Kinematic Fitting in GlueX

Kei Moriya

2011.06.01

- started looking into kinematic fit code (DKinFit)
- found some issues, possible improvements
- possible bug in translation of tracking matrix into kinematic fit matrix

DKinFit class

- based on CLAS kinematic fit code (CMU)
- written by Matt Bellis, Mike Williams
- the CLAS kfit code has been used in various publications by CMU
- rather straightforward to use
- went through the entire code, did not see anything that seemed different from the CLAS code
- documentation available at (CLAS note) http://www.jlab.org/Hall-B/notes/clas_notes03/03-017.pdf

Implementing DKinFit (1/2)

```
DKinFit *kfit = new DKinFit();
kfit->ResetForNewFit();
kfit->SetVerbose(1);
kfit->SetInitial(kd_initialStateIn);
kfit->SetFinal(kd_initialStateOut);
// kfit->SetMissingParticle(0);
kfit->Fit();
```

vector of DKinematicData

specify mass of missing particle,
comment out if no missing particle

Implementing DKinFit (2/2)

- can pull out various quantities, such as χ^2, ndf,

```
pulls for each variable
chi2 = kfit->Chi2();
ndf = kfit->Ndf();
CL = TMath::Prob(chi2,ndf);
pull = kfit->GetPull(n);
```

nth pull

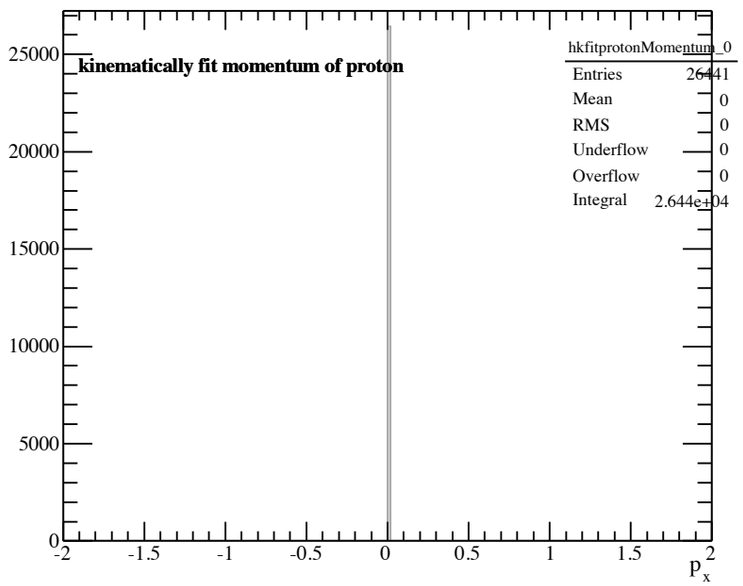
Initial State

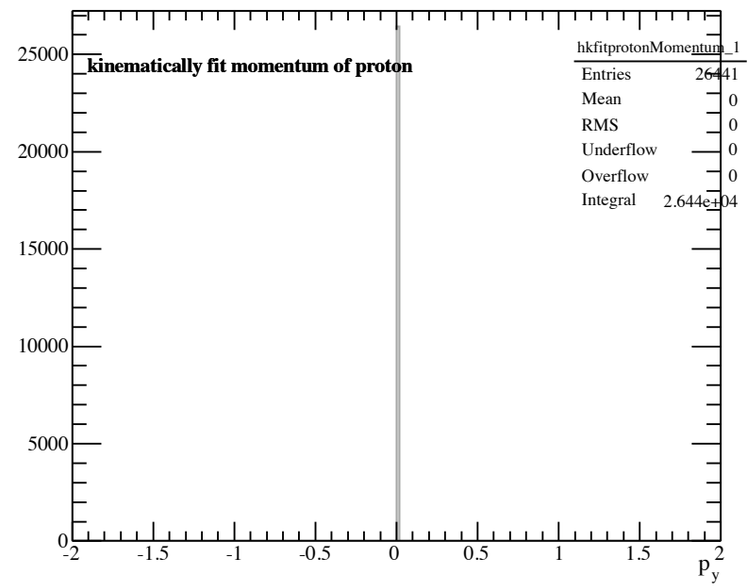
1. smearing of initial momenta is done by smearMCThrownMomentum(smear)
2. smear momentum around original value by smear, and updates error
3. this function does NOT do anything if original momentum is 0.
4. if this is the case, there is NO wiggle room for these variables since the error matrix is 0.
5. in the future, we should probably think of what to do with the error on the initial photon, in relation with the tagger.

