

FTKSIM with rahits brief updates

Anton Kapliy et al
University of Chicago
November 13, 2007

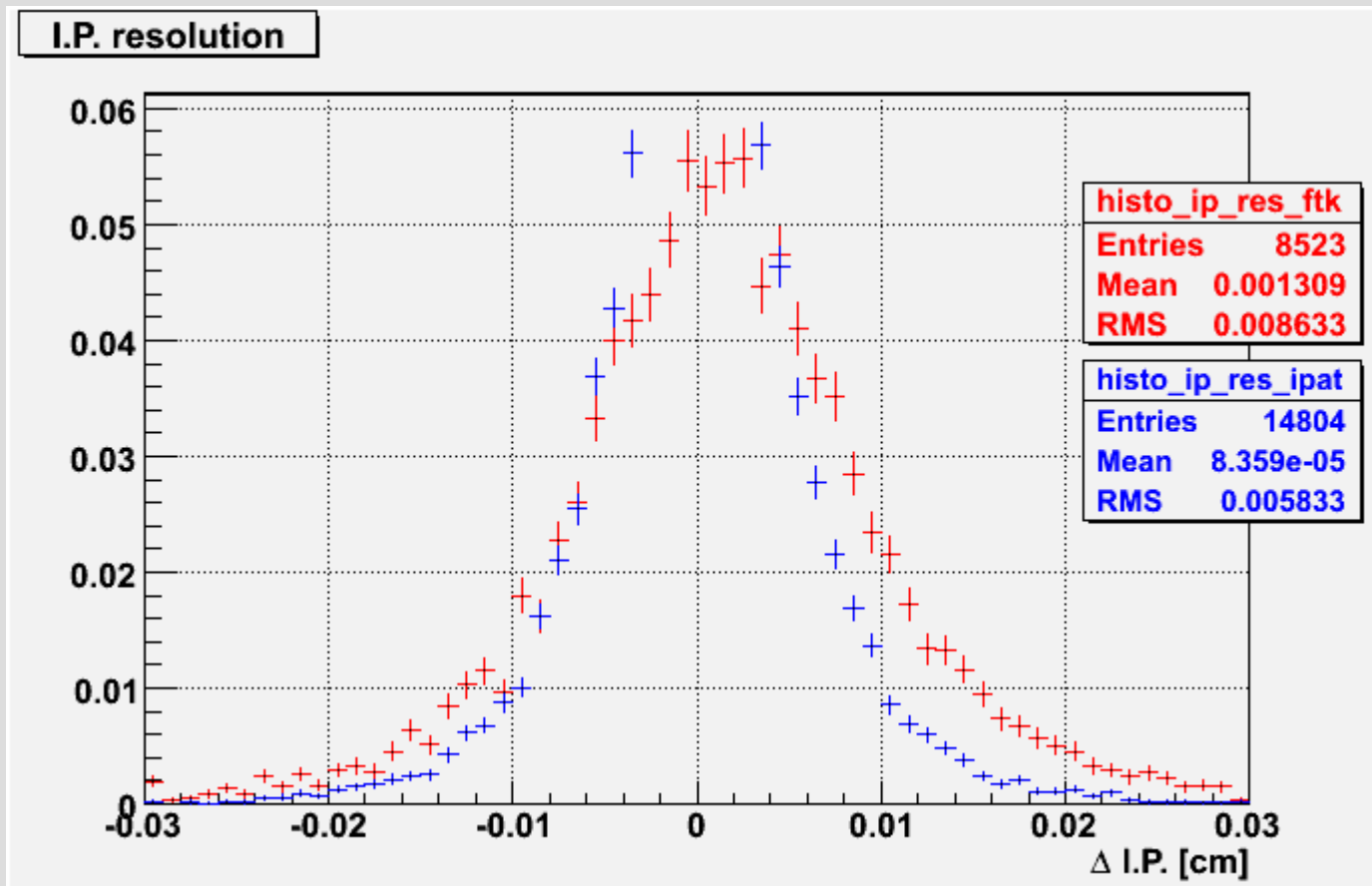
Overview

- Ftksim now works on gzipped data
 - Got rid of lseek/gzseek calls, as they are slow
- Looked at some of the issues from last time
- Bank efficiencies
- IP asymmetry: thought bad hit sorting
 - Put in bulletproof sorting (module, then strip)
 - No change at all
- Z0: looks like ipat is much better in bsmumu
 - Added some corrgen plots
- Phi dropping at +/- Pi
 - Identified an issue -> fixed somewhat

Pattern bank efficiencies (1st look)

- ~50% of training tracks fail filterHits
 - I.e., dirty tracks not fit for training purposes
- Tested pattern efficiency on independent set of single-muon tracks (non-dirty):
 - 70% of clean events matched to a patt
 - 30% of clean events didn't match to a patt
- This was done via pattgen's eff_Flag mode
- SS = 3x5; training from 100M tracks
- Size = 4M patterns / bank; 300 MB on disk
- May explain some of the lower efficiency in last week's plots

