SPD Physics & MC meeting
1 June 2022

Status of track reconstruction for SPD experiment

V. Andreev (LPI, Moscow)

Introduction

- Track reconstruction is usually divided into separate sub-tasks:

  - track finding
  - track fitting (in general on the base of Kalman filter method)

- Track finding (or pattern recognition):

  - division set of measurements in a tracking detectors into subsets
  - each subset contains measurements believed to originate from the same particle

- Track fitting:

  - starts with the measurements inside one subset as provided by the track finder

Track reconstruction (general scheme)

1. event generation (Pythia8 or another particle generators)

2. simulation => produce sim-hits in vertex and straw-tracker detectors using SPDroot

3. produce "digi" - hits => apply smearing for vertex and straw hits

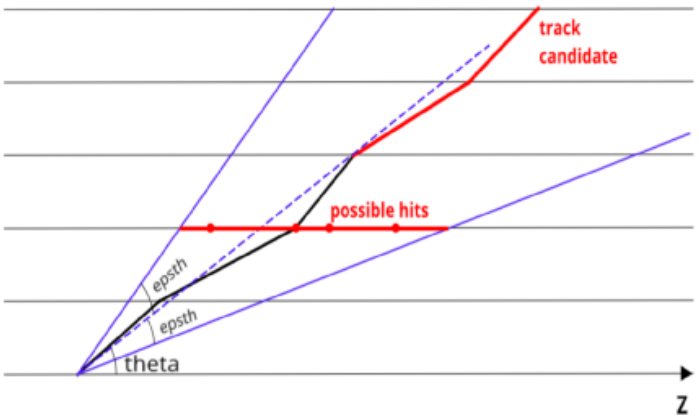
4. pattern recognition => find track candidates with the set of vertex and straw hits:

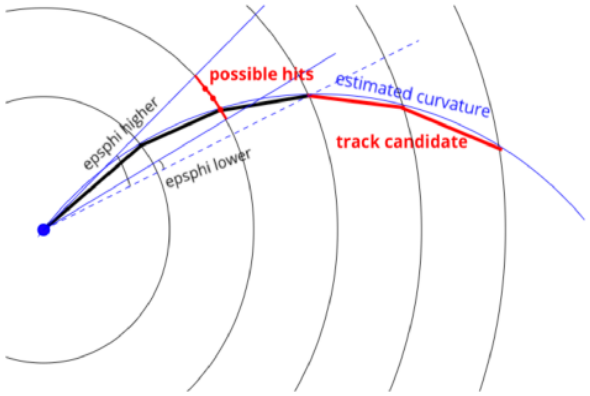
   - construct track seeds using hits in vertex detector;
   - add straw-hits from tracker detector to track candidate;
   - track candidate will contain vertex and straw tracker hits;
   - finally need to apply some cleaning procedure (remove duplicates).

5. do Kalman fit of remaining track candidates

6. write reconstructed track with the set of parameters to output root-file

Find track seeds in vertex detector





1. start from producing of 2-points seeds using 2-hits combination in the different vertex layers

2. then produce 3, 4 and 5-points seeds from 2-points seeds adding new hit from another layer with taking into account some $\Delta\theta$ and $\Delta\varphi$ – conditions:

3. after merging and cleaning procedures are applied

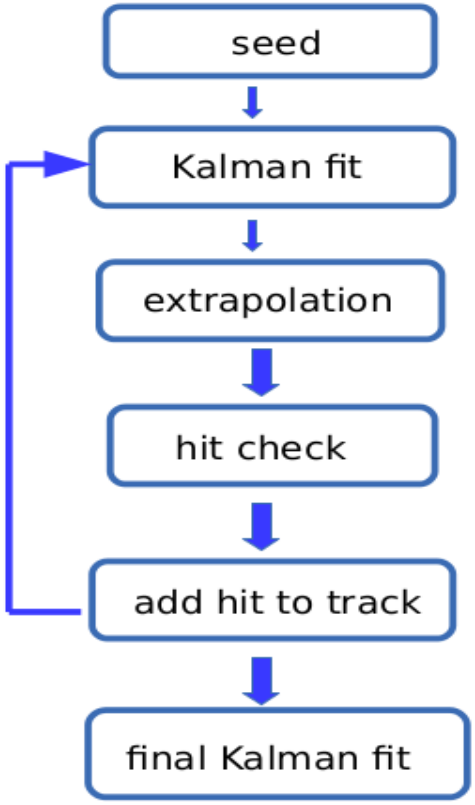
4. do estimation of primary vertex z-position

5. seed finding procedure also provides the next track seed parameters:

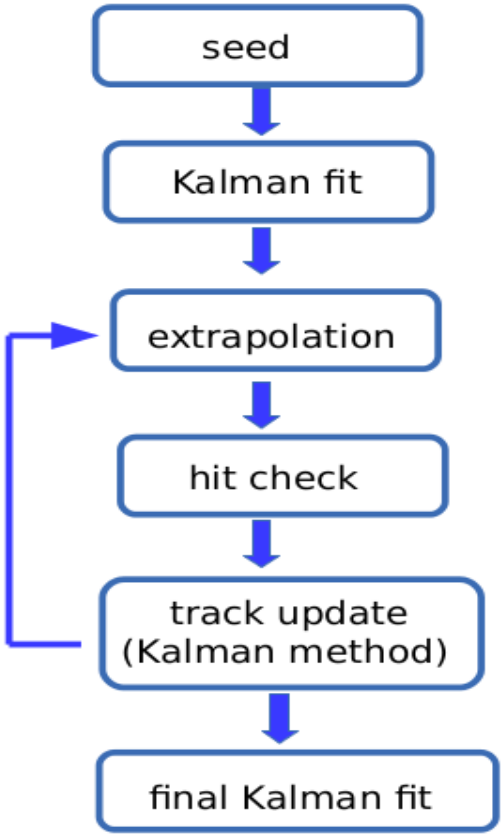
    a) charge;
    b) curvature or radius;
    c) theta;
    d) phi (phi in 1-st seed point);

Scheme for adding straw hits

Full fit



Fast fit



Three different procedures for adding straw hits to track candidate are implemented:

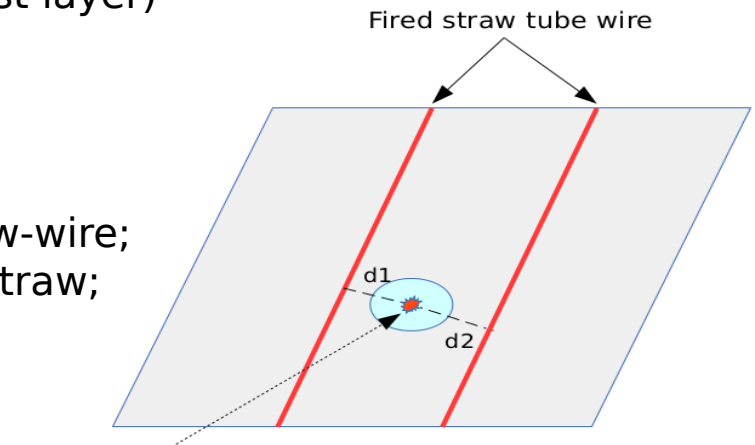
a) so called “full” fit – left scheme;
b) so called “fast” fit – right scheme.
c) Kalman tree method (uses also “fast” fit option)

Procedure of adding straw hits

1. extrapolate track to some straw plane (starting from 1-st layer)

2. check the distance of track hit position on straw plane (red point on picture) to fired straw-wires:

   a) if distance is greater than ~1.0 cm => skip this straw-wire;
   b) if distance is less than 0.5 cm => accept hit of this straw;
   c) if the distance is between 0.5 cm and 1.0 cm => check this straw-wire with taking into account error (blue area) of track extrapolation.



Hit position on straw plane of extrapolated track with error band;
d1, d2 – distance from hit position to fired wires

3. if 2 fired straw-wires (red lines on picture) on plane are compatible with the track candidate, the next reconstruction options could be used:

   a) use both hits from these fired straws for updating track parameters;
   b) if the option Kalman tree is used => then for the second hit the new track candidate is created, all previous hits are copied to the new track candidate and then extrapolation and fitting procedures are applied for all new candidates;
   c) check chi2 increment with new straw hits at the track updating procedure.

4. finally, “big” number of track candidates are produced and each track candidate contains vertex and straw hits

5. remove duplicates and do Kalman fit of remaining track candidates

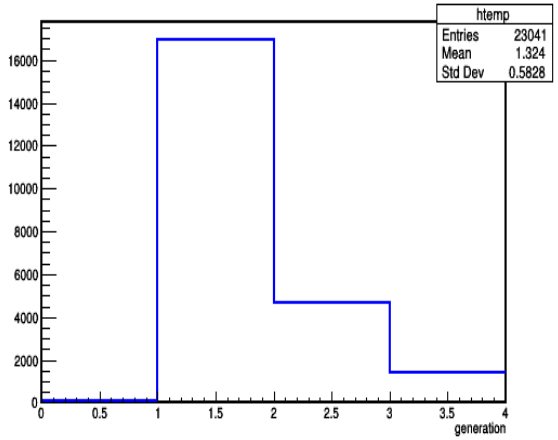
6. write reconstructed track parameters in SpdTrackRC with the same set of track parameters as for the usual SpdTrackMC

Performance of track reconstruction

1. Minimum Bias events are simulated with Pythia8 at √s = 27 GeV (MAPS and DSSD options) at z = 0.0 of primary vertex position

2. “ideal” track reconstruction – Kalman track fit uses exact MC information about hits which are belonged to the track and particle code

3. “ideal” track reconstruction can be applied to the different particle generation level:
   a) 1-st generation - only primary particles from interaction vertex are considered;
   b) 2-nd generation – primary and secondary interactions and particles are considered;
   c) 3-d generation - primary, secondary and further interactions and particles are considered.



Number of reconstructed track vs generation

4. time for reconstruction of Minimum Bias events (2.8 GHz notebook):
   a) “ideal” (known MC hits) => ~6.0 sec/event (3 generation, Its hits >=0, Its+Ts hits >= 3)
   b) “fast” fit => ~7.0 sec/event (Its hits>=2, Its+Ts hits > 3)
   c) “full” fit => ~26.0 sec/event
   d) Kalman tree + “fast” fit => ~33.0 sec/event
   e) reconstruction time is strongly depended on applied conditions (number Its hits, generation level, minimum Pt and momentum of reconstructed track)

Performance of track reconstruction (2)

5. procedure for estimation of track reconstruction efficiency:
   a) consider “ideal” reconstructed track;
   b) try to find among reconstructed tracks such track which can be considered as identical to “ideal” reconstructed track => that means this track has the some number common of vertex hits (>=2) and also has > 40 % same “straw” hits.

6. procedure for estimation of fake reconstruction rate:
   a) consider reconstructed track;
   b) try to find among “ideal” reconstructed tracks such track which can be considered as identical to reconstructed track => that means this track has the some number of common vertex hits (>= 2) and also has > 40% same “straw” hits;
   c) if such “ideal” reconstructed track is not found => this reconstructed track is considered as fake track.

7. next conditions are used for checking of the track reconstruction program:
   a) minimum Pt > 25 MeV/c and minimum P > 75 MeV/c;
   b) “ideal” reconstruction => 3 generation, Its hit >= 0, Its+Ts hits >= 3;
   c) “real” reconstruction => Its hits >=2, Its+Ts hits >= 3;

Need specially to point out that described track reconstruction procedure is valid only for finding primary vertex tracks !!

