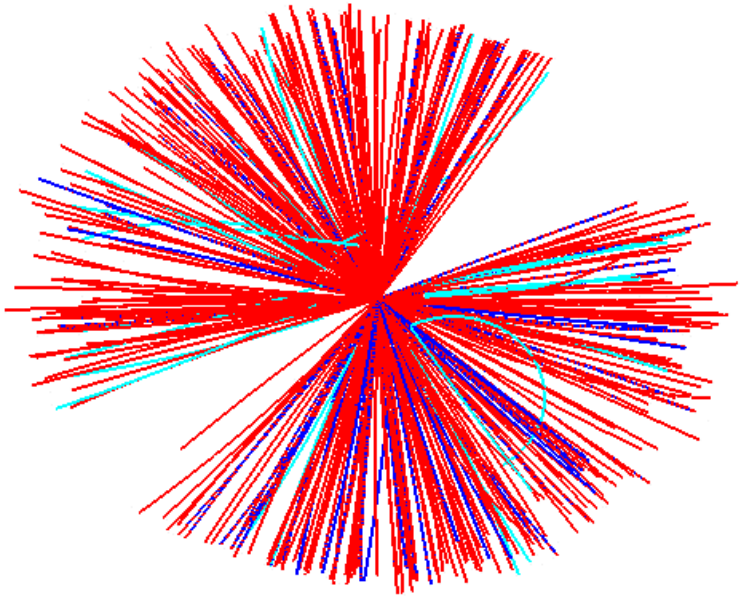
Overview of SVX Stand-Alone Tracking Software