IP asymmetry

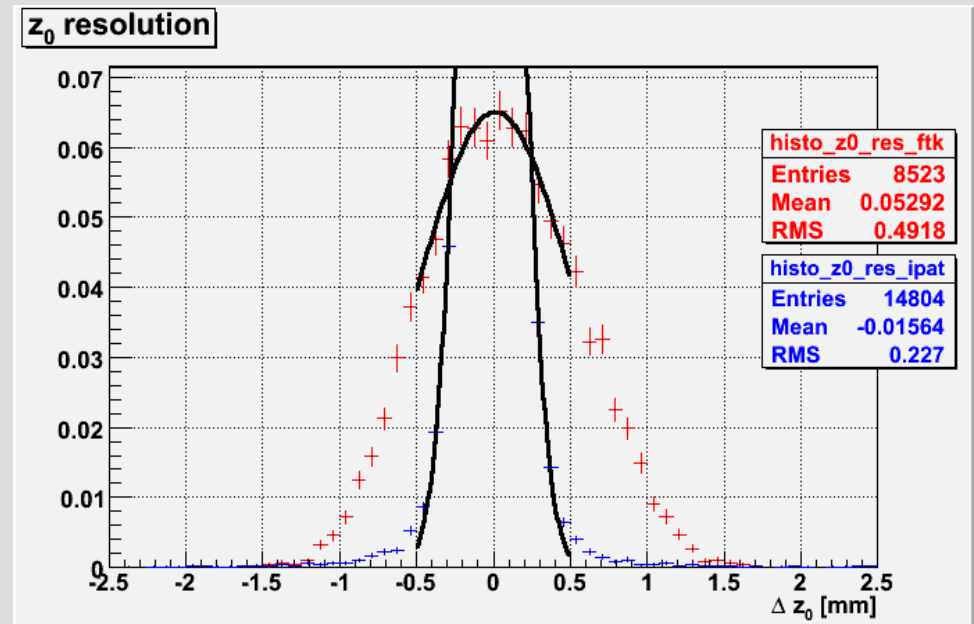
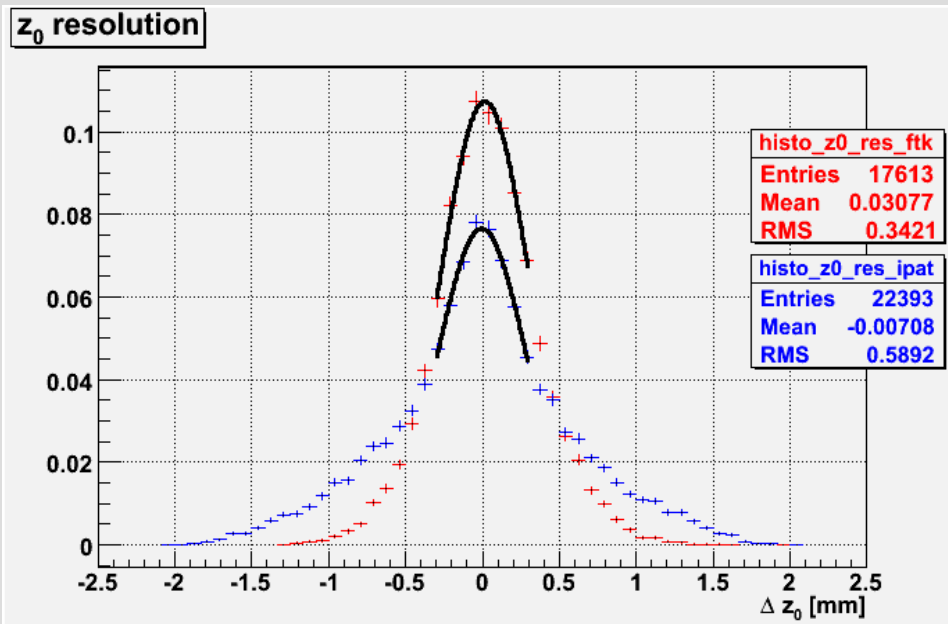


Suspected that this is due to bad sorting, but it's still present!