Printout example of reconstructed tracks

-I- <SpdMCEventHelper::PrintTracksRC> "Reco" tracks

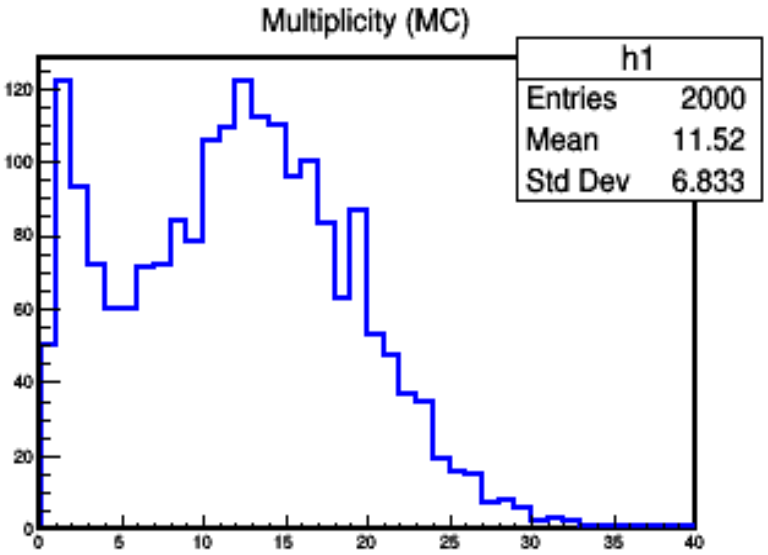
ID	M(ID)	M(PDG)	NAME	V(ID)	FIT	NH	N(Its)	N(TsB)	N(TsEC)	NP
0	0	-211	pi-	0	yes	49	4	45	0	49
1	1	211	pi+	0	yes	40	4	36	0	40
2	2	211	pi+	0	yes	51	4	0	47	51
3	3	-211	pi-	0	yes	5	3	0	2	5
4	4	211	pi+	0	yes	50	3	0	47	50
5	5	211	pi+	0	yes	28	2	0	26	28
						223	20	81	122	223

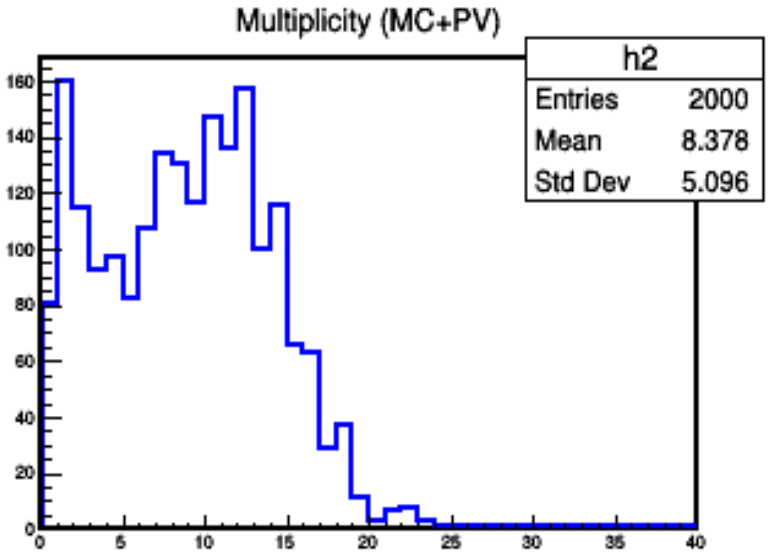
-I- <SpdMCEventHelper::PrintTracks> "Ideal" reco tracks

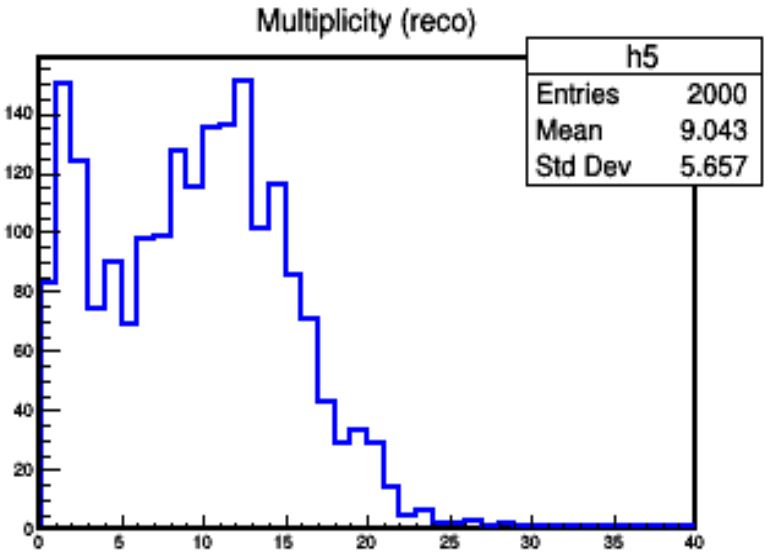
ID	M(ID)	M(PDG)	NAME	V(ID)	FIT	NH	N(Its)	N(TsB)	N(TsEC)	NP
0	17	211	pi+	0	yes	51	4	0	47	51
1	24	211	pi+	0	yes	52	3	0	49	52
2	25	-321	K-	0	yes	3	3	0	0	3
3	28	211	pi+	0	yes	28	2	0	26	28
4	29	211	pi+	0	yes	42	4	38	0	42
5	30	-211	pi-	0	yes	52	4	48	0	52
6	55	13	mu-	7	yes	51	0	51	0	51
7	63	211	pi+	10	yes	51	3	0	48	51
8	64	-211	pi-	10	yes	50	3	0	47	50
						380	26	137	217	380

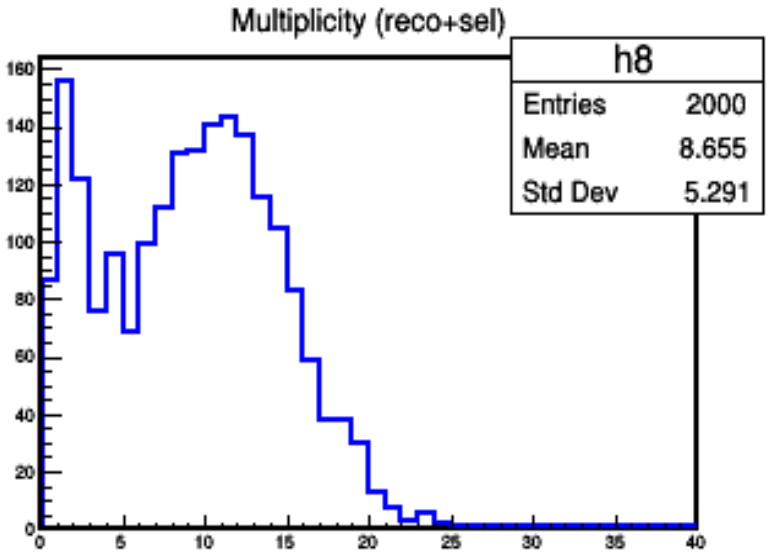
Secondary tracks

Reconstructed track multiplicity (MAPS)

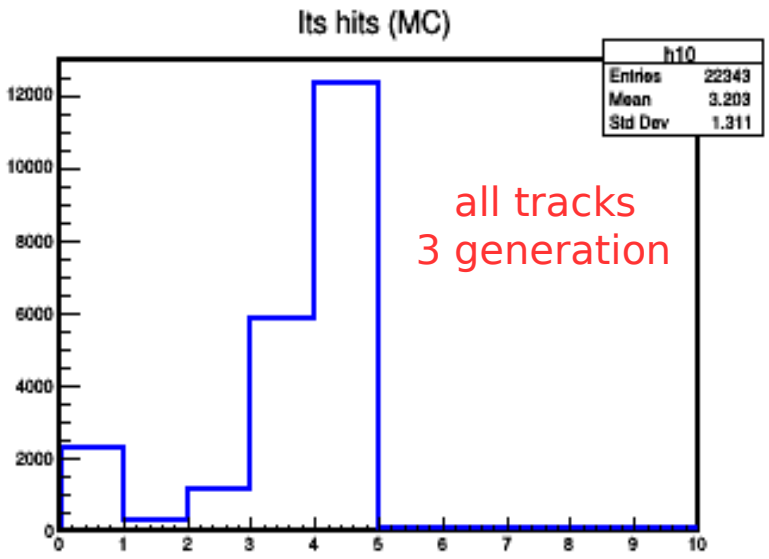


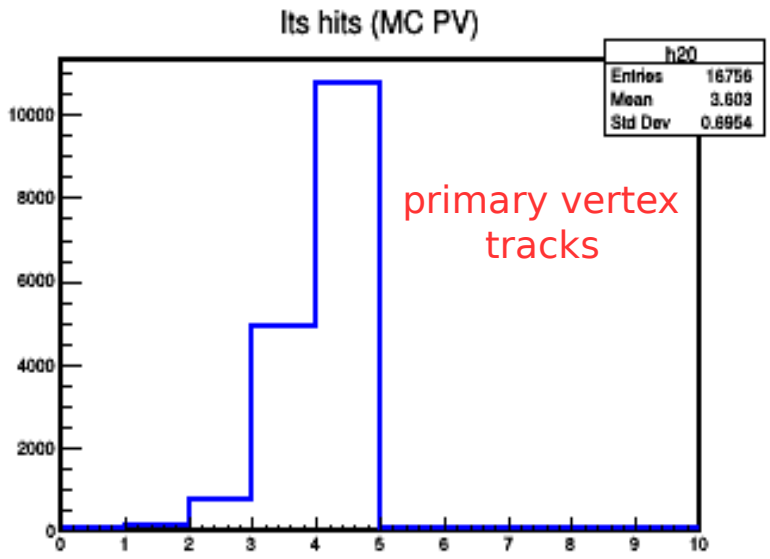


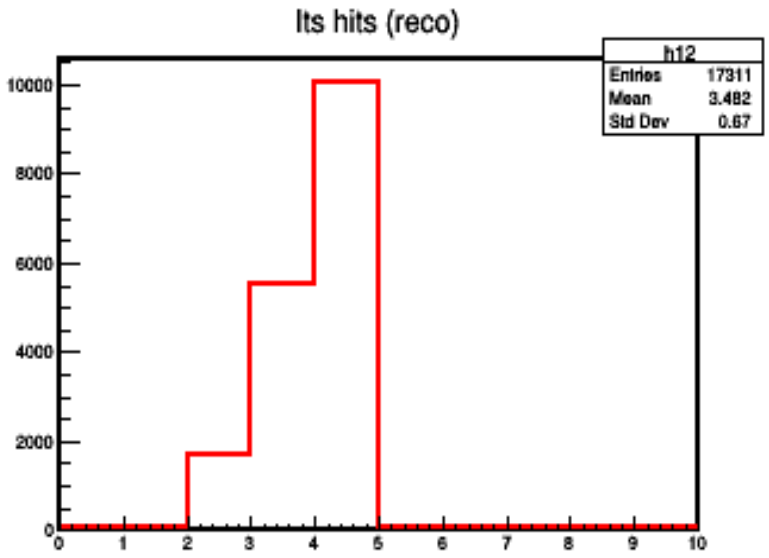


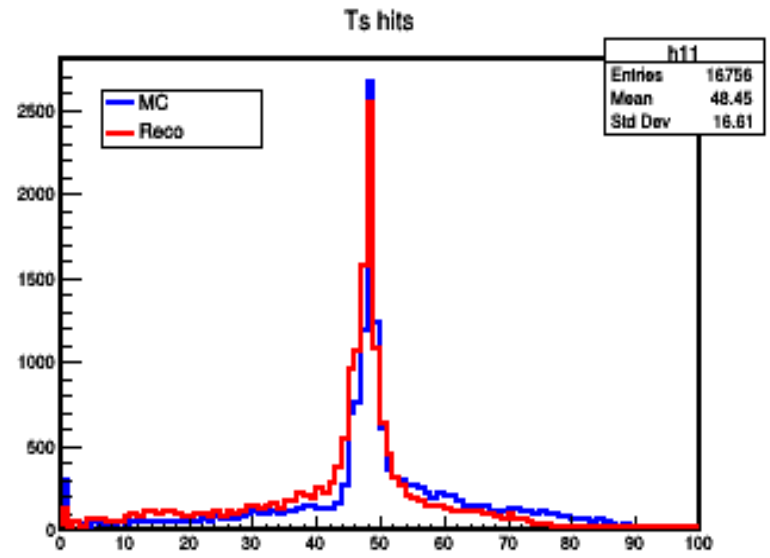


Number of Its and Ts hits in reconstructed tracks (MAPS)

