New Developments in VertexReco

Rudi Frühwirth, Kirill Prokofiev, Thomas Speer, Pascal Vanlaer, Wolfgang Waltenberger .

ACAT'02 , Moscow, June 24th – 28th, 2002

- Recap: Linear fitting
- Robustifications
- Results
- Outlook

Recap: Linear Fitters

- Kalman Fit
  - $\exists$ (ORCA Implementation)
- Linear Fit
  - Linearizes tracks at the `impactPoints`.
  - Iterative.
  - $\exists$ (ORCA Implementation)
- Simplified Kalman Fit
  - Kalman Fit, with 3 “frozen” track parameters.
  - “Full” (non-linear) track model.
  - Problems in forward region.
  - Iterative. Fast.
  - $\nexists$ (ORCA Implementation)

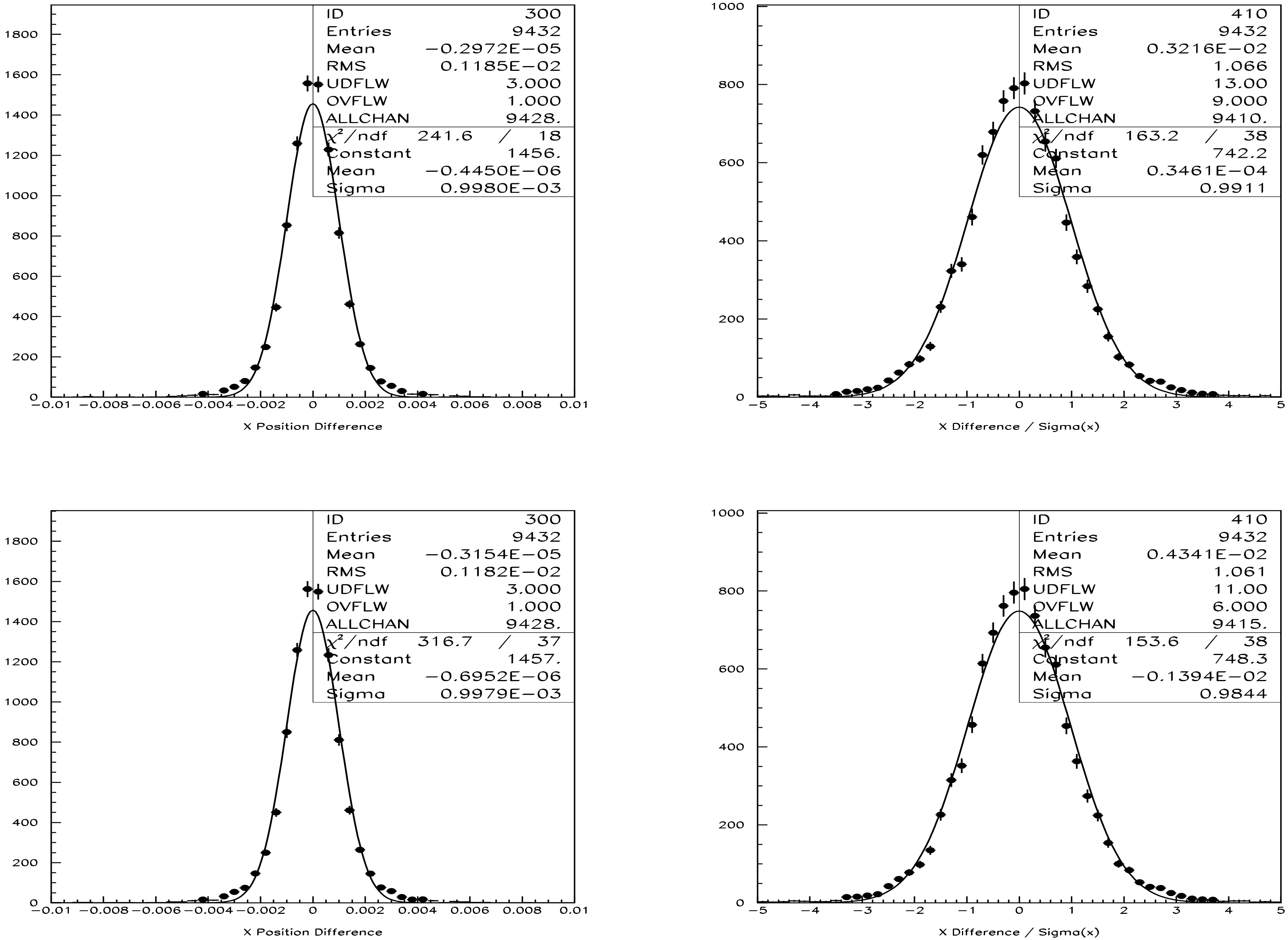
Differencies in CPU time / precision are currently under study.

Recap: Linear Fitters (contd)

Kirill Prokofiev, Thomas Speer

Kalman

Linear



Why robust estimators?

