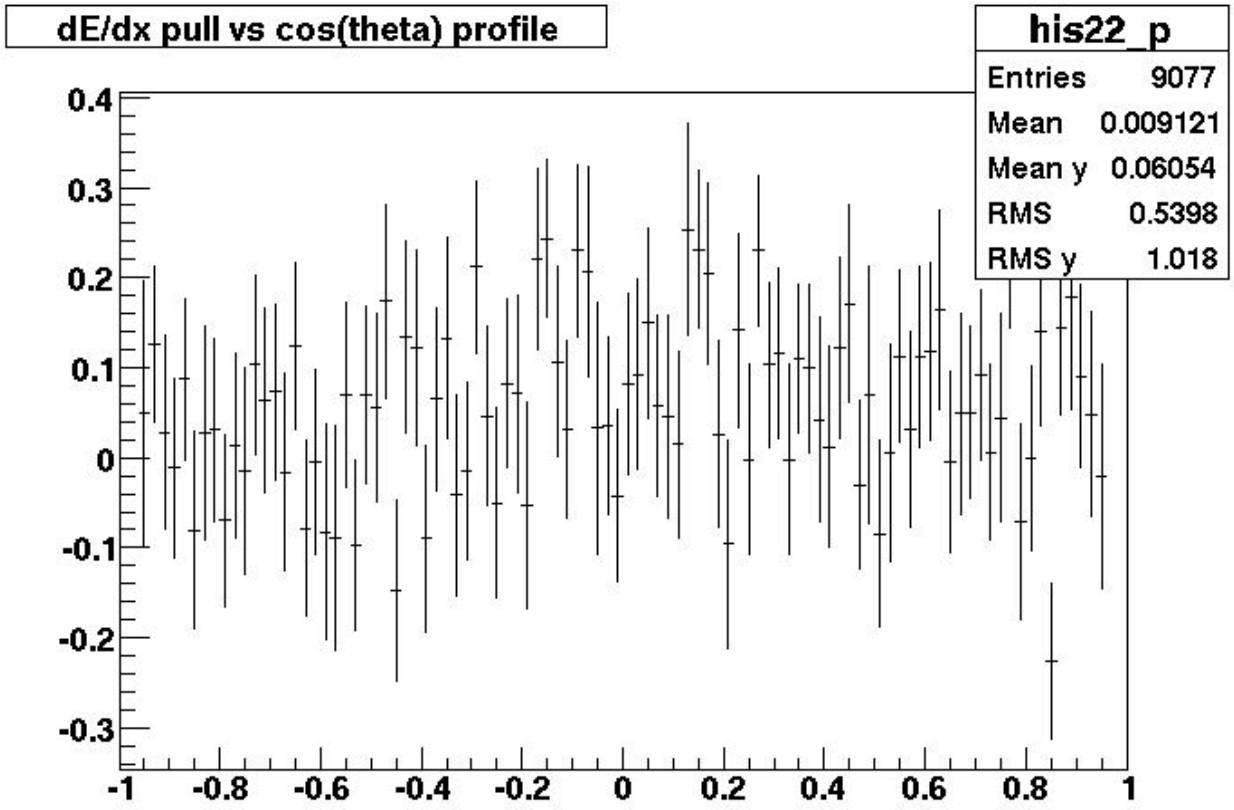

dE/dx pull vs cos(theta) profile
his22_p
Entries 27124
Mean 0.006814
Mean y 0.1033
RMS 0.5425
RMS y 1.076
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

Using p(reco) at DCH

dEdxExpPi computed with p reco computed at the DCH (first DCH hit)
dE/dx reco computed from (truncated) weighted mean in PacTrunc
Comments: the pull is not good at low p values, though it has improved with respect to the when p is computed at the origin

