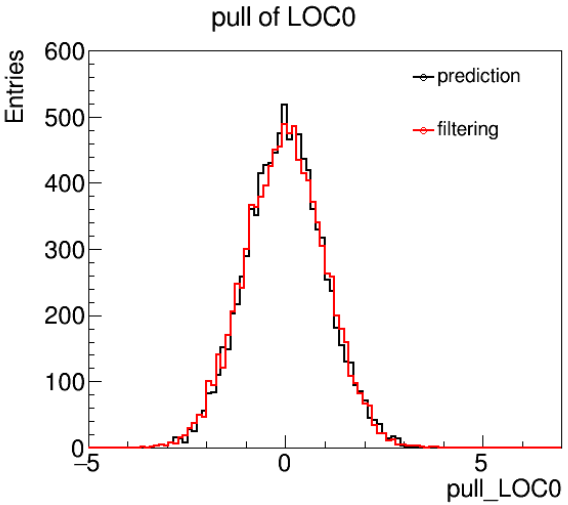
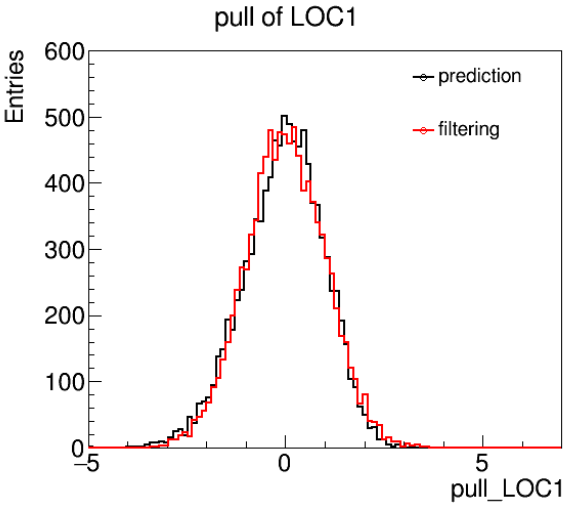
Track parameter pull from KalmanFittter in ACTS