Z0 resolution

Training sample

bsmumu



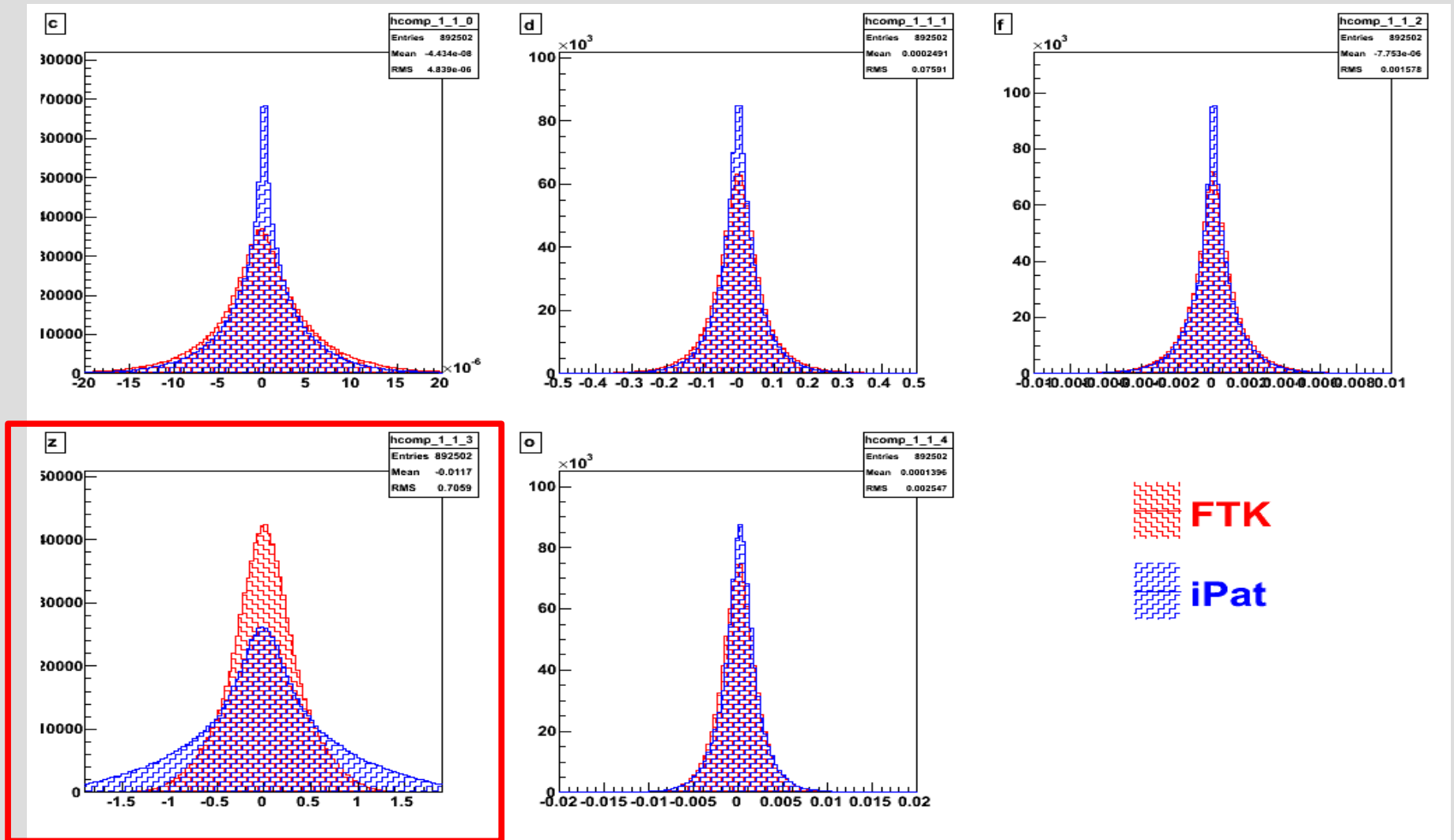
Sigmas from the Gaussian fit:

FTK: 0.29 mm
IPAT: 0.28 mm – looks very non-gaussian

FTK: 0.5 mm
IPAT: 0.17 mm - got a lot better

I'm not sure these number are of any use... But in any case, IPAT looks a lot better with bsmumu sample (compared to single muon)