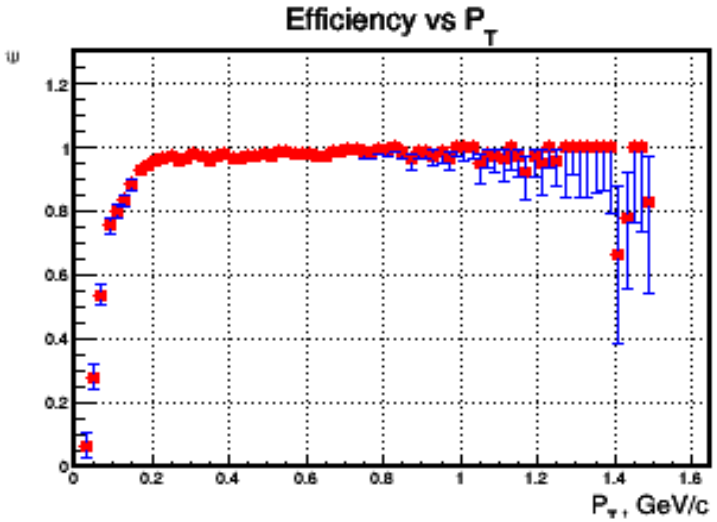


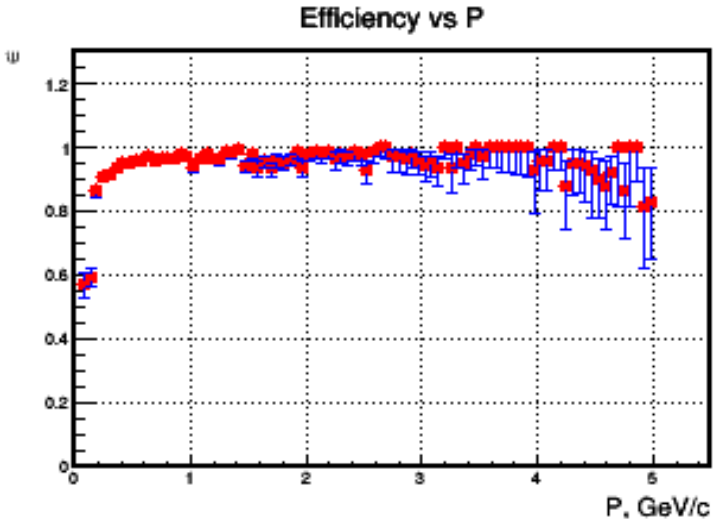


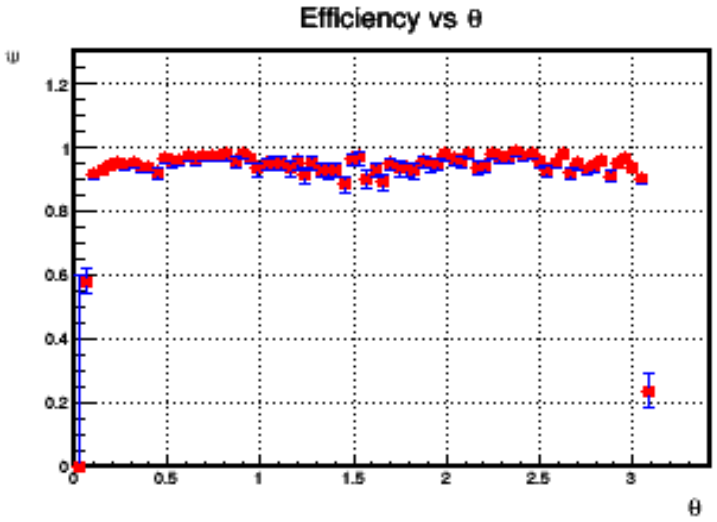


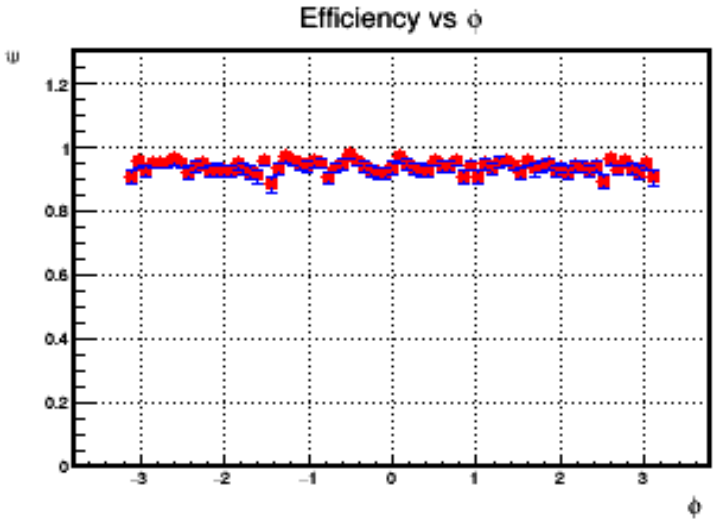


Track reconstruction efficiency (MAPS)





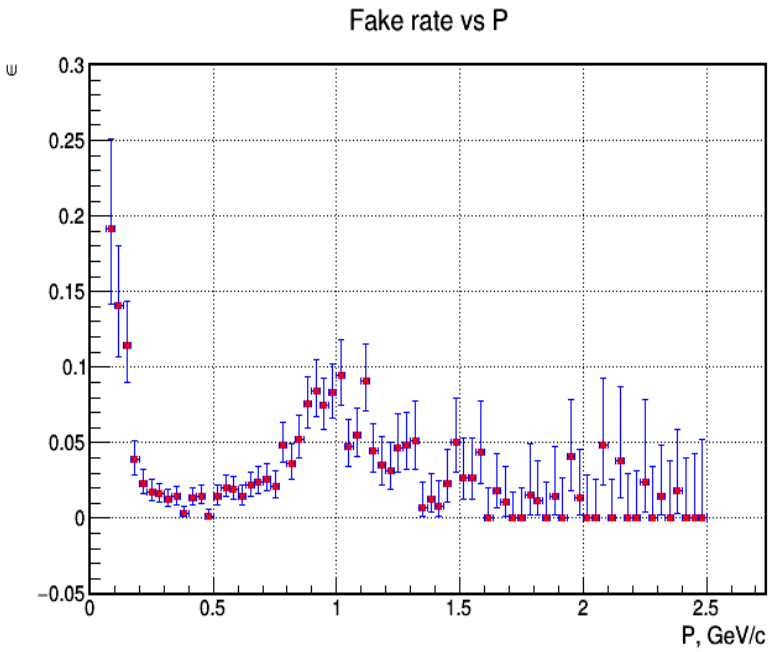


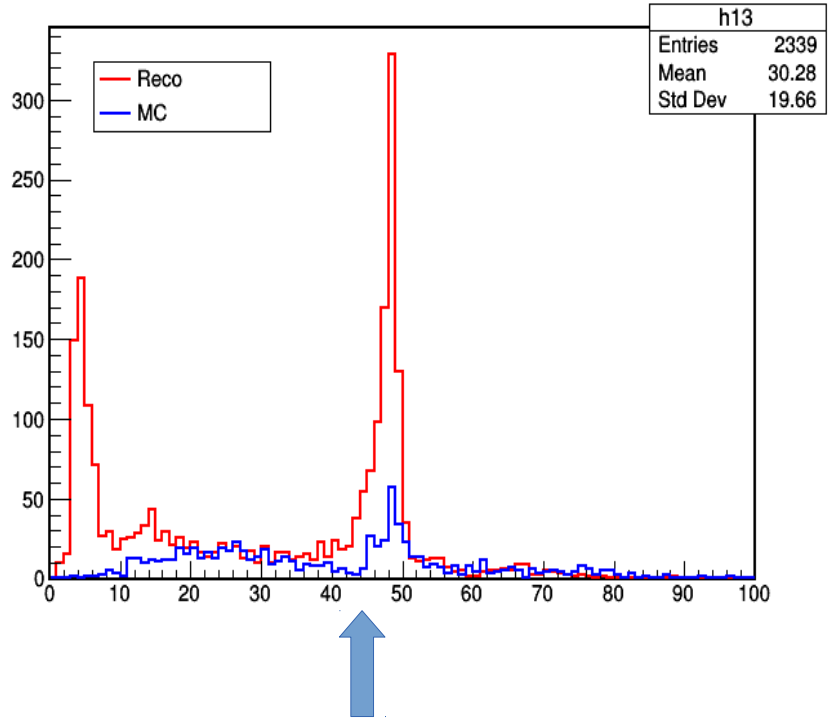


Total track reconstruction efficiency

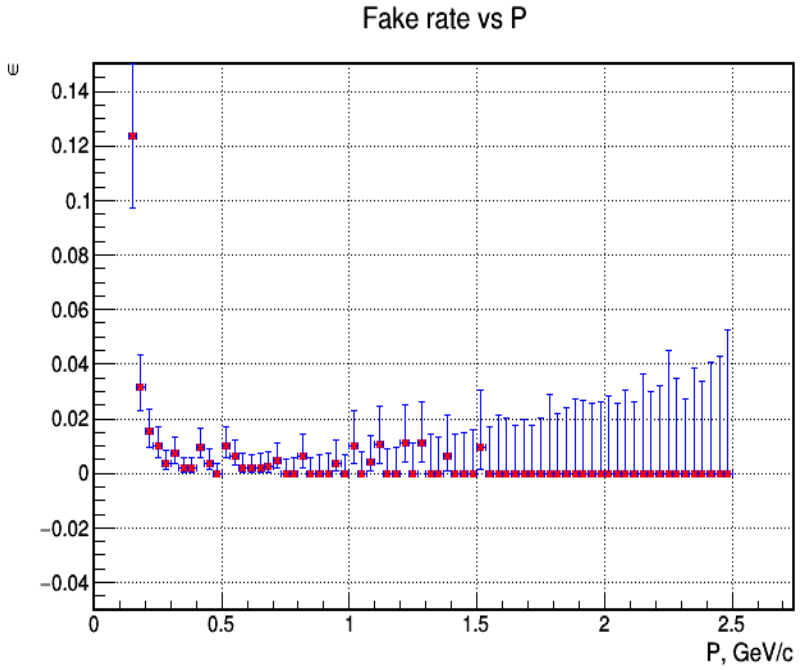
Pt (MeV)	> 25	> 50	> 100	> 150	> 200
Eff (%)	94.1	94.7	96.1	97.0	97.5

Fake rate



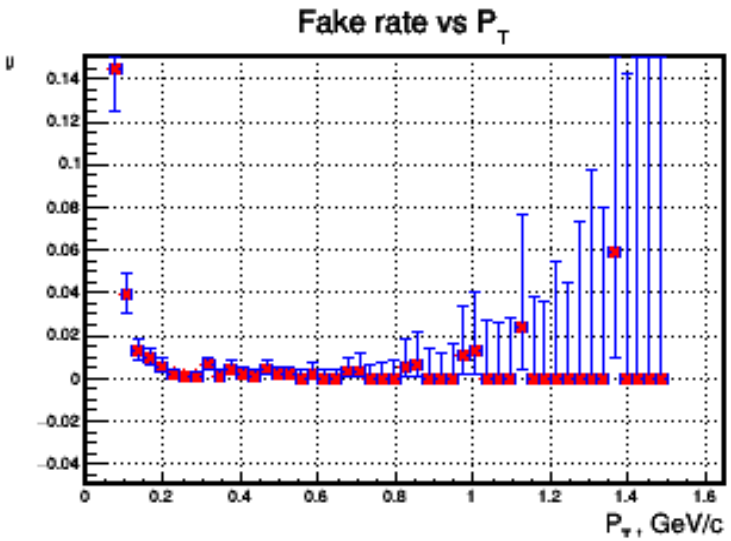


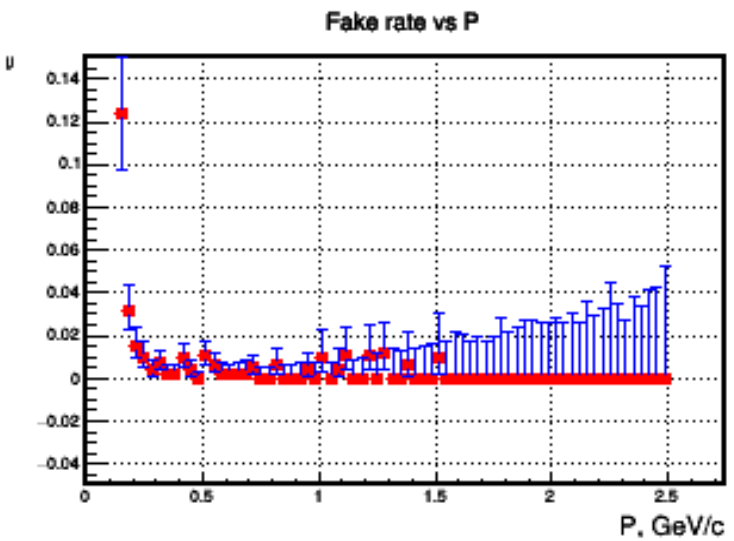
Distribution of Ts hits for reconstructed tracks with Its hits = 2