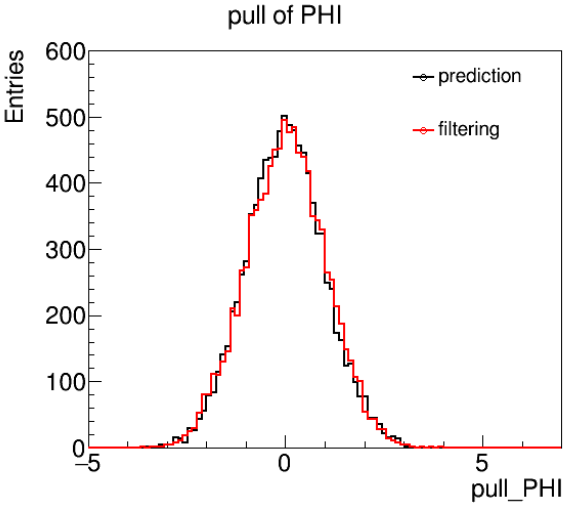
Xiaocong Ai
UC Berkeley

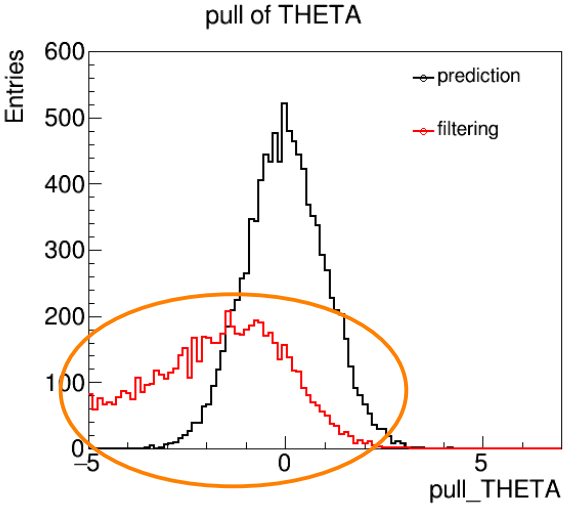
Jul 4, 2019

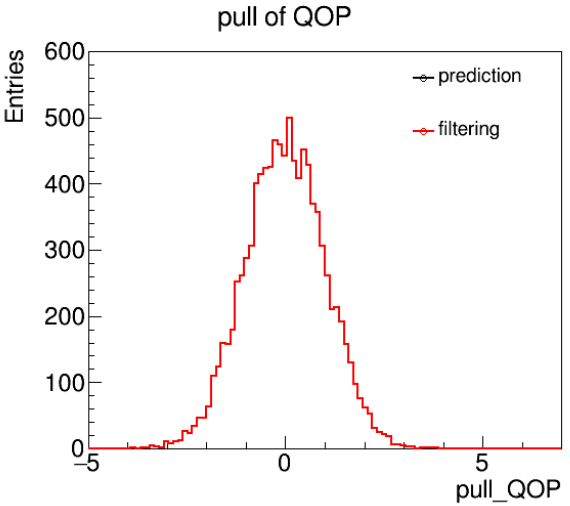
Problem: biased pull for filtering









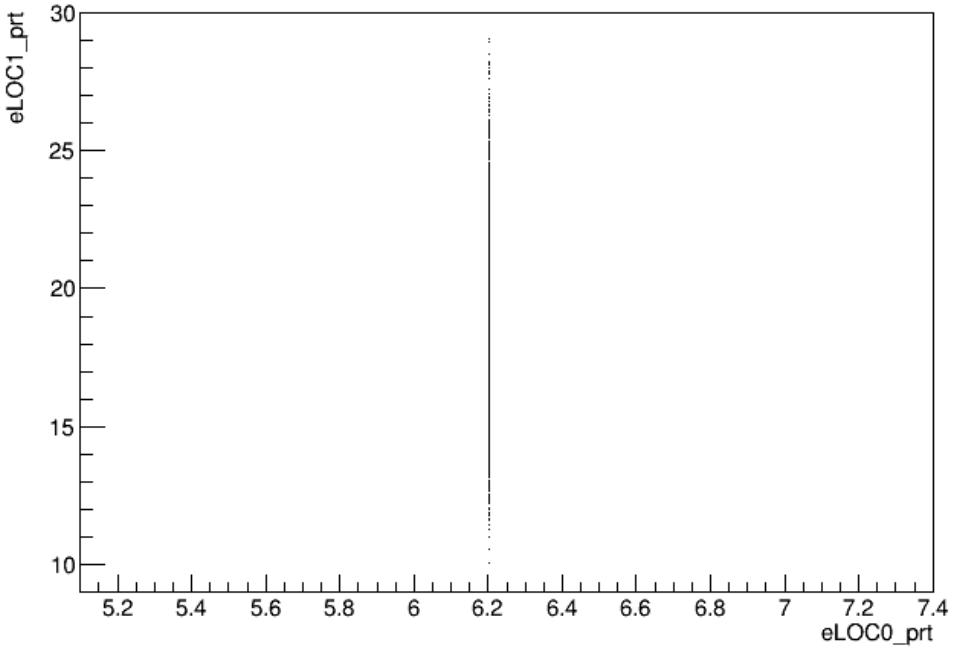


- Initial track parameter and measurements taken from truth smeared with Gaussian
- 10000 tracks have the same first measurement surface

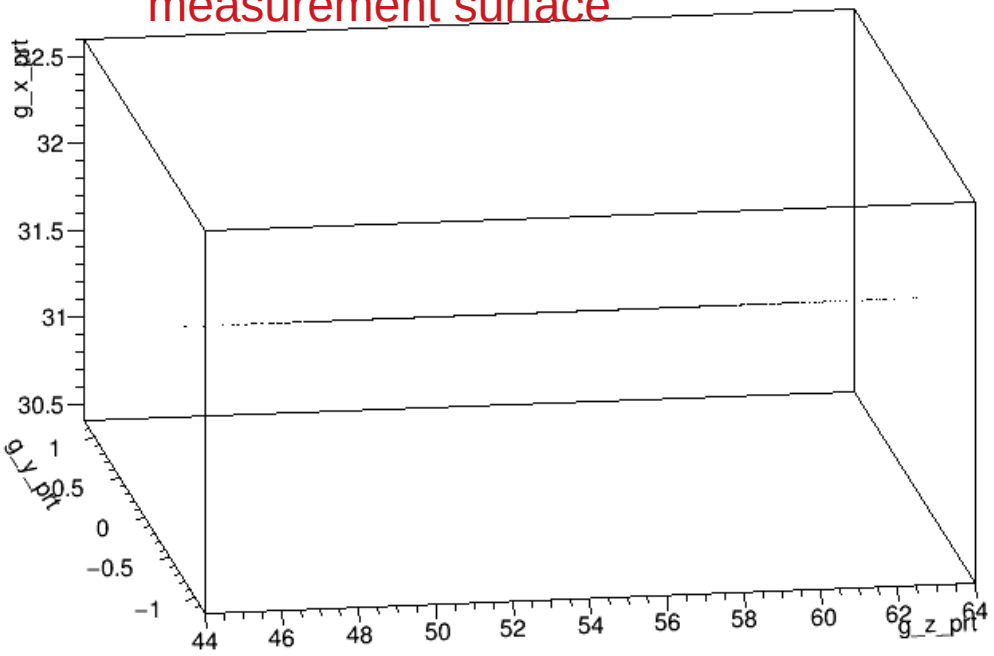
Pull of theta from filtering at first detector layer is already biased...