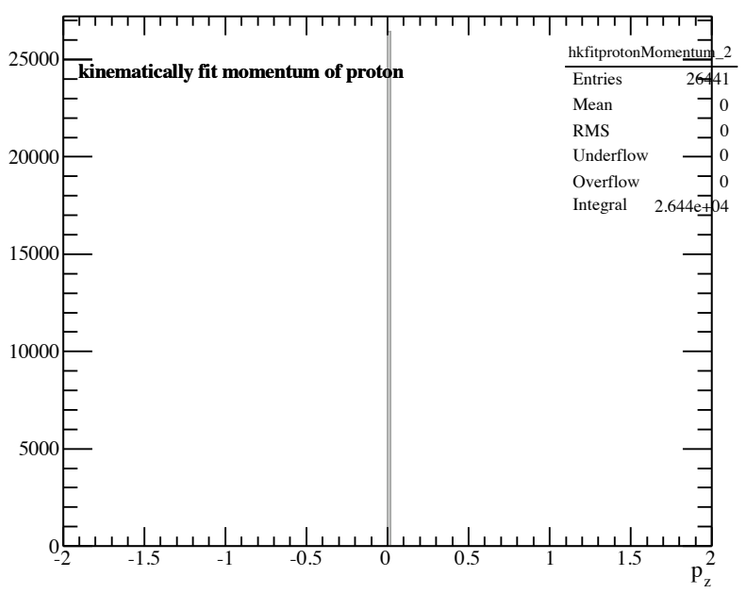
Fitting to p π^+ π^-

1. try fitting to exclusive p π^+ π^- events.