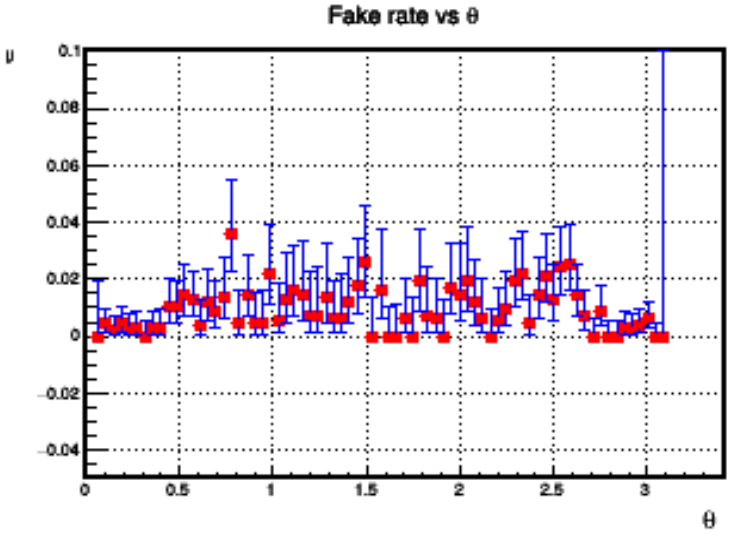


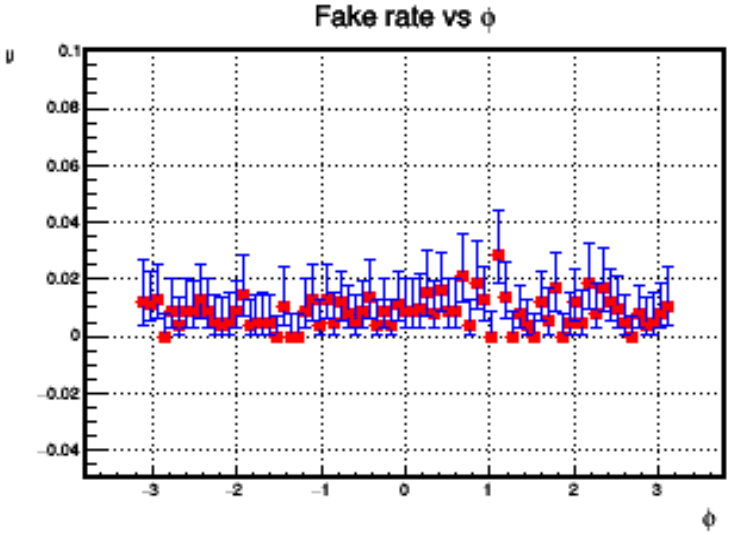
Fake rate after applying additional cuts for track with Its hits = 2 => Ts hits > 8

Fake rate (MAPS)





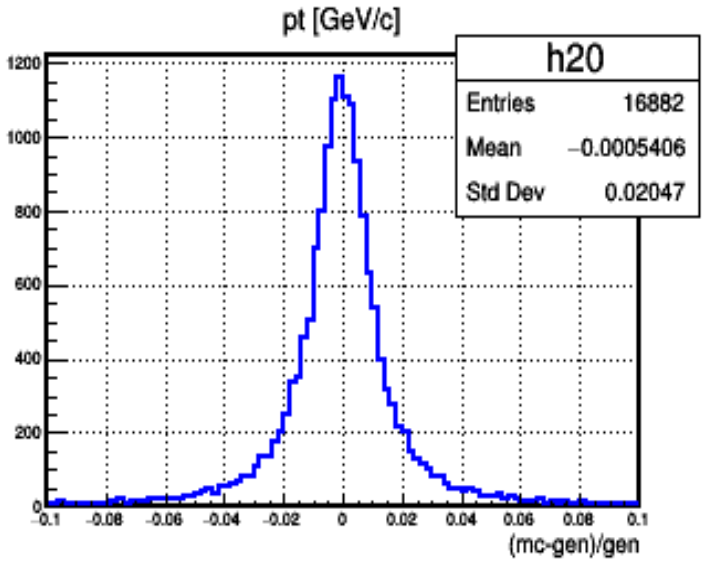


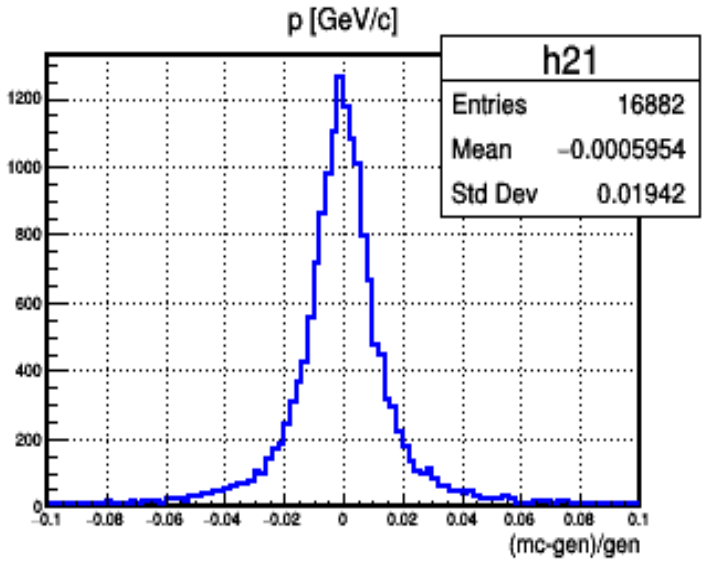


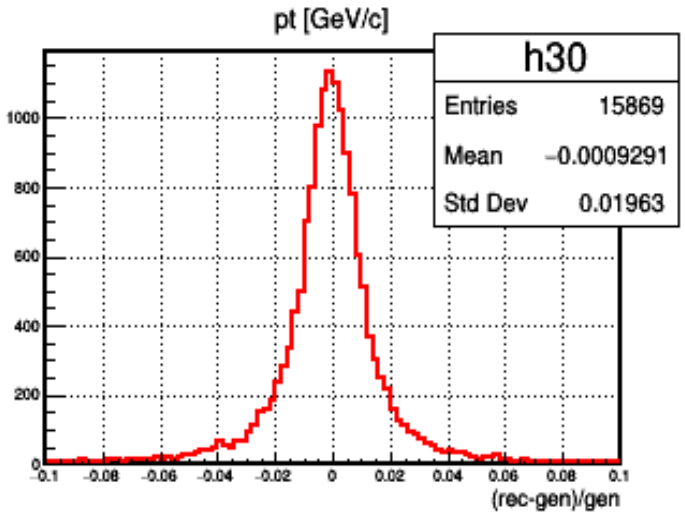
Pt (MeV)	> 25	> 50	> 100	> 150	> 200
Fake1 (%)	3.23	3.2	2.78	2.72	2.67
Fake2 (%)	0.87	0.82	0.46	0.36	0.31

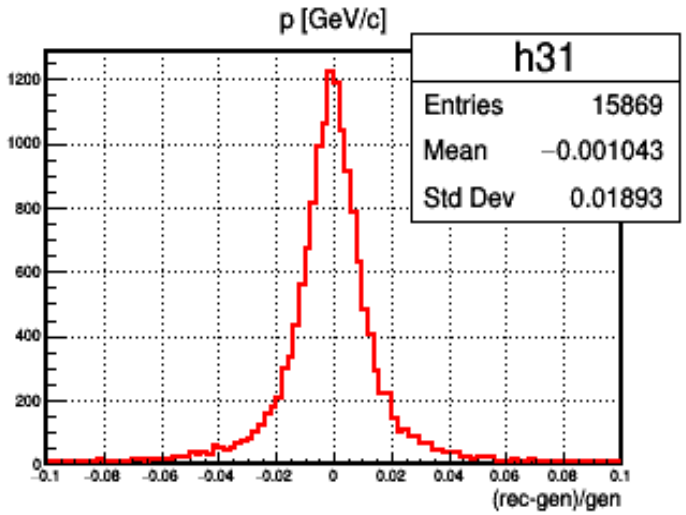
Its=2+Ts > 8

Pt and momentum resolution (MAPS)

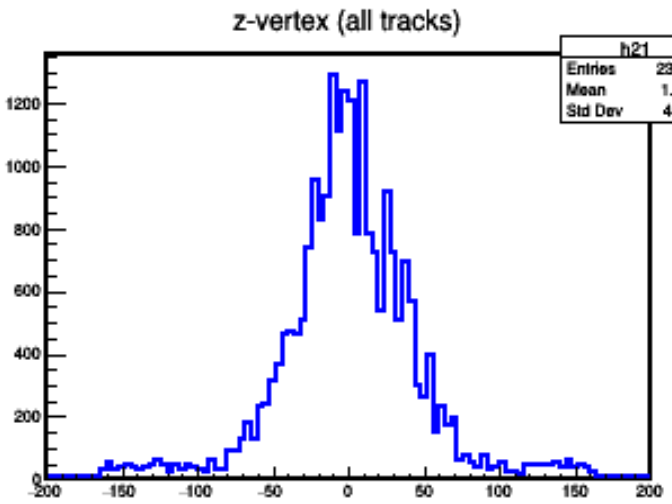


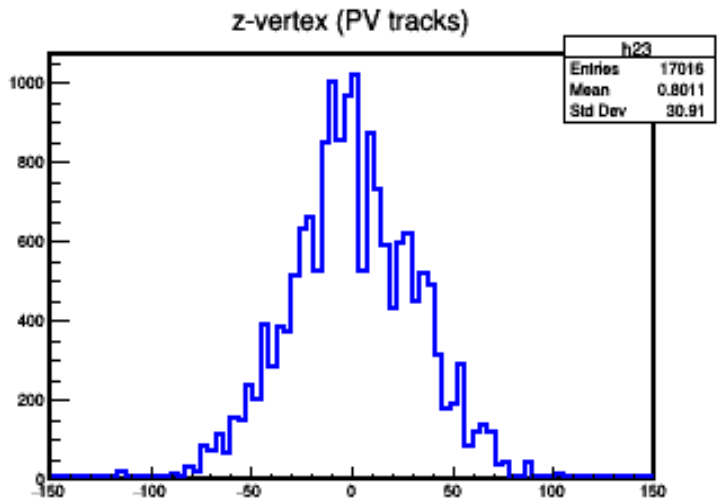


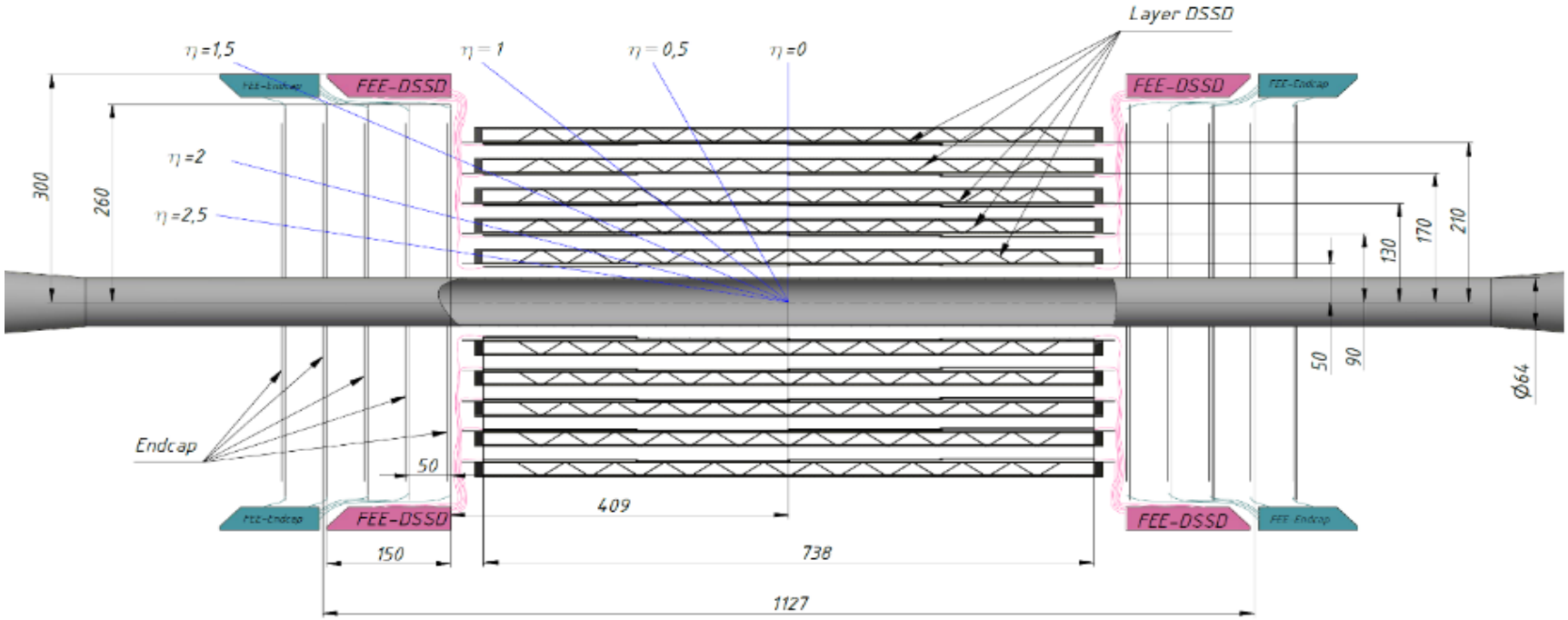




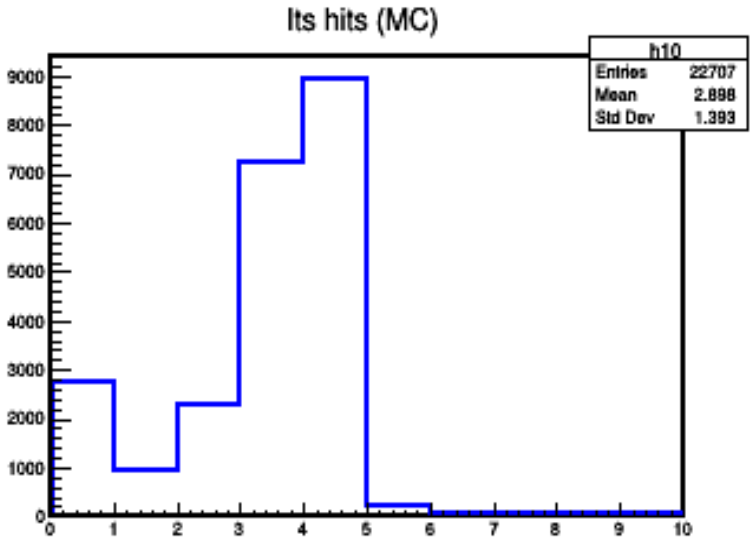
Z of primary vertex distribution (σ= 30.0 cm)