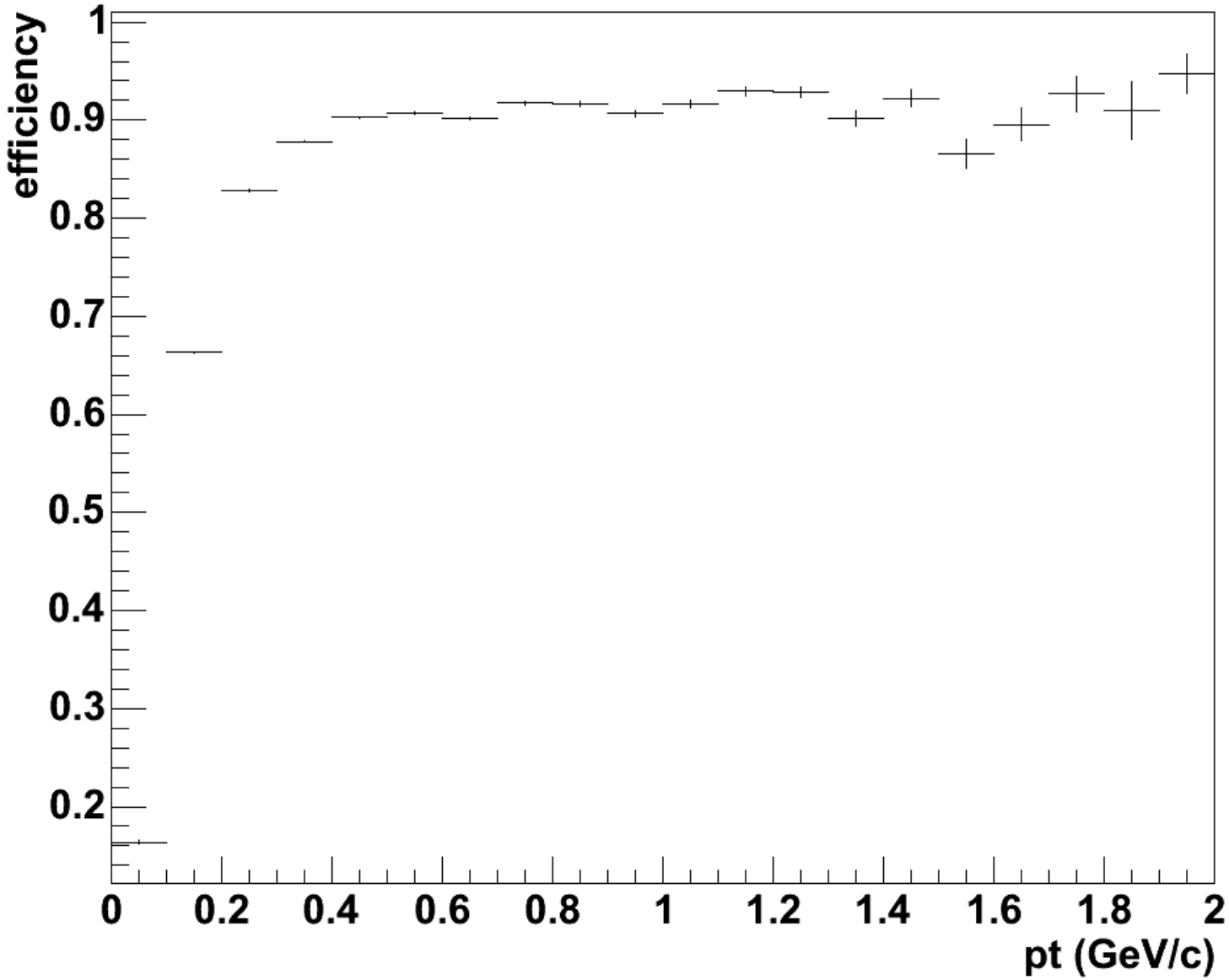
Alan Dion

Stony Brook University

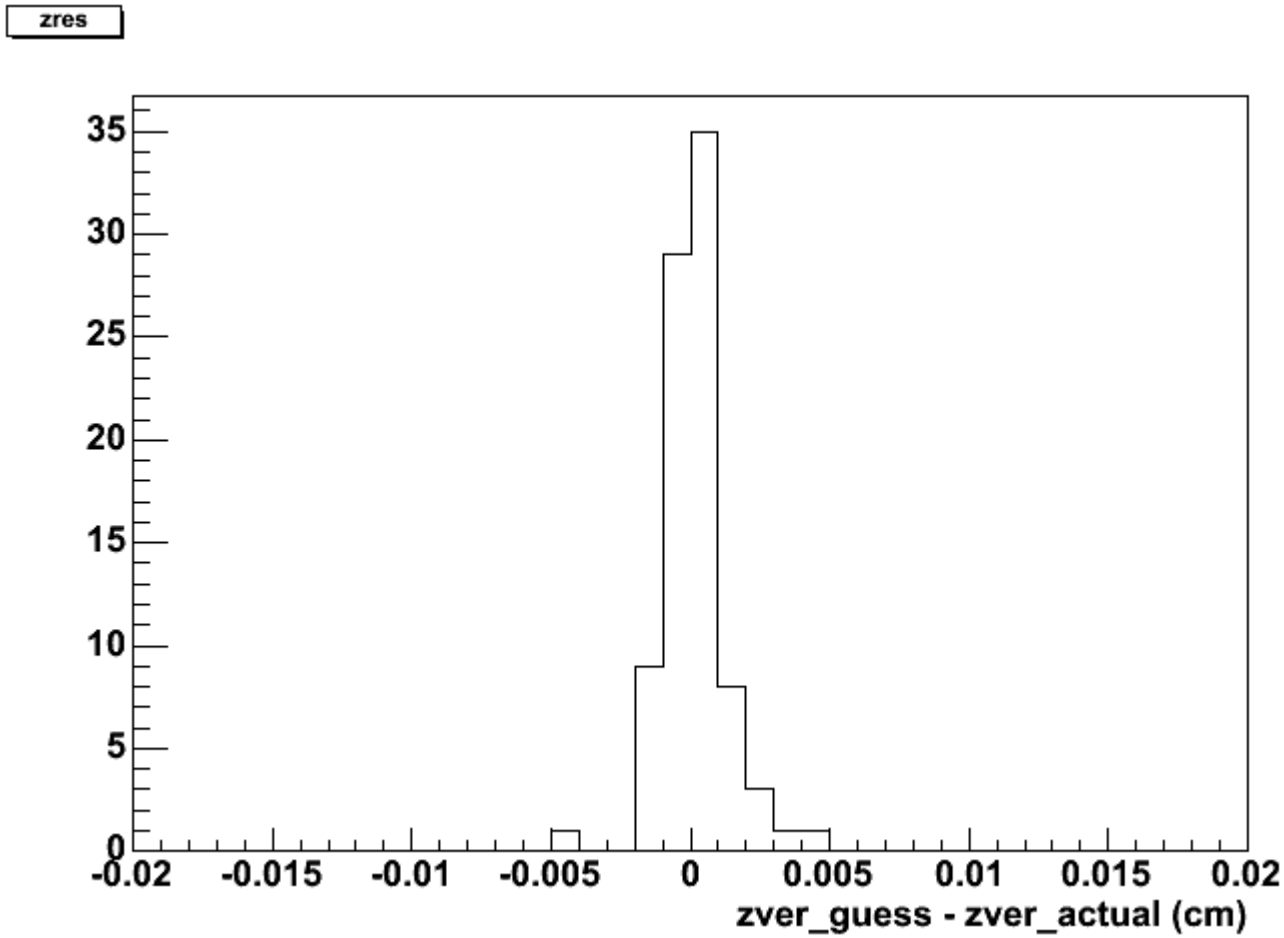
2005.04.13

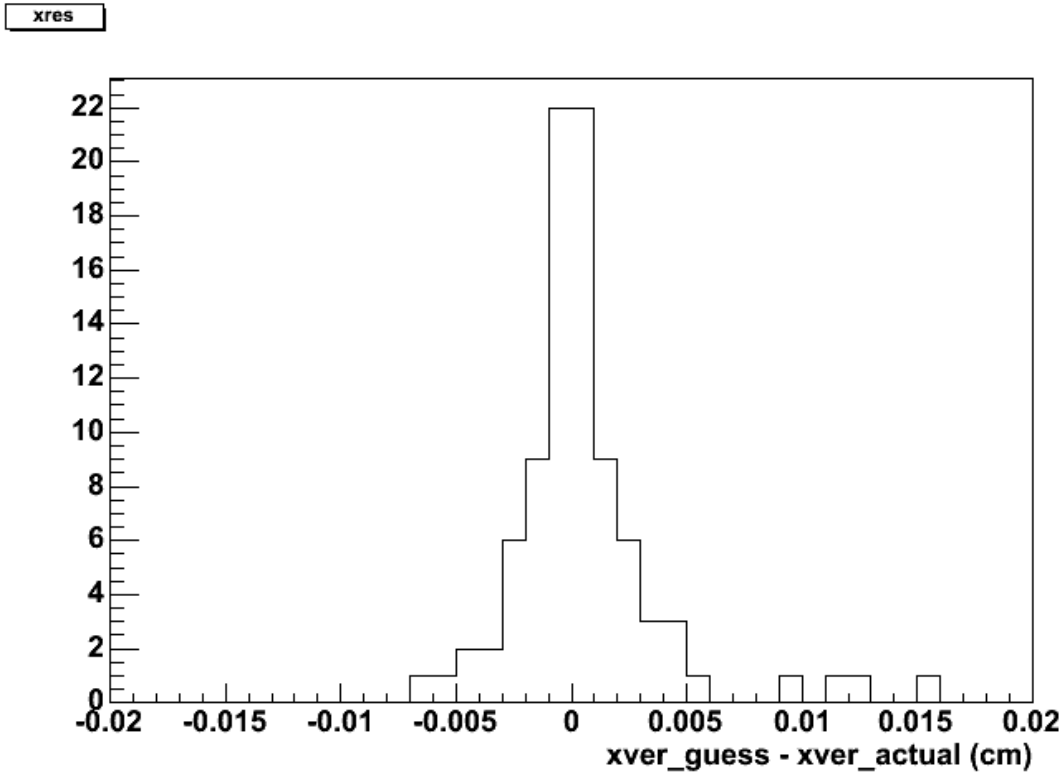
Reconstruction Efficiency

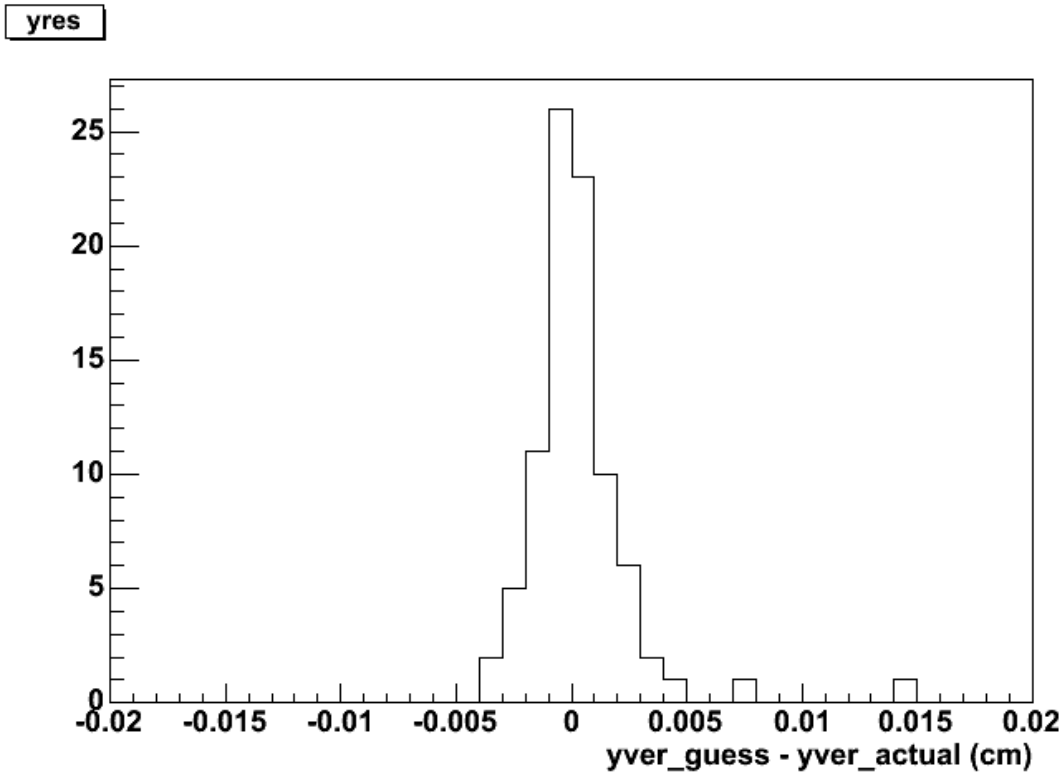