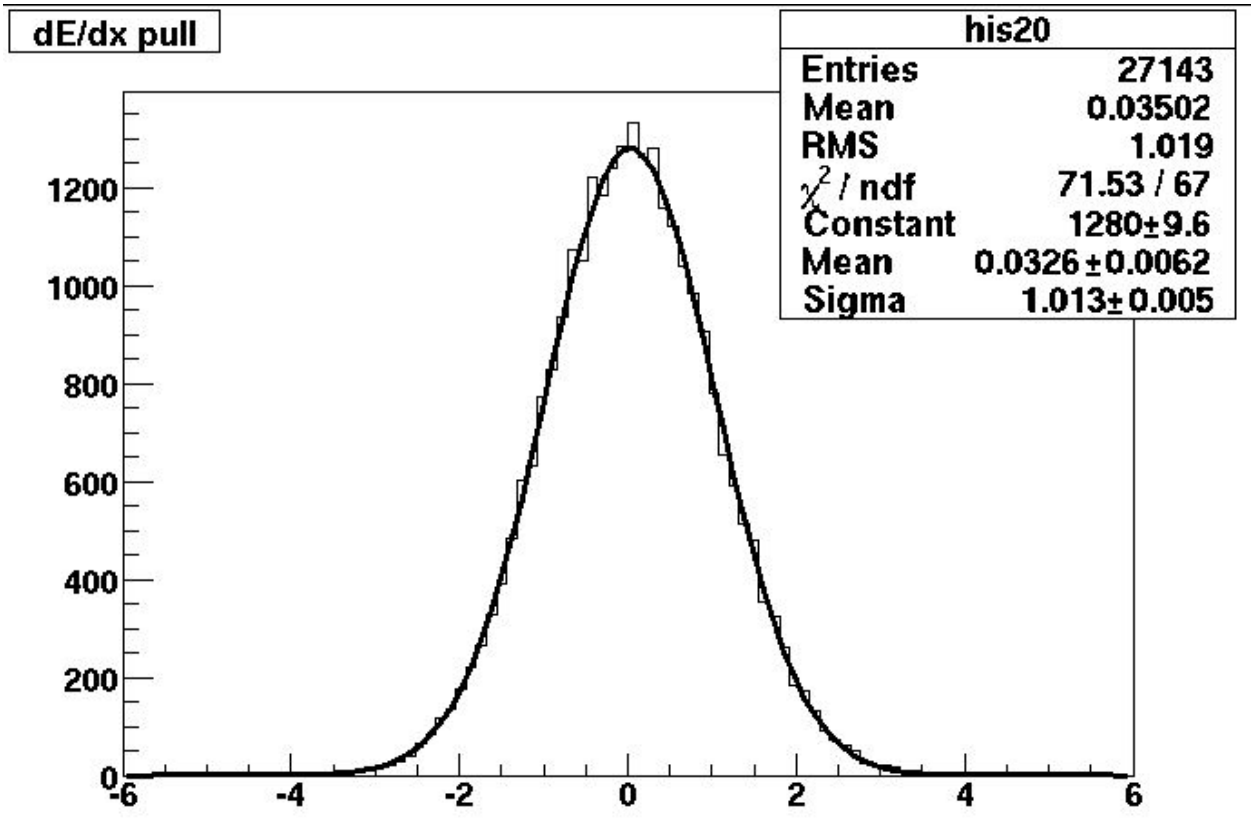


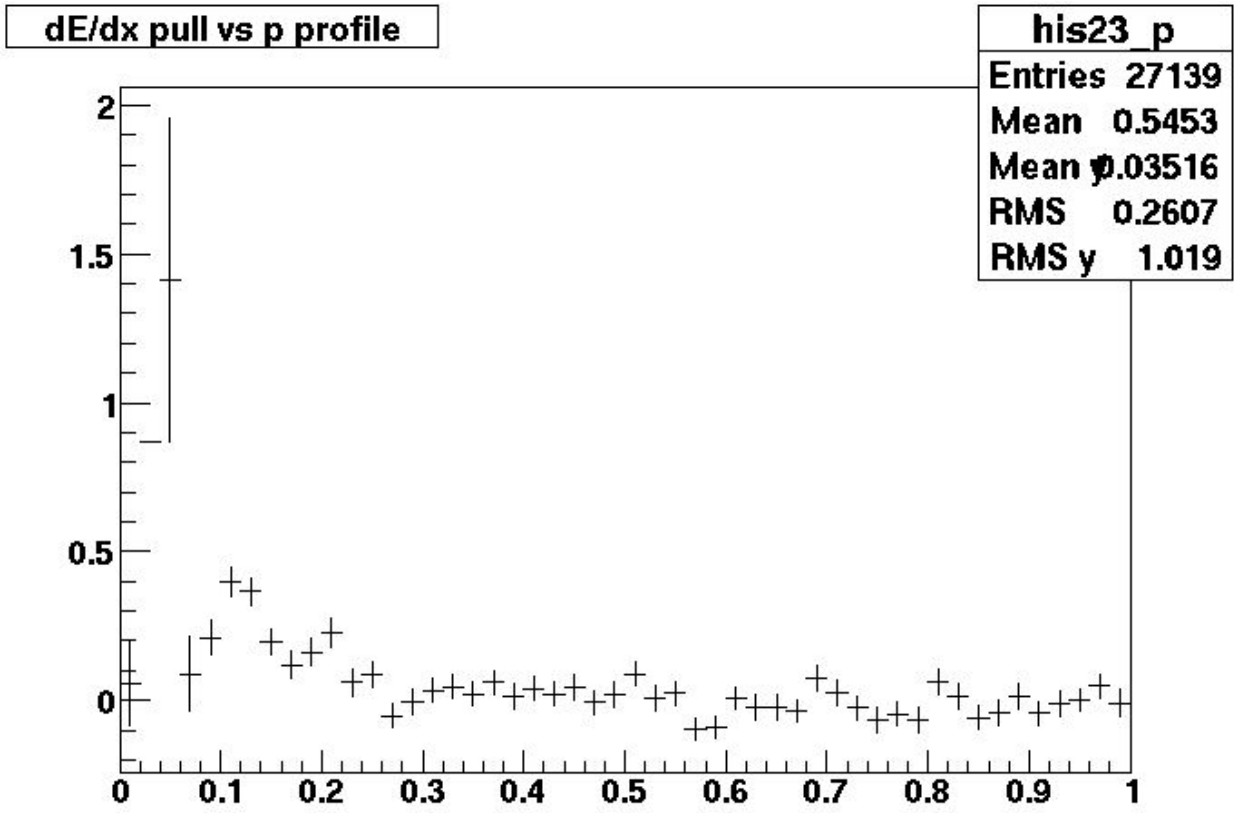

dE/dx pull vs p profile
his23_p
Entries 27129
Mean 0.5455
Mean y 0.03777
RMS 0.2606
RMS y 1.028
2
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