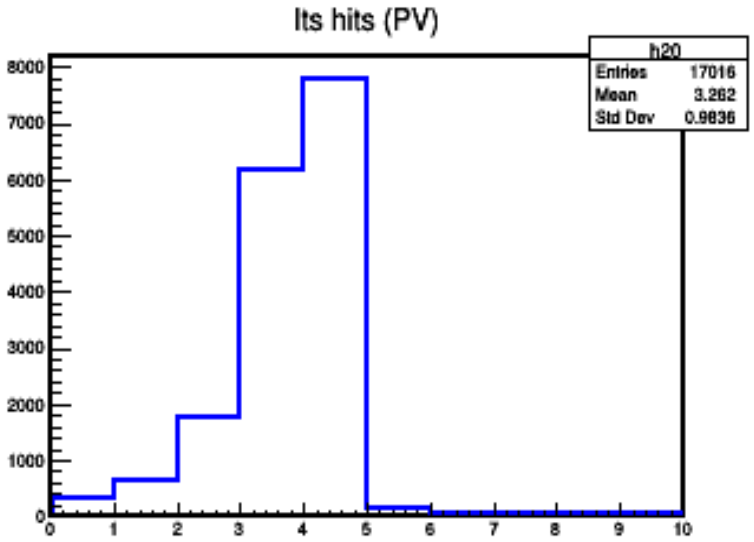


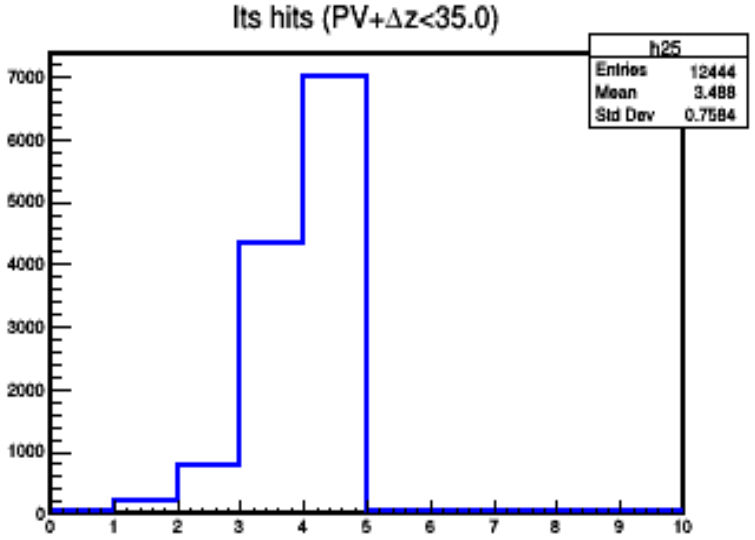


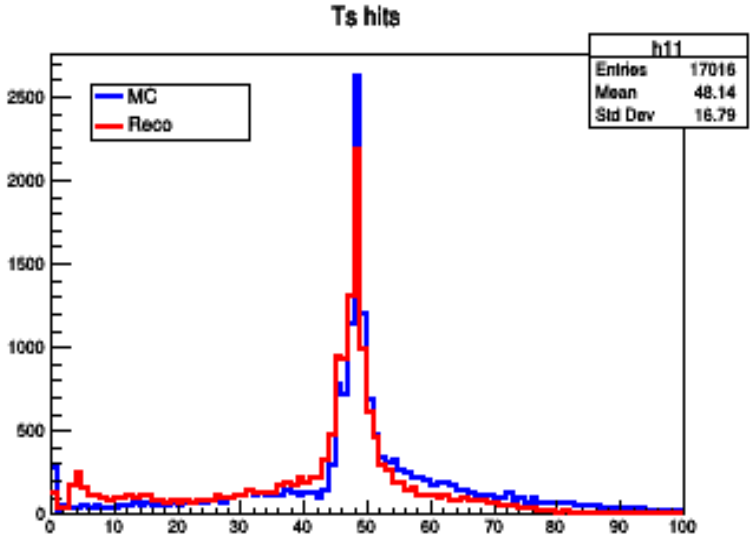


Number of Its and Ts hits in reconstructed tracks (MAPS)

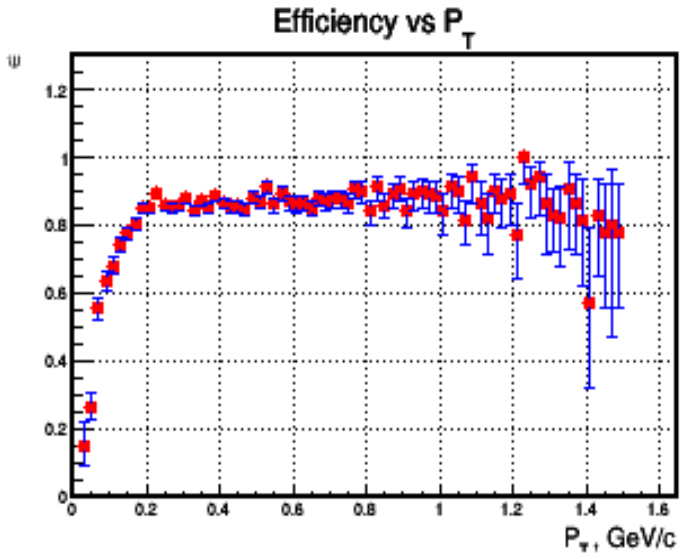


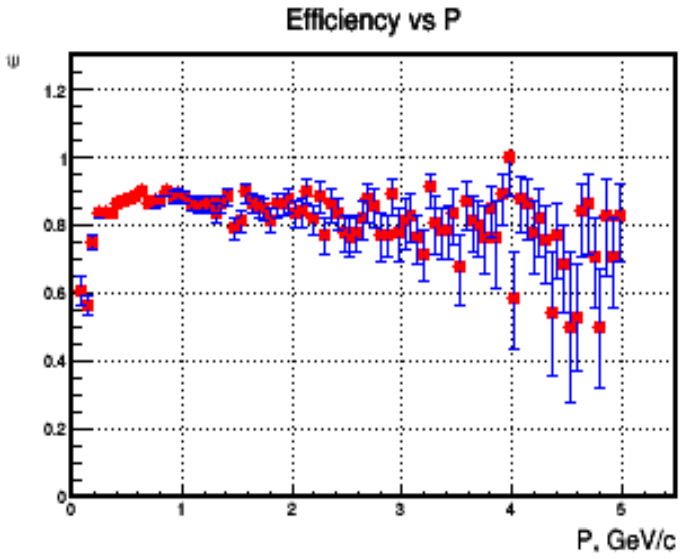


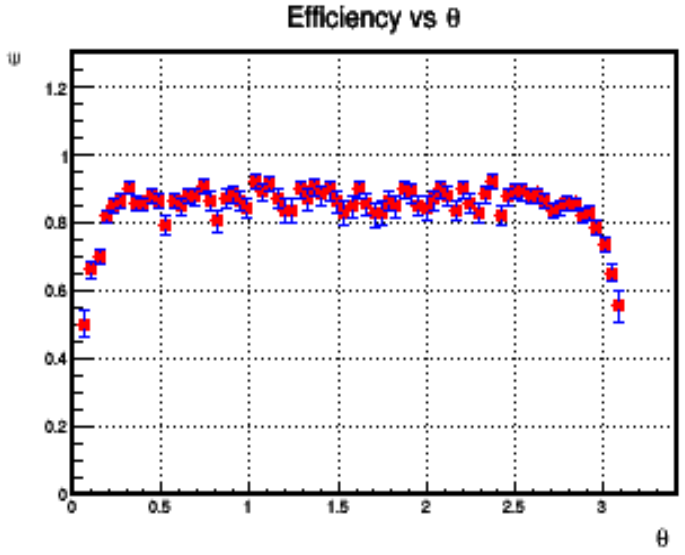


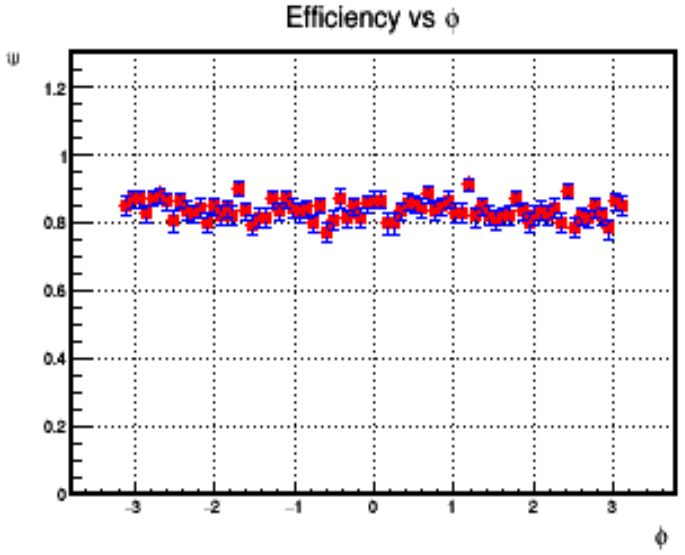


Reconstruction efficiency all z-position of PV (MAPS)