2. generate using genr8, process through hdgenat, then reconstruction





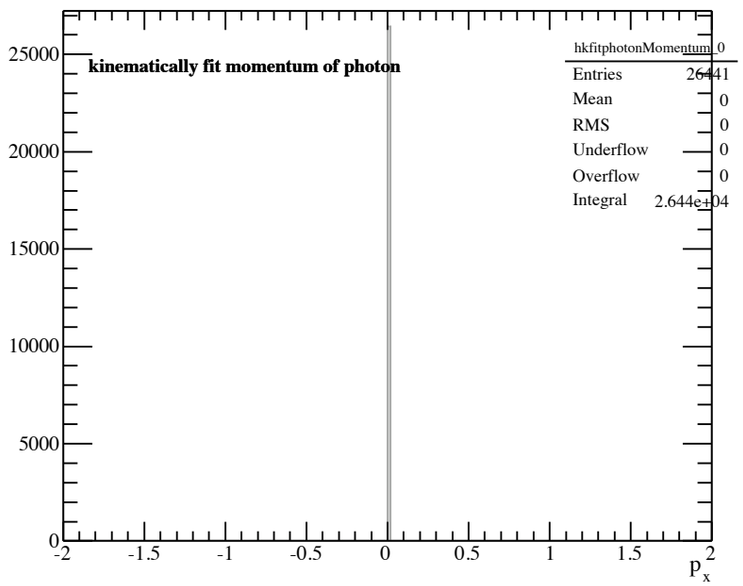


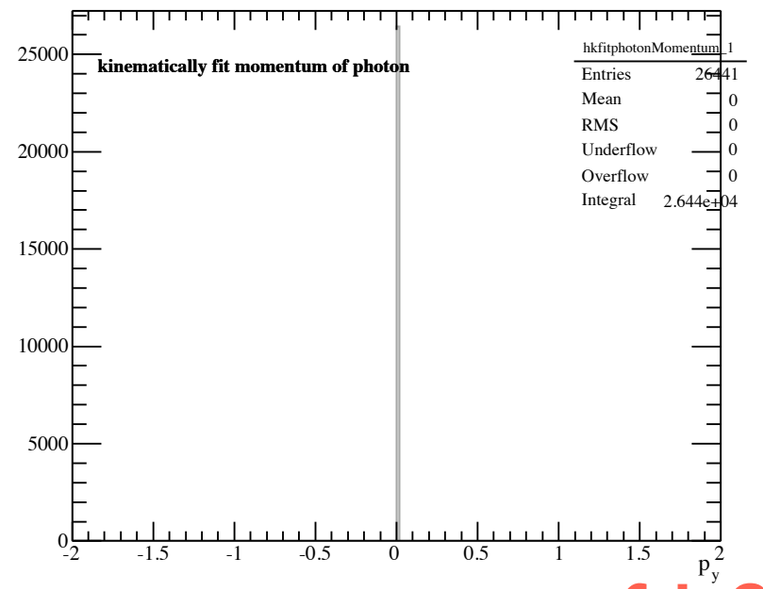
momentum of kfit initial proton does not change from 0 since the error matrix is 0

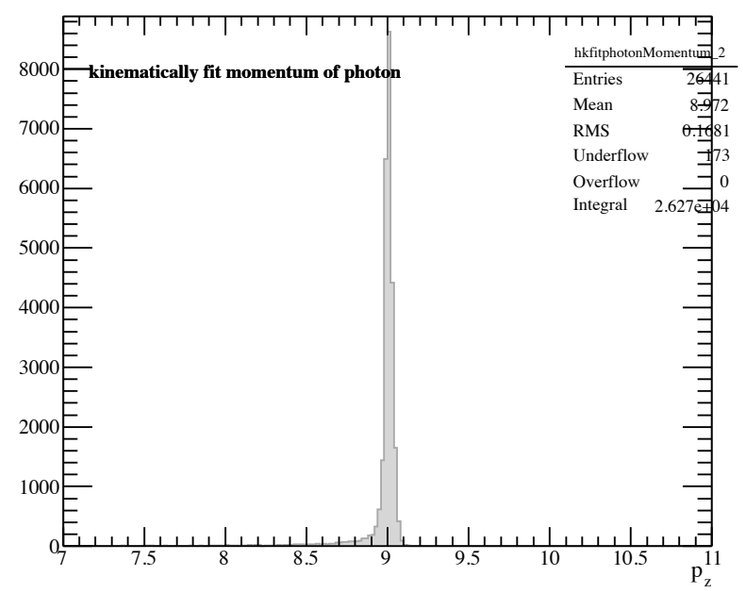
Fitting to p π^+ π^-

1. try fitting to exclusive p π^+ π^- events.