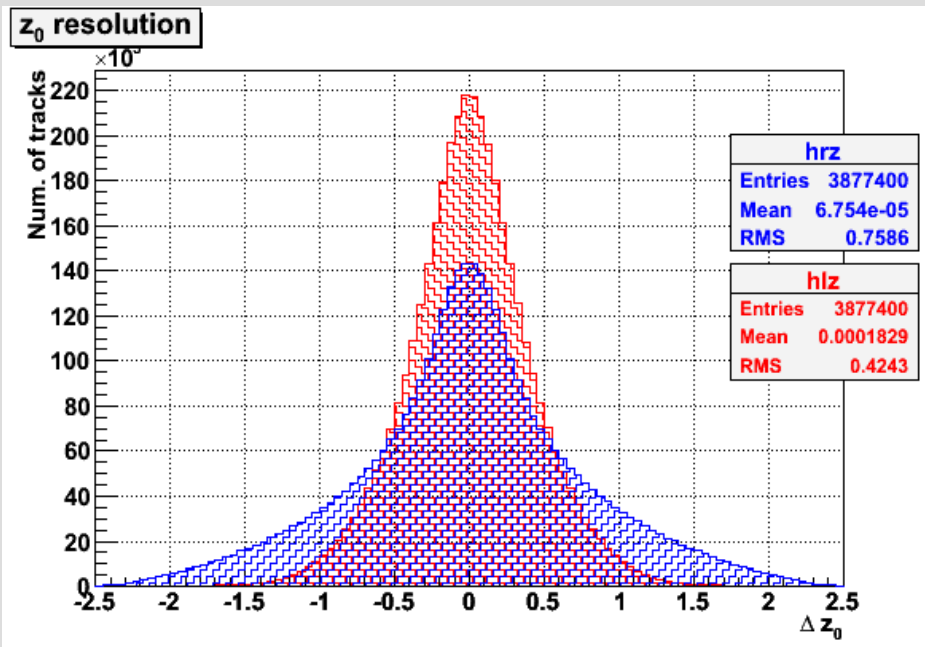
Z0 resolution: from corrgen



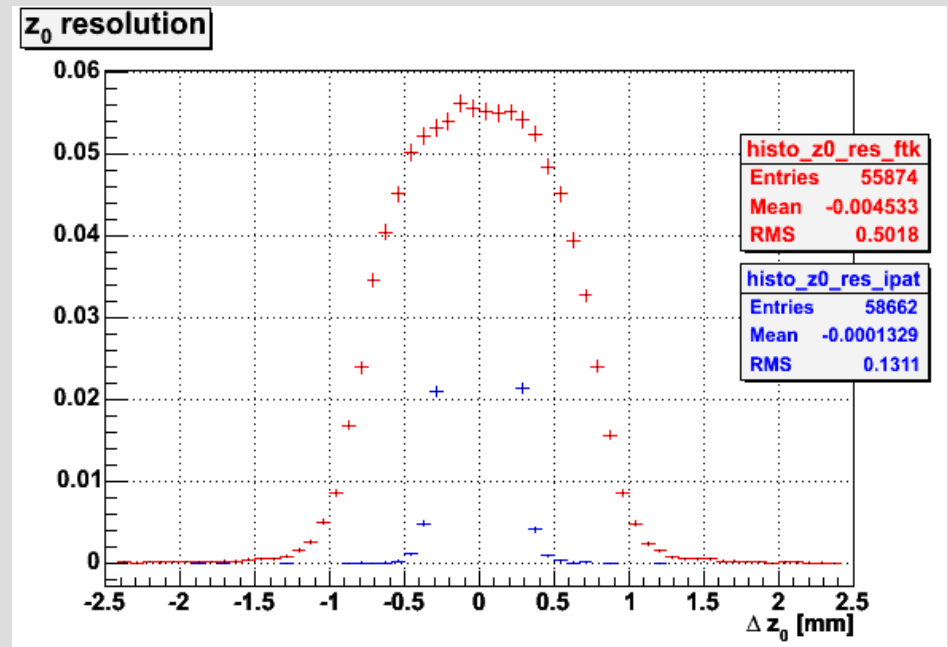
Same as left plot on previous page (training data)

Z0 resolution: Guido's v 0.7.1

Training (from corrigen)



Bsmumu (from ftksim)



Overall similar effect to what I see (ipat becomes a lot better on bsmumu)