Reconstructed primary tracks – the errors are event-by-event standard deviations, weighted by multiplicity



Vertex Resolution




xres
22
20
18
16
14
12
10
8
6
4
2
0
-0.02
-0.015
-0.01
-0.005
0
0.005
0.01
0.015
0.02
xver_guess - xver_actual (cm)


yres
25
20
15
10
5
0
-0.02
-0.015
-0.01
-0.005
0
0.005
0.01
0.015
0.02
yver_guess - yver_actual (cm)

Why do we need pT?

Gamma-jet correlations

Track-matching to central-arm tracks

Other “stand-alone” measurements? SVX has high azimuthal acceptance

Limitations

Multiple scattering — low pT

Measurement resolution — high pT

Current limit still ~ 20%

Toy Monte Carlo

To gain better understanding of limitations on p_T determination, I developed a toy event generator/detector response with the following properties:

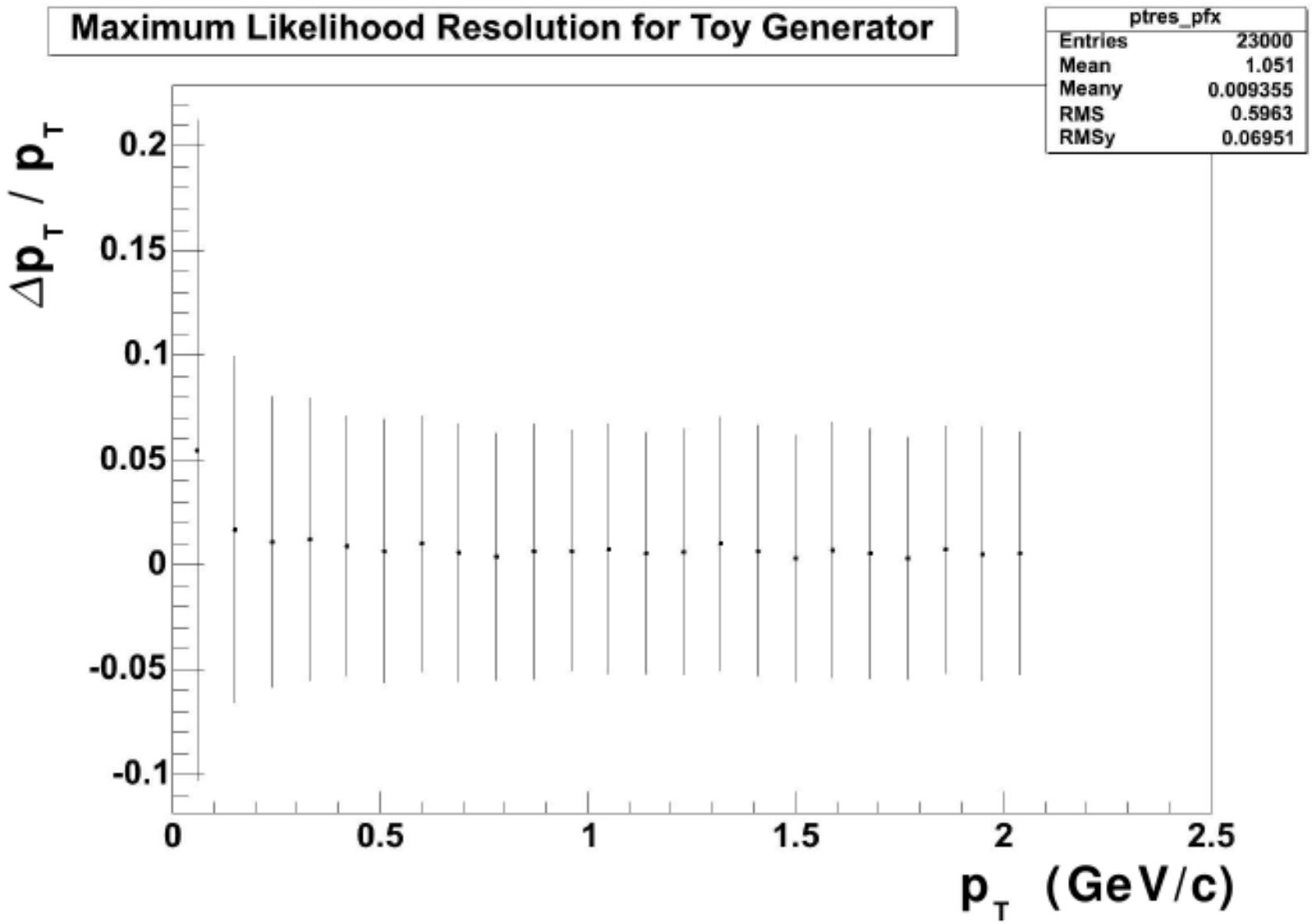
multiple scattering: Gaussian approximation to Moliére