2. generate using genr8, process through hdgenat, then reconstruction





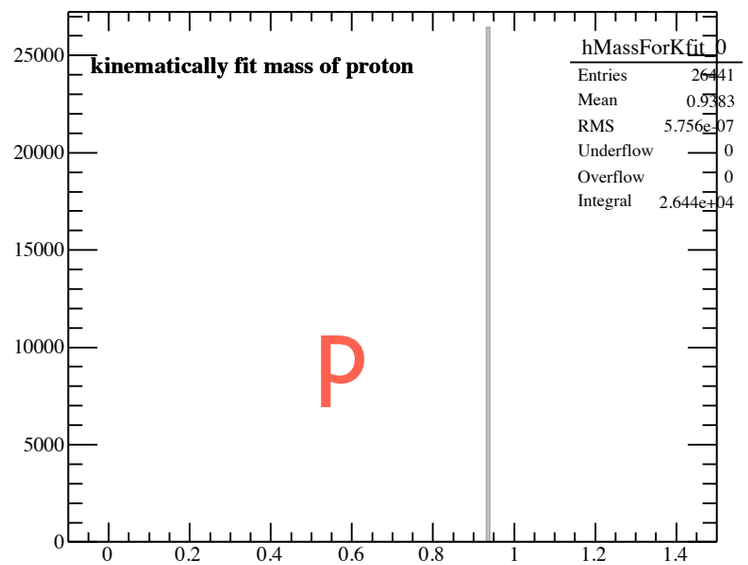


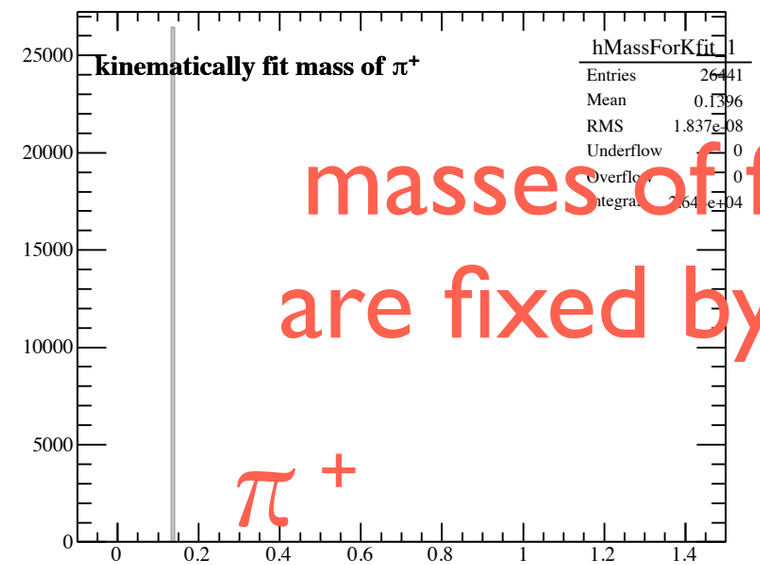
momentum of kfit initial photon changes in the z-direction
(used smear of 0.001)

Fitting to p π^+ π^-

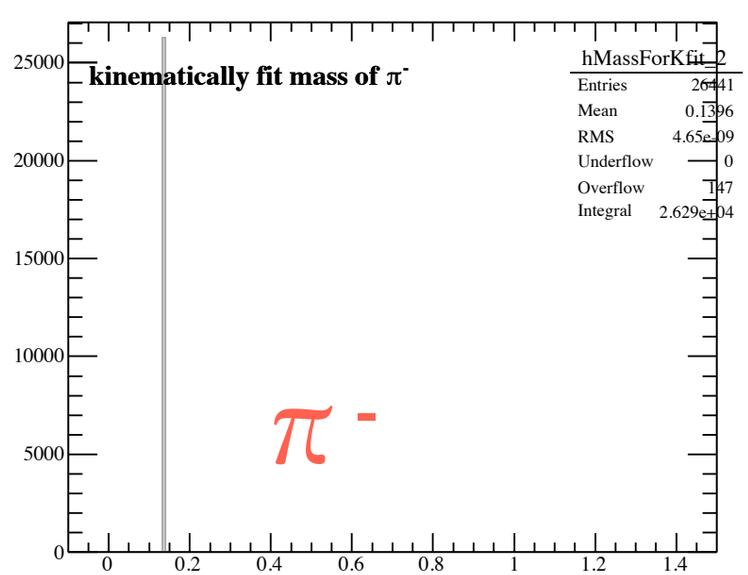
1. try fitting to exclusive p π^+ π^- events.