A test with only θ smeared

loc0: loc1 at first measurement surface



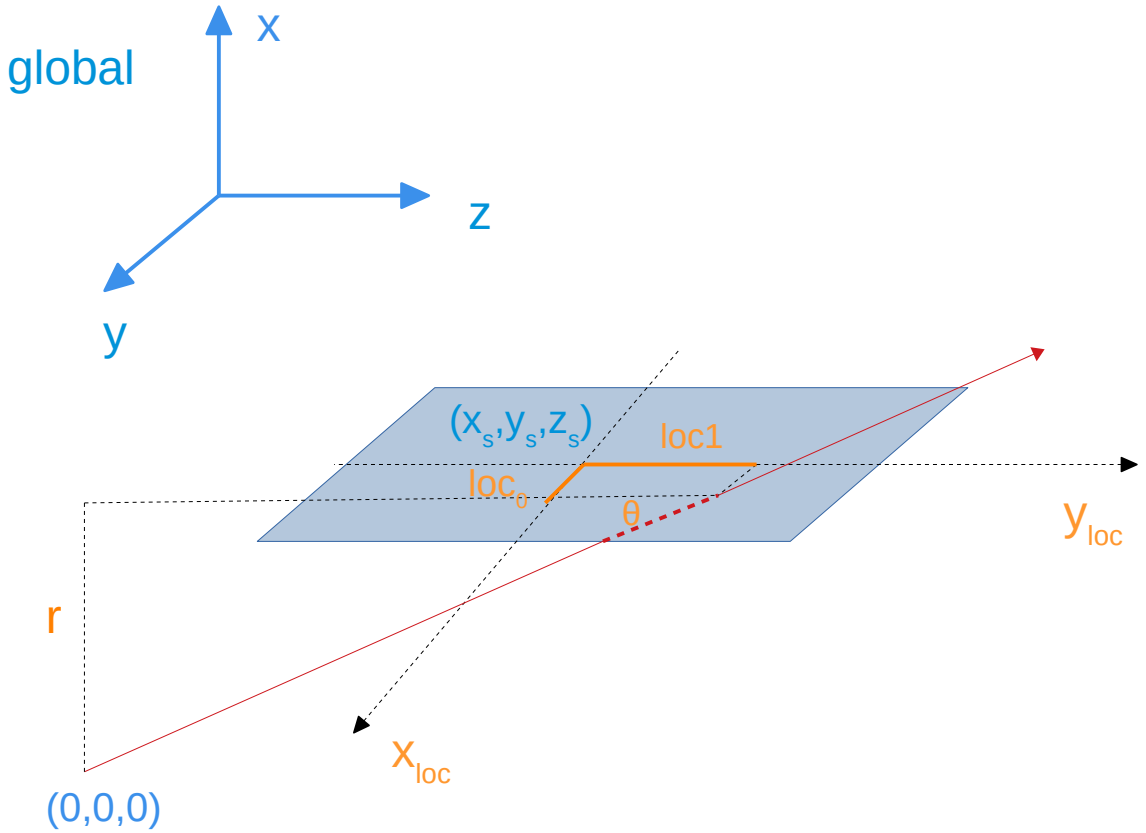
globalX: globalY: globalZ at first measurement surface



In this case:

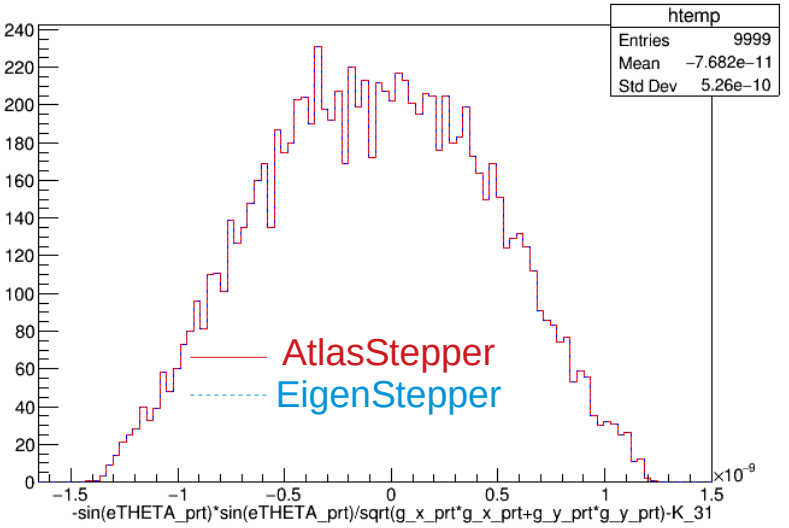
- Measurement is the same with truth hit, measurement covariance =0
- The predicted loc0, Φ, q/p on the first measurement surface will be fixed
- Only variance of loc1 and θ is present