curv

d

phi

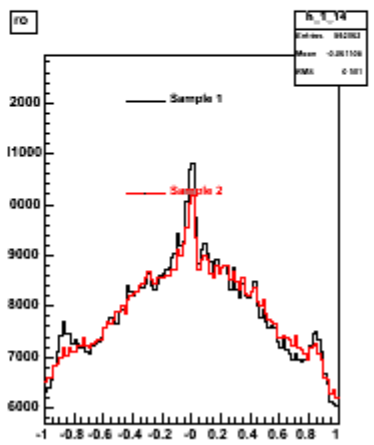
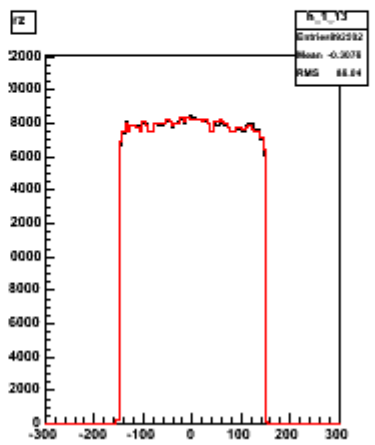
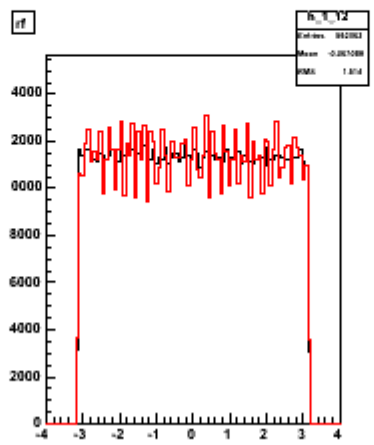
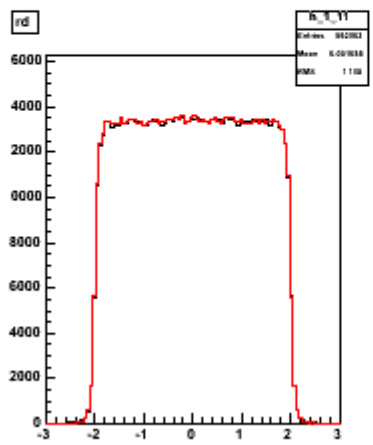
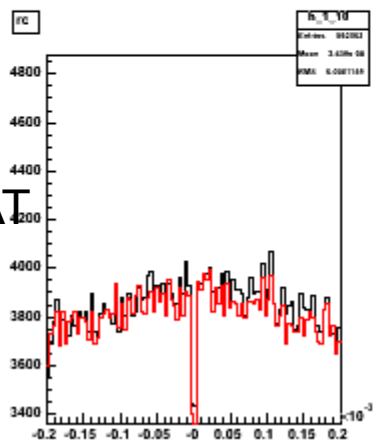
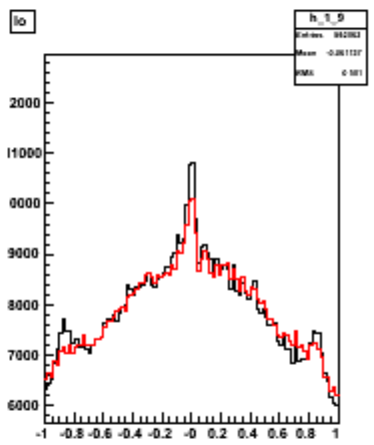
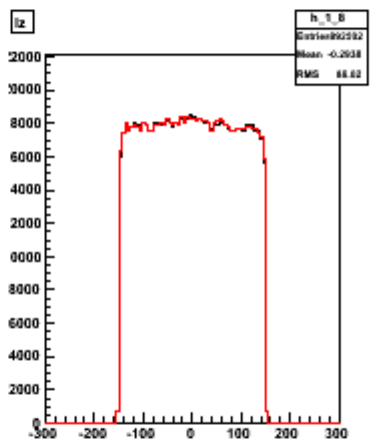
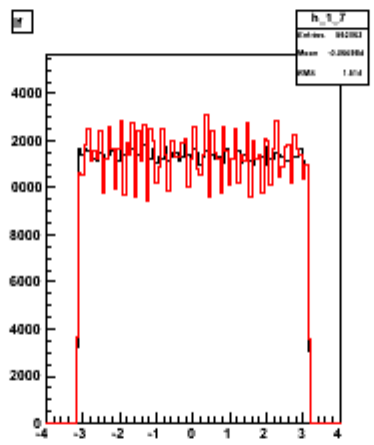
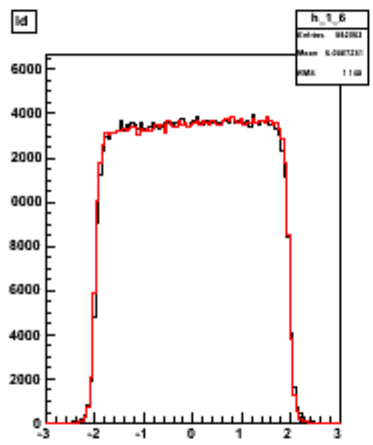
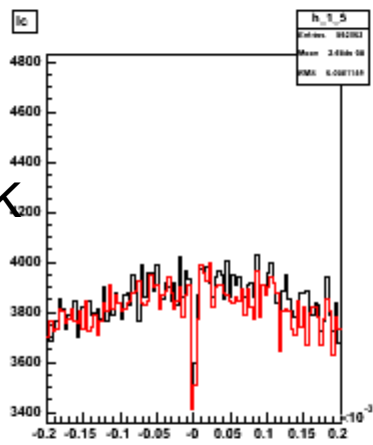
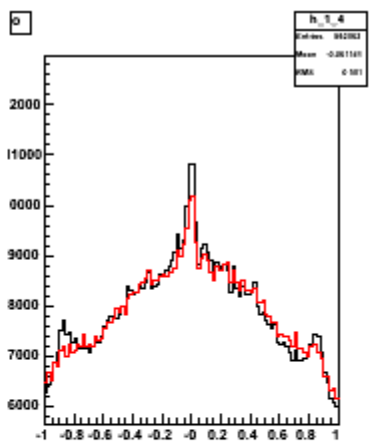
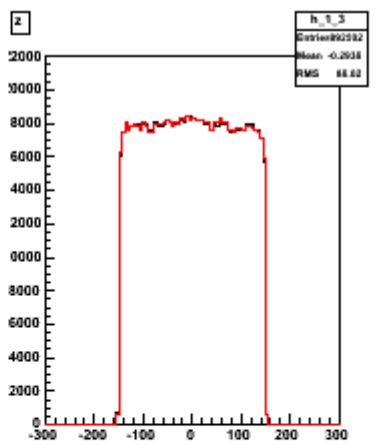
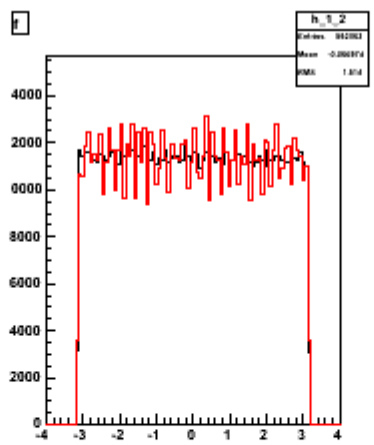
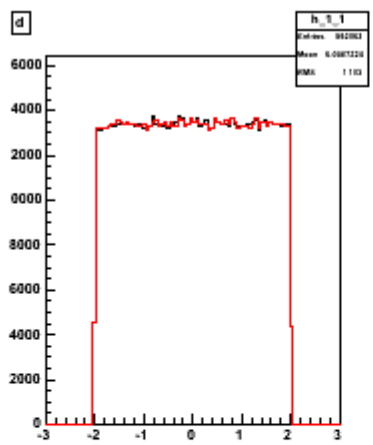
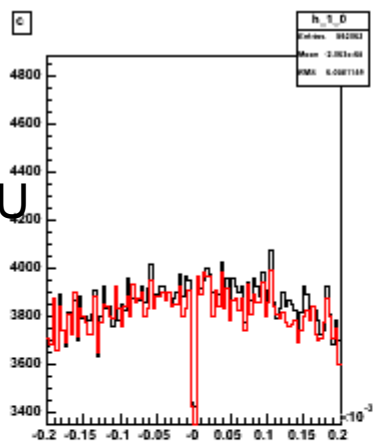
z

cot(theta)