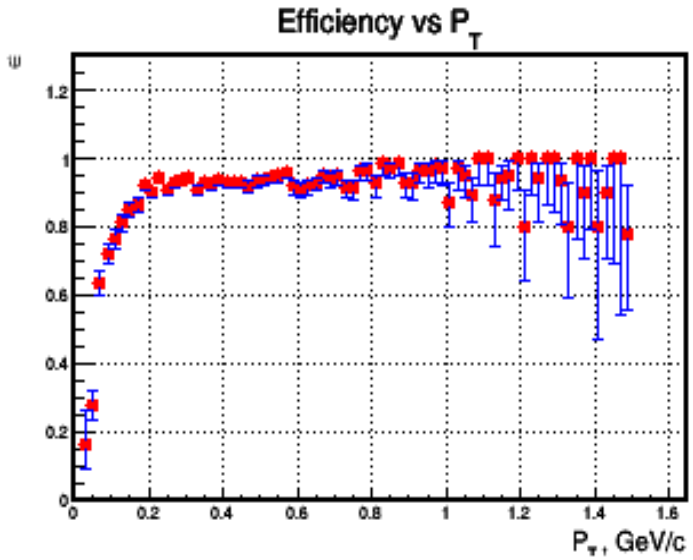


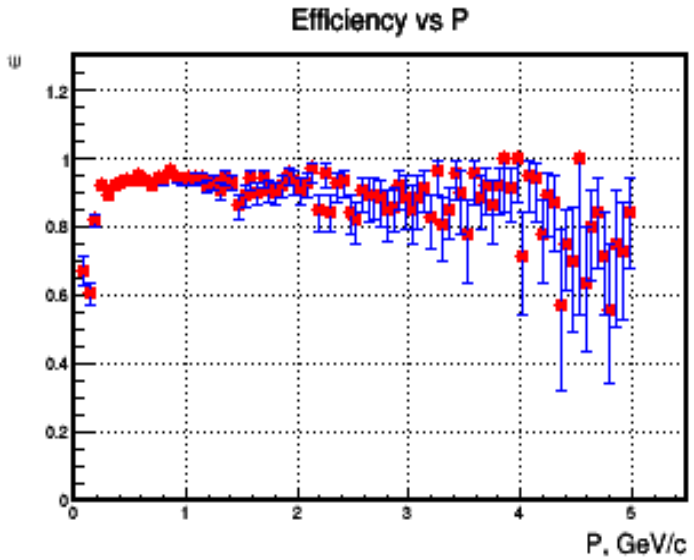


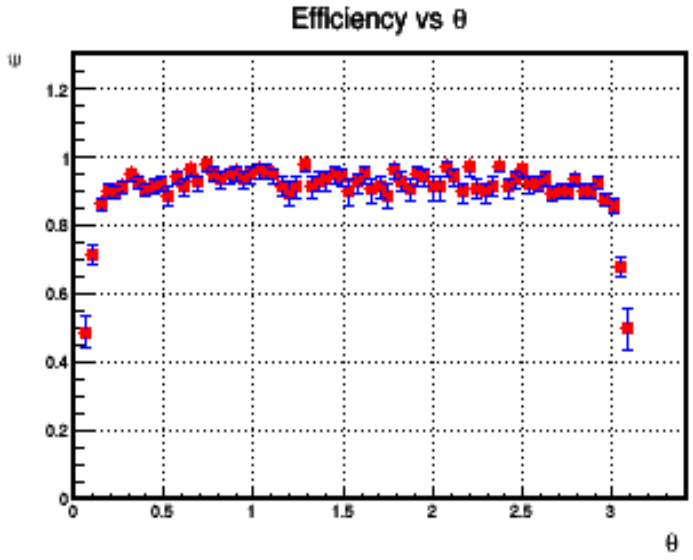


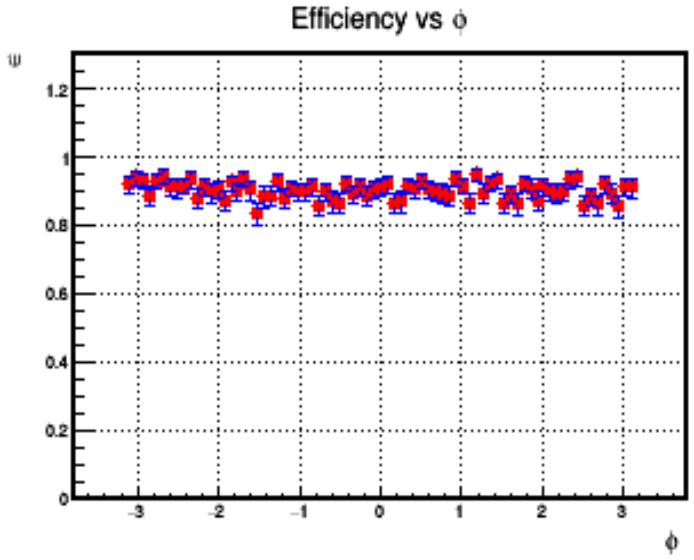
Δz(cm)	all	< 60	< 50	< 40	< 30	< 20	< 10
Eff(%)	83.9	87.3	88.8	89.9	91.0	91.4	91.9

Track reconstruction efficiency ΔZ < 35.0 cm (MAPS)









Total reconstructions efficiency ~90.4 % and ~93.2 % (Pt>200 MeV/c)

Summary and plans

1. 2-nd version of track reconstruction is ready for implementation to SPDroot

2. add new “fast” fit option to reconstruction algorithm

3. some cleaning and simplification of code structure are done

4. reconstruction algorithm shows good track reconstruction efficiency and low fake rate

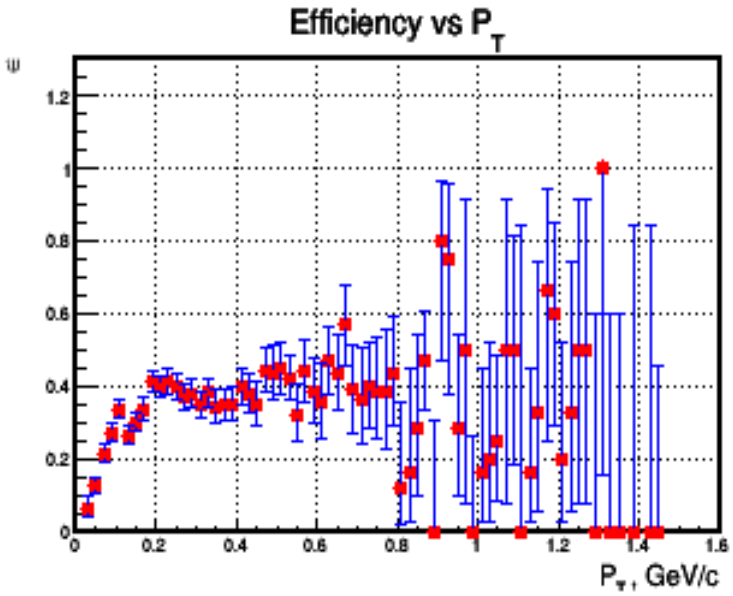
Future plans:

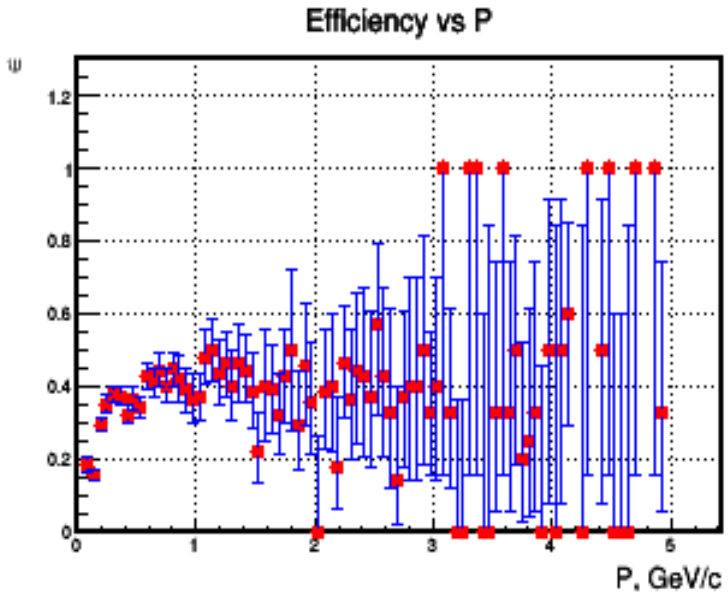
1. develop the finding procedure for secondary tracks:
   a) track finding without vertex detector hits
   b) select primary and secondary interactions

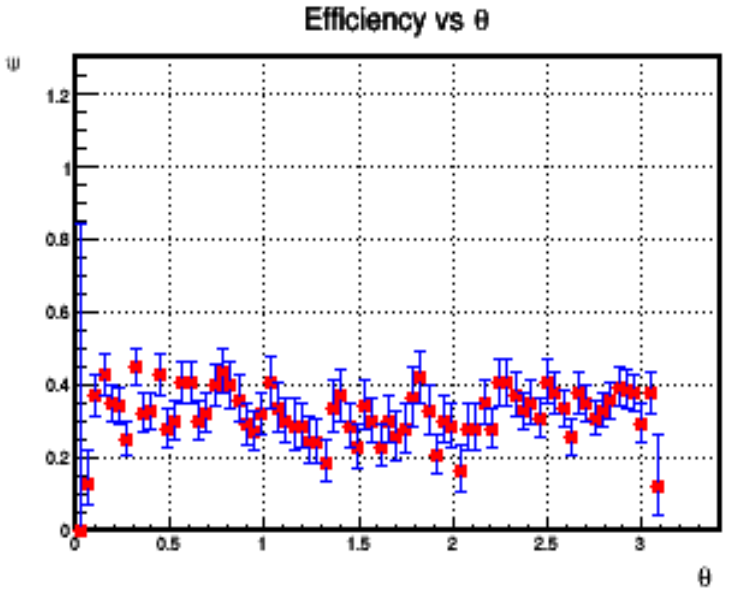
2. implement additional algorithm for reconstruction of primary vertex

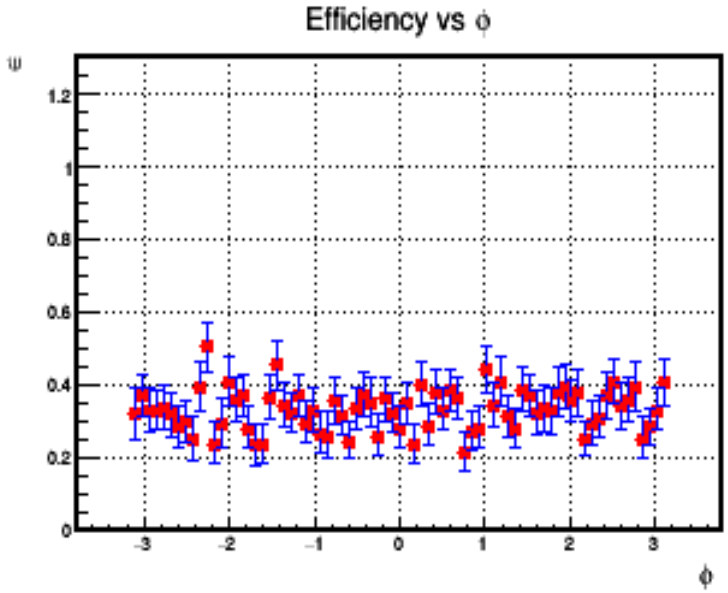
Backup slides

Reconstruction efficiency for secondary tracks (MAPS)

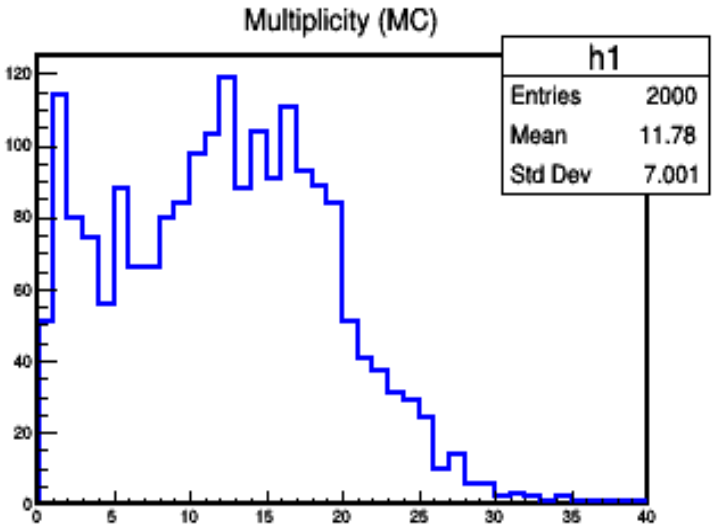


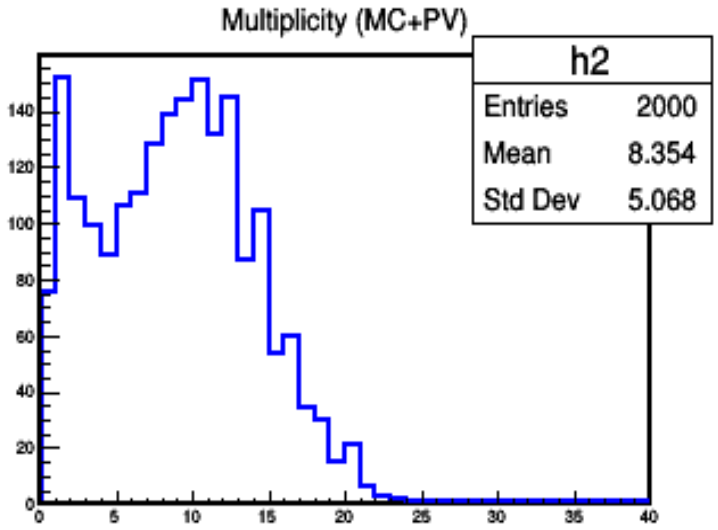


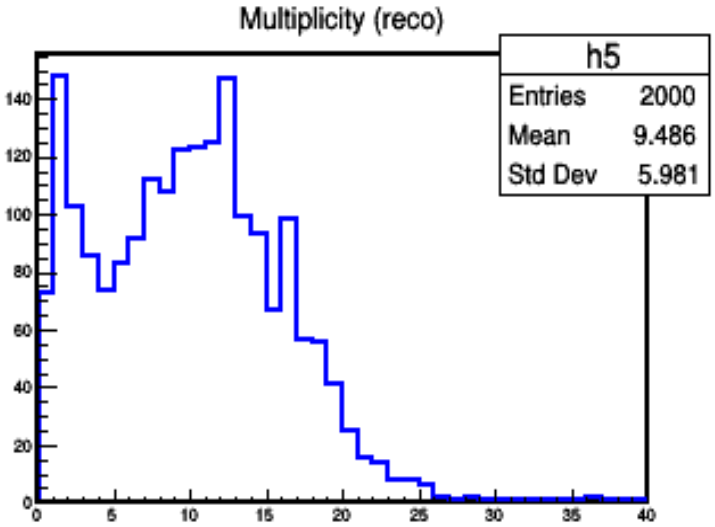


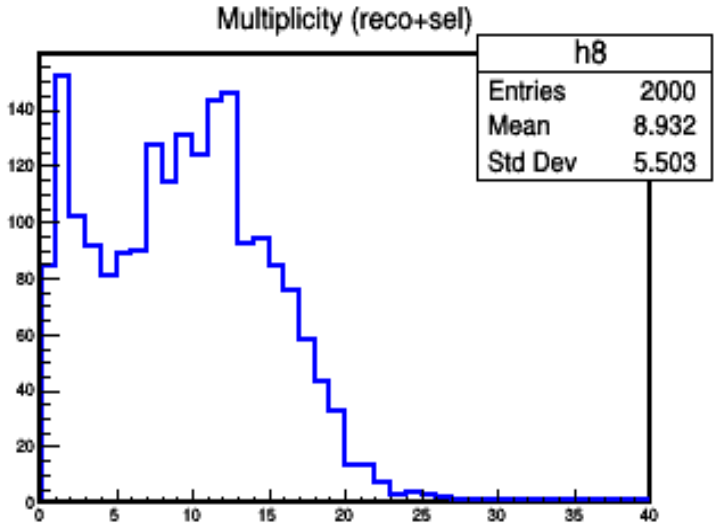


Total multiplicity (DSSD)

