Beam transport: Checks in PLUTO

J. Biernat, T. Hennino,I. Froehlich, P.Salabura

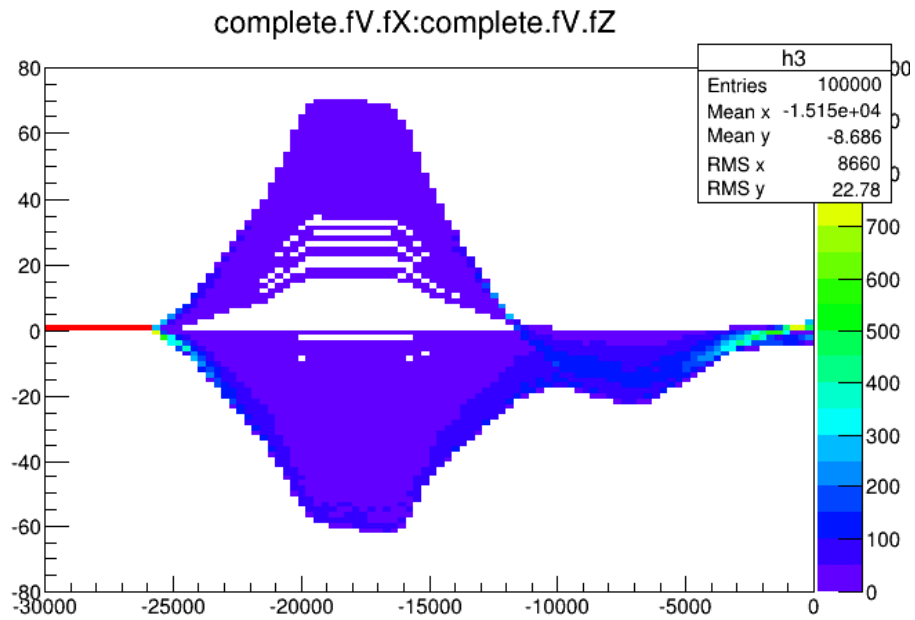
Element positions along the beam line

step number	length of element (meters)	integrated length from production target (meters)	integrated length from HADES target (mm, counted backward)	step name
0	0	0	-33645,24	production target
1	0,9	0,9	-32745,24	in Q1
2	1	1,9	-31745,24	out Q1
3	1	2,9	-30745,24	in Q2
4	1	3,9	-29745,24	out Q2
5	0,6	4,5	-29145,24	in FOPI
6	0	4,5	-29145,24	FOPI
7	1	5,5	-28145,24	FOPI
8	0,5	6	-27645,24	FOPI
9	0	6	-27645,24	out FOPI
10	1,58	7,58	-26065,24	in dipole 1
11	1,47262	9,05262	-24592,62	out dipole1
12	3	12,05262	-21592,62	in Q3
13	0,6	12,65262	-20992,62	out Q3
14	0,8	13,45262	-20192,62	in Q4
15	0,6	14,05262	-19592,62	out Q4
16	1,5	15,55262	-18092,62	nominal interm. focus
17	1	16,55262	-17092,62	detector 1
18	0,5	17,05262	-16592,62	in Q5
19	0,6	17,65262	-15992,62	out Q5
20	0,8	18,45262	-15192,62	in Q6
21	0,6	19,05262	-14592,62	out Q6
22	3	22,05262	-11592,62	in dipole 2
23	1,47262	23,52524	-10120	out dipole2
24	2,81	26,33524	-7310	in Q7
25	1	27,33524	-6310	out Q7 (Q7 length = 1m)
26	0,91	28,24524	-5400	detector 2
27	1,9	30,14524	-3500	in Q8
28	1	31,14524	-2500	out Q8 (Q8 length = 1 m)
29	0,6	31,74524	-1900	in Q9
30	0,4	32,14524	-1500	out Q9 (Q9 length = 0.4 m)
31	0,5	32,64524	-1000	interm. Point
32	1	33,64524	0	HADES target
33	0	33,64524	0	HADES target

This excell table gives positions in mm going from target backward

Effect of Dp/p =+-8%

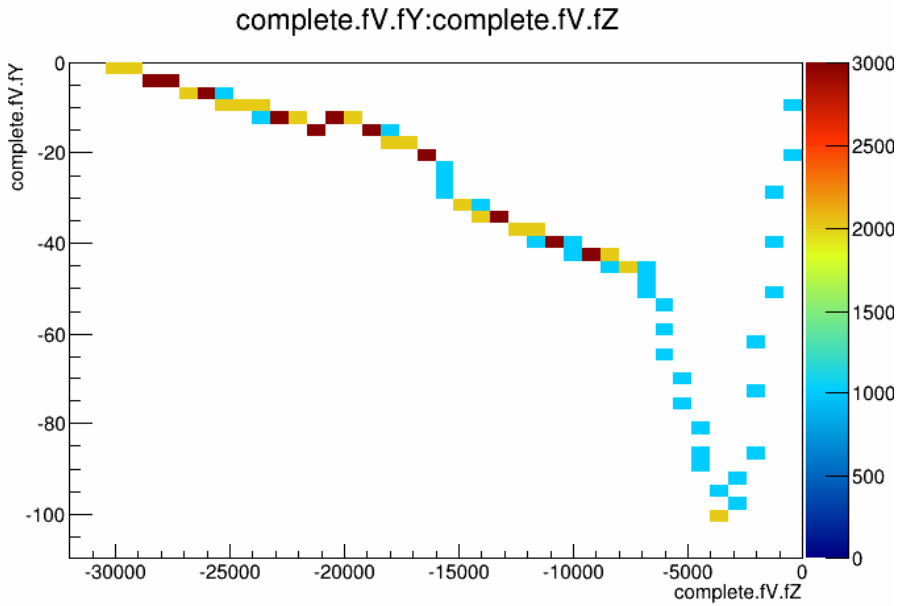
```
//Init the reaction for c.m. sampling:
PBeamLineSimulation *sim = new PBeamLineSimulation("beam", "Beam line simulation");
sim->SetReaction("p + p");
//Read the data file
sim->InitBeamLine("pibeam_set6.data");
//Place (virtual) detector along the beam line. Units are in mm:
sim->AddDetector("det1", -3000.0);
sim->AddDetector("det2", -6000.0);


for (double dist = -30000; dist<0; dist+=300)
        sim->AddDetector("complete", dist);
//open the ROOT file and type:
//data.Draw("complete.fV.fX:complete.fV.fZ >>h3(100,-30000,0,100,-80,80)","","colz");


//Choose which module should be the target:
sim->TargetIsElement(33);
//Select the global momentum:
sim->SetGlobalMomentum(3.0);

//Beam profile at the production target
//position in mm:
sim->Do("_beam_x = 0;  _beam_y = 0;");
//divergence in px/pz and py/pz:
//sim->Do("_beam_px = 0; _beam_py = 0.05;");
sim->Do("_beam_px = 0; _beam_py = 0;");
//momentum spread:
sim->Do("_beam_dp = 0.16*sampleFlat() - 0.08; "); //+- 8%
//sim->Do("_beam_dp = -0.08; ");

//Add and enable module:
makeDistributionManager()->Add(sim);

PReaction my_reaction("_T1 = 2.2","p","p","p p eta [dilepton [e+ e-] g]", "beam_line");
//PReaction my_reaction("_T1 = 2.2","p","p","p p", "beam_line");

my_reaction.Print();   //The "Print()" statement is optional
my_reaction.Loop(1000);
```