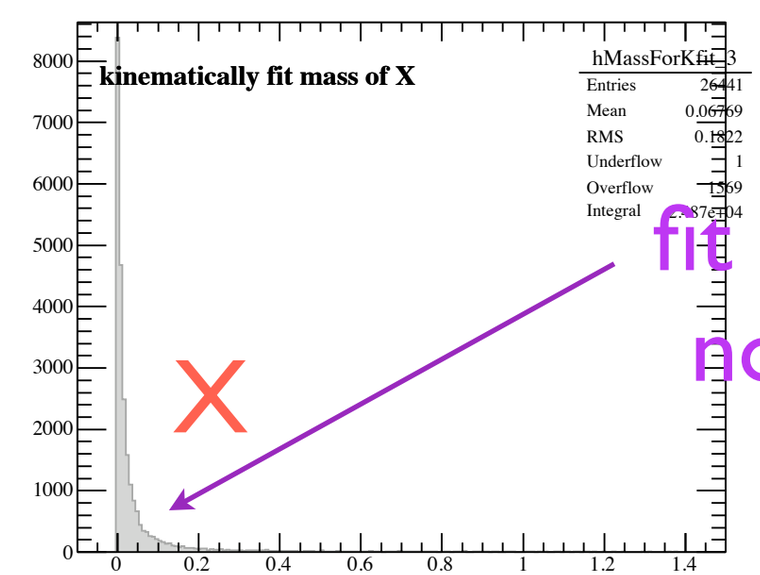
2. generate using genr8, process through hdgenat, then reconstruction





