Recent Changes in GENFIT, Update on V^0s

Tobias Schlüter (2015-01-19)
LMU München

Recent Developments in Genfit

1. issues with fits that only use axial wires fixed
2. energy loss formulas corrected
3. smaller `CDCRecoHit`

Axial-wire only fits

Fitting tracks without p_z (i.e. only axial wires) lead to weird jumps

- Kalman fit iterates forward and backward, uses result of previous pass as starting value for next pass
- in particular, the covariance matrix is reused
- in order to not bias the fit, the previous covariance matrix is multiplied by a large factor

If only axial wires are in the track, p_z is not determined. The corresponding covariances remain large.

- these elements would grow from iteration to iteration
- finally hitting numerical limits

Fixed by limiting growth of elements of covariance matrix.

- if someone has time ...a track representation that uses p_T instead of p that is switchable from 4D (i.e. no p_Z) to 5D (full momentum vector) would be even nicer

Energy loss corrected

After redoing the validation plots last year I had observed a bias at low momenta. I finally had the opportunity to investigate. The issue involved a combination of two items:

- a very silly typo
- some wrong algebra in the transformation from energy loss to momentum loss

These items were fixed, and the bias is gone. It also improved the situation for the FOPI experiment who are using very slow tracks (protons below 100 MeV).

- Bethe-Bloch formula still looks weird next to PDG
- FOPI guys (S. Dorheim) are looking into it

Smaller `CDCRecoHit`

The current `CDCRecoHit` is unnecessarily large.

- it stores a 7×7 covariance matrix with mostly zeros
- because, in terms of storage, wire positions are treated as measured values with zero error

Fairly wasteful. In order to prepare the new `CDCRecoHit` (which includes correct drift-time treatment), I tried to fix this.

- Failed to do this in a backwards-compatible manner.

Measured values are `protected` members of the base class `genfit::AbsMeasurement`, their use is defined in the intermediate class `WireMeasurement`.

- no way to define schema evolution rule that replaces the intermediate class

So the new `CDCRecoHit` will need a new name.

Improvement

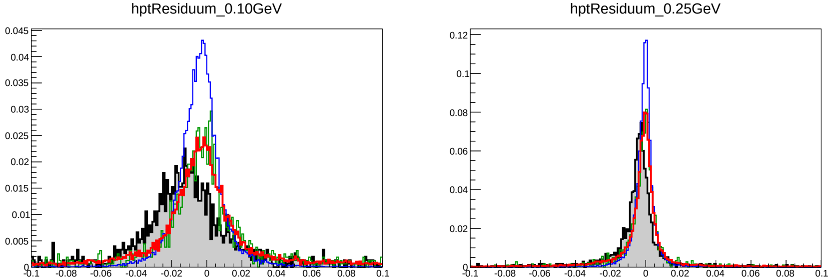

Figure : p_T residuals for $p_T = 100\,\mathrm{MeV}$ (left) $p_T = 250\,\mathrm{MeV}$ (right) with different versions of the software. Black: old reference. Green: Current software, default statistics. Red: Current software, 10× increase in statistics. Blue: Current software, 10× increase in statistics, MC reco.

There are other differences, and I will address them in the following.

Comparison to MC-based reconstruction

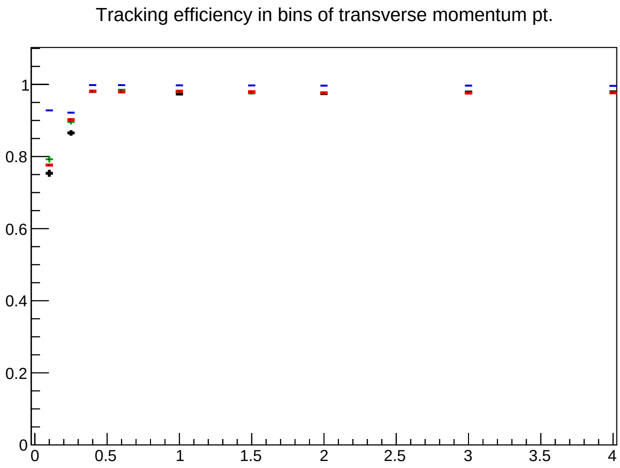

Figure : Tracking efficiency as function of p_T with different versions of the software. Black: old reference. Green: Current software. Red: Current software, 10× increase in statistics. Blue: Current software, 10× increase in statistics, MC reco.