- Robust estimators are insensitive to outlying observations
- Outlier recognition improved
- Lower bias in the overall fit
- Better separation between primary vertices and secondary vertices
- Good starting point for more complex algorithms (Multivertex fit)

Examples of robust estimators

- Median (very old)
- L_1=Least Sum of Absolute Values (Edgeworth)
- M-estimators (Huber)
  ORCA implementation planned
- Adaptive estimators (Peterson et al., Frühwirth and Strandlie)
  Implemented with lin. annealing schedule
- LTS=Least Trimmed Sum of Squares (Rousseeuw)
  Implemented

Examples of robust estimators

- LMS=Least Median of Sqares (Rousseeuw)
  Coordinate-wise implementation
- MVE=Minimum Volume Ellipsoid (Rousseeuw)
- MCD=Minimum Covariance Determinant (Rousseeuw)
  Implemented

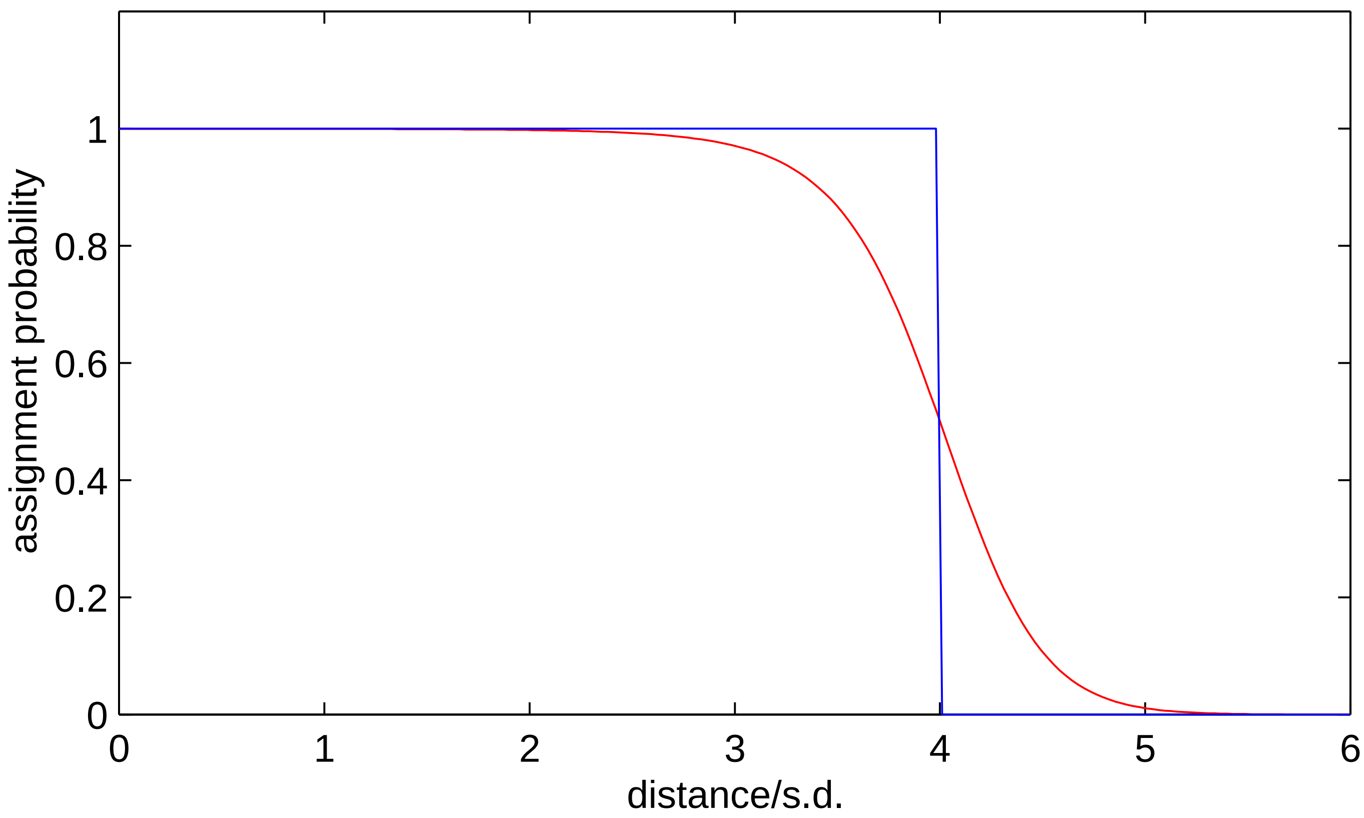
Adaptive estimators

- These are very similar to *M-estimators* and differ only in the computation of the weights. If μ is the current estimate (e.g. of location) and m_i a (multivariate) observation with cov. matrix Σ, the weight w_i is given by

$$w_i = \frac{\varphi(m_i; \mu, \Sigma)}{c_i + \varphi(m_i; \mu, \Sigma)}$$

The constant c_i defines the cut beyond which the weight quickly drops to 0.

Adaptive estimators (contd)



Weight of a univariate observation with a cut at 4σ

Adaptive estimator (contd)

ORCA implementation:

$$w_i = \frac{1}{1 + e^{\frac{\chi^2 - \chi^2_{\text{cutoff}}}{2T}}}$$

T = Temperature, default: 1

Deterministic annealing:

$$T = T \cdot r$$

for every iteration step.

$\chi_{\text{cutoff}} \equiv$ “cut off” parameter, defaults to 3 $\sigma \approx$ 1 % cut

LTS estimator

- Instead of minimizing the total sum of the squared residuals, the sum of the h smallest squared residulas is minimized ($[n/2]+1 < h < n$). The breakdown point is roughly equal to the trimming proportion.
- The LTS estimate can be computed exactly only by an exhaustive search. However, there is a very good approximate iterative algorithm (FAST-LTS).