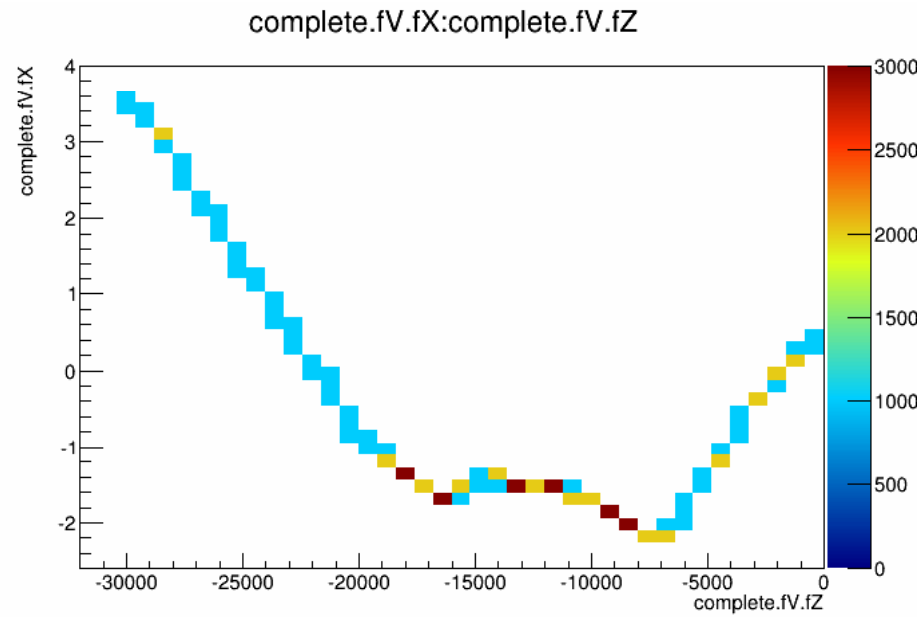


units in Z and X are in mm

Pion production point: X:1mm, Y:1 mm

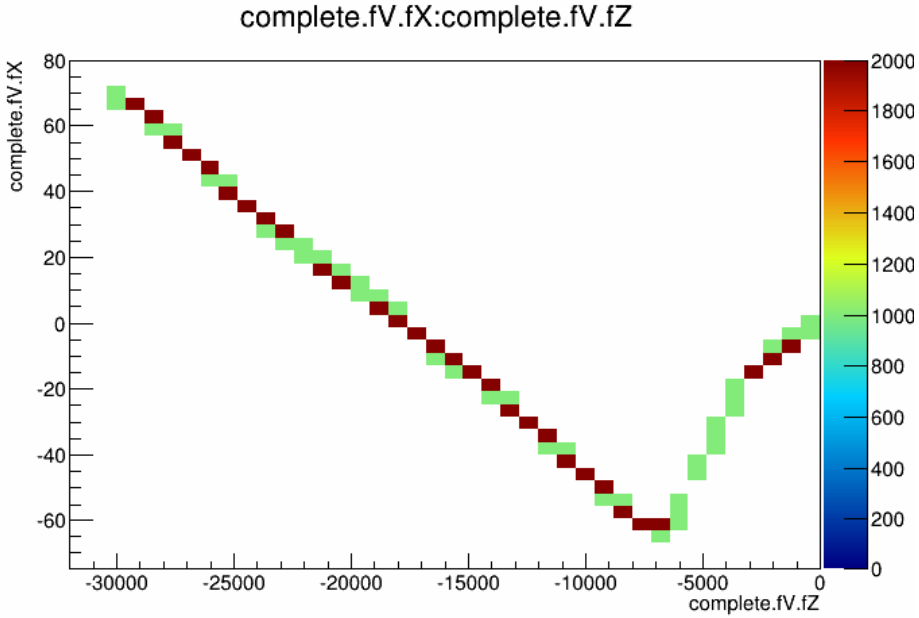
```
 //Beam profile at the production target    //position in mm:
sim->Do("_beam_x = 1.;  _beam_y = 1. ;
//divergence in px/pz and py/pz:
//sim->Do("_beam_px = 0; _beam_py = 0.05;");
sim->Do("_beam_px = 0; _beam_py = 0;");
//momentum spread:
//sim->Do("_beam_dp = 0.16*sampleFlat() - 0.08; "); //+- 8%
sim->Do("_beam_dp = 0.00;
```

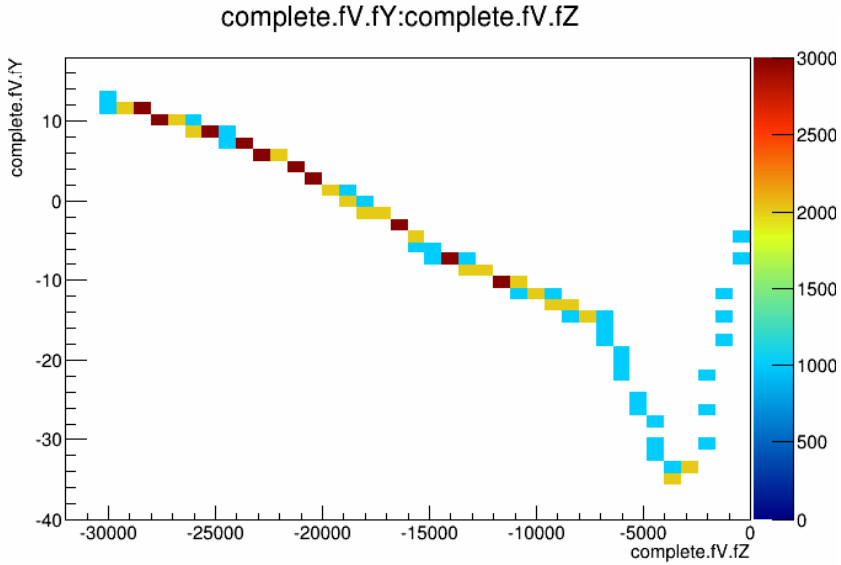




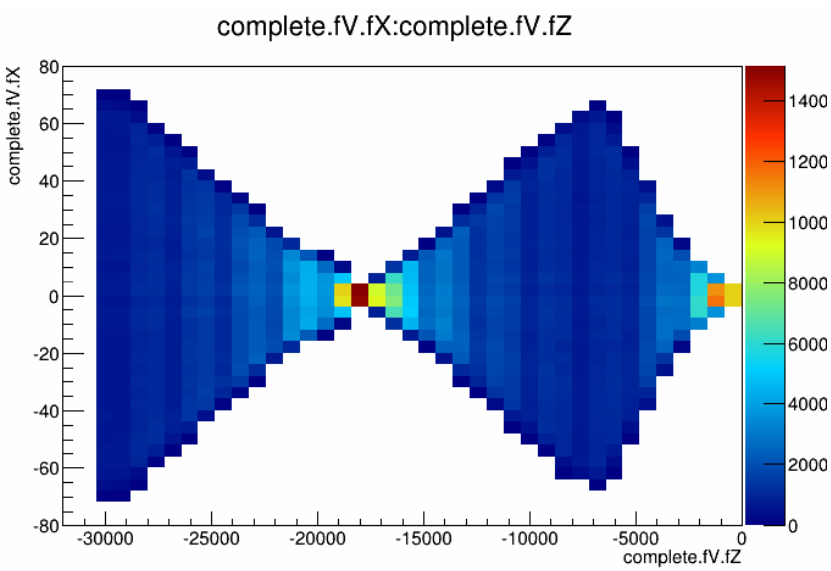
Pion emission angles: $\Delta\Theta$=10, $\Delta\phi$=20 mrad