has position resolution of SVX

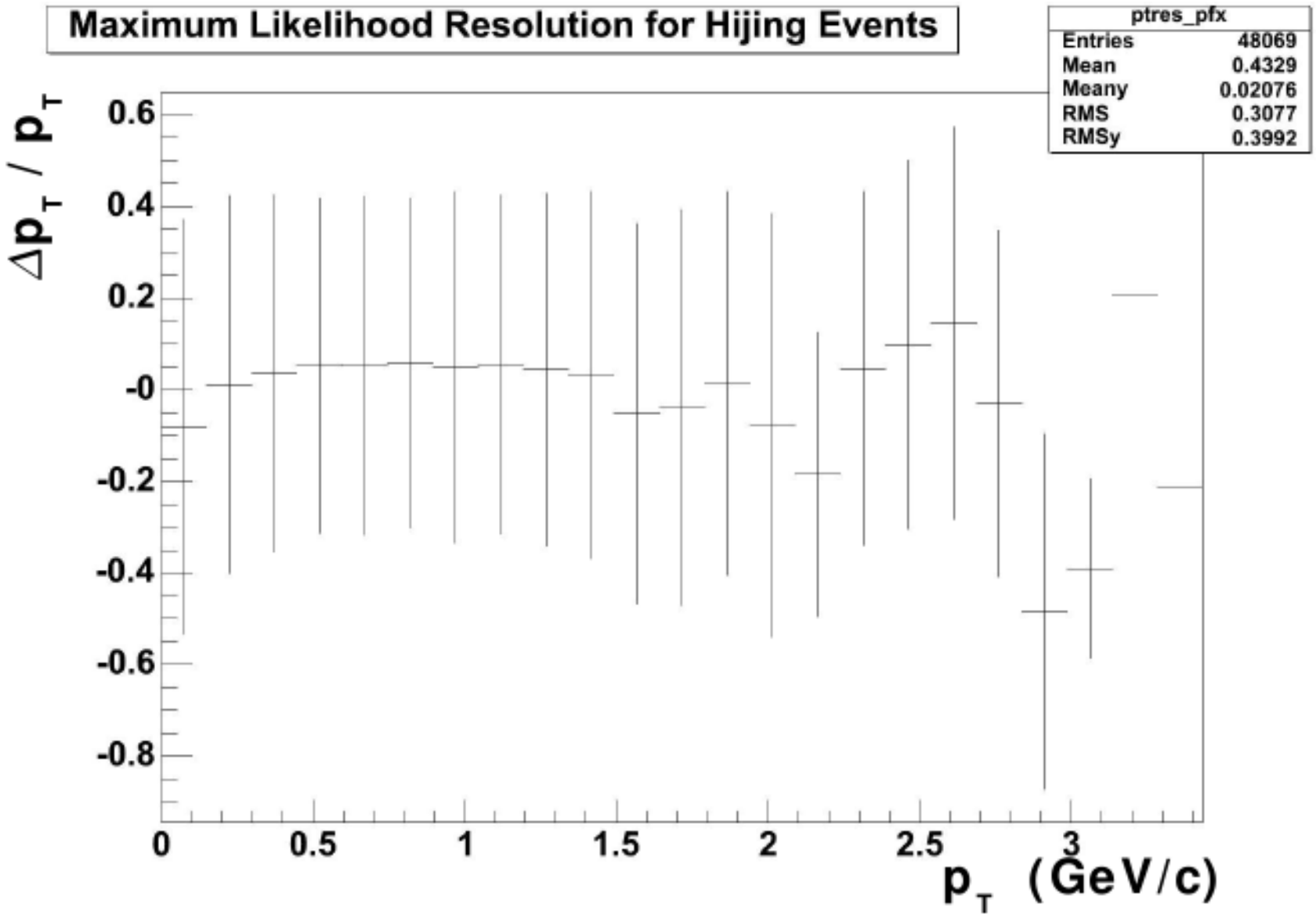
no energy loss

I used a brute force algorithm to determine the most likely p_T for each track, knowing exactly the parameters of the event generator. The best possible p_T resolution is ~7%.

True Maximum Likelihood Method



Modified Maximum Liklihood Method for PISA Data



Taking Care of the Multiple Scattering – Kalman Filter

The Kalman filter expresses a dynamic system in a state-space representation, and sequentially updates a linear projection for the system

State Equation

$$\boldsymbol{\xi}_{t+1} = \boldsymbol{F}\boldsymbol{\xi}_t + \boldsymbol{v}_{t+1}$$

Measurement Equation

$$\boldsymbol{y}_t = \boldsymbol{a}(\boldsymbol{x}_t) + [\boldsymbol{H}(\boldsymbol{x}_t)]'\boldsymbol{\xi}_t$$

Update Equations

$$\xi_{t/t} = \xi_{t/t-1} + \boldsymbol{P}_{t/t-1}\boldsymbol{H}(\boldsymbol{x}_t)[[\boldsymbol{H}(\boldsymbol{x}_t)]'\boldsymbol{P}_{t/t-1}\boldsymbol{H}(\boldsymbol{x}_t) + \boldsymbol{R}(\boldsymbol{x}_t)]^{-1}$$
$$\times [\boldsymbol{y}_t - \boldsymbol{a}(\boldsymbol{x}_t) - [\boldsymbol{H}(\boldsymbol{x}_t)]'\xi_{t/t-1}]$$

$$\boldsymbol{P}_{t/t} = \boldsymbol{P}_{t/t-1} - \boldsymbol{P}_{t/t-1}\boldsymbol{H}(\boldsymbol{x}_t)[[\boldsymbol{H}(\boldsymbol{x}_t)]'\boldsymbol{P}_{t/t-1}\boldsymbol{H}(\boldsymbol{x}_t) + \boldsymbol{R}(\boldsymbol{x}_t)]^{-1}$$
$$\times [\boldsymbol{H}(\boldsymbol{x}_t)]'\boldsymbol{P}_{t/t-1}$$

$$\xi_{t+1/t} = \boldsymbol{F}(\boldsymbol{x}_t)\xi_{t/t}$$

$$\boldsymbol{P}_{t+1/t} = \boldsymbol{F}(\boldsymbol{x}_t)\boldsymbol{P}_{t/t}[\boldsymbol{F}(\boldsymbol{x}_t)]' + \boldsymbol{Q}(\boldsymbol{x}_t)t$$

Implementation of Kalman Filter for the SVX

For now only the 2-D information from the transverse plane is used.

