Kinematic Fitting With Missing Particles

BABAR Analysis Tools Workshop, October 1. 2005

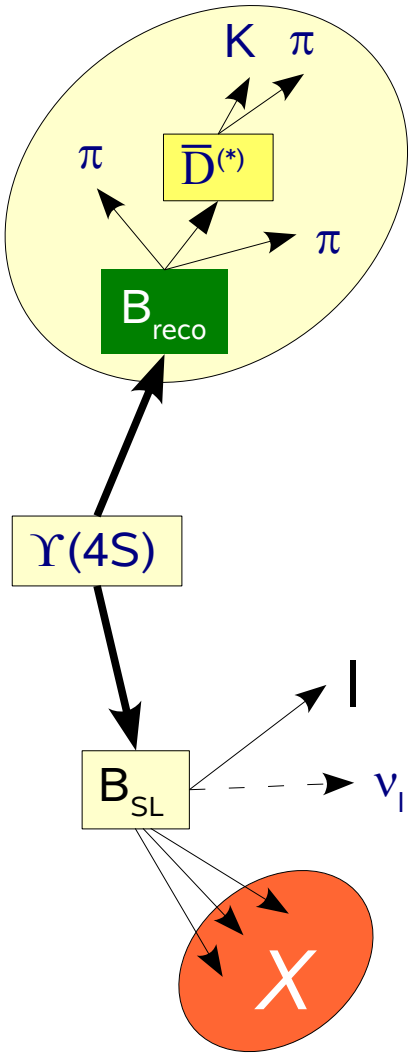
TECHNISCHE UNIVERSITÄT DRESDEN

Verena Klose, Jan Erik Sundermann

- Motivation and use case
- “KinFitter” package, what's available?
- Implementation example
- Performance
- Summary

Motivation And Use Case

- Measurement of the inclusive hadronic mass spectrum m_X in semileptonic decays $B \rightarrow X l \nu$
- m_X-resolution dominated by missing particles
  $\Rightarrow$ can't use uncertainties from reconstruction
- *What we need:*
  kinematic fitter that can handle user defined covariance matrices and four-vector parametrizations



“KinFitter” Package

- “General” tool for kinematic fitting with user defined error matrices (e.g. missing particles)
- Iterative minimization of χ^2 with non-linear constraints
- Flexible in changing 4-vector parametrization, e.g.
  - cartesian, spherical, abc
- ... and constraints, e.g.
  - energy and momentum conservation,
  - mass constraint (also with gaussian shape)

“KinFitter” Package

- C++ implementation based on ROOT classes
- Currently also used for studies in the B_{reco} V_{ub} analysis (Wolfgang Menges):
  - They developed a Beta interface similar to those of the other BABAR fitter packages
  - Currently located in “VubRecoilUser”
  - Could be cleaned up and moved to the “KinFitter” package if interest exists

Fit Strategy ($B \to X_c l \nu$)

- Perform a 2C-Fit:
  - measured B_{reco}, e, X ($\Rightarrow$ 11 parameters) and
  - unmeasured ν ($\Rightarrow$ 3 parameters)
  - use 5 constraints:
    - a) energy and mom. conservation: $P_X + P_l + P_\nu + P_{Breco} = P_{\Upsilon(4S)}$
    - b) constrain the two B meson masses: $\sqrt{P^2_{Breco}} = \sqrt{(P_X + P_l + P_\nu)^2}$
- B_{reco} and e covariance matrix taken from reconstruction
- X-System covariance matrix derived from MC in bins Multiplicity$_X$ and θ_X. Assumed to be uncorrelated.

X-System Parametrization

- Comparison of two X-system parametrizations:

1. **cartesian:**
   - X-system parametrization: E, px, py, pz
2. **abc:**
   - X-system parametrization: a, b, c, d

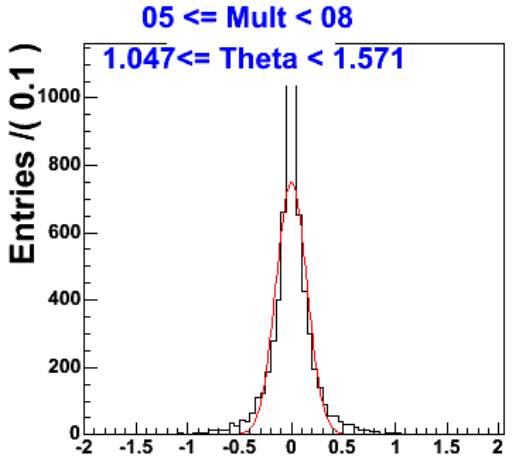
$$\boldsymbol{p}_{fit} = a\,\boldsymbol{p}_{meas} + b\,\boldsymbol{u}_b + c\,\boldsymbol{u}_c$$

$$E_{fit} = \sqrt{p_{fit}^2 + d^2\, m_{meas}^2}$$

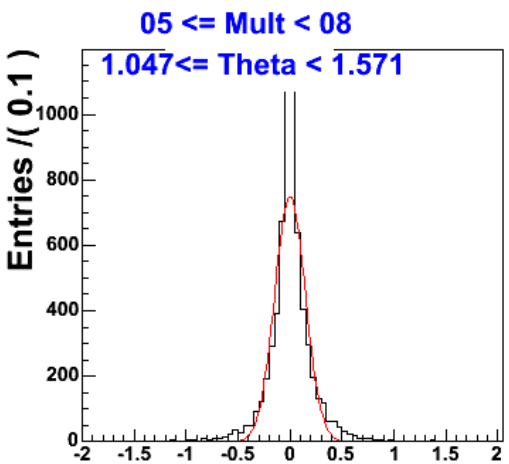
   - measured 4-vector corrected with mean of resolution distributions