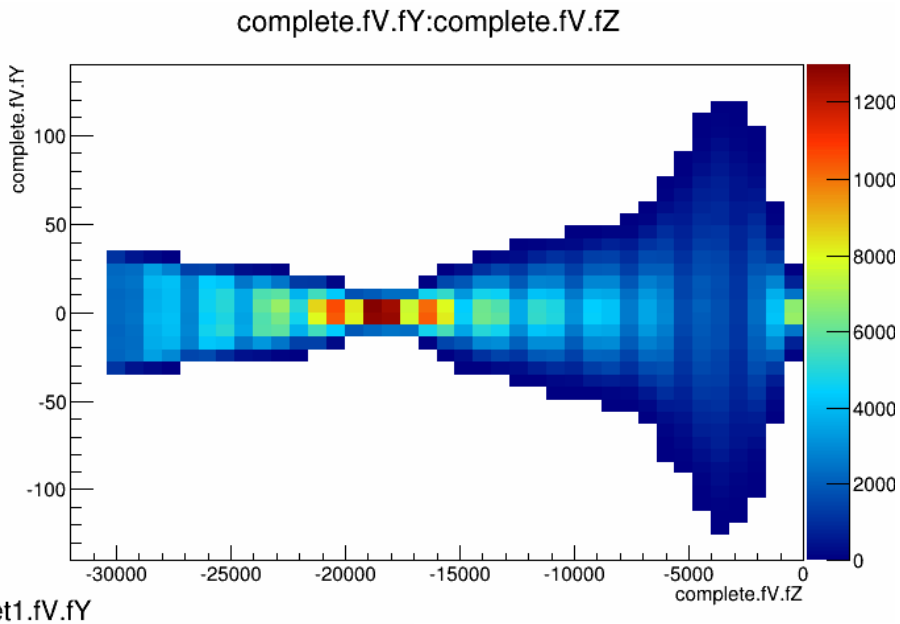
```
//Beam profile at the production target    //position in mm:
sim->Do("_beam_x = 0..;  _beam_y = 0. ;
//divergence in px/pz and py/pz:
//sim->Do("_beam_px = 0; _beam_py = 0.05;");
sim->Do("_beam_px = 10; _beam_py = 20;");  // mrad  !!
//momentum spread:
//sim->Do("_beam_dp = 0.16*sampleFlat() - 0.08; "); //+- 8%
sim->Do("_beam_dp = 0.00;
```

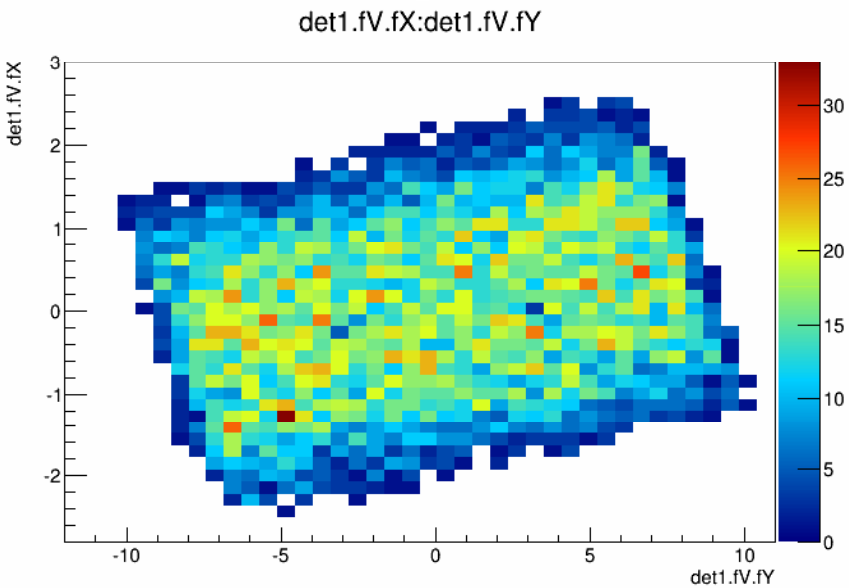




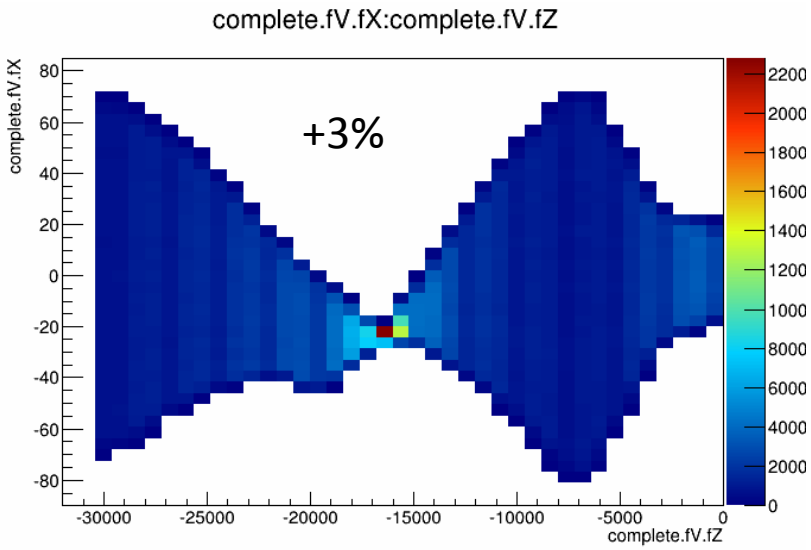
Dp=0.0, Full smearing $\Delta\Theta,\phi$ and $\Delta X_0 Y_0$