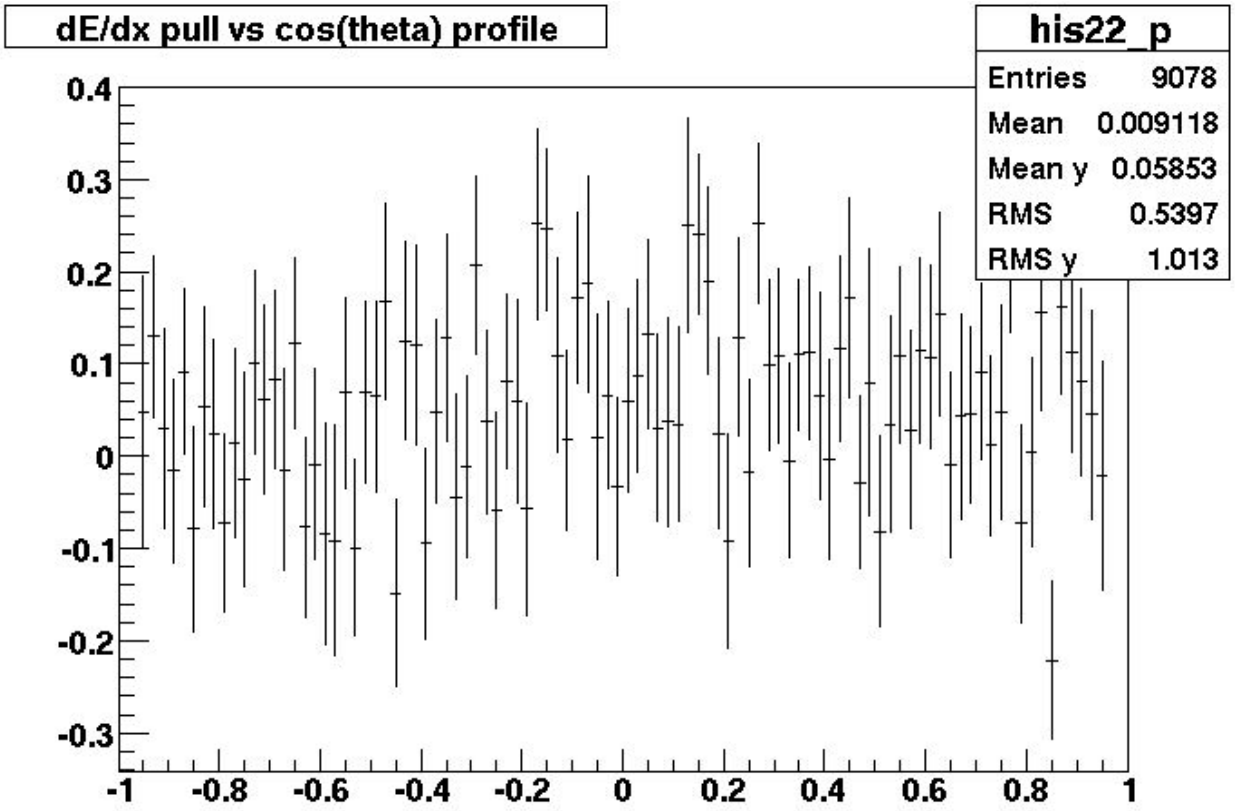

dE/dx pull vs cos(theta) profile
his22_p
Entries 9077
Mean 0.009121
Mean y 0.06054
RMS 0.5398
RMS y 1.018
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

Using p(true) at DCH

dEdxExpPi computed with p true (1st PacSimHit of DCH)
dE/dx reco computed from (truncated) weighted mean in PacTrunc
Comments: the pull is still not perfect because the true p changes slightly along the track due to the energy loss. A small bias at positive values is visible at very low momenta.