LTS estimator (contd)

The FAST-LTS algorithm:

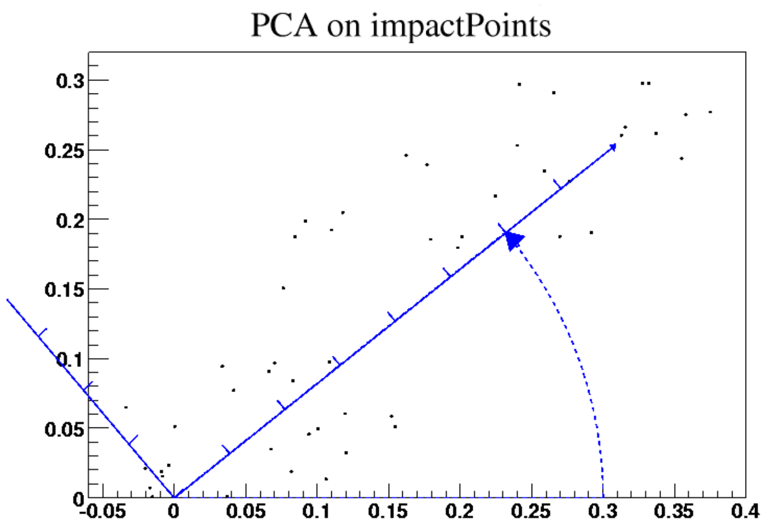
- Compute squared residuals with current estimate
- Recompute estimate using observations corresponding to the h smallest squared residuals
- Iterate until convergence

LMS estimator

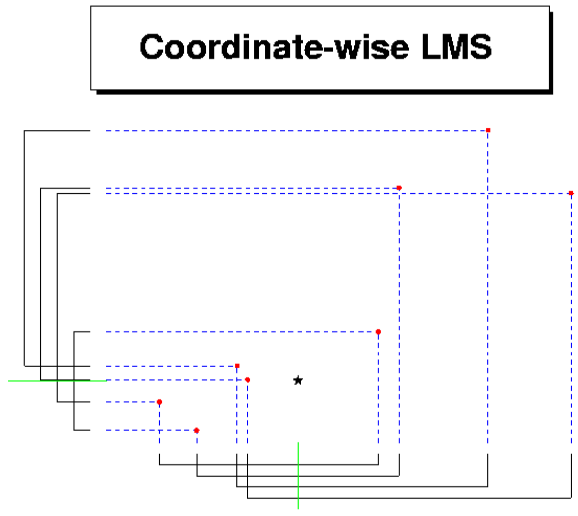
- The LMS estimator minimizes the median of the squared residuals. Its breakdown point is $([n/2]-p+2)/n$.
- In the univariate case, the LMS estimate of the location is the midpoint of the shortest interval covering one half of the sample.
- In the case of a fitting a line to data points in the plane, the LMS line is the narrowest band (measured in the y-direction) which covers half of the sample. This line passes through two of the data points (exact fit property).

LMS estimator (contd)

ORCA implementation: (non-robust)
Principal Component Analysis + coordinate-wise LMS



+



MCD estimator

- Multivariate location estimator, mean of the $h = [n/2] + 1$ points for which the determinant of the covariance matrix is minimal. Its breakdown point is $([n/2] - p + 1)/n$.
- Gives a covariance estimate at the same time.
- Computationally similar to the LTS.
- Can be generalized to other values of h.

LTS/MCD in ORCA

LTS and MCD have been generalized in ORCA to work with any kind of distance measure, the most general weight being

$$\frac{1}{w_{\text{tot}}} = \frac{c_{\text{track}}}{w_{\text{track}}} + \frac{c_{\text{vertex}}}{w_{\text{vertex}}}$$