Some mathematics



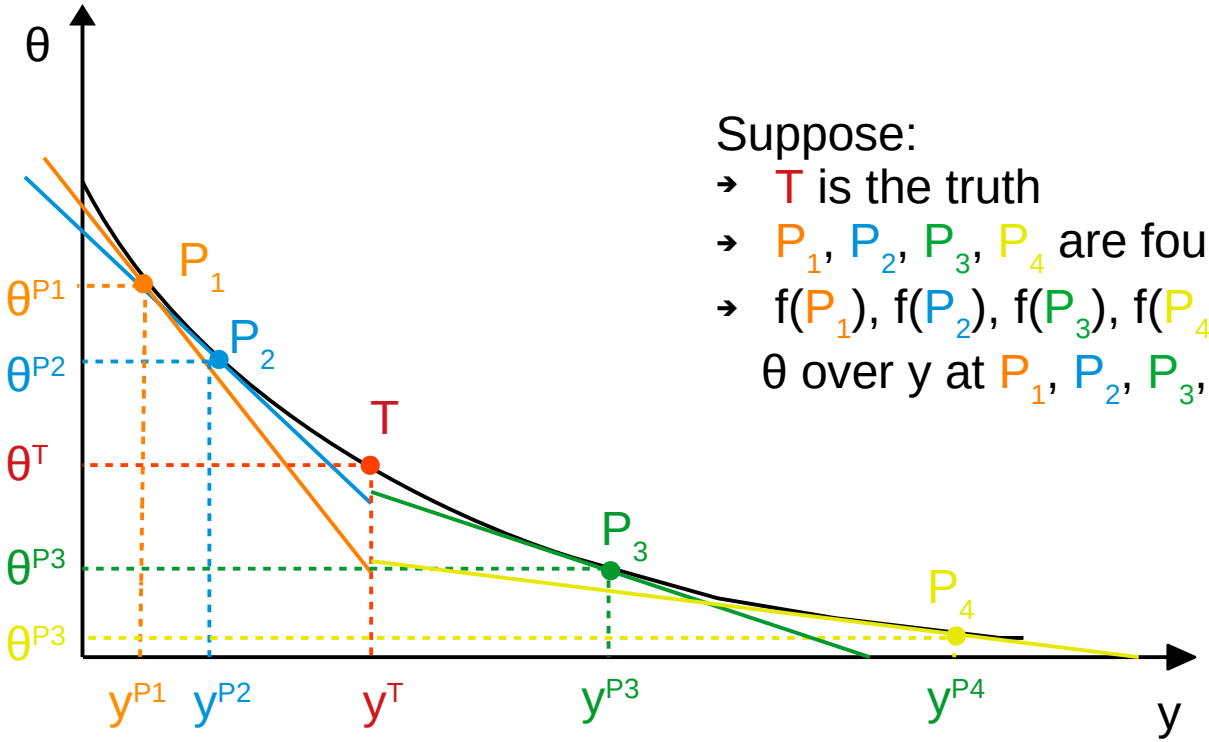
(x_s, y_s, z_s) is the global coordinate of the surface, $loc_1 = r\cot\theta - Z_s$
=> $d\theta/dloc1 = -\sin^2\theta/r$

$d\theta/dloc1$ calculated in this way is the same with filtering gain matrix element K(3,1) !
=> error propagation is validated



Initial parameter have:
- fixed position = (0,0,0), phi = 0
- smeared θ

What's happening in the filtering



Suppose:

- T is the truth
- P_1, P_2, P_3, P_4 are four possible prediction
- $f(P_1)$, $f(P_2)$, $f(P_3)$, $f(P_4)$ are the derivative of θ over y at P_1, P_2, P_3, P_4, respectively.

When we use information at prediction to estimate the value of θ at truth, we will get:

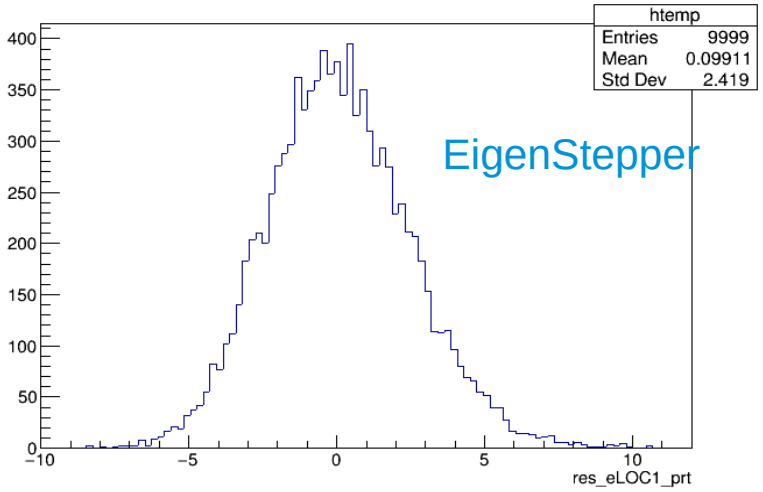
- $\theta^{P1} + f(P_1)*(y^{P1}-y^T) < \theta^T$
- $\theta^{P2} + f(P_2)*(y^{P2}-y^T) < \theta^T$
- $\theta^{P3} + f(P_3)*(y^{P3}-y^T) < \theta^T$
- $\theta^{P4} + f(P_4)*(y^{P4}-y^T) < \theta^T$