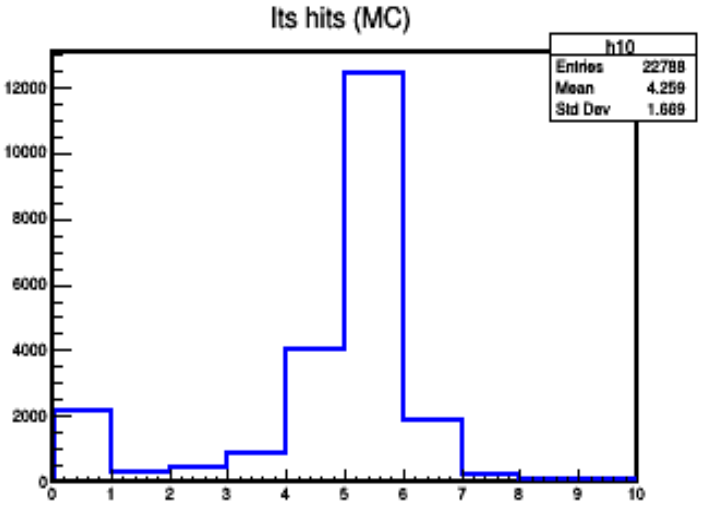


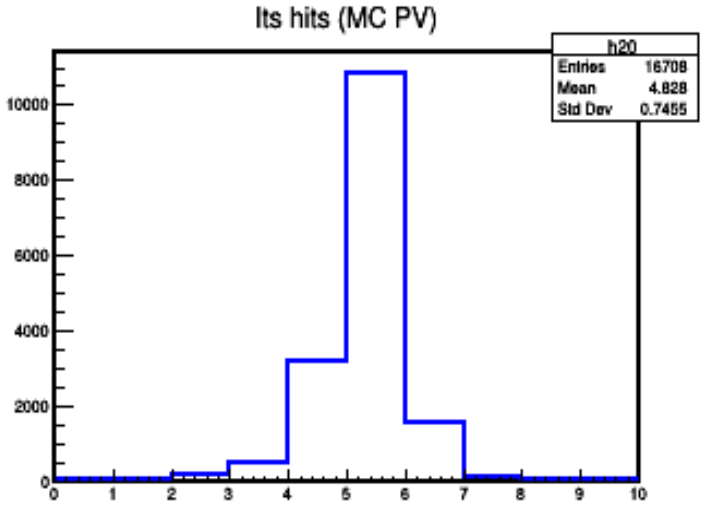


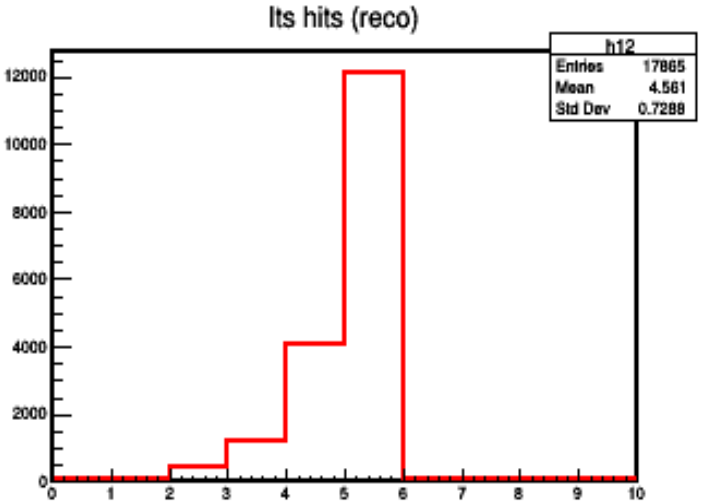


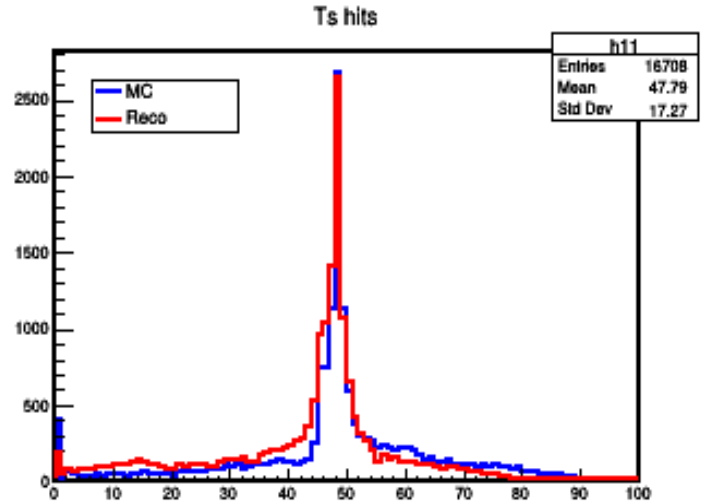


Its and Ts hits (DSSD)