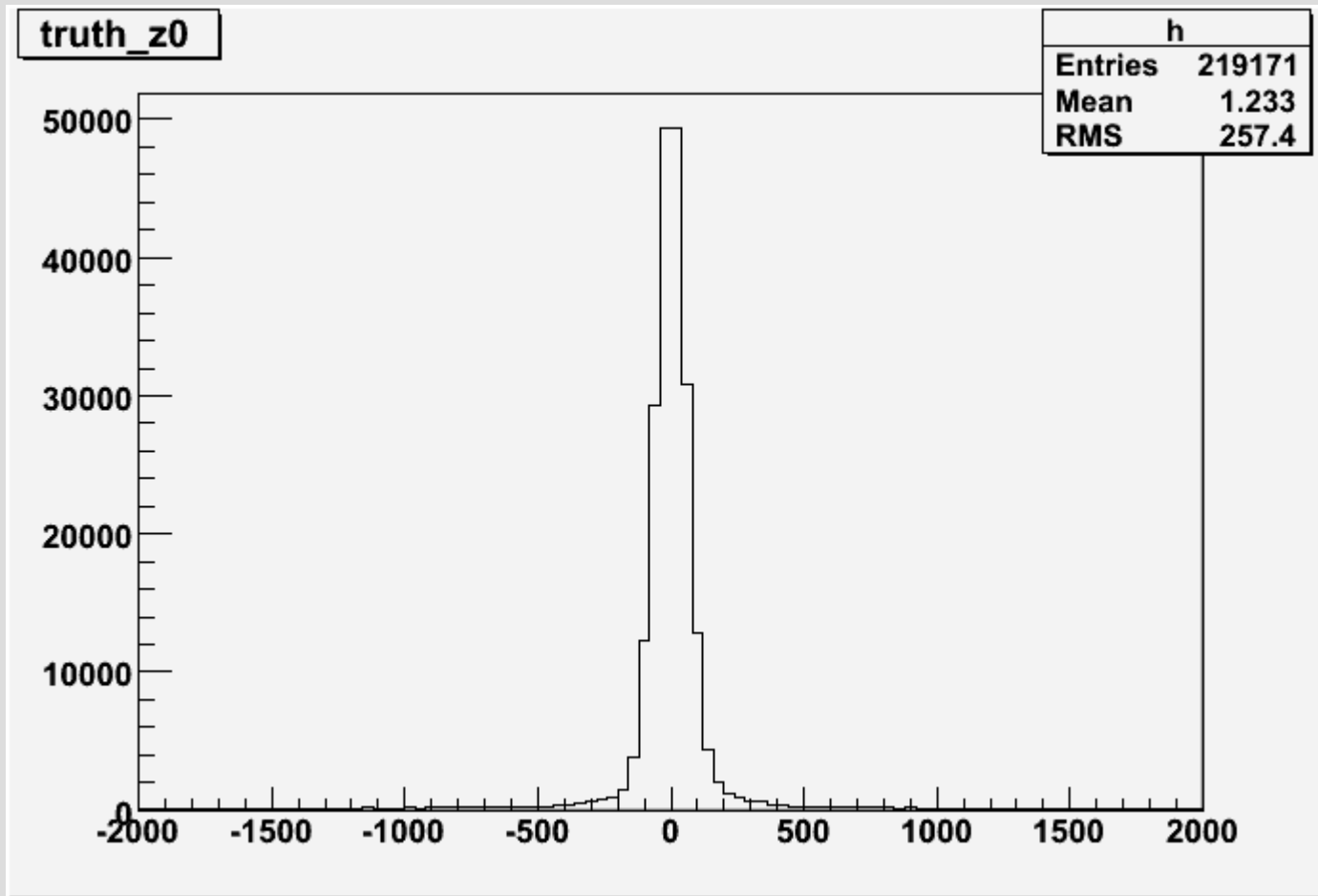
TRU

FTK

IPAT



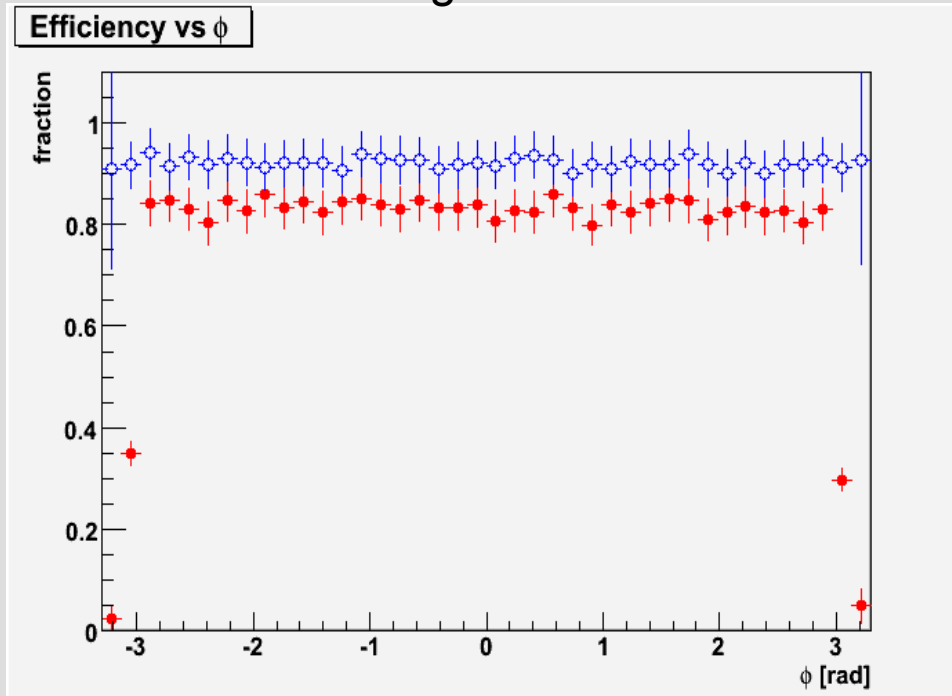
Z0 for bsmumu (truth)