X-System Resolution

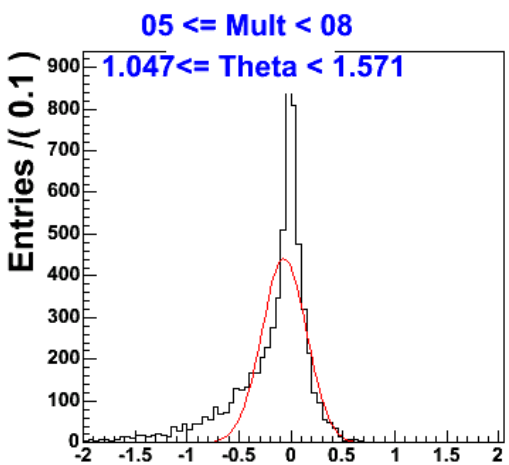
cartesian:



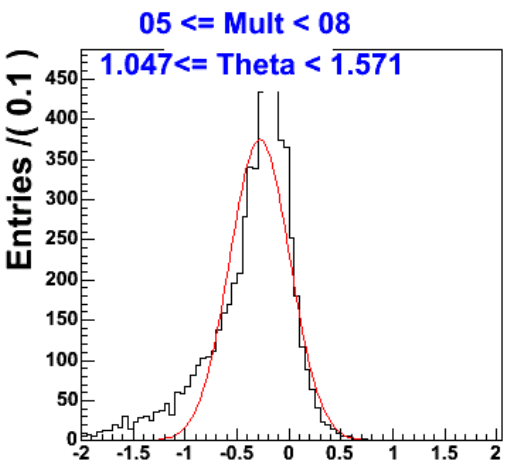
px(reco) - px(true)



py(reco) - py(true)

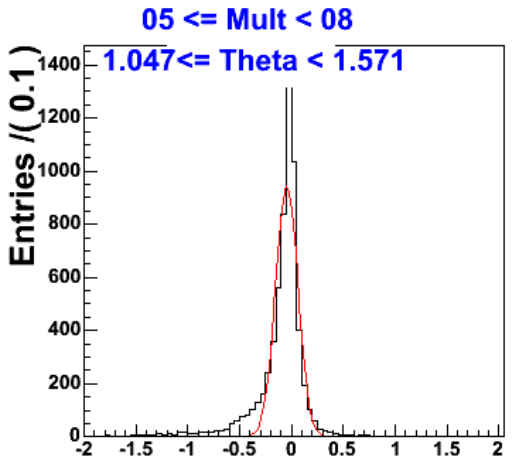


pz(reco) - pz(true)

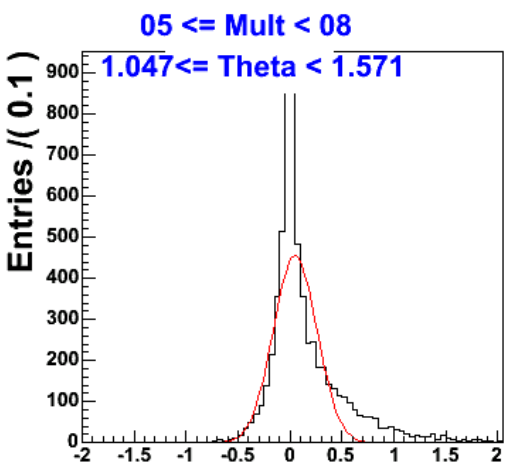


E(reco) - E(true)

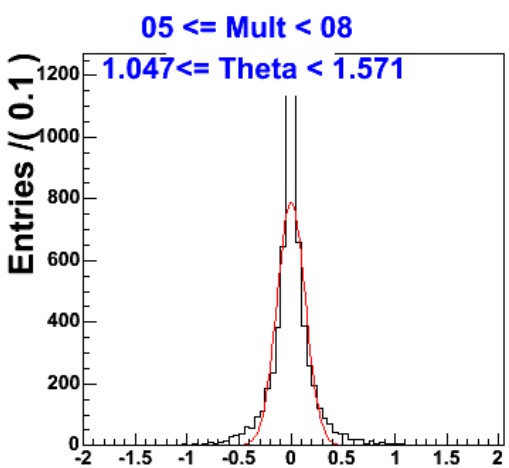
abc:



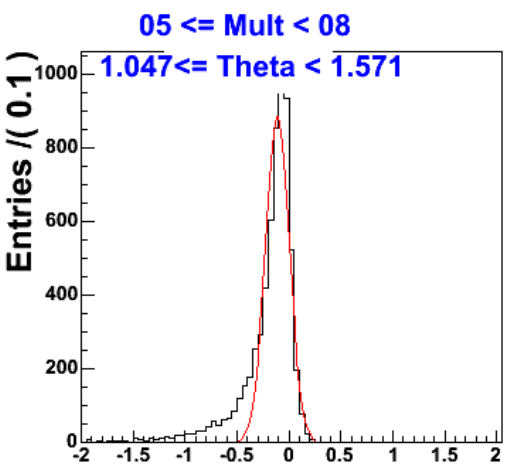
a(reco) - a(true)



b(reco) - b(true)



c(reco) - c(true)



d(reco) - d(true)

Implementation In ROOT (I)

Definition of fit particles in cartesian coordinates:

```
TFitParticleECart     breco( "Breco", "Breco", &vecP4BReco, &covBReco);
TFitParticleMCCart  l( "Lepton", "Lepton", &vecP3L, 511e-6, &covL);
TFitParticleECart     x( "X", "X", &vecP4X, &covX);
TFitParticleMCCart  nu( "Nu", "Nu", &vecP3Nu, 0., NULL);
```

Definition of constraints:

```
TFitConstraintEp pX( "pX", "pX conservation", 0,
                     TFitConstraintEp::pX, vecP4Y4S.X() );
pX.addParticles( &breco, &l, &x, &nu );
(...)
TFitConstraintM  MC( "Mass", "Mass constraint", NULL, NULL, 0.);
MC.addParticle1( &breco );
MC.addParticles2( &l, &x, &nu );
```

Implementation In ROOT (II)

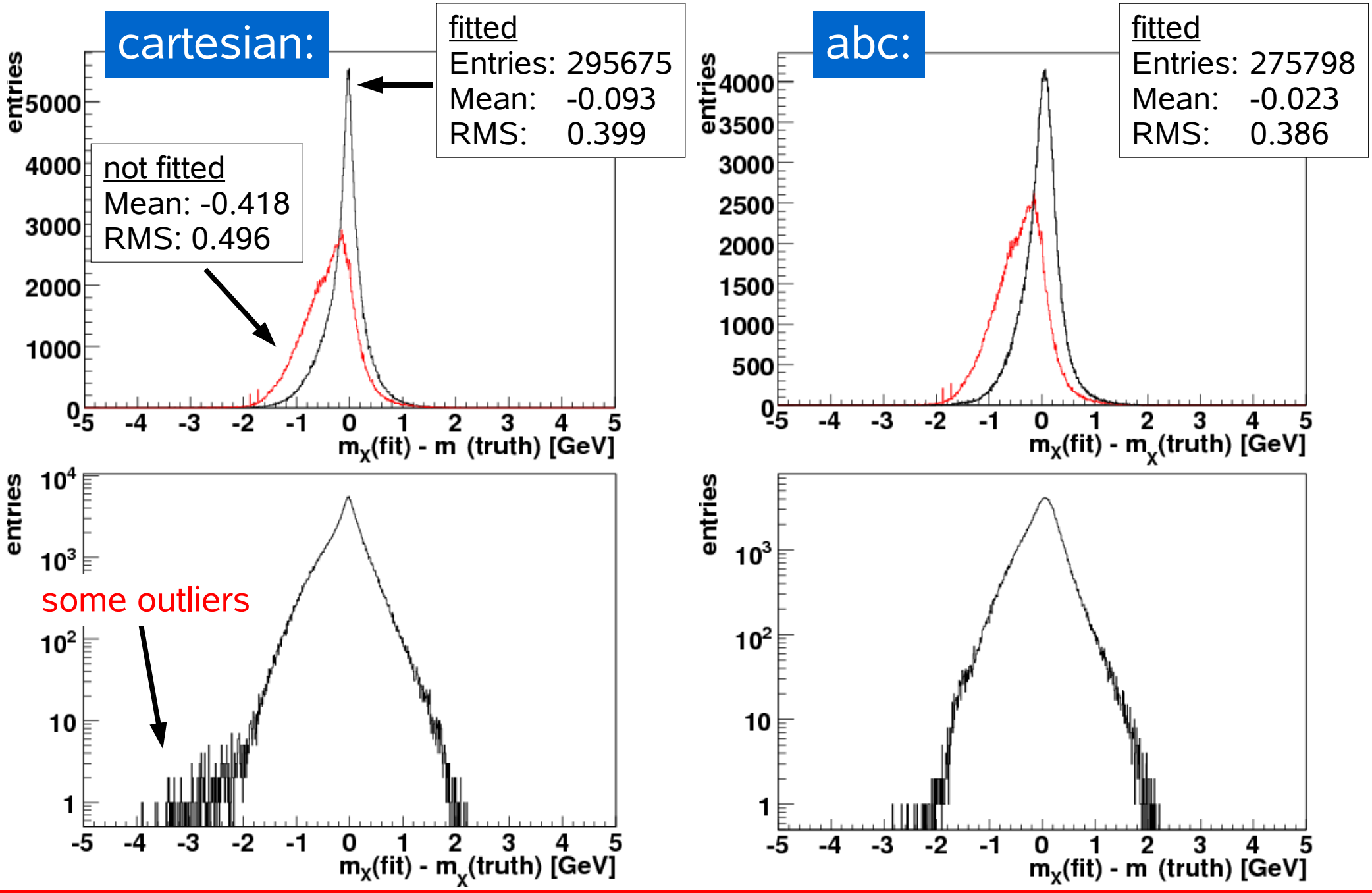
Define fitter setup (measured, unmeasured particles, and constraints):

```
TKinFitter fitter;
fitter.addMeasParticles( &breco, &l, &x );
fitter.addUnmeasParticle( &nu );
fitter.addConstraints( &pX, &pY, &pZ, &E, &MC );
```