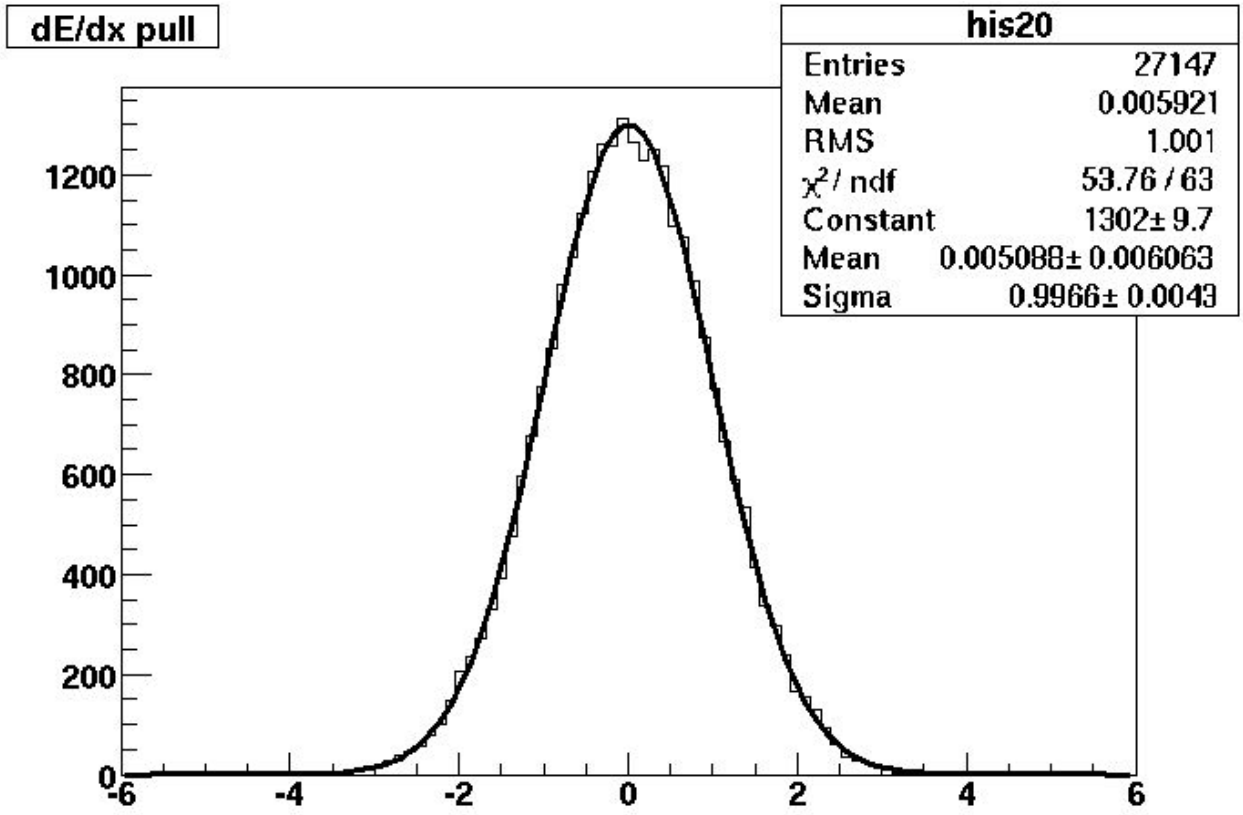


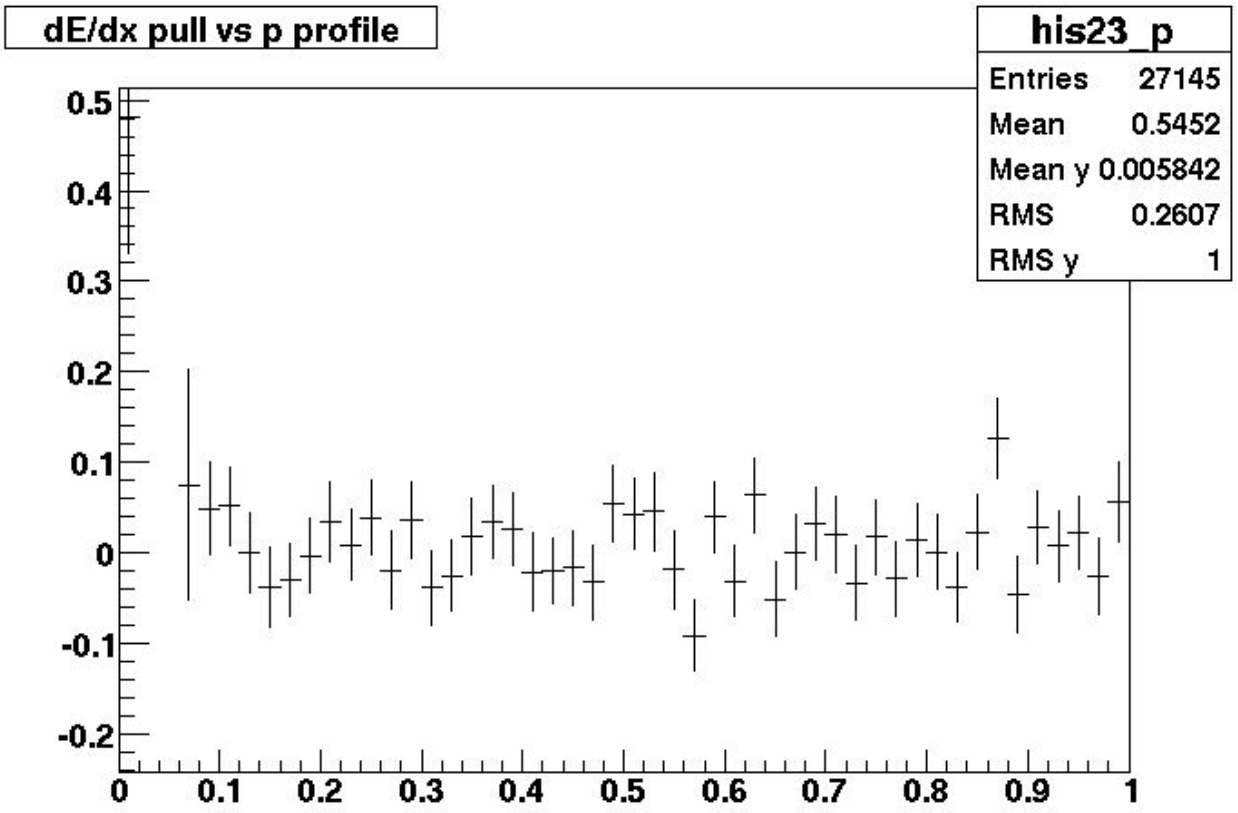

dE/dx pull vs p profile
his23_p
Entries 27139
Mean 0.5453
Mean y 0.03516
RMS 0.2607
RMS y 1.019
2
1.5
1
0.5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