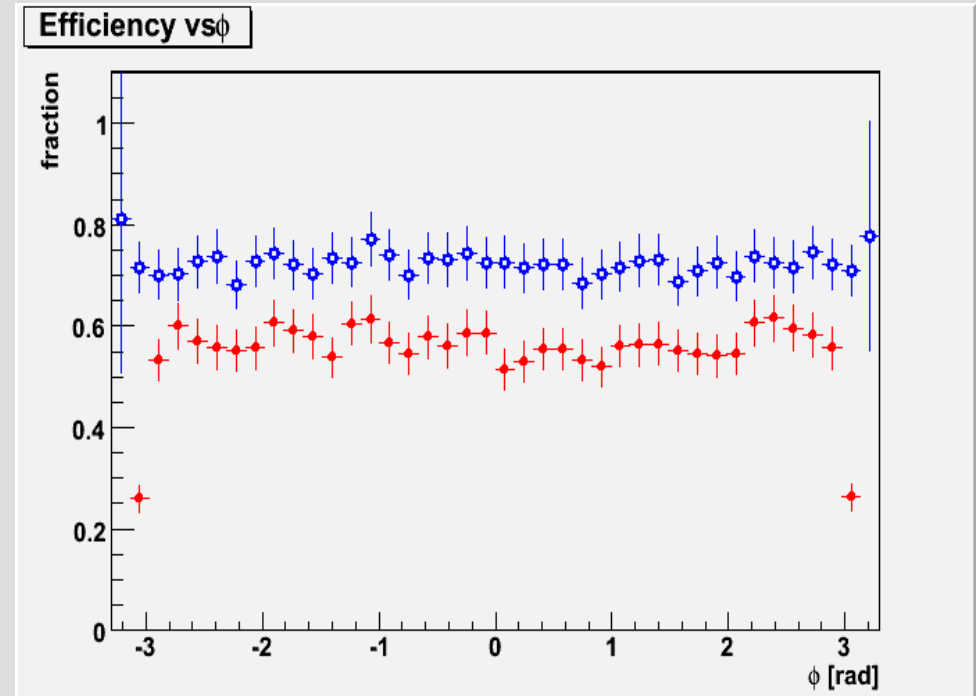
Phi eff drop at +/- PI

FROM LAST WEEK:

Training data



Bsmumu



(One of) sources of problem (FC)

$$P[i] = C[i] + Q[i][j] * X[j]$$

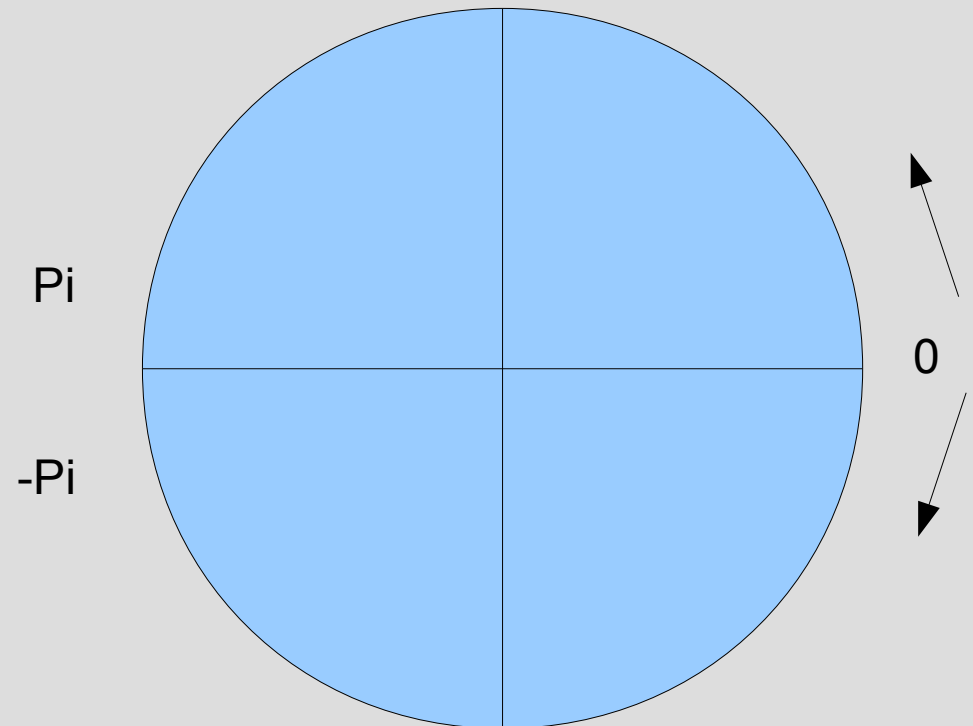
$X[j]$ = 14 coordinates
(3 * 2D PIX + 8 * 1D SCT)

$P[i]$ = 5 track parameters
(curv, phi, cot, z0, d)

$C[i], Q[i][j]$ are computed per sector
from training data in corrgen

For a sector around P_i , some tracks
have $-P_i$, some tracks have $+P_i$.
During averaging, it's screwed up.