masses of final particles are fixed by kinematic fit



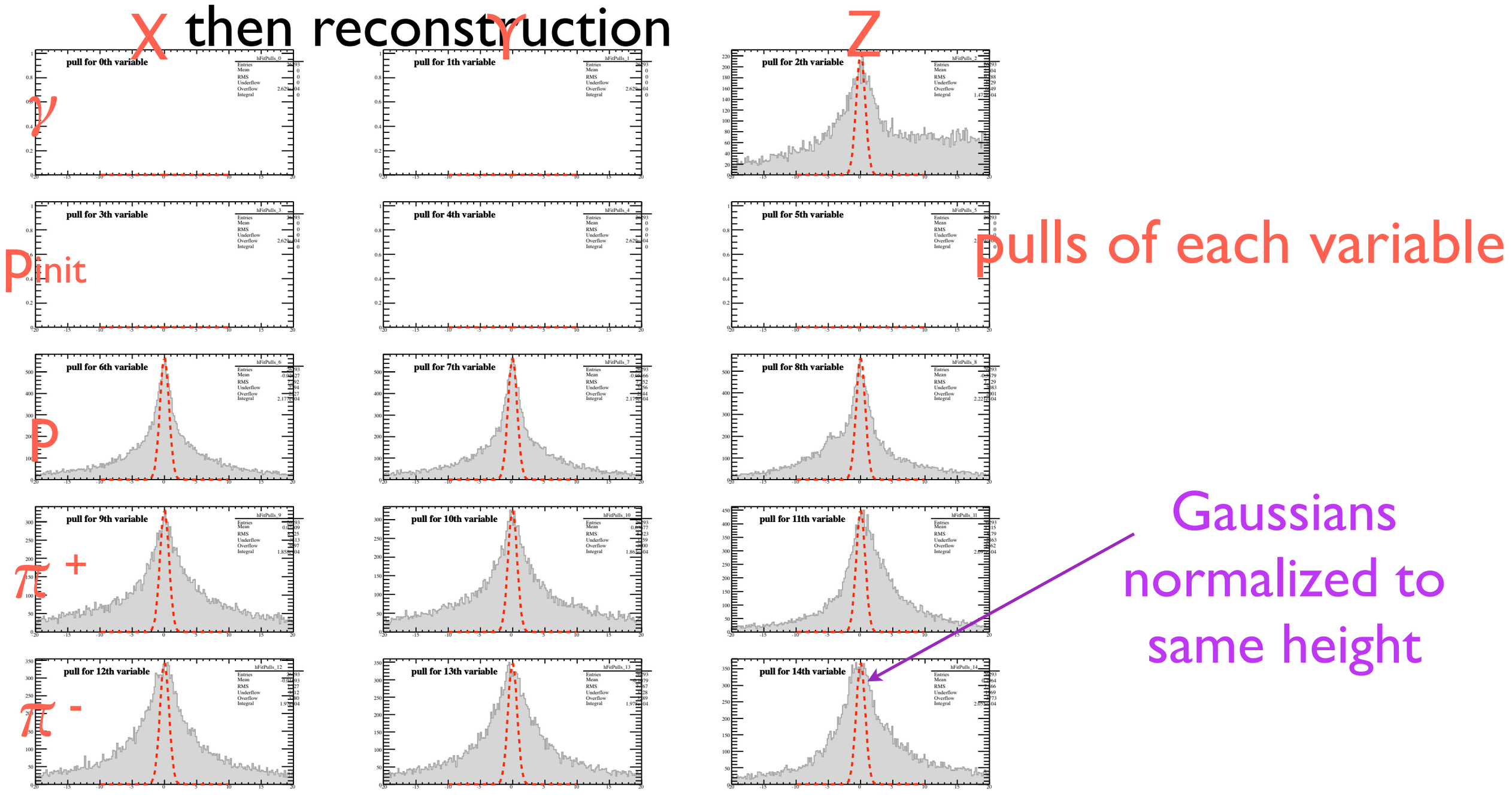


fit probably did not converge

Fitting to p π^+ π^-

1. try fitting to exclusive p π^+ π^- events.

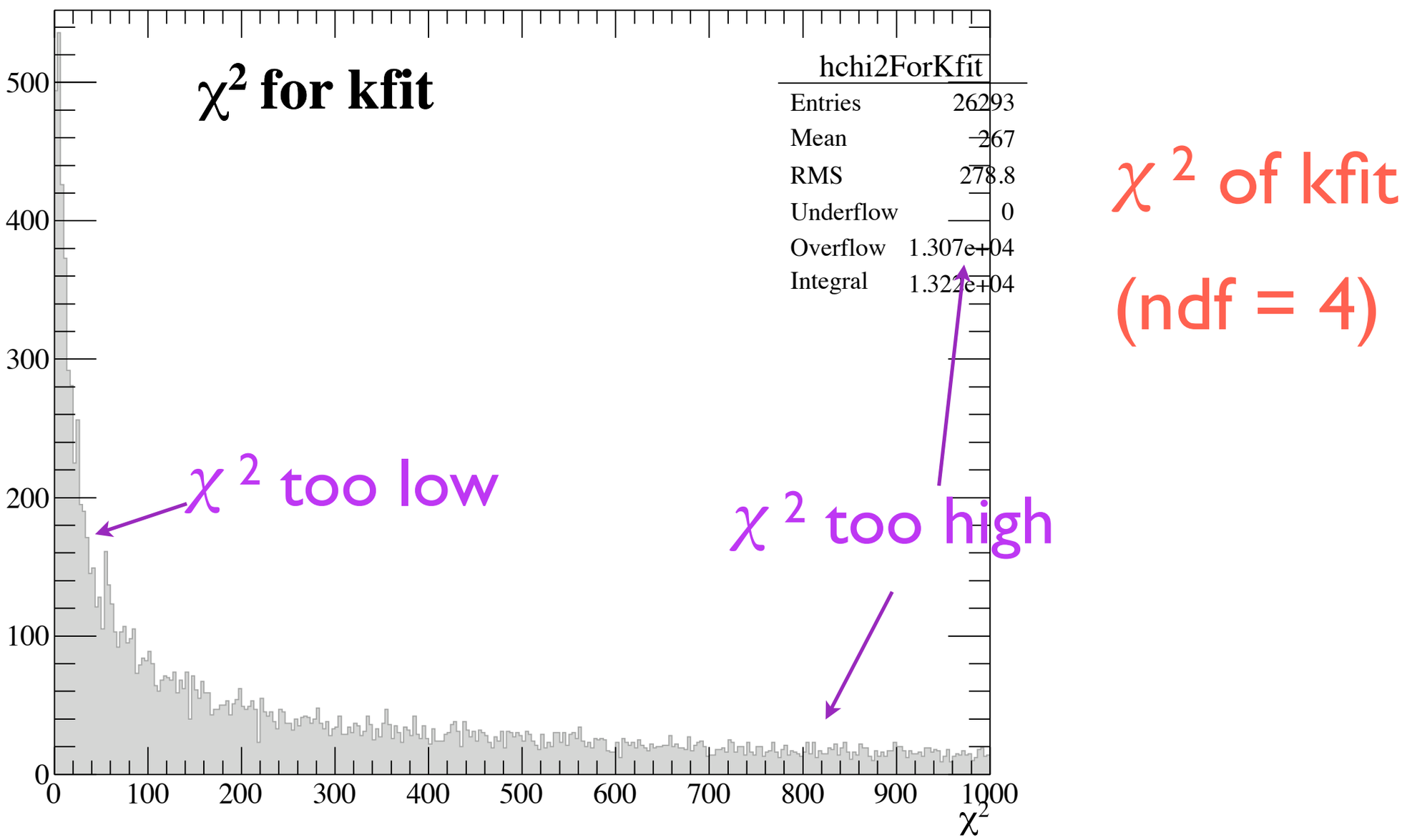
2. generate using genr8, process through hdgenat, then reconstruction