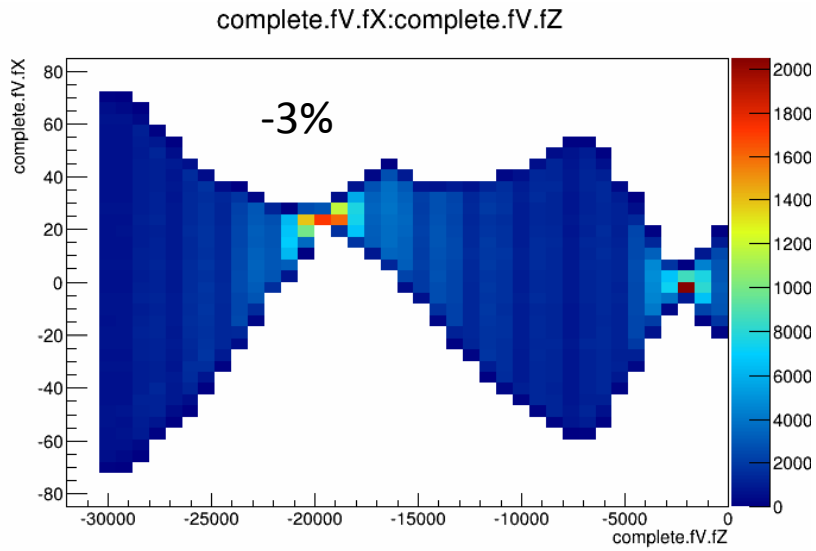


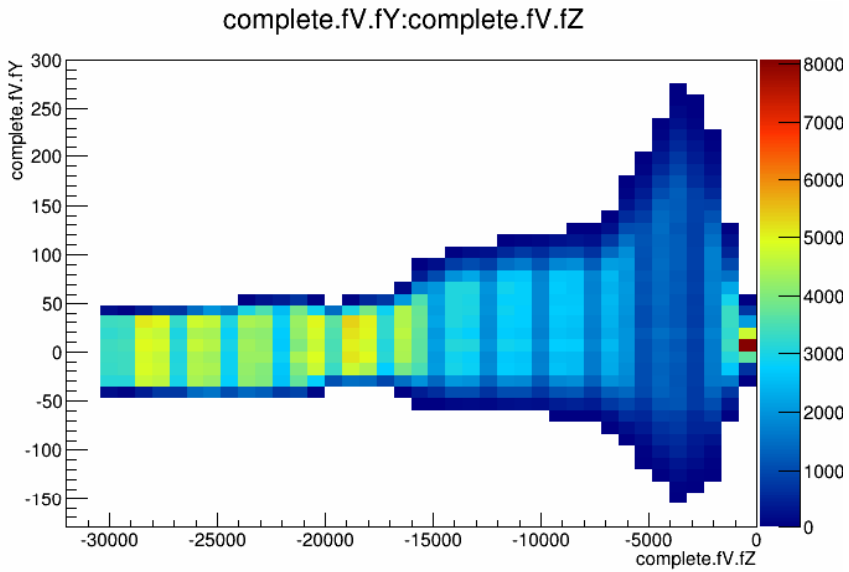


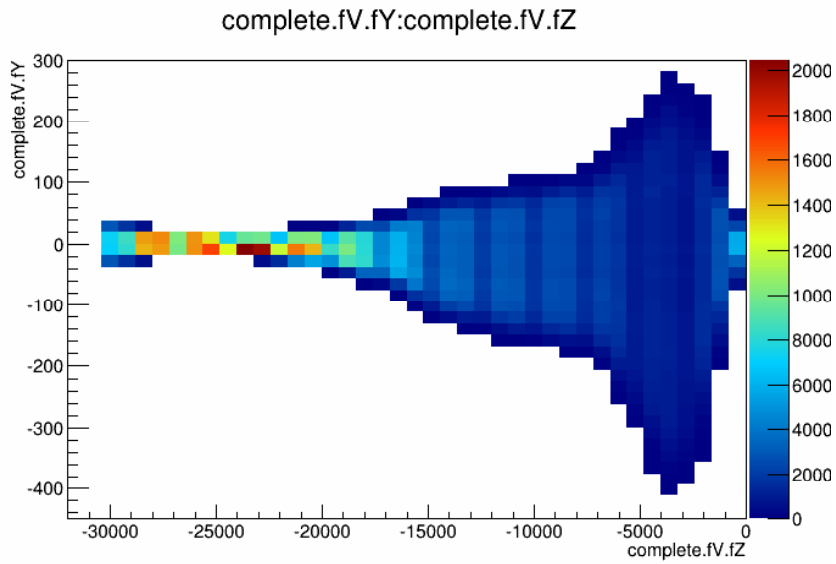


Dp=±3%, Full smearing $\Delta\Theta$, $\Delta\phi$ and $\Delta X_0, Y_0$



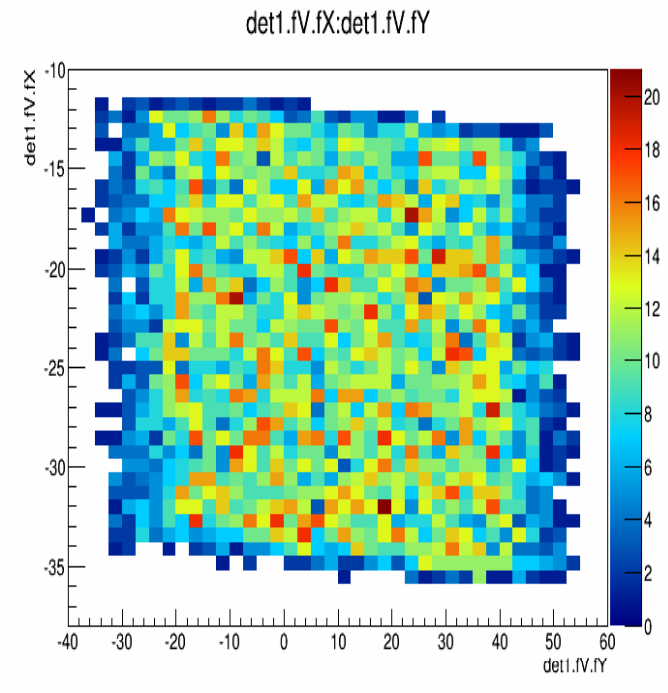




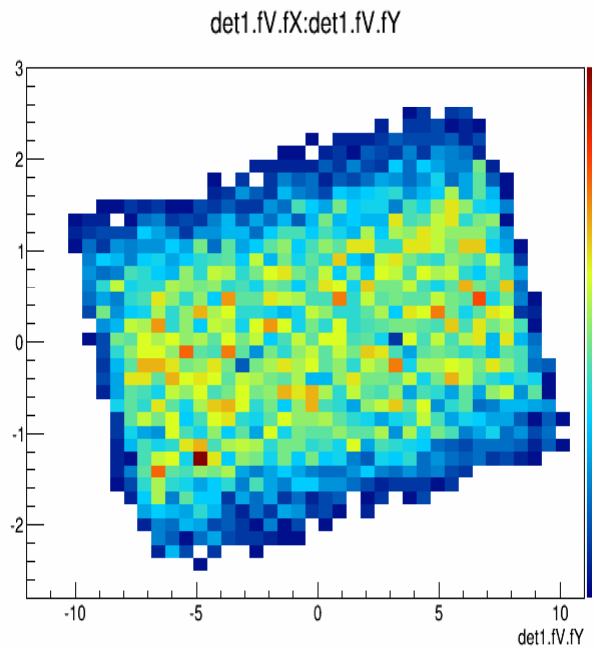


Dp=0%,±3%, Full smearing $\Delta\Theta,\phi$ and $\Delta X_0 Y_0$

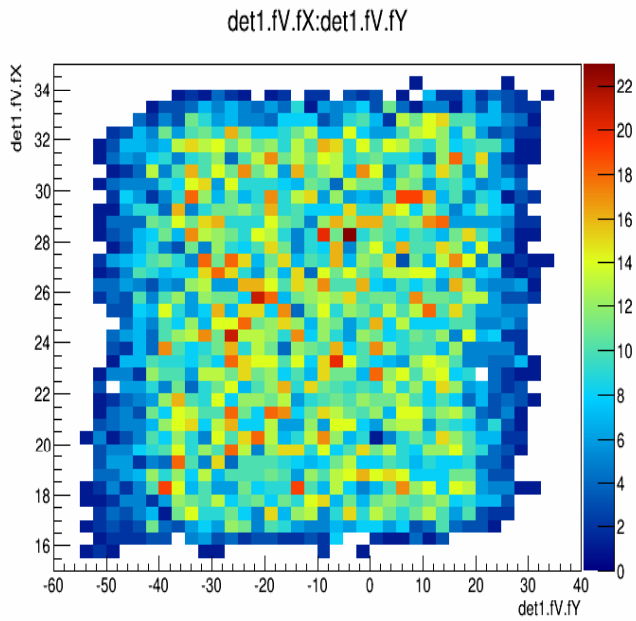
+3%



0%

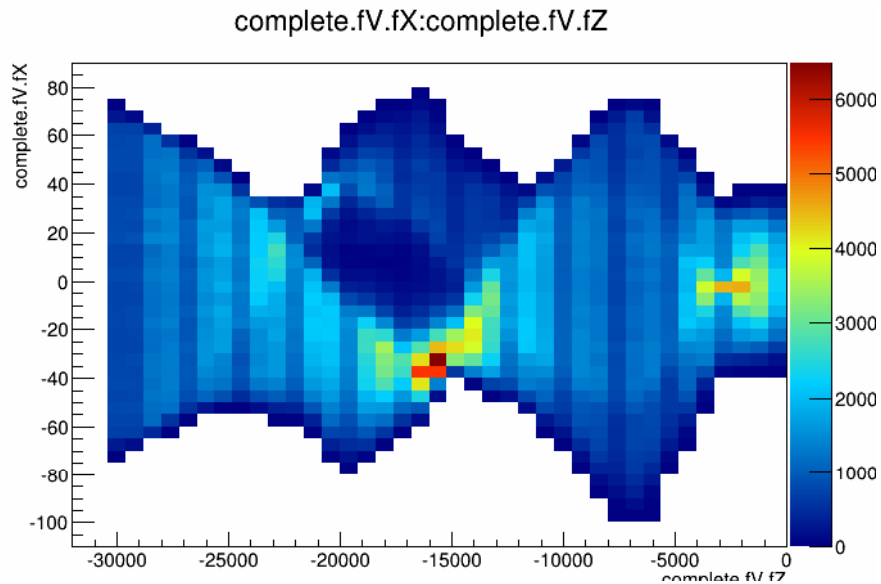


-3%

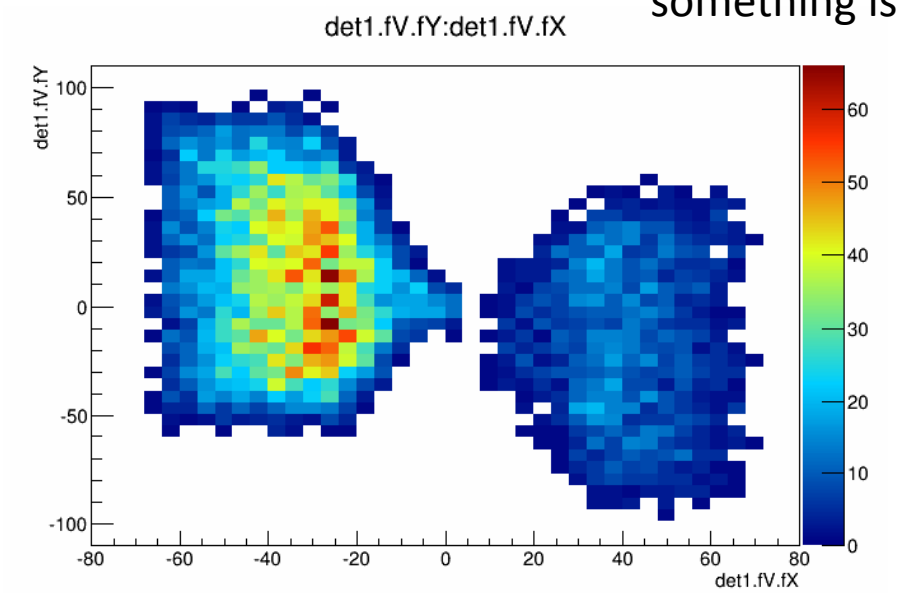


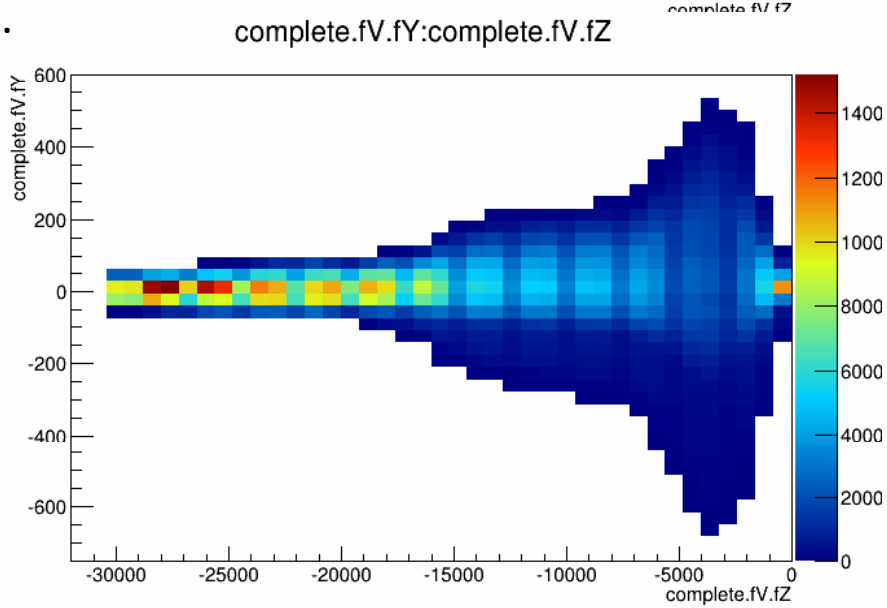
Full smearing Dp, $\Delta(\Theta,\phi)$ and $\Delta(X_0Y_0)$

```
 //Place (virtual) detector along the beam line. Units are in mm:
sim->AddDetector("det1", -18092.6);
sim->AddDetector("det2", -6310.0);