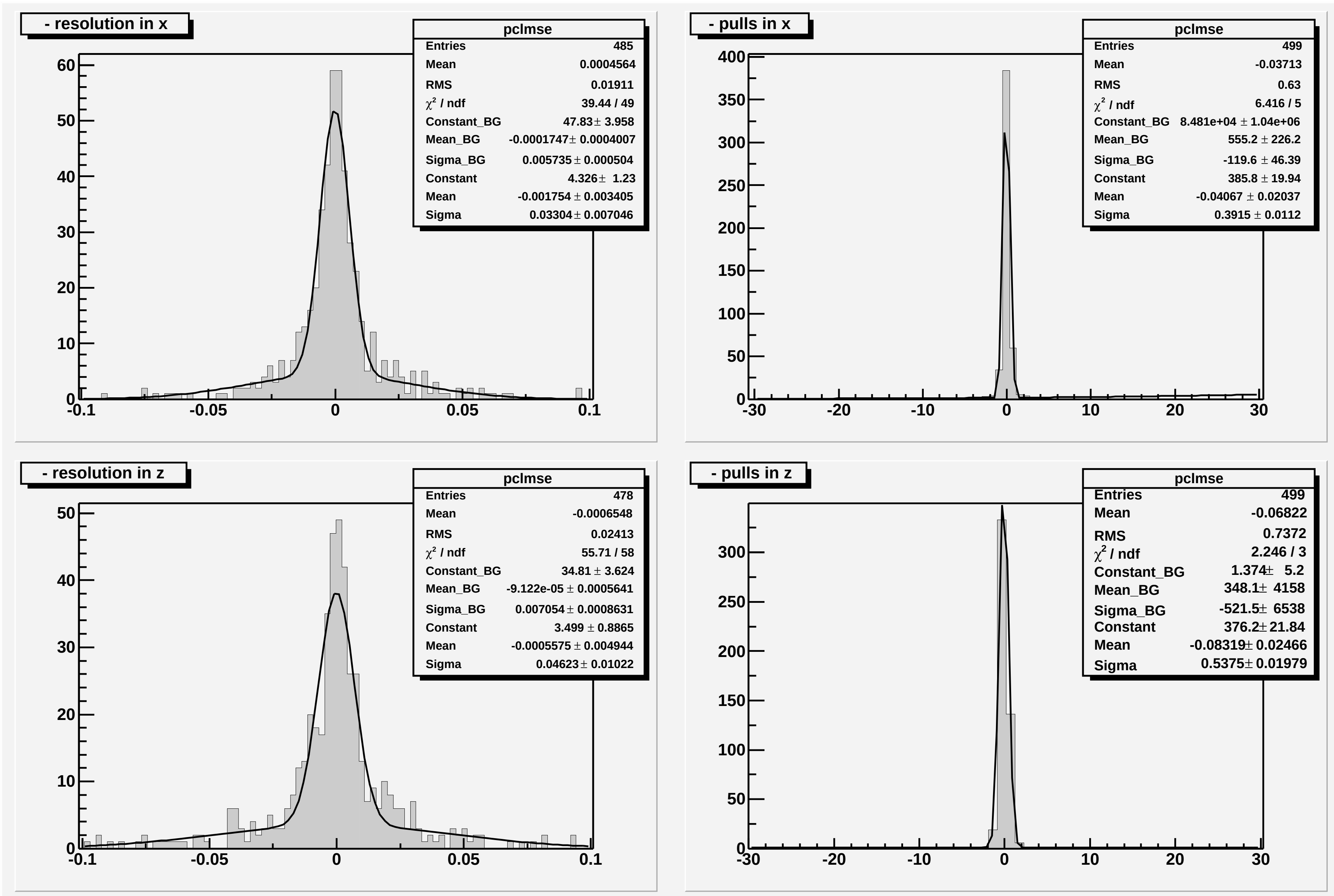
- LTS: $c_{\text{track}} = 1, c_{\text{vertex}} = 0$
- MCD: $c_{\text{track}} = 0, c_{\text{vertex}} = 1$

Results

Four simulation experiments:

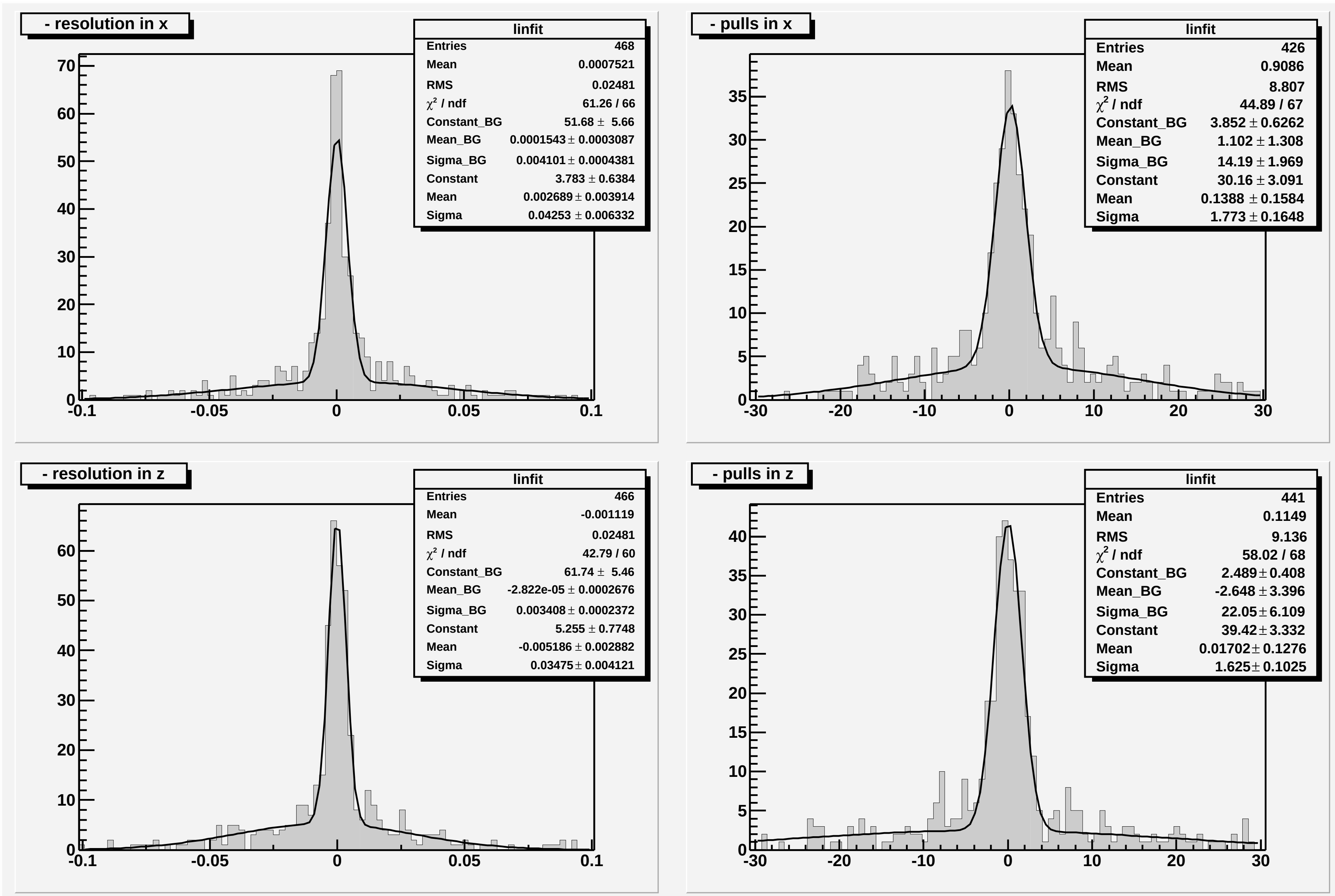
- VertexGun events with “well-behaved” outliers:
  25 tracks, 5 outliers, P=20 $\pm$ 2 GeV. $\sigma_x = 2$ mm, $\sigma_{x(Outlier)} = 5$ mm, $\sigma_{dxdir} = 5$ mrad, $\sigma_{dxdir(Outlier)} = 8$ mrad.
- VertexGun events with “evil” outliers:
  25 tracks, 5 outliers, P=20 $\pm$ 2 GeV. $\sigma_x = 1$ mm, $\sigma_{x(Outlier)} = 10$ mm, $\sigma_{dxdir} = 5$ mrad, $\sigma_{dxdir(Outlier)} = 25$ mrad.
- Pythia events: $c\bar{c}$ - 100 GeV, $\eta < 1.4$ - NoPU
- Pythia events: $q\bar{q}$ - 100 GeV, $1.4 < \eta < 2.4$ - NoPU