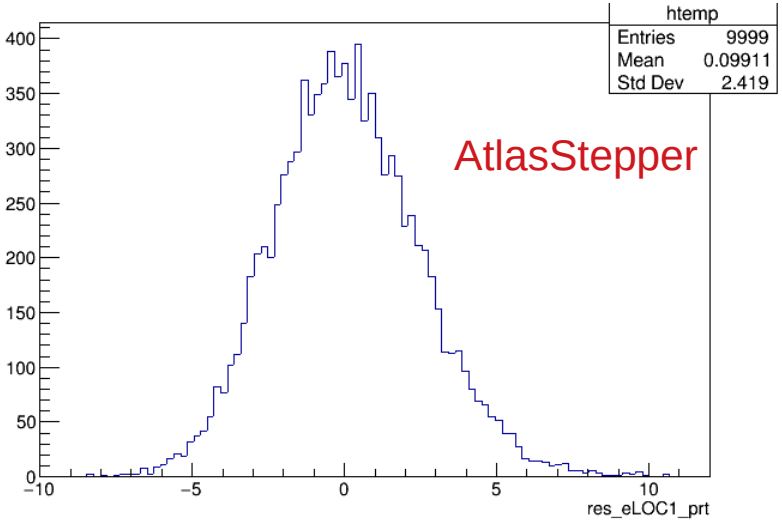
$$K_k = C_k^{k-1} H_k^T (V_k + H_k C_k^{k-1} H_k^T)^{-1}$$
$$x_k = x_k^{k-1} + K_k \left(m_k - h_k(x_k^{k-1})\right)$$
$$C_k = (1 - K_k H_k)\, C_k^{k-1}$$

i.e. the estimated value of θ will always be smaller than θ^T !