for (double dist = -30000; dist<0; dist+=300)
sim->AddDetector("complete", dist);
//Choose which module should be the target:
sim->TargetIsElement(33);
 //Select the global momentum:
 sim->SetGlobalMomentum(3.0);
//Beam profile at the production target    //position in mm:
sim->Do("_beam_x = 1.*sampleFlat()-0.5;  _beam_y = 1.0*sampleFlat()-0.5;"); //+- 1mm
//divergence in px/pz and py/pz:
sim->Do("_beam_px = 20.*sampleFlat()-10.; _beam_py = 100.*sampleFlat()-50.;");
//momentum spread:
 sim->Do("_beam_dp = 0.12*sampleFlat() - 0.06; "); //+- 6%
```

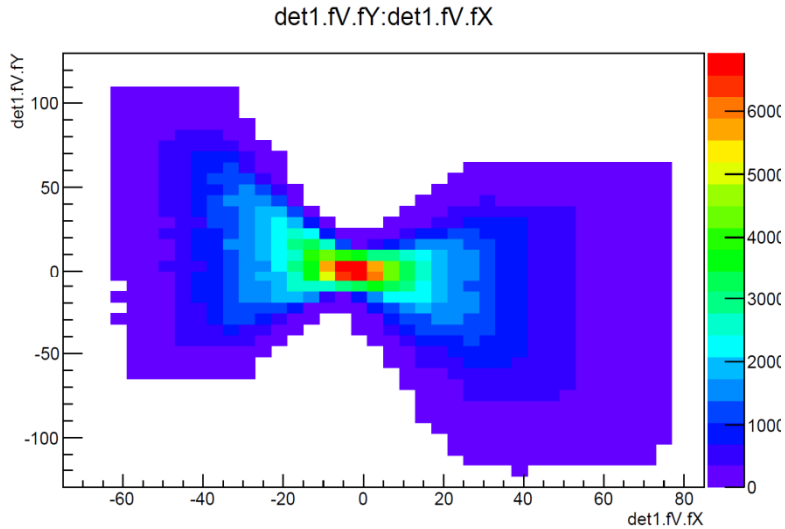


something is wrong..





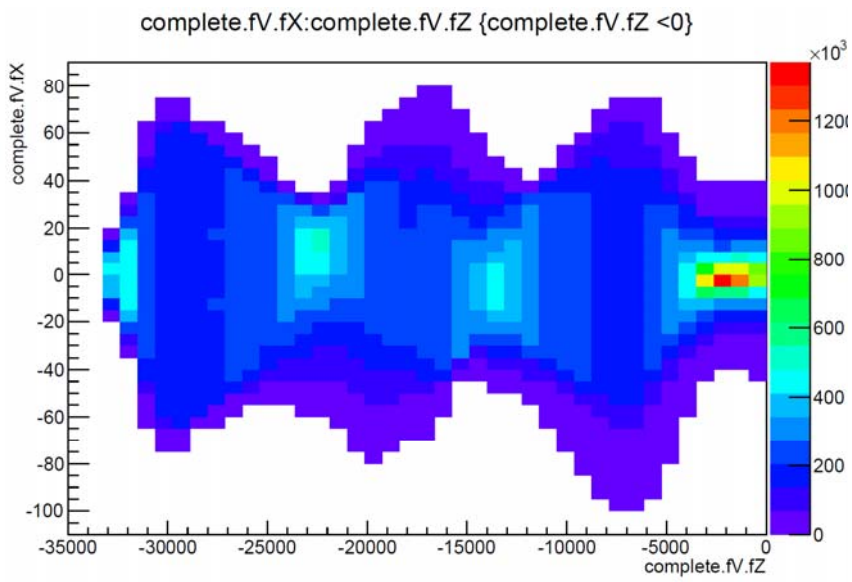
Full smearing Dp, $\Delta\Theta,\phi$ and $\Delta X_0 Y_0$