PCLMSFit



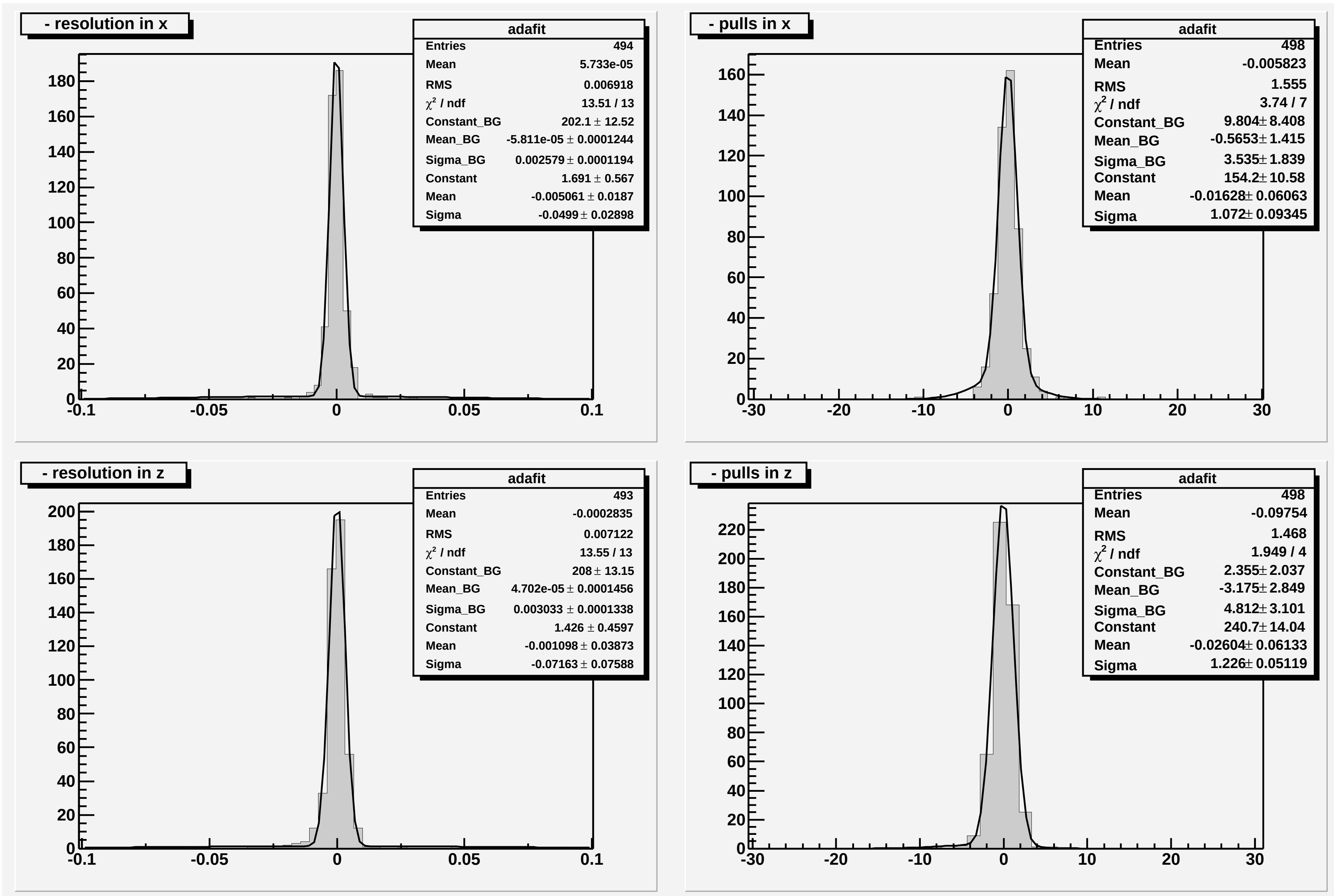
- resolution in x
pclmse
Entries 485
Mean 0.0004564
RMS 0.01911
χ² / ndf 39.44 / 49
Constant_BG 47.83 ± 3.958
Mean_BG -0.0001747 ± 0.0004007
Sigma_BG 0.005735 ± 0.000504
Constant 4.326 ± 1.23
Mean -0.001754 ± 0.003405
Sigma 0.03304 ± 0.007046
- pulls in x
pclmse
Entries 499
Mean -0.03713
RMS 0.63
χ² / ndf 6.416 / 5
Constant_BG 8.481e+04 ± 1.04e+06
Mean_BG 555.2 ± 226.2
Sigma_BG -119.6 ± 46.39
Constant 385.8 ± 19.94
Mean -0.04067 ± 0.02037
Sigma 0.3915 ± 0.0112
- resolution in z
pclmse
Entries 478
Mean -0.0006548
RMS 0.02413
χ² / ndf 55.71 / 58
Constant_BG 34.81 ± 3.624
Mean_BG -9.122e-05 ± 0.0005641
Sigma_BG 0.007054 ± 0.0008631
Constant 3.499 ± 0.8865
Mean -0.0005575 ± 0.004944
Sigma 0.04623 ± 0.01022
- pulls in z
pclmse
Entries 499
Mean -0.06822
RMS 0.7372
χ² / ndf 2.246 / 3
Constant_BG 1.374 ± 5.2
Mean_BG 348.1 ± 4158
Sigma_BG -521.5 ± 6538
Constant 376.2 ± 21.84
Mean -0.08319 ± 0.02466
Sigma 0.5375 ± 0.01979