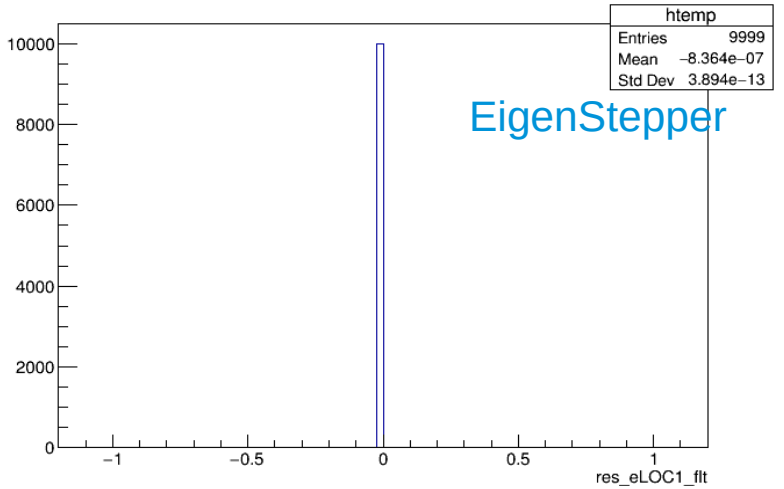
Residual of loc1

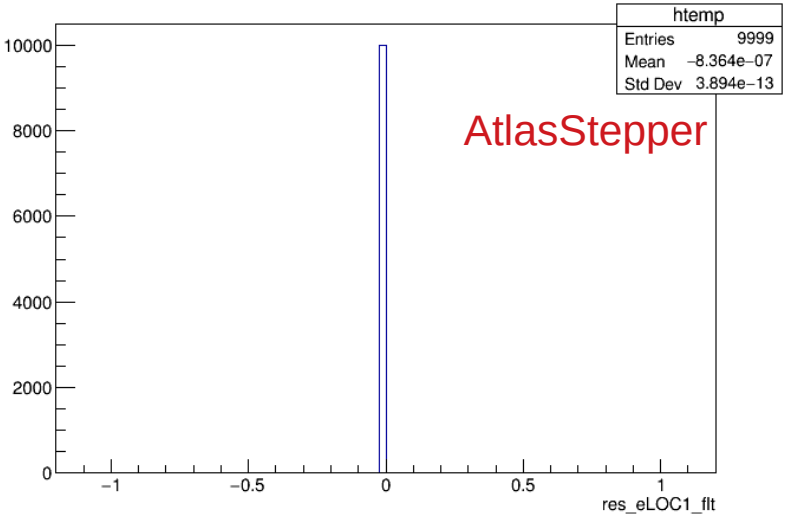
resid_loc1_**pred**





resid_loc1_**filtering**

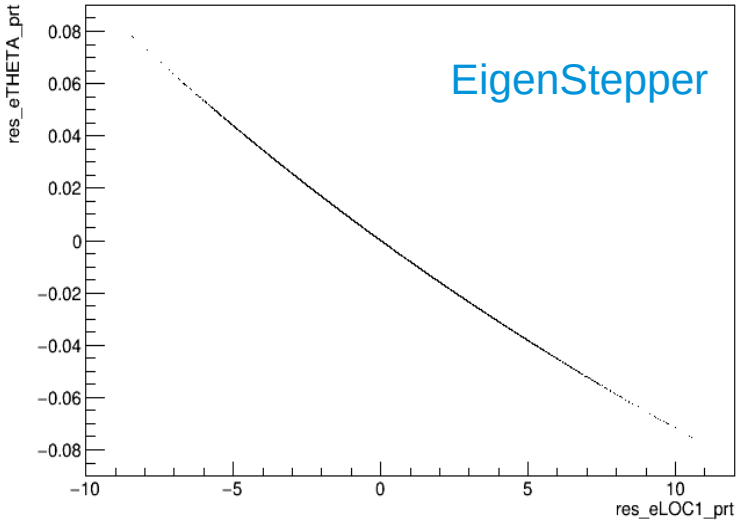




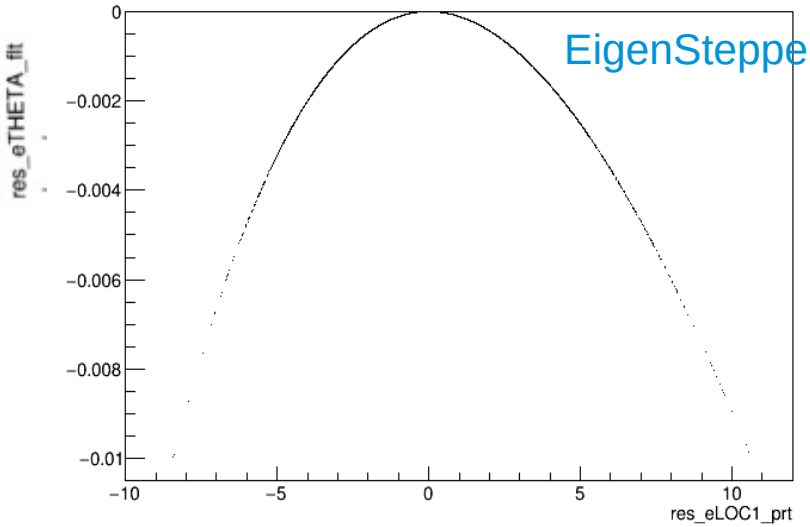
gain matrix K(1,1) = 1
=> filtered loc1 is the same with truth as expected