Fitting to p π^+ π^-

1. try fitting to exclusive p π^+ π^- events.

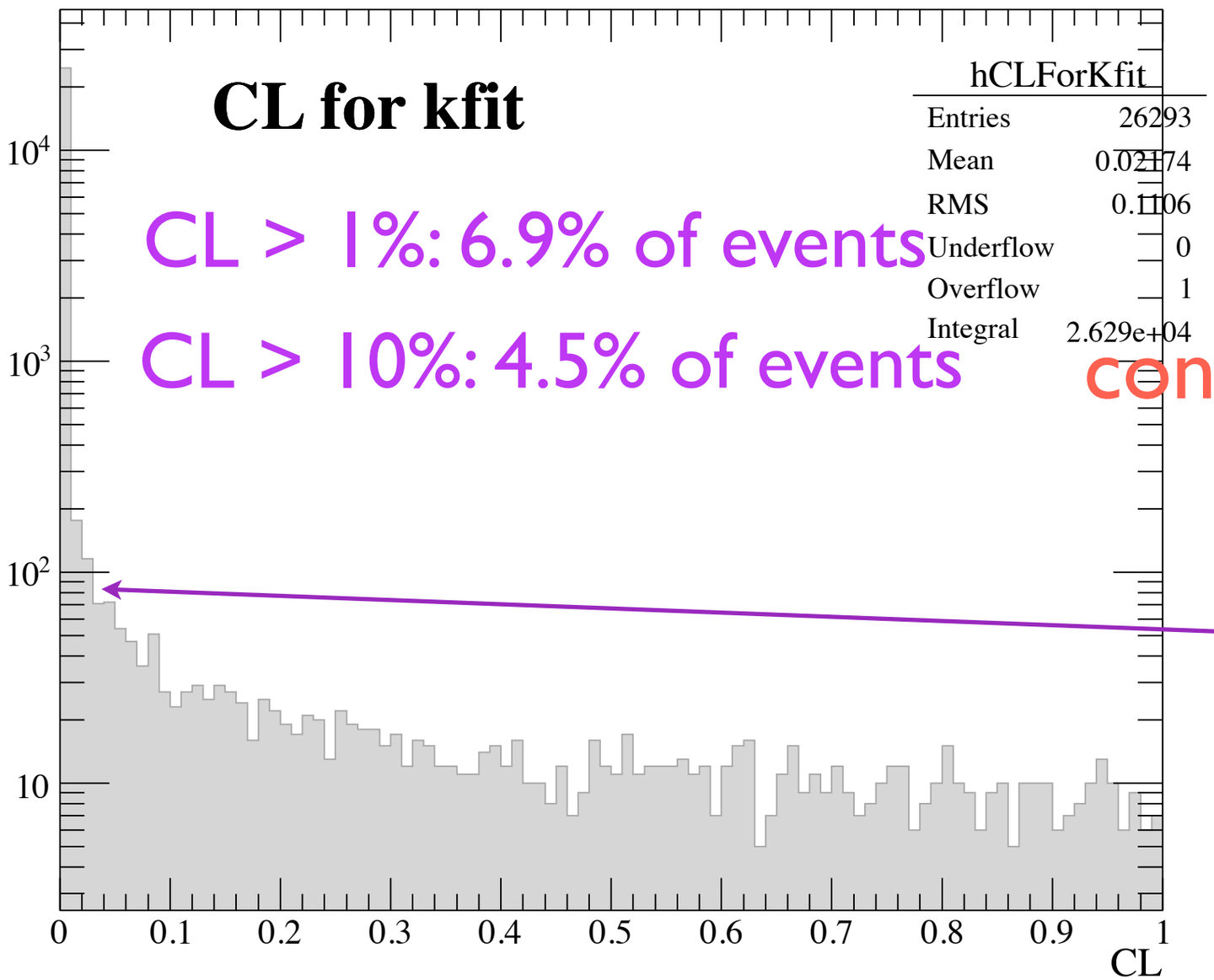
2. generate using genr8, process through hdgenat, then reconstruction