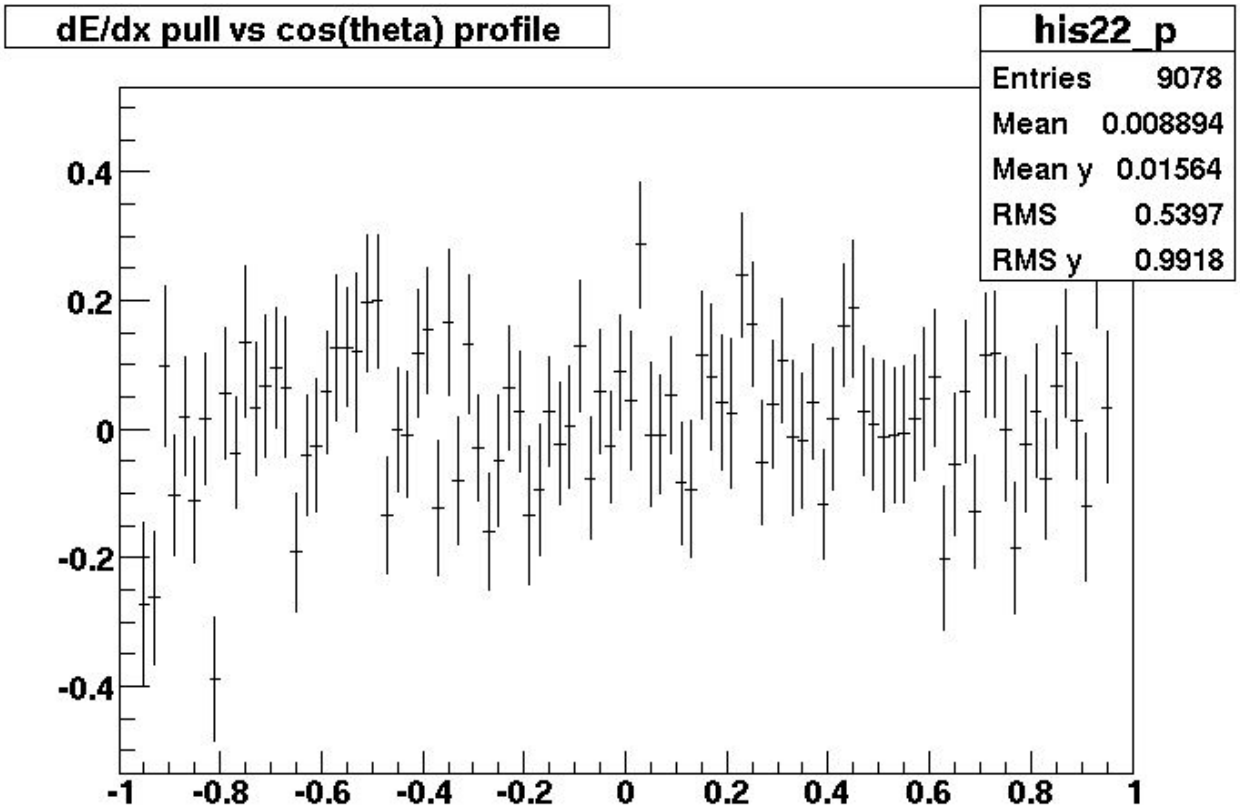


Using p(true) at DCH and only the info of first (sim)hit

dEdxExpPi computed with p true (1st PacSimHit of DCH)
dE/dx reco computed from Gauss(dEdxExpPi,sigma) where sigma=error_1st_DchHit/ sqrt(nhit)
Comments: the pull is perfect at all p values, including the lowest ones.


dE/dx pull
his20
Entries 27147
Mean 0.005921
RMS 1.001
χ² / ndf 53.76 / 63
Constant 1302 ± 9.7
Mean 0.005088 ± 0.006063
Sigma 0.9966 ± 0.0043
1200
1000
800
600
400
200
0
-6
-4
-2
0
2
4
6


dE/dx pull vs p profile
his23_p
Entries 27145
Mean 0.5452
Mean y 0.005842
RMS 0.2607
RMS y 1
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1


dE/dx pull vs cos(theta) profile
his22_p
Entries 9078
Mean 0.008894
Mean y 0.01564
RMS 0.5397
RMS y 0.9918
0.4
0.2
0
-0.2
-0.4
-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1

Comments

The main reason of the bad shape of the dE/dx pull in FastSim at very low momenta is the fact that the expected dE/dx is calculated using the reconstructed momentum at the origin, instead of p at the DCH. Using p at the DCH reduces the bias by at least a factor 3 [see here].
Even using p at the DCH a small bias of the pull remains. Is it due to the fact that p at the DCH is not evaluated well enough? No, because the bias remains even when the true p at the DCH (the p of the first SimHit) is used [see here].
The remaining bias is due to the fact that the particle momentum changes in a non negligible way within the DCH portion of the track. Therefore if the pull is built from dE/dx_measured - dE/dx_expected where dE/dx_expected is the value expected at the first hit of the DCH, and dE/dx_measured is a weighted average over the DCH hits of the track, then there is a small (positive) bias for tracks with small betagamma.

Possible solution:
Note: the small bias at low betagamma is not due to a problem in the dE/dx measurement, but rather in the calculation of the expected dE/dx.

Possible solutions:

option	measured dE/dx	expected dE/dx	comments
1	Measured dE/dx of first DCH hit. The error is calculated as "error of 1st hit/sqrt(# of dE/dx hits)"	dE/dx of the first hit	Pro: Pulls good at low betagamma. Cons: dE/dx measurement perhaps unnecessarily simplified
2	Weighted average of dE/dx of the hits of the track	dE/dx of the first hit	As it has been so far. Cons: small bias of pulls at low betagamma
3	Weighted average of dE/dx of the hits of the track	weighted average of dE/dx of the track	Pro: Pulls good at all betagamma. Pro: It may be implemented so that even in case of hit-merging the pull is good. Cons(?): It needs to know the PacSimHit corresponding to a reco hit.