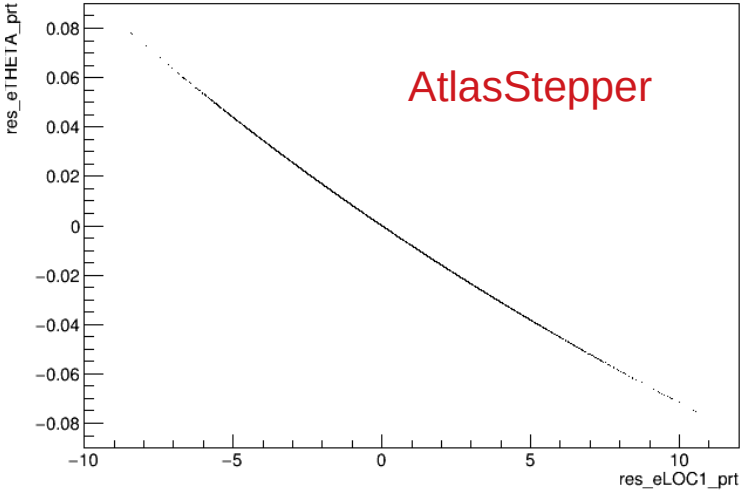
Residual of θ

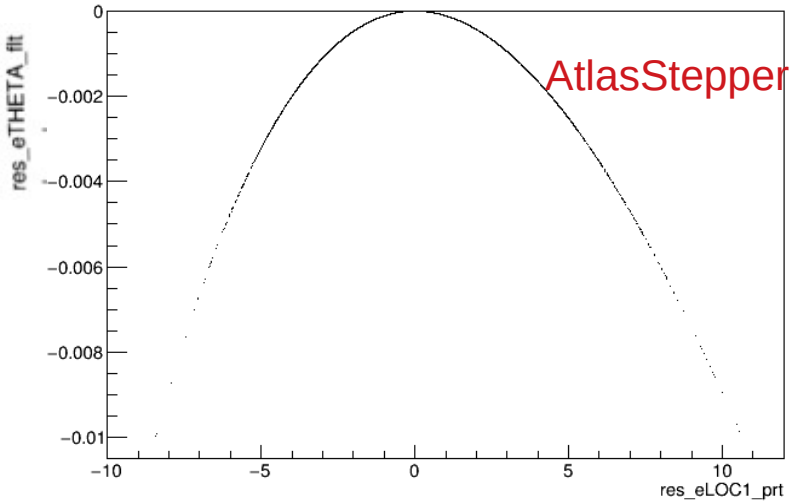
resid_θ_**pred** vs. resid_loc1_pred



resid_θ_**filtering** vs. resid_loc1_pred





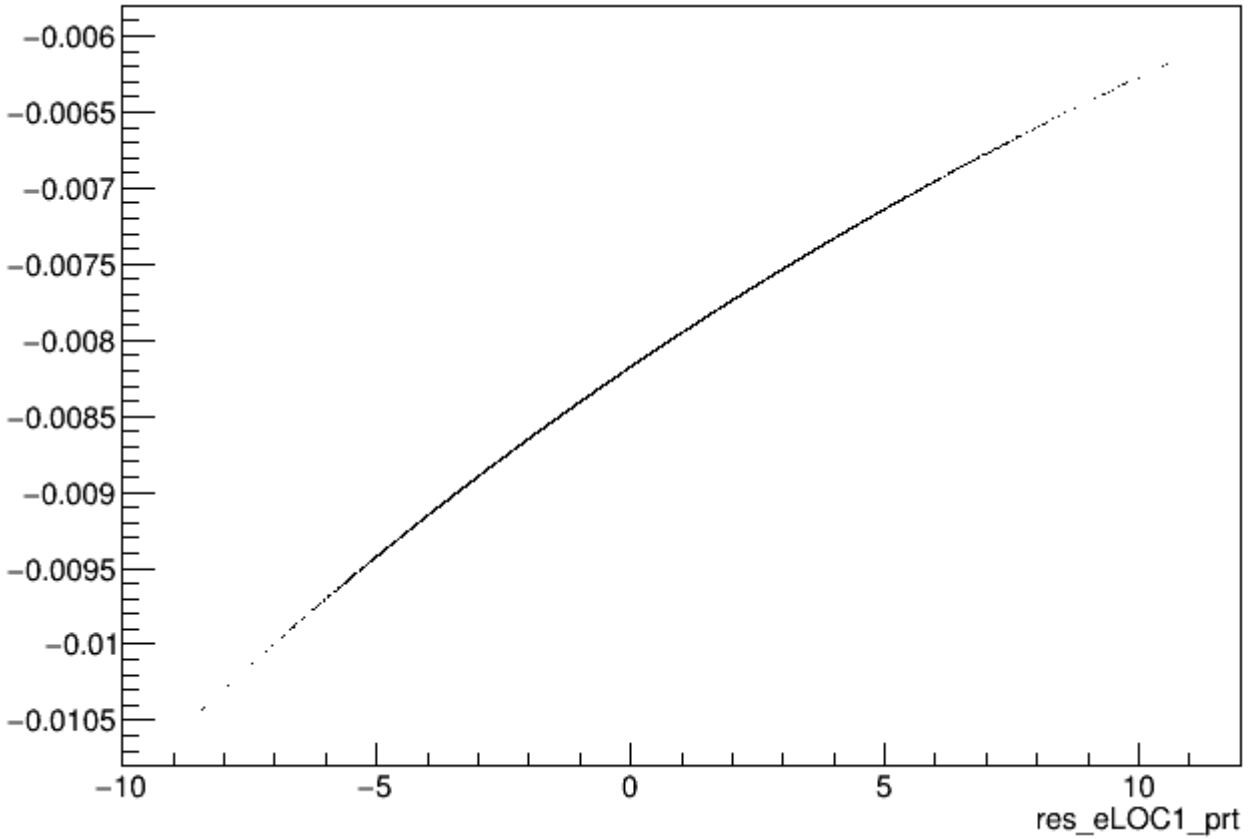


Filtered θ is smaller than truth with no exception (the mathematics tells us the same thing)!

Discussion

- No difference between AtlasStepper and EigenStepper
- The bias of the pull seem to have something to do the non-linear correlation between the track parameters.
  - quite similar to the problem which motivates Runge-Kutta technique?
  - Is this already taken into account in the filtering formalism?