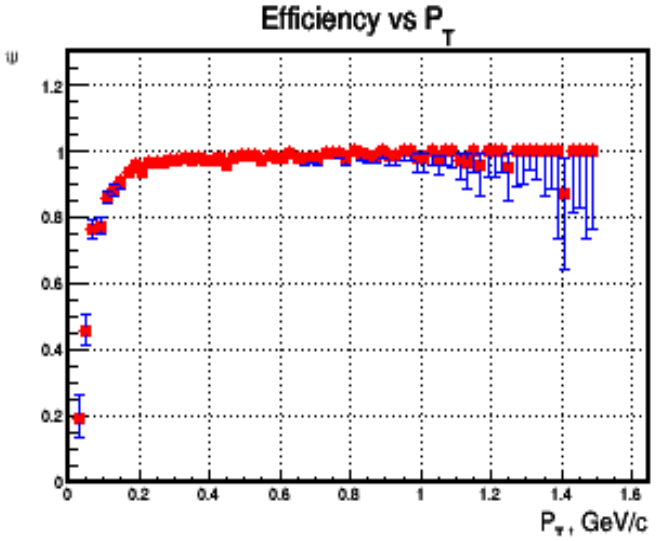


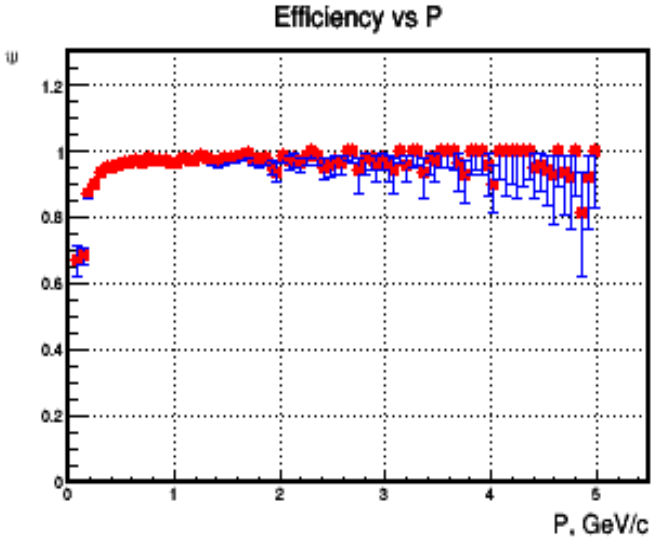


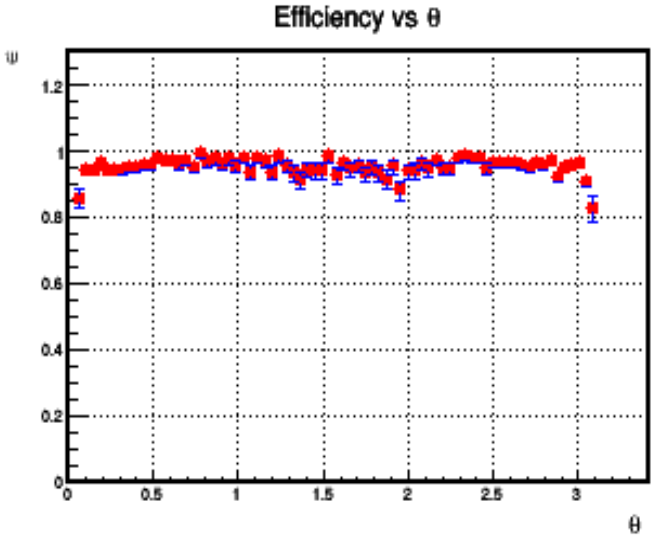


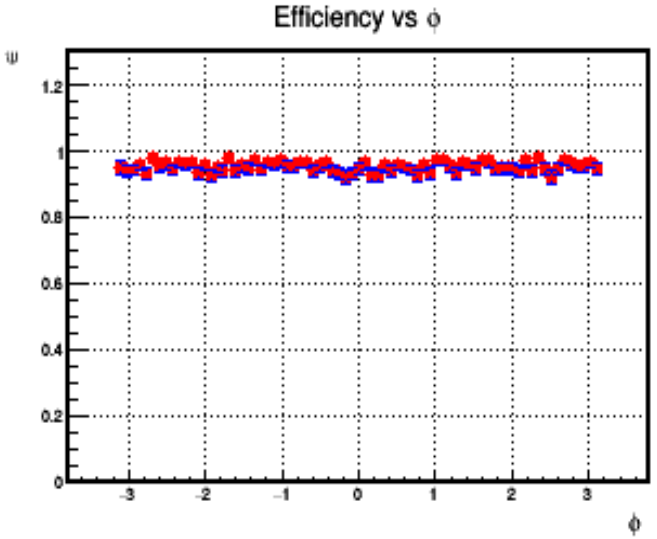


Track reconstruction efficiency DSSD option





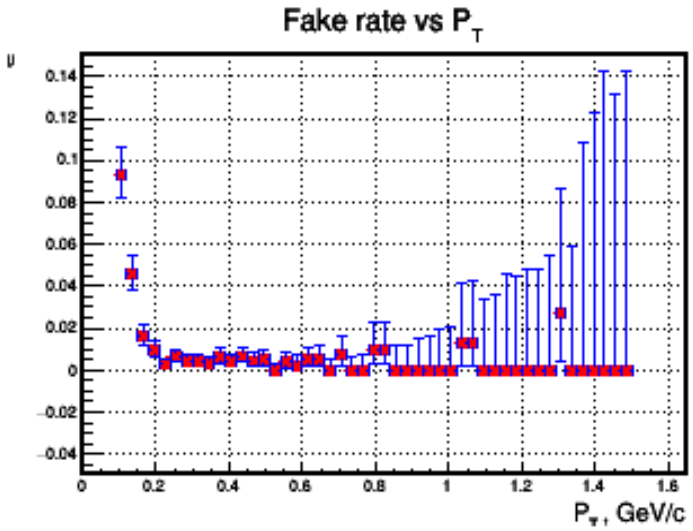


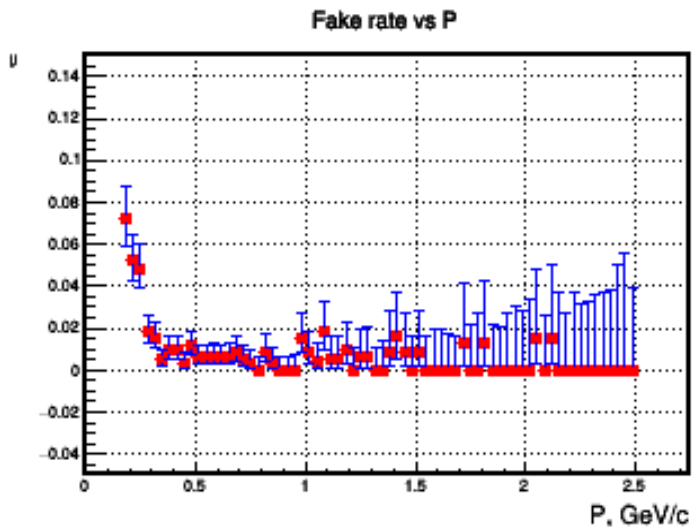


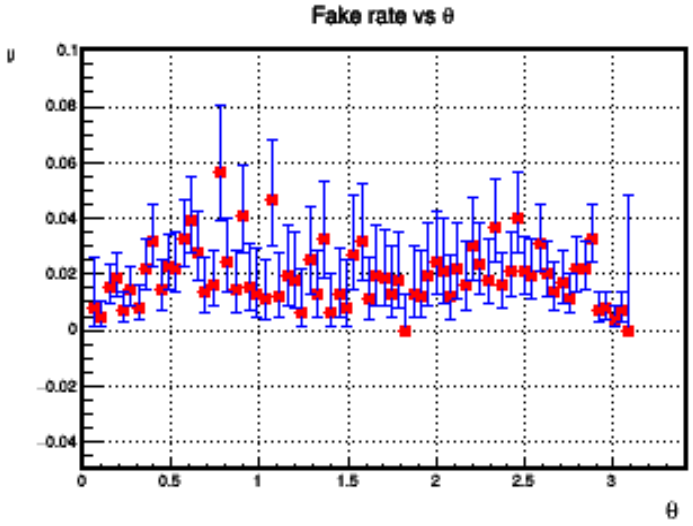
Total track reconstruction efficiency

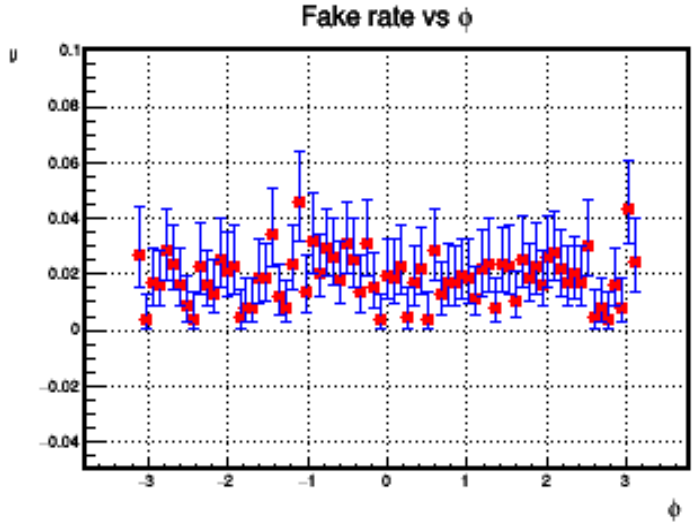
Pt (MeV)	> 25	> 50	> 100	> 150	> 200
Eff (%)	95.5	95.9	96.8	97.4	97.6

Fake rate (DSSD)

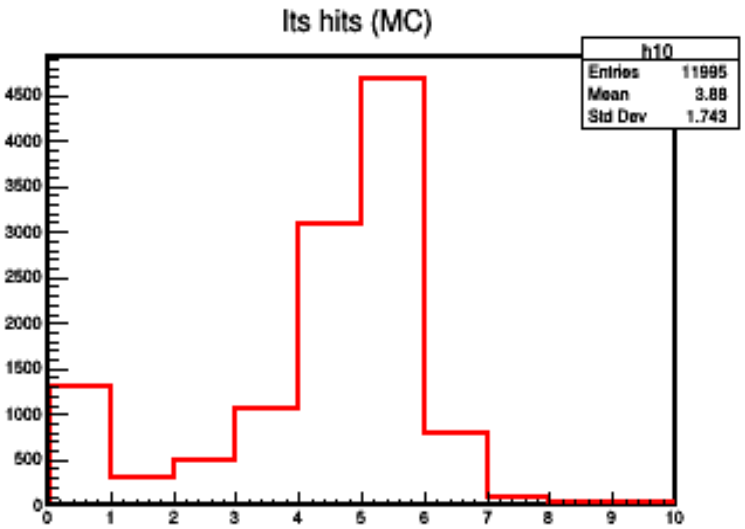


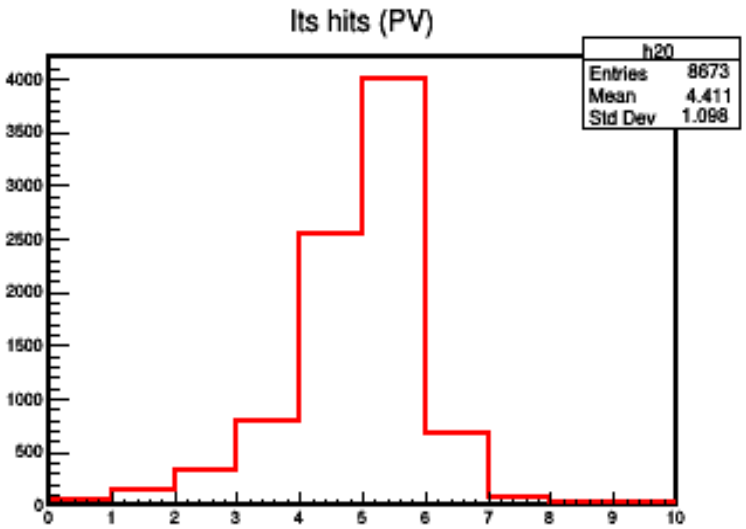


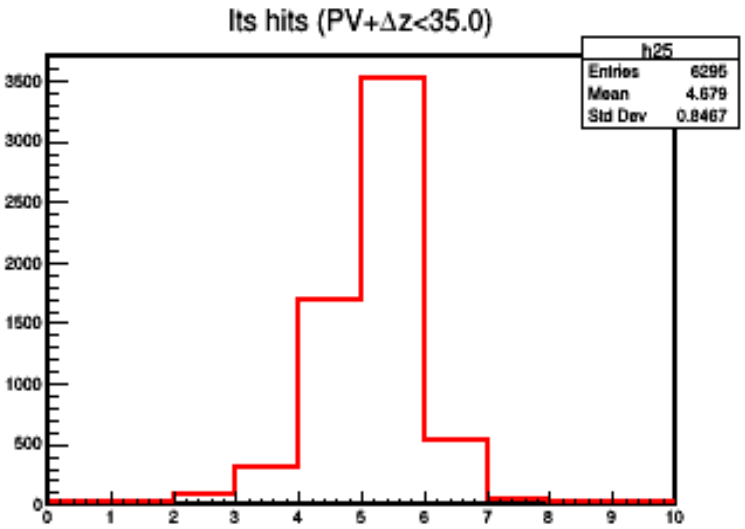


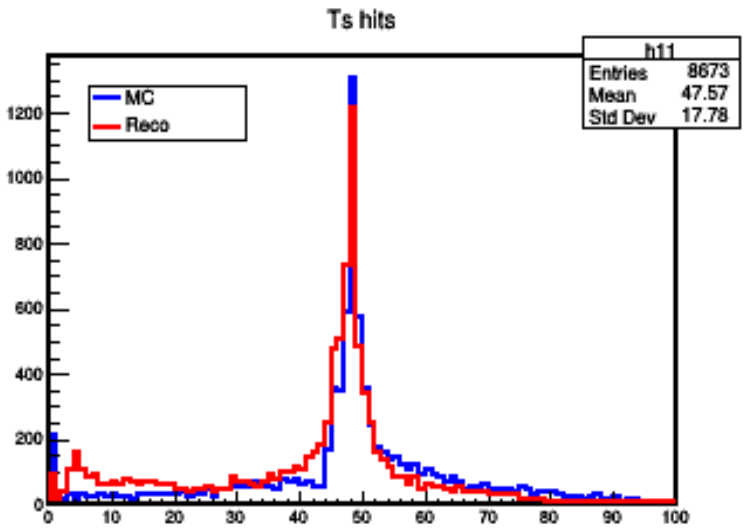


Its and Ts hits (DSSD) (z Gaussian distribution)

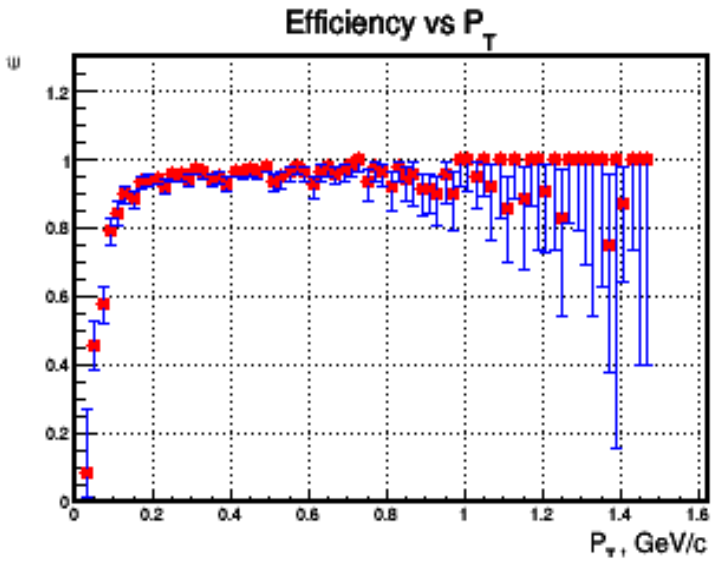


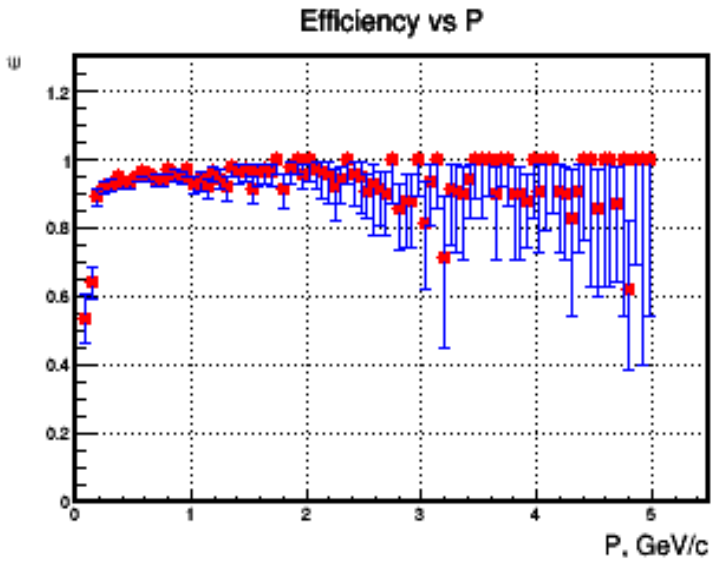


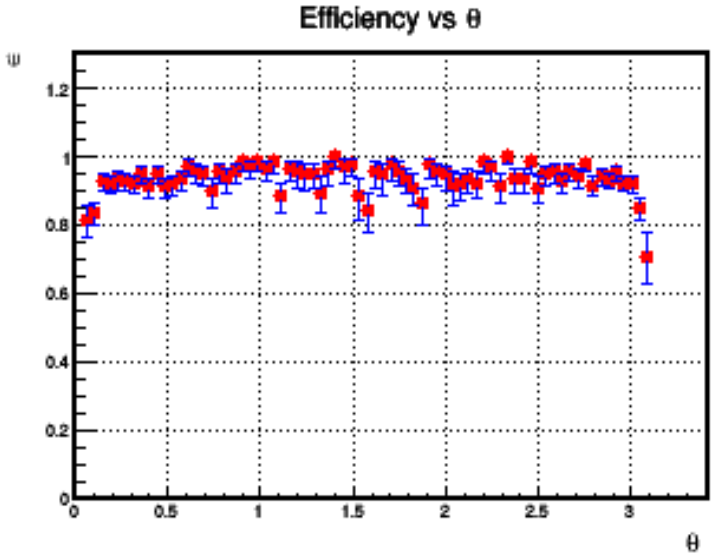


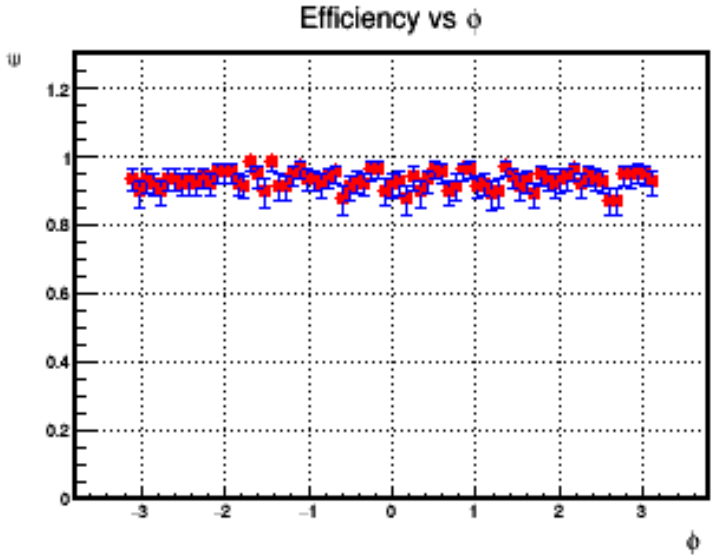


Track reconstruction efficiency (DSSD) (z Gaussian)









Total track reconstruction efficiency = 93.3 %