Fitting to p π^+ π^-

1. try fitting to exclusive p π^+ π^- events.

2. generate using genr8, process through hdgenat, then reconstruction



CL > 1%: 6.9% of events

CL > 10%: 4.5% of events

confidence level of kfit

most events have very low confidence level

Possible Improvements

1. set up flag for convergence
2. fitting to neutrals?
3. function that returns how many pulls are available
4. function that uses timing information to add extra degrees of freedom is available, but have not looked into this yet.
5. setting of covariance matrix for missing particle is in CLAS coordinates, meaningless.

Error Matrices

1. Errors of kinematic fit are derived from error matrix of each particle, which is in the $\{p_x,p_y,p_z,E,x,y,z\}$ (DKinematicData) basis.
2. These errors are taken from the track fitting program, where fitting is done in the $\{q/p_T,\varphi,\tan\lambda,D,z\}$ (tracking) basis.
3. Propagation of these errors is handled in DTrackFitterKalmanSIMD::FitTrack(void) where the 5x5 tracking cov. matrix is converted to the 7x7 DKinematicData basis cov. matrix via $C_{DKinematicData} = J\, C_{tracking}\, J^T$, where J is 7x5

Error Matrices

1. The actual code to do this is

```
fitparams.setErrorMatrix(Get7x7ErrorMatrix(errMatrix));
```

member of DTrackFitter of type DKinematicData which holds the tracking fit result

converts 5x5 tracking matrix into 7x7 DKinematicData cov. matrix via J (7x5)

5x5 tracking matrix

2. The matrix J is given within the function DTrackFitterKalmanSIMD::Get7x7ErrorMatrix and each component is $J_{ij} = \partial y_i / \partial x_j$ where y_i is in the DKinematicData basis, and x_j is in the tracking basis.

Error Matrices

I have gone through each component of J, and think there may be two bugs:

1. $\partial E/\partial(q/p_T)$ ← should be units of E^2

Within the code:

J(state_E, state_q_over_pt) = q*pt*p_sq/E3;

dimensionless

should be $-q\, p_T^3\, (1+\tan^2\lambda)/E$

2. $\partial E/\partial(\tan\lambda)$ ← should be units of E

Within the code:

J(state_E, state_tanl) = pt_sq*tanl_/E3;

units of E^{-1}

should be $p_T^2 \tan\lambda/E$