LinearFit

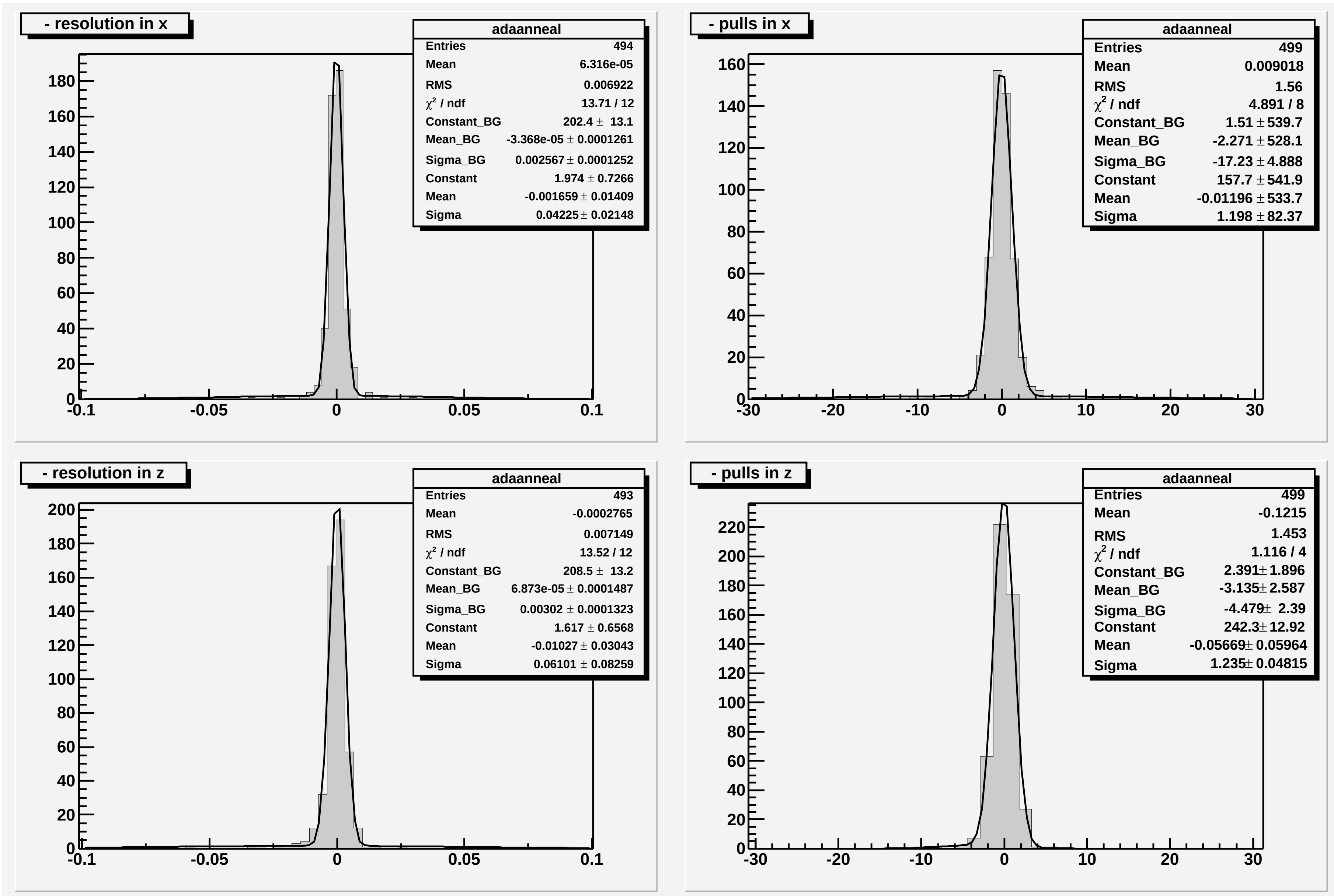


- resolution in x
linfit
Entries 468
Mean 0.0007521
RMS 0.02481
χ² / ndf 61.26 / 66
Constant_BG 51.68 ± 5.66
Mean_BG 0.0001543 ± 0.0003087
Sigma_BG 0.004101 ± 0.0004381
Constant 3.783 ± 0.6384
Mean 0.002689 ± 0.003914
Sigma 0.04253 ± 0.006332
- pulls in x
linfit
Entries 426
Mean 0.9086
RMS 8.807
χ² / ndf 44.89 / 67
Constant_BG 3.852 ± 0.6262
Mean_BG 1.102 ± 1.308
Sigma_BG 14.19 ± 1.969
Constant 30.16 ± 3.091
Mean 0.1388 ± 0.1584
Sigma 1.773 ± 0.1648
- resolution in z
linfit
Entries 466
Mean -0.001119
RMS 0.02481
χ² / ndf 42.79 / 60
Constant_BG 61.74 ± 5.46
Mean_BG -2.822e-05 ± 0.0002676
Sigma_BG 0.003408 ± 0.0002372
Constant 5.255 ± 0.7748
Mean -0.005186 ± 0.002882
Sigma 0.03475 ± 0.004121
- pulls in z
linfit
Entries 441
Mean 0.1149
RMS 9.136
χ² / ndf 58.02 / 68
Constant_BG 2.489 ± 0.408
Mean_BG -2.648 ± 3.396
Sigma_BG 22.05 ± 6.109
Constant 39.42 ± 3.332
Mean 0.01702 ± 0.1276
Sigma 1.625 ± 0.1025