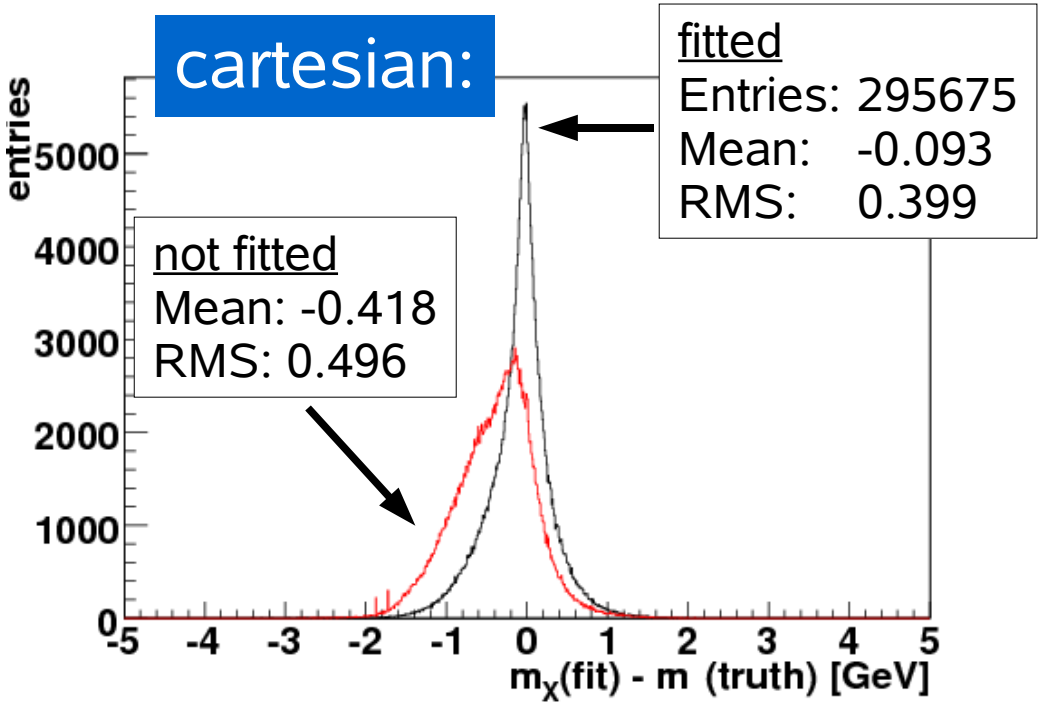
Perform the fit, define convergence criteria:

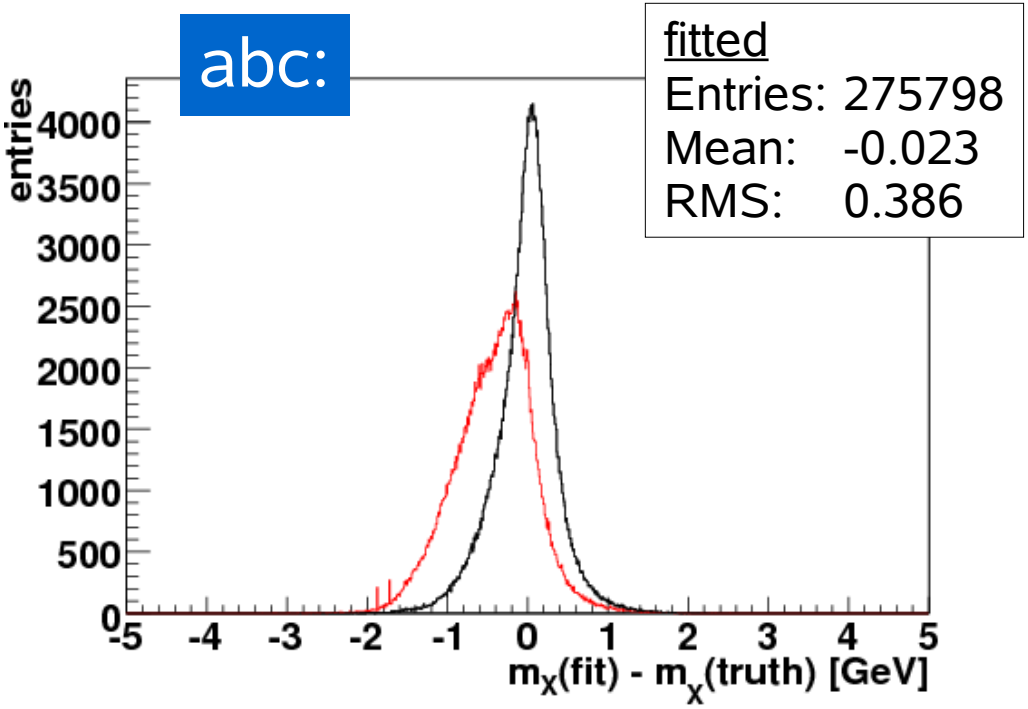
`fitter.setMaxNbIter( 50 );`	Maximum number of iterations
`fitter.setMaxDeltaS( 5e-5 );`	Maximum change of χ^2 between iterations
`fitter.setMaxF( 1e-4 );`	Maximum sum of constraints
`fitter.setVerbosity(1);`	Output level
`fitter.fit();`	Perform fit