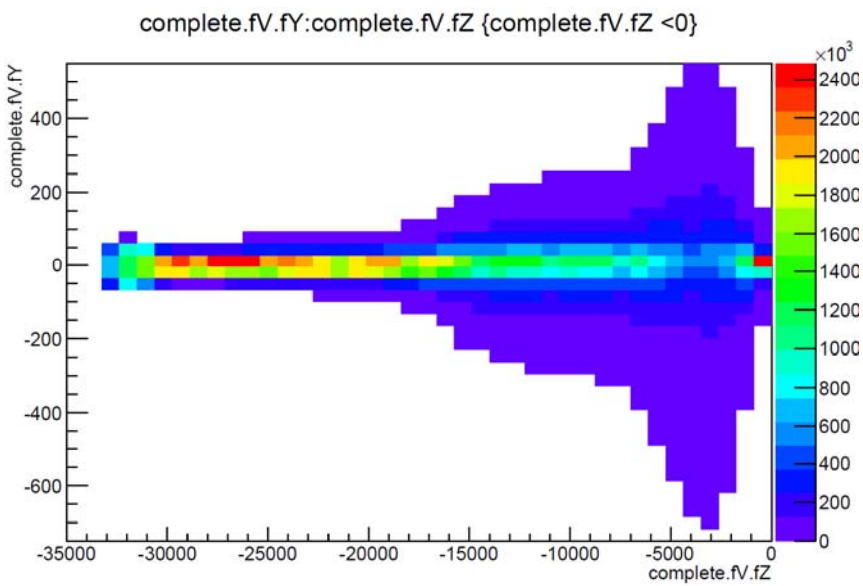


problem with momentum sampling found and removed

calculation of the output momentum from dp to p was not correct

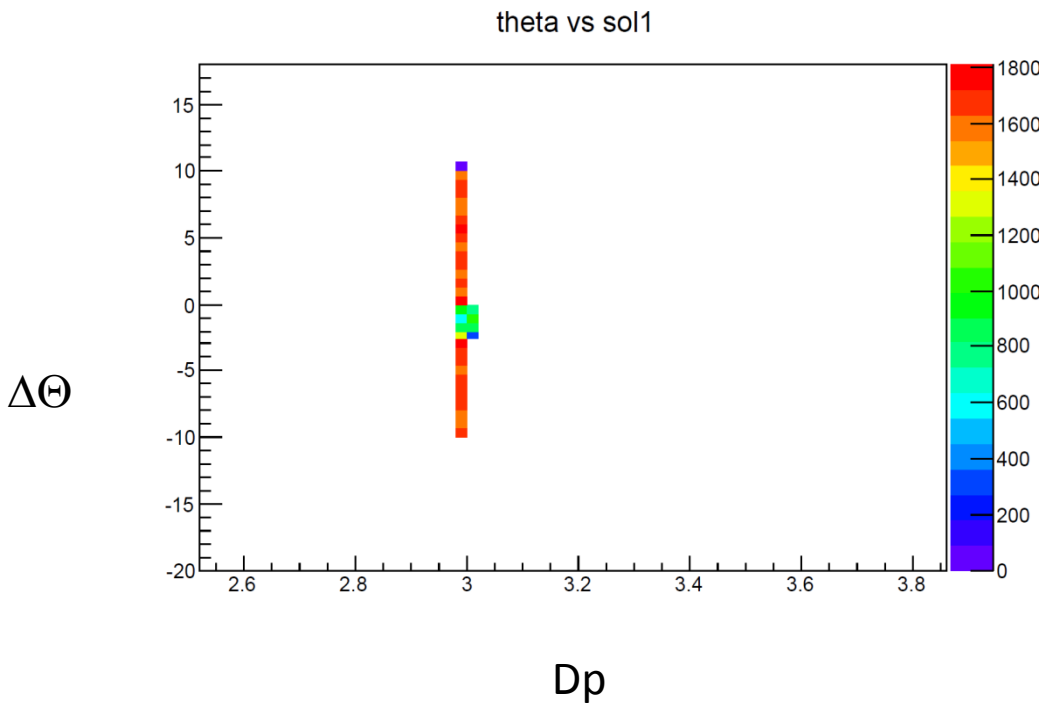
Now all looks fine !





Test of mementum reconstruction

- delta(p)/p = 3 % emission angles $\Delta\Theta$ uniform between -10 mrad and + 10 mrad i.e. px/pz -0.01 to 0.01 , no x and Y smearing

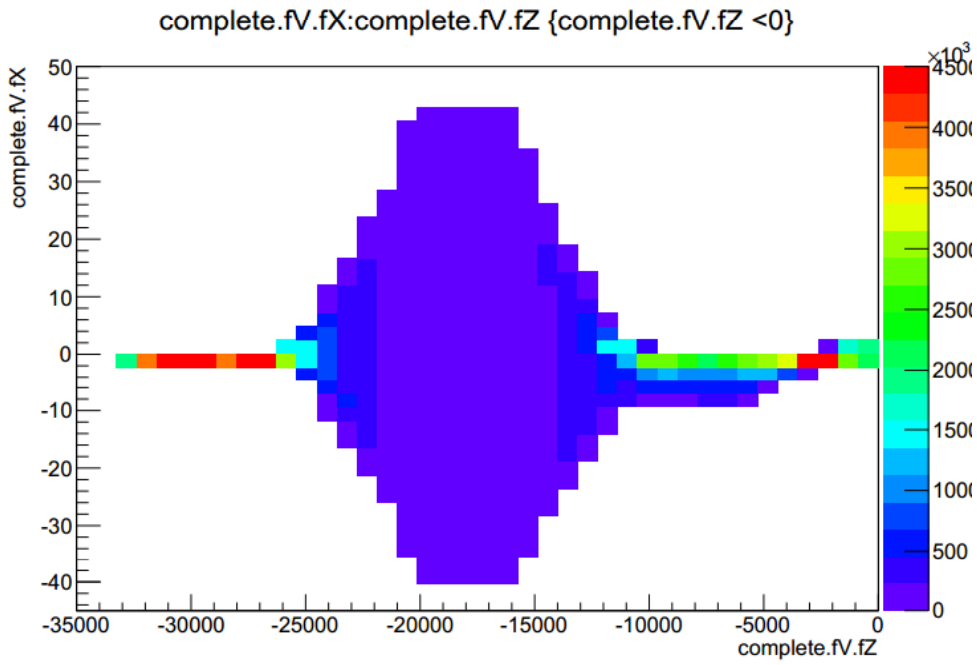


New Simulation

```
{
    //Init the reaction for c.m. sampling:

    PBeamLineSimulation *sim = new PBeamLineSimulation("beam", "Beam line simulation");
    sim->SetReaction("pi- + p");
    //Read the data file
    sim->InitBeamLine("pibeam_set6_mod.data");
    //Place (virtual) detector along the beam line. Units are in mm:

    sim->AddDetector("det1", -17092.6);

    sim->AddDetector("det2", -5400.0);

    sim->AddDetector("det3",-300);

    sim->AddDetector("det4",-10);

    for (double dist = -36000; dist<0; dist+=100)

            sim->AddDetector("complete", dist);

    //open the ROOT file and type:

    //Choose which module should be the target:

    sim->TargetIsElement(33);

    //Select the global momentum:

    sim->SetGlobalMomentum(3.0);

    sim->Do("_beam_y = 1.0*sampleFlat()-0.5;"); //+- 1mm
    sim->Do("_beam_x = 1.0*sampleFlat()-0.5;"); //+- 1mm

    //divergence in px/pz and py/pz:

    sim->Do("_beam_px = 0.02*PUtils::sampleFlat() - 0.01;");

    sim->Do("_beam_py = 0.1*PUtils::sampleFlat() - 0.05;");

    //momentum spread:
    sim->Do("_beam_dp = 0.12*sampleFlat() - 0.06;"); //+- 6%

    //Beam profile at the production target
    //position in mm:

    //Add and enable module:

    makeDistributionManager()->Add(sim);

PReaction my_reaction("_T1 = 1.3","pi-","p","n w [e+ e-]","pi_p_n_rho0");

    my_reaction.Print();   //The "Print()" statement is optional

    my_reaction.Loop(500000);
```