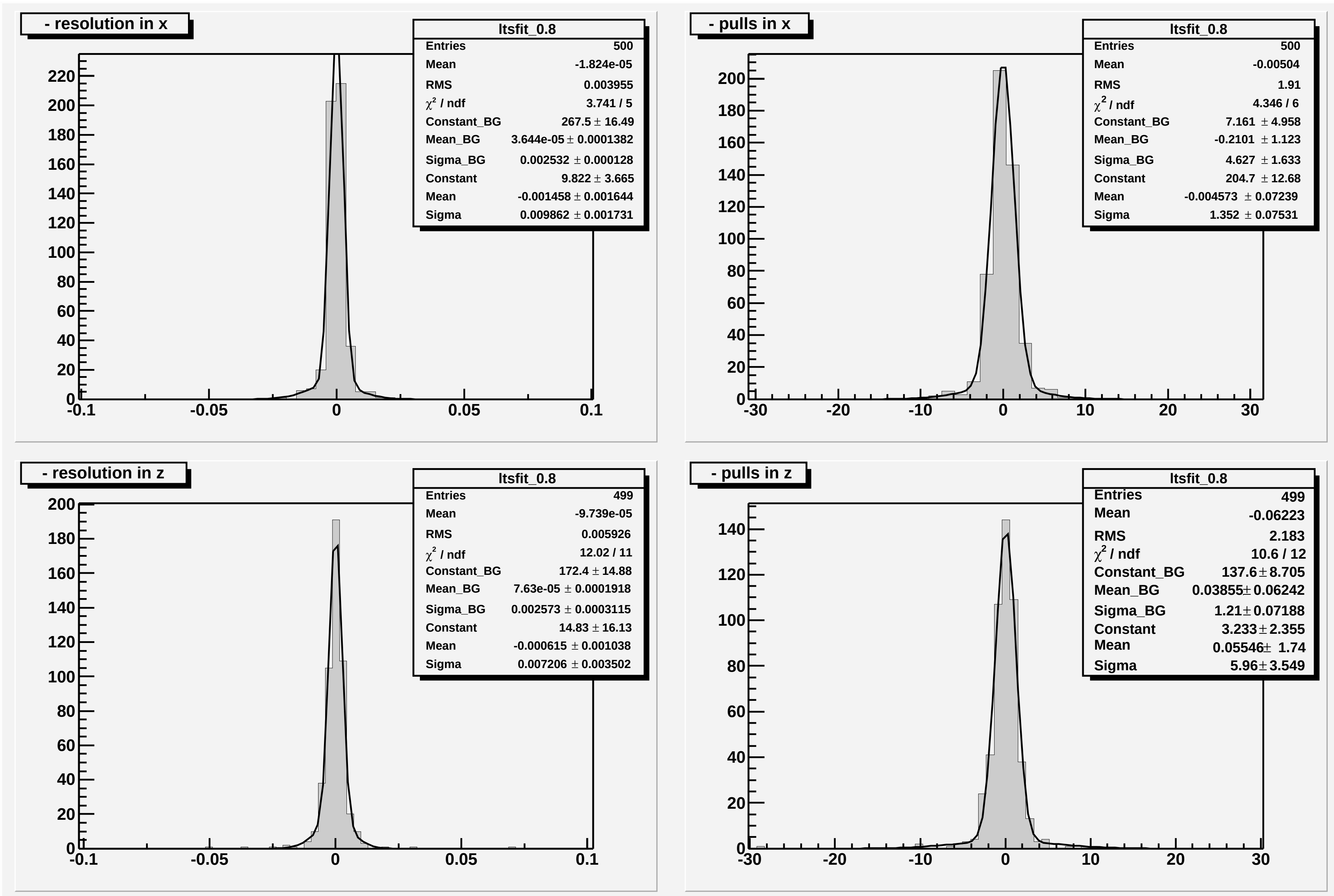
AdaptiveFit



AdaptiveFit + annealing



LTSFit (80%)


- resolution in x
ltsfit_0.8
Entries 500
Mean -1.824e-05
RMS 0.003955
χ² / ndf 3.741 / 5
Constant_BG 267.5 ± 16.49
Mean_BG 3.644e-05 ± 0.0001382
Sigma_BG 0.002532 ± 0.000128
Constant 9.822 ± 3.665
Mean -0.001458 ± 0.001644
Sigma 0.009862 ± 0.001731
- pulls in x
ltsfit_0.8
Entries 500
Mean -0.00504
RMS 1.91
χ² / ndf 4.346 / 6
Constant_BG 7.161 ± 4.958
Mean_BG -0.2101 ± 1.123
Sigma_BG 4.627 ± 1.633
Constant 204.7 ± 12.68
Mean -0.004573 ± 0.07239
Sigma 1.352 ± 0.07531
- resolution in z
ltsfit_0.8
Entries 499
Mean -9.739e-05
RMS 0.005926
χ² / ndf 12.02 / 11
Constant_BG 172.4 ± 14.88
Mean_BG 7.63e-05 ± 0.0001918
Sigma_BG 0.002573 ± 0.0003115
Constant 14.83 ± 16.13
Mean -0.000615 ± 0.001038
Sigma 0.007206 ± 0.003502
- pulls in z
ltsfit_0.8
Entries 499
Mean -0.06223
RMS 2.183
χ² / ndf 10.6 / 12
Constant_BG 137.6 ± 8.705
Mean_BG 0.03855 ± 0.06242
Sigma_BG 1.21 ± 0.07188
Constant 3.233 ± 2.355
Mean 0.05546 ± 1.74
Sigma 5.96 ± 3.549

Final comparison - VertexGun

“nice” VertexGun

Name	t(msecs)	Res(x)	Pulls(x)	Res(z)	Pulls(z)
LinearFitter	15	67μ	1.4	67μ	1.4
LTSFitter (20% trim)	57	64μ	1.2	53μ	1.2
MCDFitter (20% trim)	57	65μ	1.2	52μ	1.2
PCLMSFitter (err.mat)	12	260μ	0.3	129μ	0.26
AdaptiveFitter	29	59μ	1	50μ	0.97
Adaptive + anneal	28	59μ	1	49μ	0.97

“evil” VertexGun

Name	t(msecs)	Res(x)	Pulls(x)	Res(z)	Pulls(z)
LinearFitter	15	101μ	4.2	84μ	4.2
LTSFitter (20% trim)	58	29μ	1	24μ	1
MCDFitter (20% trim)	58	29μ	1.1	24μ	1
PCLMSFitter (err.mat)	22	134μ	0.26	76μ	0.076
AdaptiveFitter	31	28μ	0.91	24μ	0.94
Adaptive + anneal	30	28μ	0.92	24μ	0.95

Final comparison - Pythia events

$c\bar{c}$ - 100 GeV

Name	t(msecs)	Res(x)	Pulls(x)	Res(z)	Pulls(z)
LinearFitter	12	58μ	3.5	42μ	1.9
LTSFitter (20% trim)	45	27μ	1.5	30μ	1.3
MCDFitter (20% trim)	45	36μ	1.8	32μ	1.6
PCLMSFitter (err.mat)	8	78μ	0.4	113μ	0.54
AdaptiveFitter	30	26μ	1.2	26μ	1.3
Adaptive + anneal	18	26μ	1.2	31μ	1.3

$q\bar{q}$ - 100 GeV

Name	t(msecs)	Res(x)	Pulls(x)	Res(z)	Pulls(z)
LinearFitter	18	39μ	2.1	39μ	1.9
LTSFitter (20% trim)	67	25μ	1.1	29μ	1.1
MCDFitter (20% trim)	67	21μ	1.2	27μ	1.2
PCLMSFitter (err.mat)	12	74μ	0.66	90μ	0.64
AdaptiveFitter	44	21μ	1.1	28μ	1.1
Adaptive + anneal	24	22μ	1	28μ	1.1

Outlook

- Test robust fitters on multi-vertex fits.
- Test robust fitters on Finding-thru-Fitting algorithms.