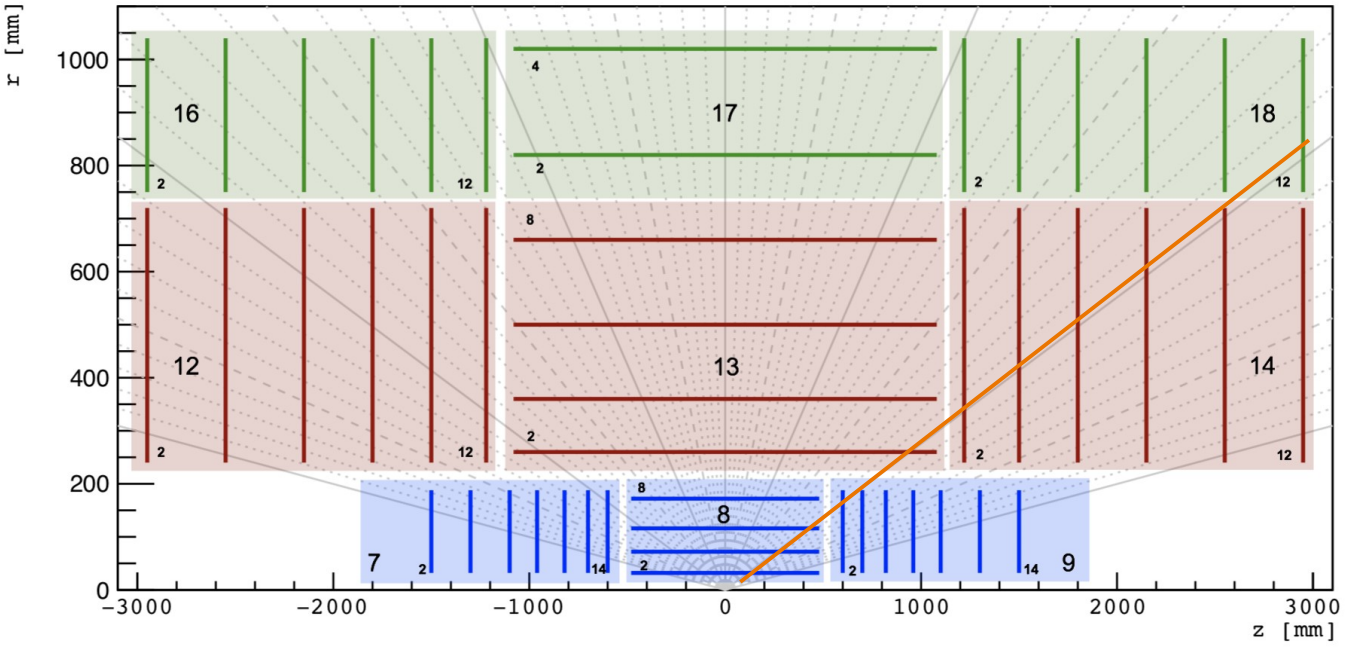
Gain matrix


−0.006
−0.0065
−0.007
−0.0075
−0.008
−0.0085
−0.009
−0.0095
−0.01
−0.0105
−10
−5
0
5
10
res_eLOC1_prt

Global setup

- No material
- No Bfield
- GenericGeometry
- Simulation:
  - 10000 single muon generated with ParticleGun at the same direction, position, pT to disentangle the bias caused by missed measurement surface during propagation (i.e. prediction/filtering won’t be done for those measurements)
- Inputs for an KalmanFitter instance
  - Truth particle parameter (Loc0, Loc1, phi, theta, q/p) smeared with Gaussian is passed to KalmanFitter as initial parameter
  - Truth hits positions (Loc0, Loc1) smeared with Gaussian are passed to KalmanFitter as measurements

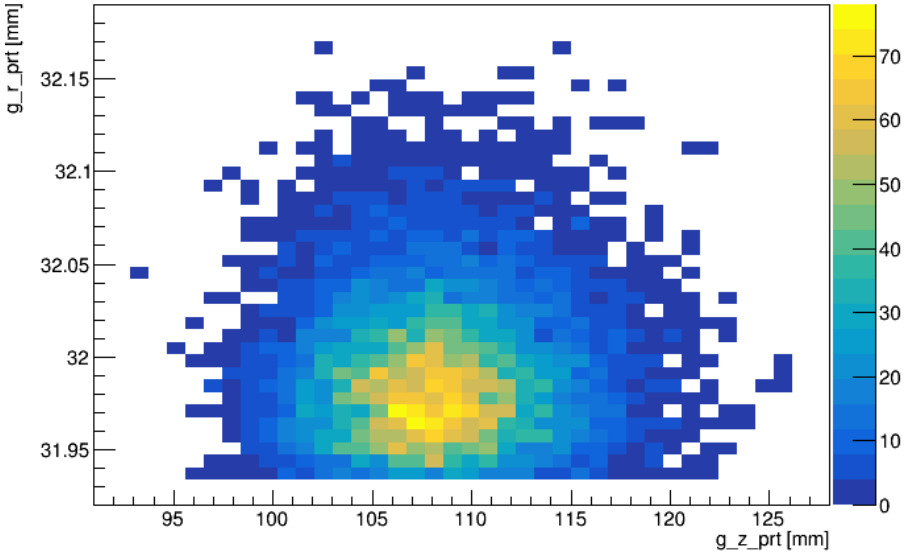
Test 1 setup



d0 = 0, z0 =0
eta = 1.6
phi =0
pT = 10 GeV/c

9995 particles propagated to first detector surface at: Volume/layer/moduleID = 8/2/136

global_r: global_z predicted at first surface



local_x: local_y predicted at first surface