Between layers, the tracks are ~circular. Parametrize a circle in terms of global r, ϕ :

$$\phi = \arccos\left(\frac{r^2 + r_c^2 - R^2}{2\, r\, r_c}\right) + \phi_c$$

$r \equiv$ *radius from center of barrel*, $\phi \equiv$ *azimuthal angle*

$r_c \equiv$ *distance to center of circle from center of barrel*

$R \equiv$ *radius of circle*

$\phi_c \equiv$ *azimuthal angle from center of barrel to center of circle*

$\phi \equiv$ *azimuthal angle from center of barrel*

Expand $\phi(r)$ around the radius of one of the layers

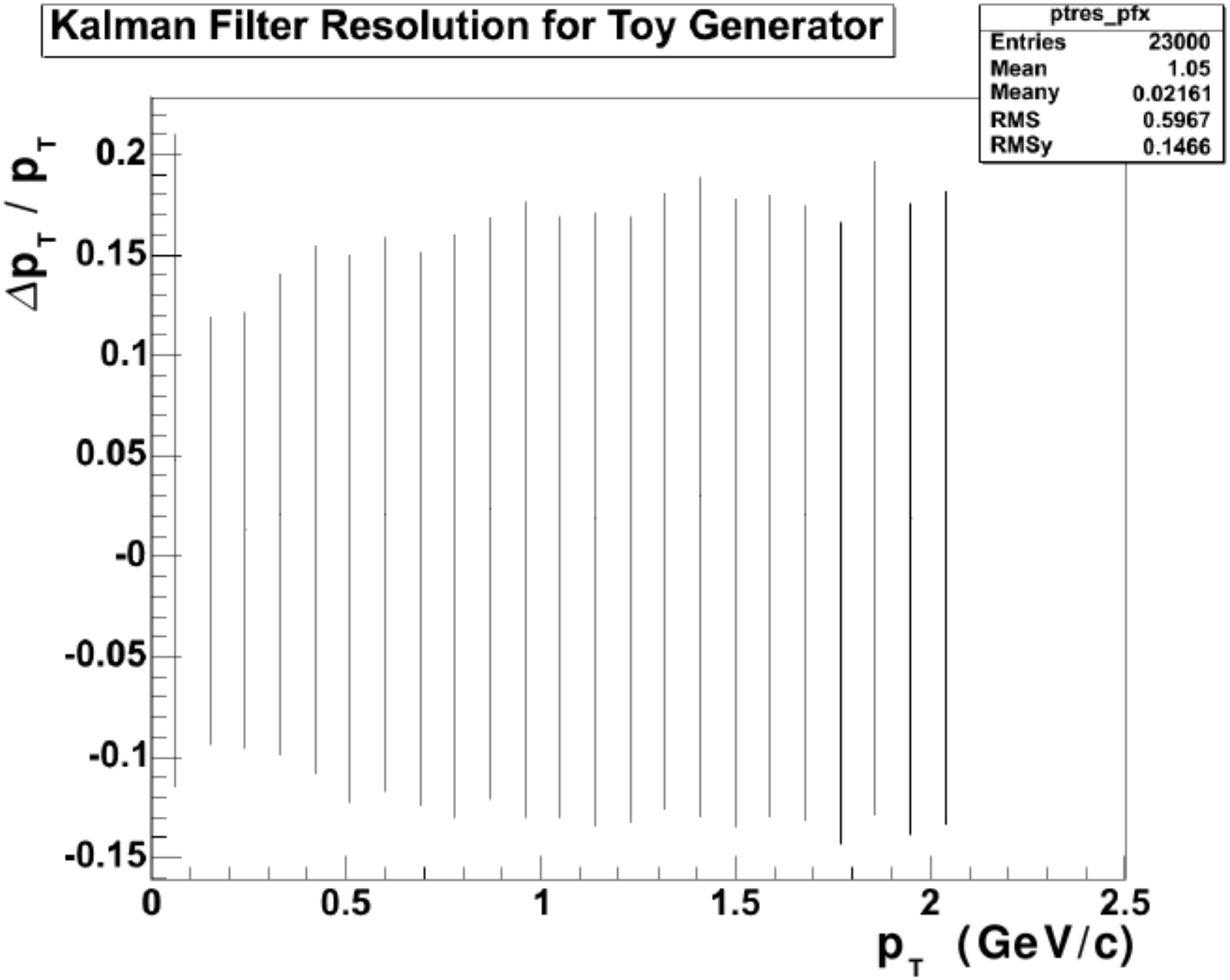