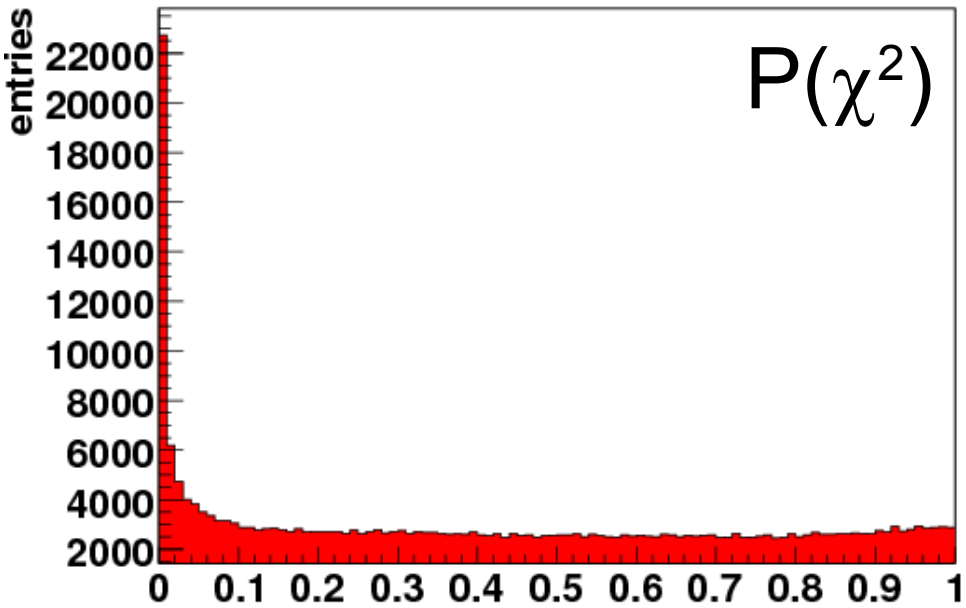
Fit Results: m_X-Resolution



Fit Results: m_X-Resolution



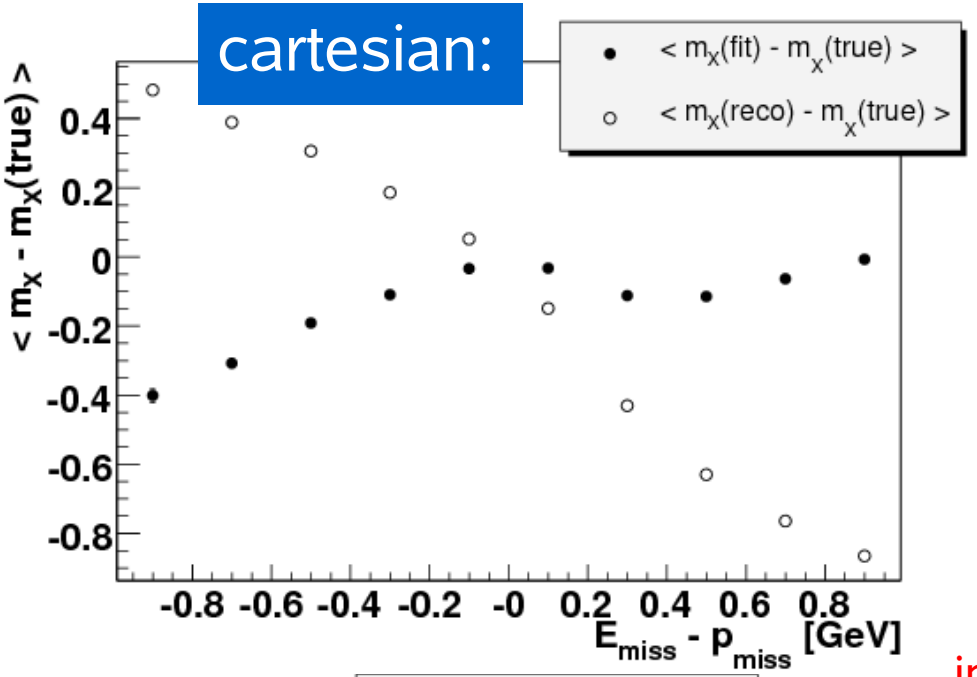


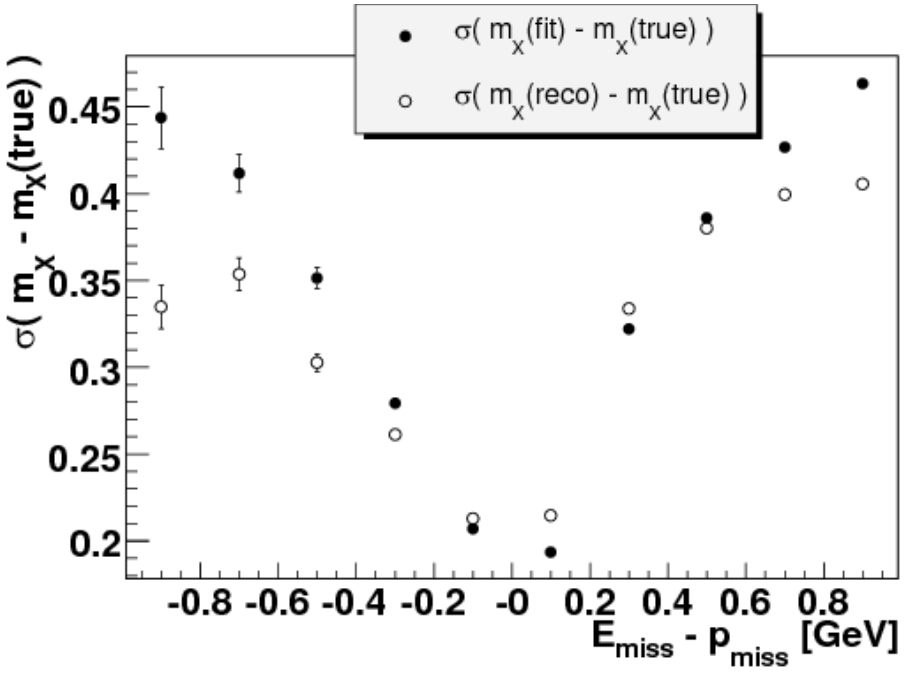


- ~10% fits with high χ^2
- $P(\chi^2)$ reasonably flat between 0.1-1

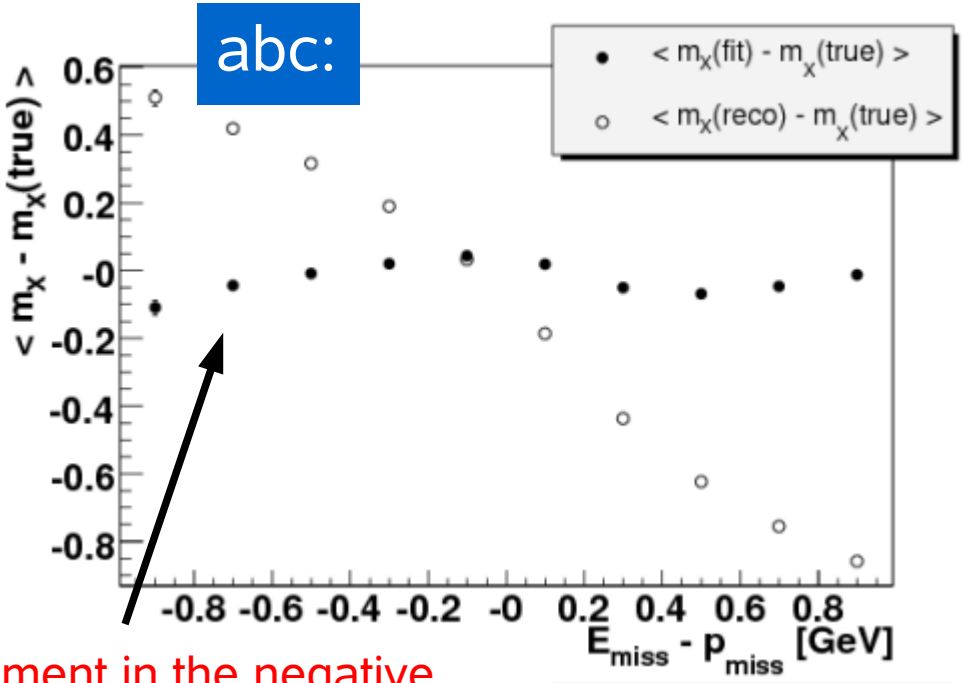
Bias And Resolution (E_{miss}-p_{miss} dependence)

cartesian:

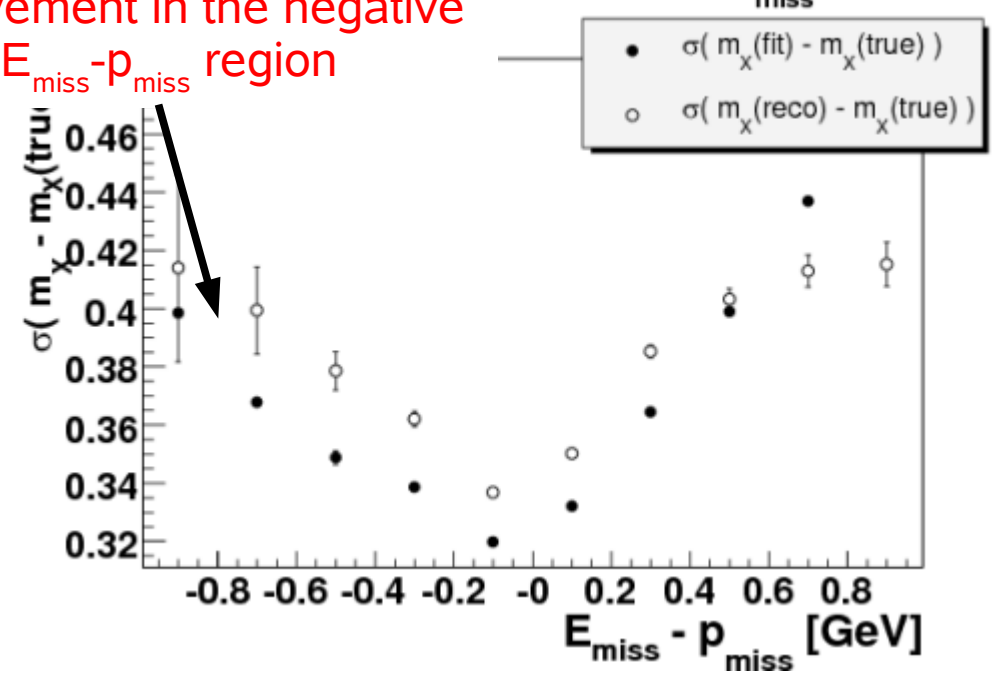




abc:

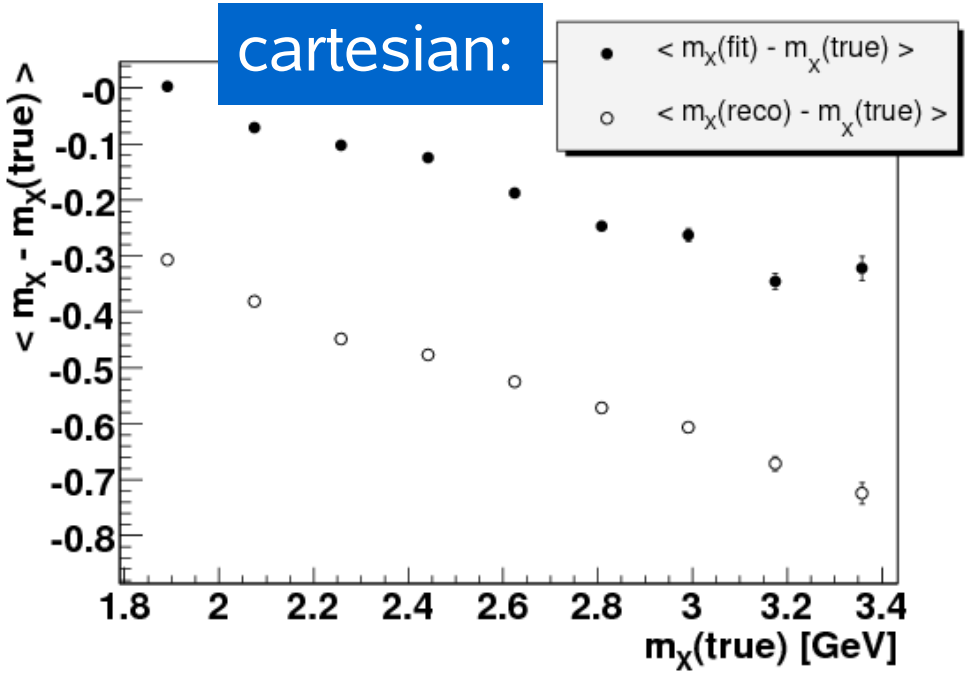


improvement in the negative E_{miss}-p_{miss} region

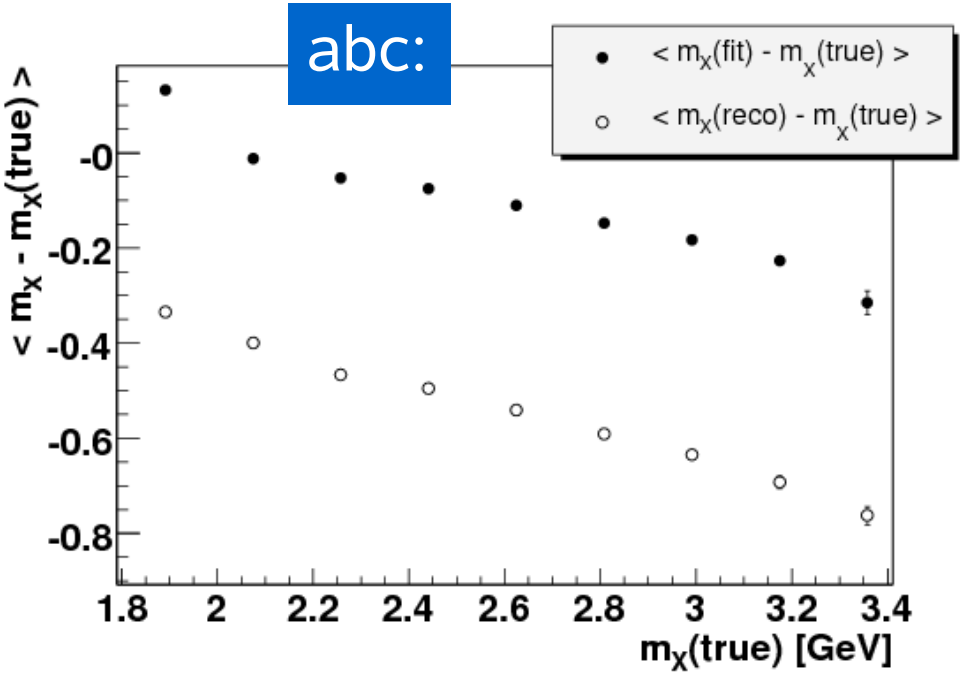


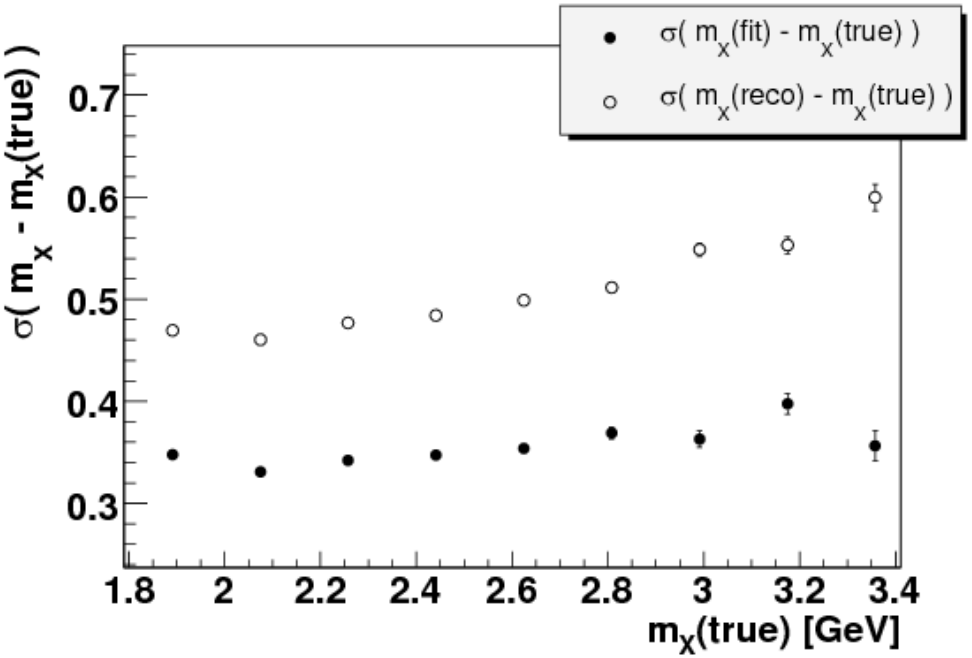
Bias And Resolution ($m_{X,true}$ dependence)

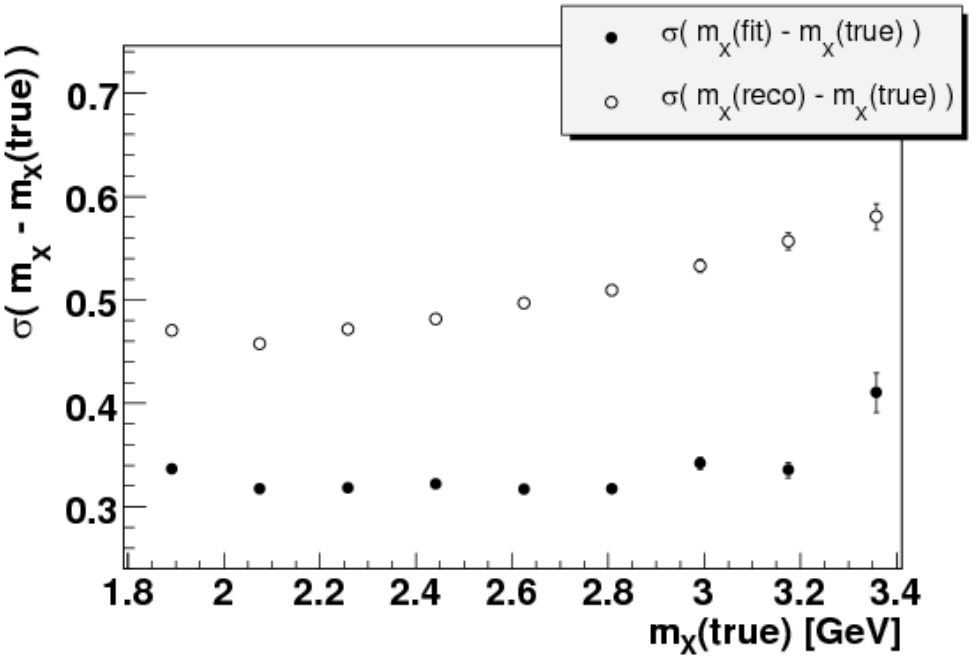
cartesian:



abc:







Summary

- Kinematic fitter package
  - iterative minimization of χ^2 with non-linear constraints
  - flexible in changing four-vector parametrization
- Software available via CVS package *"KinFitter"*
- Documentation see BAD #1061
- Online class documentation:
  http://www.slac.stanford.edu/~jsunderm/kinfitterdoc/USER_Index.html