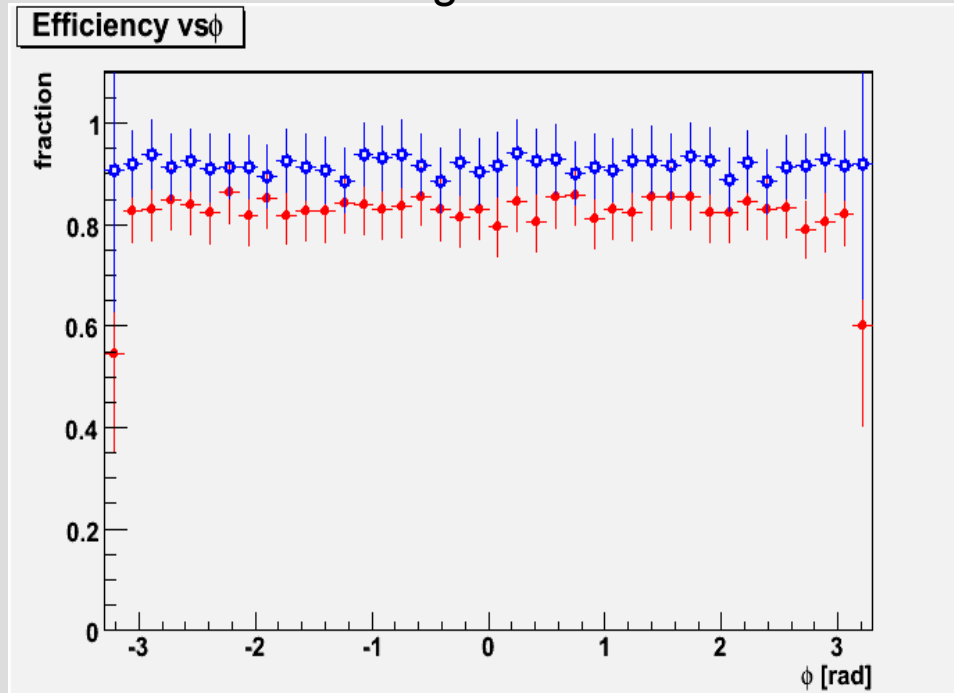
Solution: for all these sectors, add $2 * \pi$ to negative-phi training tracks



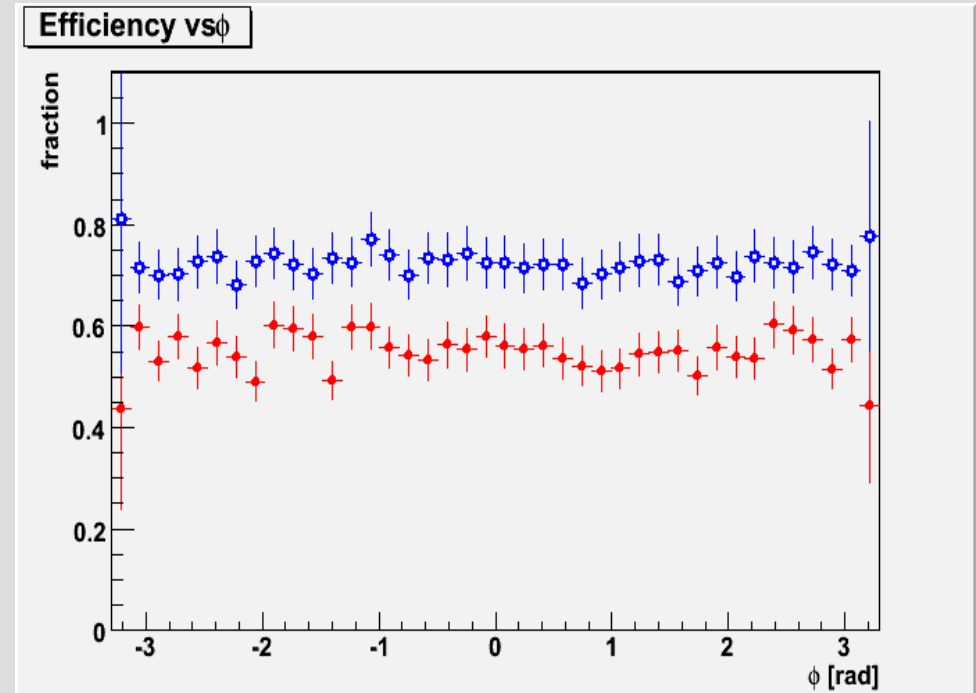
Phi after the fix

FROM LAST WEEK:

Training data



Bsmumu



Got a lot better around $|\text{Pi}|$, but still some drop in efficiency. Need to investigate!
I see no reason why efficiency should be any lower around $|\text{Pi}|$