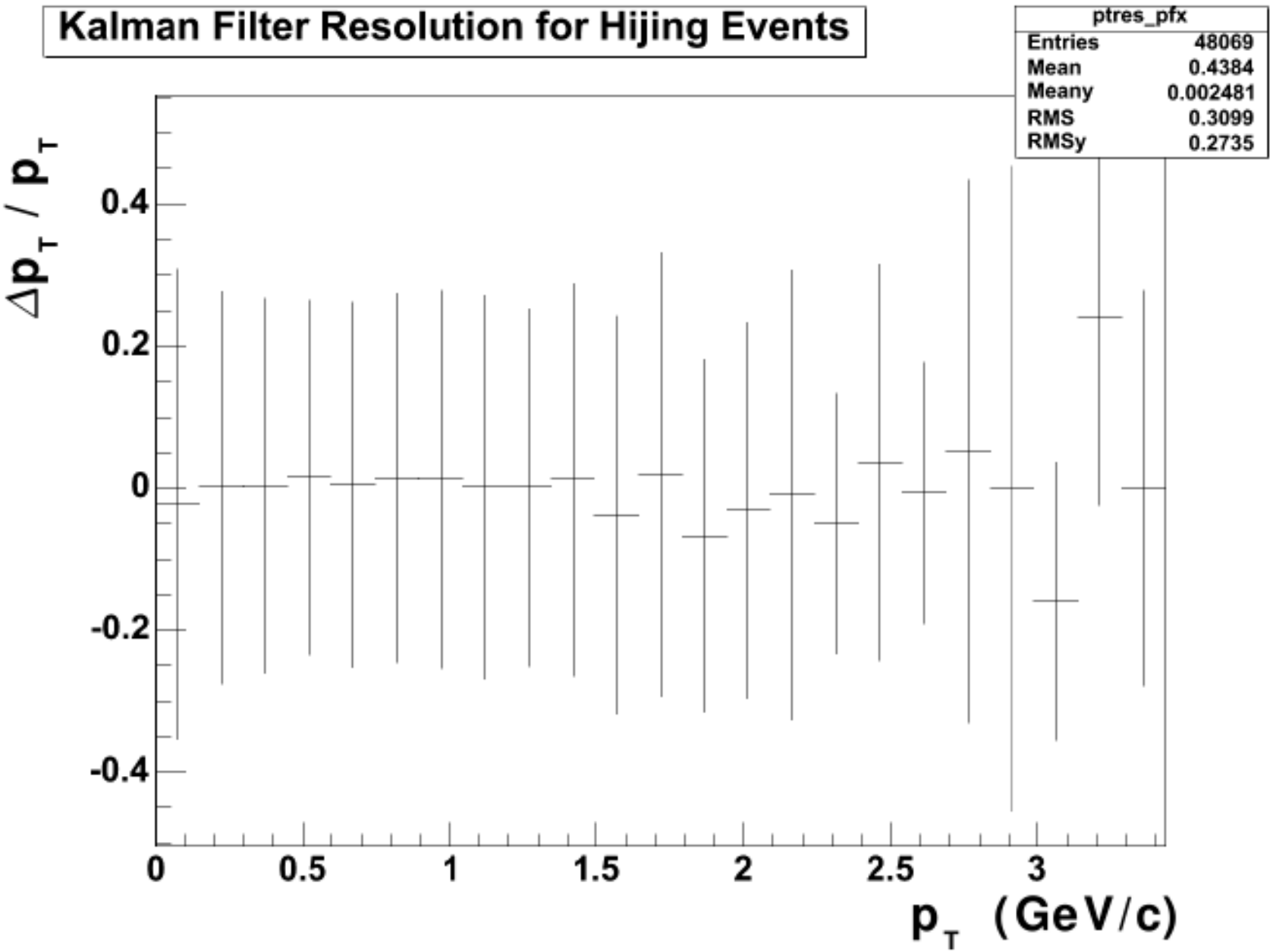
$$\phi(r) = \sum_{n=0}^{2} \beta_n (r - r_i)^n$$

$$\beta_1 = \frac{r_c^2 - r_i^2 - R^2}{\pm r_i \sqrt{2r_i^2 r_c^2 + 2r_i^2 R^2 - r_i^4 - r_c^4 + 2r_c^2 R^2 - R^4}}$$

$$\beta_2 = \frac{3r_i^4 r_c^2 - 3r_i^4 R^2 - r_i^6 - 3r_i^2 r_c^4 - 2r_i^2 r_c^2 R^2 + 5r_i^2 R^4 + r_c^6 - 3r_c^4 R^2 + 3r_c^2 R^4 - R^6}{\pm r_i^2 (2r_i^2 r_c^2 + 2r_i^2 R^2 - r_i^4 - r_c^4 + 2r_c^2 R^2 - R^4)^{3/2}}$$

$$R = \frac{(\beta_1^2 r_i^2 + 1)^{3/2}}{r_i \beta_2 + 2\beta_1 + r_i^2 \beta_1^3}$$

Kalman Filter Method for Toy Generator



Kalman Filter Method for PISA Data