Results

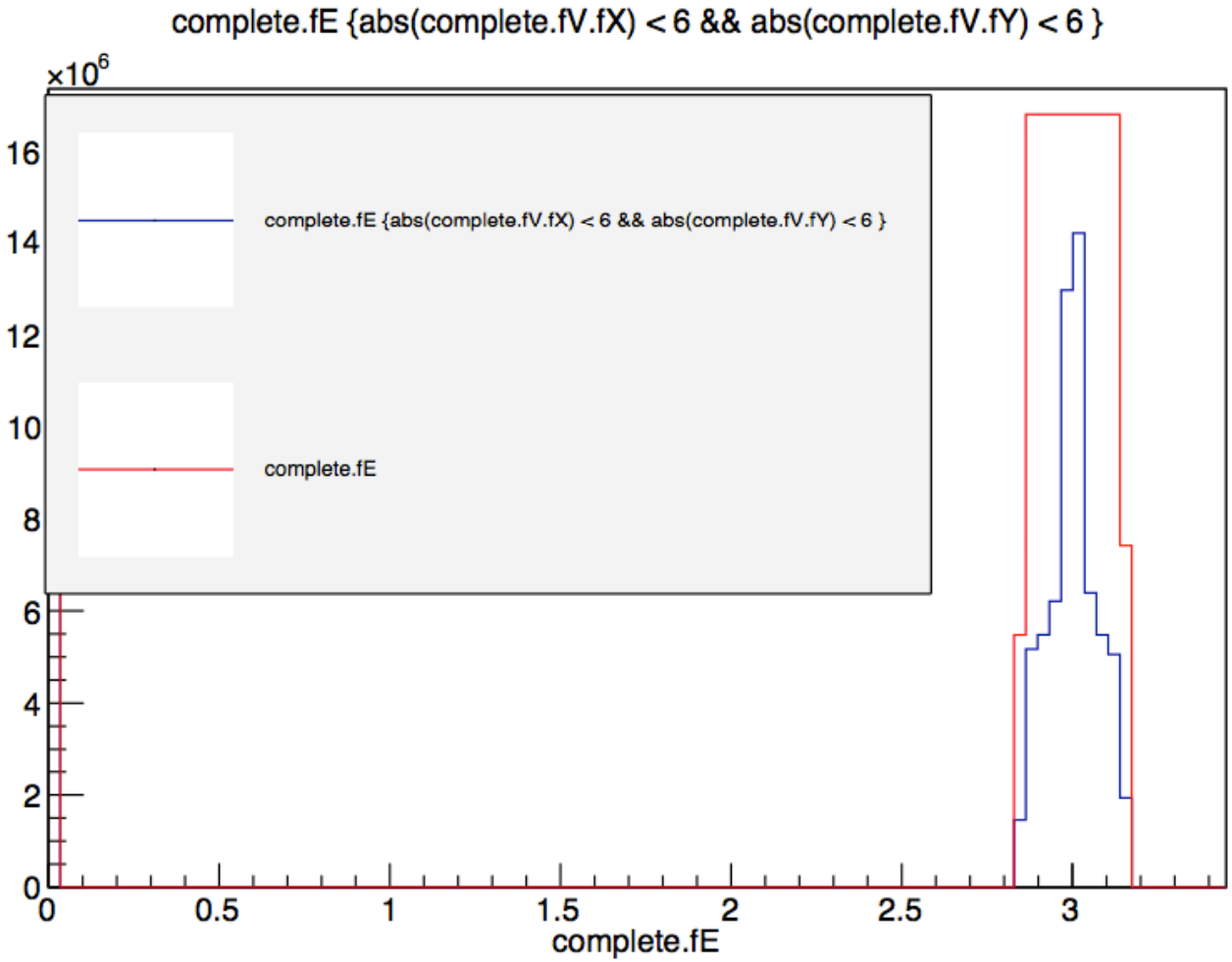
- Remark to syntax in PLUTO macro : settings of x, y /px ,py need to be set in separate lines
- Results for beam profile are as expected



Momentum reconstruction at HADES target

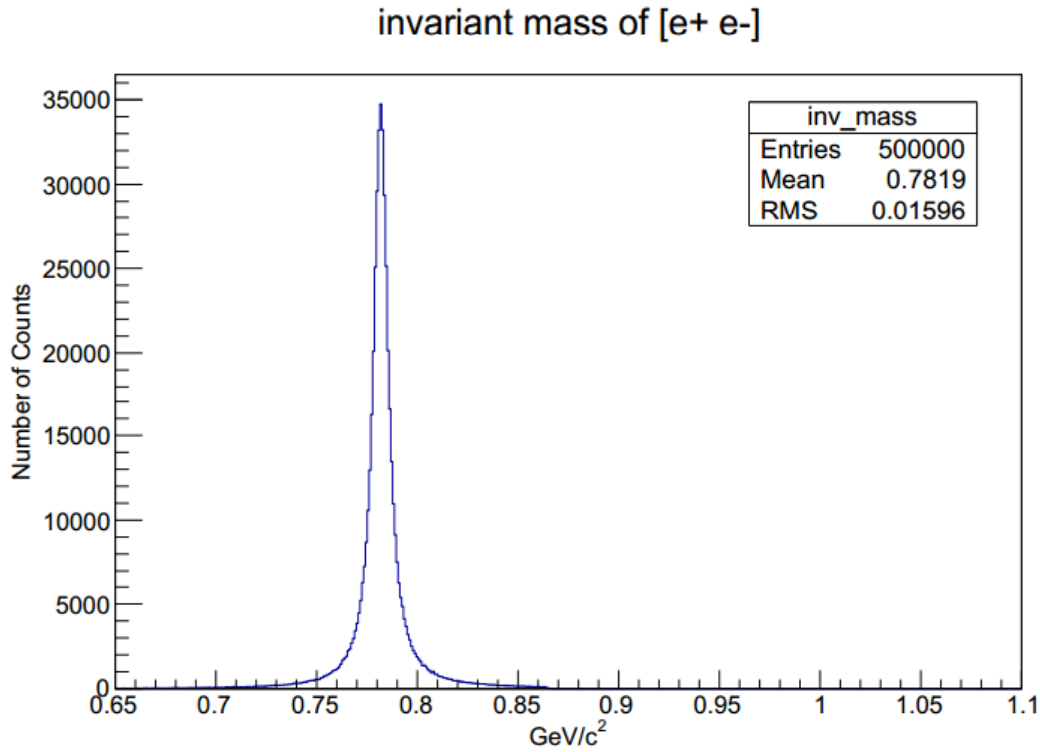
- A reconstruction of p_x, p_y ,p_z was preformed
- To check the reconstructed values of momentum a so called target detector was implemented on the beam line simulation (1 cm before the target)
- To simulate the acceptance of the beam line, a condition was introduced ($x < 6$ & $y < 6$)
- A start detector was also added to the beam line (a hit from the start detector says that this was a valid event)
- Simulated reaction: $\pi\uparrow- + p \rightarrow n\ \omega \rightarrow n\ e\uparrow+\ e\uparrow-$
- Calculations of the mass of neutron was preformed for three situations (fixed momentum = 1.3, momentum form target detector, momentum coming from reconstruction)