Comparison to MC-based reconstruction II

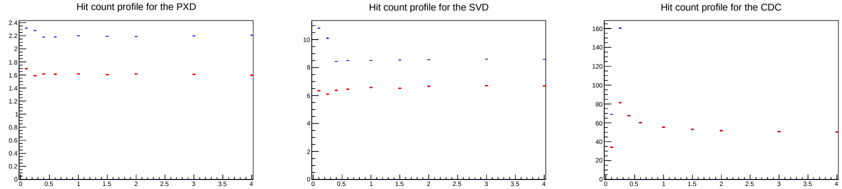

Figure : No. hits associated with reconstructed track for diff. p_T with different versions of the software. (L to R: PXD, SVD, CDC; counting s.t. two SVD hits per layer.) Red: Current software, 10× increase in statistics. Blue: Current software, 10× increase in statistics, MC reco.

(Overshoots in lowest bins due to curlers.)

- CDC track-finder (Trasan) is almost 100% efficient, except for curlers (red points fall on blue points in rightmost plot)
- VXDTF on the other hand misses roughly one third of hits
- this probably explains improved resolution and higher efficiency of MC-based reco (but ...)

Closer look at efficiencies

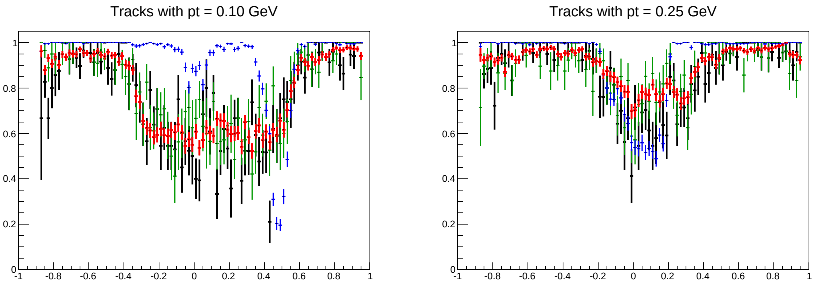

Figure : Efficiency as function of $\cos\theta$ for $p_T = 100\,\mathrm{MeV}$ (left) $p_T = 250\,\mathrm{MeV}$ (right) with different versions of the software. Black: old reference. Green: Current software, default statistics. Red: Current software, $10\times$ increase in statistics. Blue: Current software, $10\times$ increase in statistics, MC reco.

Note that the efficiencies of MC reco drop below the real reco???

Back to this plot

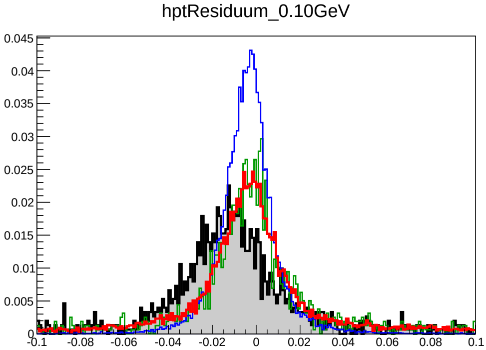

Figure : p_T residuals for $p_T = 100\,\text{MeV}$.

There's a bias in the MC reco! Note that this is after extrapolation back through the beampipe. My working hypothesis is that large multiple scattering and the VXDTF don't interact nicely.

Update on V^0 finder

1. introduced new cuts, as requested by software group (mass cut inside beam pipe, different χ^2 cuts inside and outside)
2. now throws lots of warnings because `mdst::V0` was updated with no synchronisation (such things shouldn't happen)
3. unfortunately, P-values look as bad as they looked at last B2GM (actually, exactly the same as far as I can tell)

A few quality indicators Old Slide

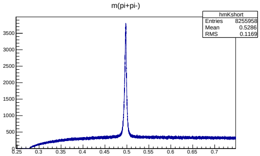

Figure : K^0_S mass peak

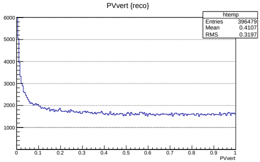

Figure : P-val of true K^0_S vertices. 50e3 in 1st bin!

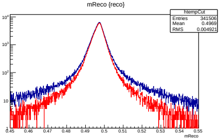

Figure : K^0_S mass peak with cut P-value $> 1\%$

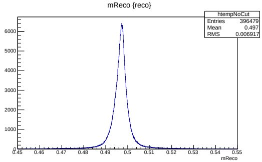

Figure : Mass of matched true K^0_S. $\sigma_1 = 1.4\,\mathrm{MeV}$, $\sigma_2 = 3.7\,\mathrm{MeV}$ + wings

- left plots: sharp K^0_S peak reconstructed
- middle plot: P-value of true K^0_S is flat except for $\approx 13\%$ in first bin (this is not good, would happen if 6.7% of tracks reconstructed badly)
- right plot: low P-value correlates with bad mass reconstruction
- (not shown) very little correlation of this bad feature with low p_T of K^0_S or $\pi^\pm$