Acceptance cut

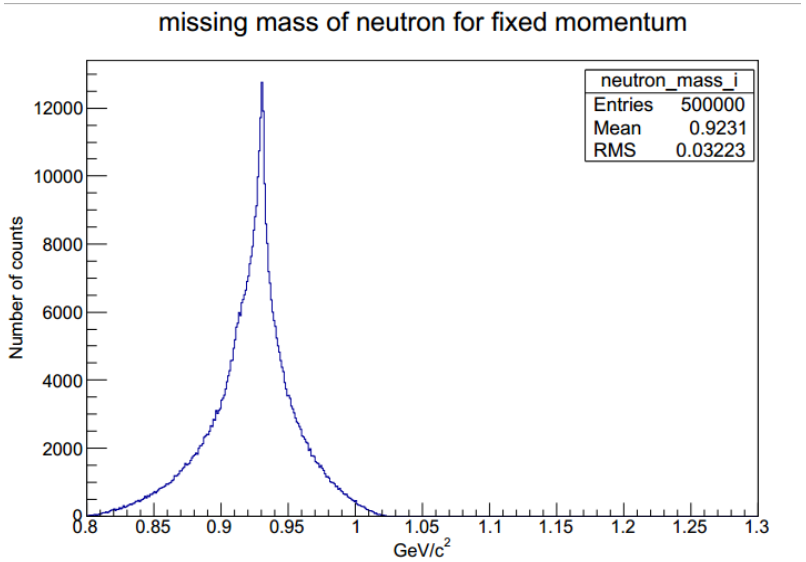


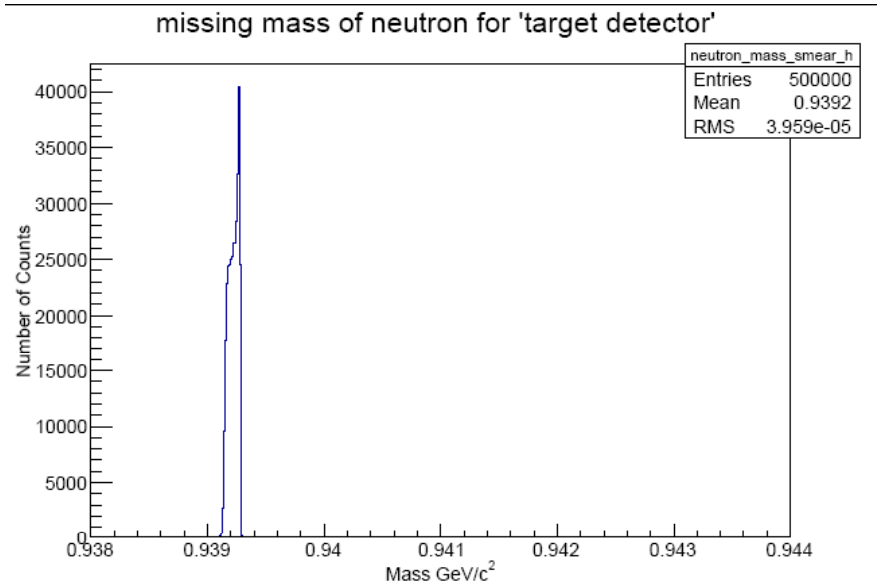
Results

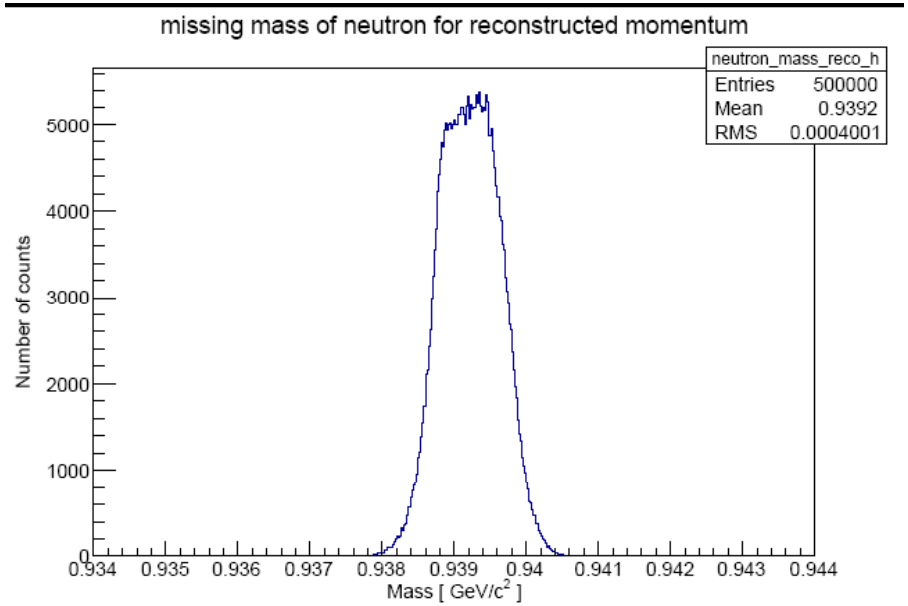
- Invariant mass of e+ e-



Neutron mass calculations

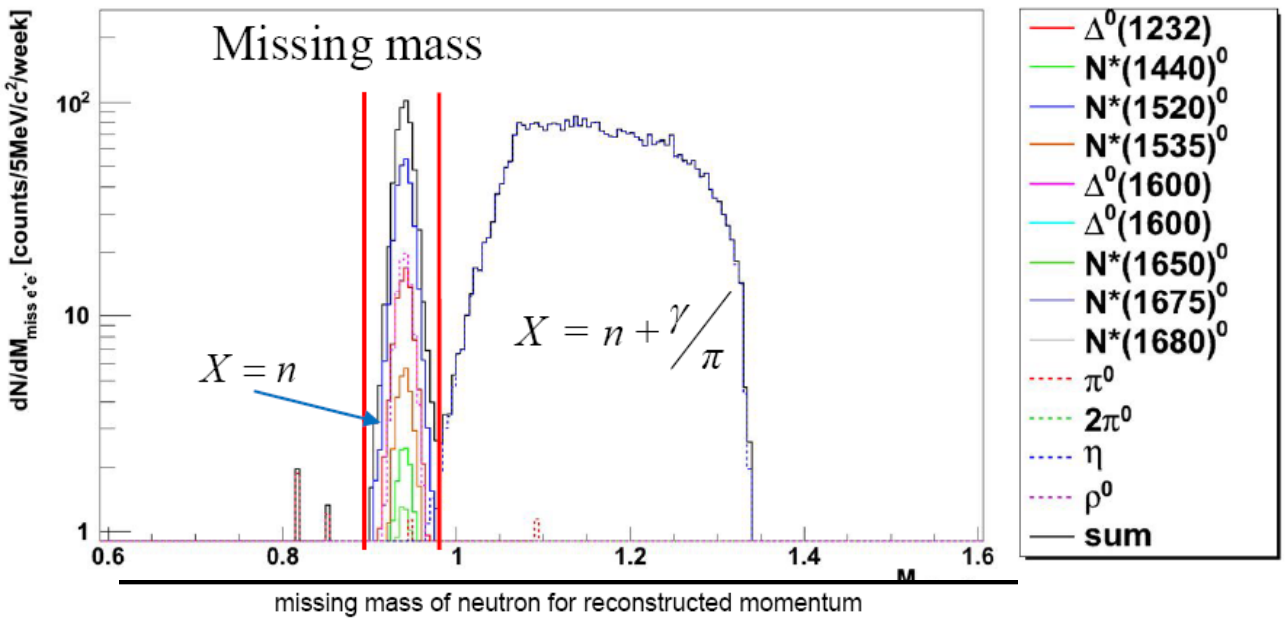




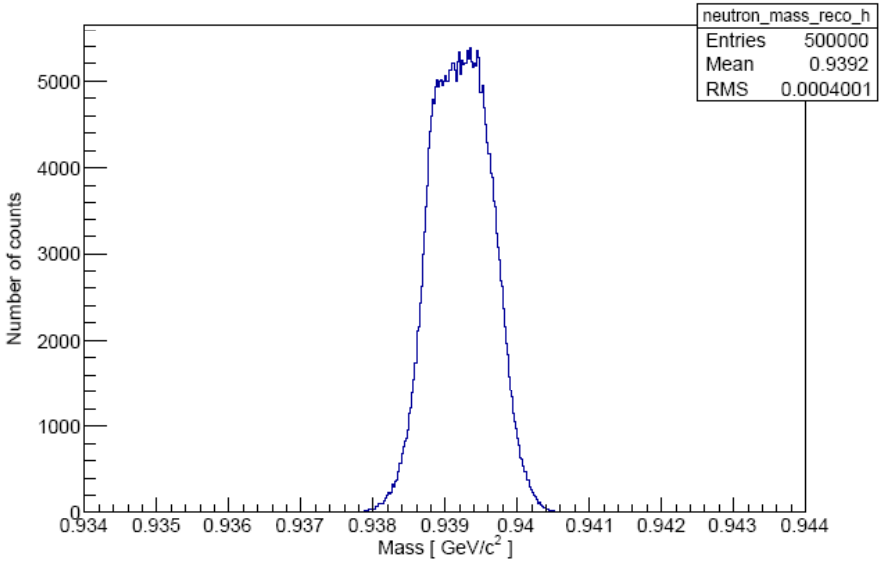


- pion beam acceptance included
- hits on start detector required (30 cm in front of the target 16x16 mm^2 active area)

Comparison of missing mass resolutiuon from HADES and pion beam reconstruction procedure



HADES resolution ;
beam momentum fixed



no HADES resolution;
beam momentum
reconstructed from hits
on tracker

Conclusions

- The issue of missing momentum in the simulation is fixed;
- Calculations of neutron missing mass show that the algorithm works as it should (same results coming from reconstruction and target detector)