GENFIT 2

Johannes Rauch

Physik Department E18
Technische Universität München
Germany

Belle II F2F Tracking Meeting, Dec 11th to 13th, 2013

Outline

New Features

Main new Features

Technische Universität München

General:

- More user friendly code (I hope).
- Better data-structure: TrackPoints with Measurements and FitterInfos instead of GFBookkeeping.
- Full Tracks need more memory/disk space, but pruning function allows to strip off unneeded data.
- Easy track merging, adding and removing hits, fast refitting.
- Built-in consistency checks to spot and avoid errors.
- Time of flight calculation.
- Improved visualization.
- Easier Measurement implementation, can use H-Matrices provided by GENFIT.
- H-Matrix for slanted strips.

Main new Features

Fitters:

- FitStatus objects in track (which fitter was used, number of iterations, fit converged?, NdF, χ^2, p-values).
- Kalman filter with reference track, which can also re-sort TrackPoints.
- Possibility to fit tracks partially (Kalman).
- optional square-root formalism (Kalman).
- Possibility to save ThinScatterers in TrackPoints (not yet used, but useful for GBL).

Configurable Convergence Criteria:

- Minimum and maximum number of iterations.
- Converged if:
  - Δ pVal $< 1\mathrm{E}-3$
  - relative change in $\chi^2 < 20\,\%$
  - (default values, configurable)

Main new Features

TrackReps:

- More extrapolation options, possible to stop at material boundaries.
- Get list of materials crossed in last extrapolation.
- Get path / radiation length crossed in last extrapolation.
- Get time of flight of last extrapolation.

TrackCand:

- TrackCandHits, from which one can inherit and save all necessary information (already in GENFIT 1).
- In the new MeasurementFactory, pointers to the TrackCandHits are provided to the Measurement (fka RecoHit) constructors, so these information can be used to construct the measurements.
  This is now used to set the left/right ambiguity of the CDCRecoHits directly in the CDCRecoHit constructor.

Migration Status

Migration Status

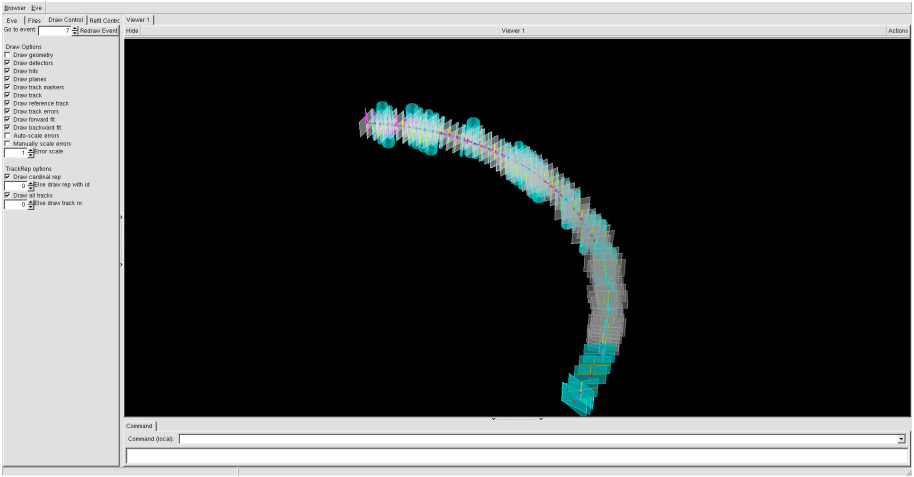
Migration completed:

- **genfit2** as package in basf2 (via svn:externals).
- New externals v00-05-00 w/o GENFIT.
- Doxygen: `https://belle2.cc.kek.jp/internal/software/development/namespacegenfit.html`.
- Problem: svn:externals don't work well with gitsvn, can be worked around.
- Still some problems and crashes: work in progress.

New Module:

- **GenfitVisModule** passes genfit::Tracks to genfit::EventDisplay.
- Visualize tracks, refit tracks, change fitters and settings, activate debug output, ...

GENFITVisModule



Validation Status

Tracking Validation

Standard Validation:

- Validation scripts to check pull distributions (x, y, z, p_x, p_y, p_z), p-value distributions, resolutions and efficiency.
- PXD + SVD + CDC fitting with GENFIT's Kalman filter and RKTrackRep.
- 1.000 tracks with fixed momentum and theta.
- Nightly results at: `https://belle2.cc.kek.jp/internal/validation/index.html`

Advanced Options:

- In the steering script (`01_steering_genfitStudy.py`), various selections can be made:
  - Turn off/on PXD, SVD, CDC separately.
  - Switch the fitter.
  - Turn on the full test, 100.000 tracks over the whole phase-space.

GENFIT 1 vs. GENFIT 2

GENFIT 2

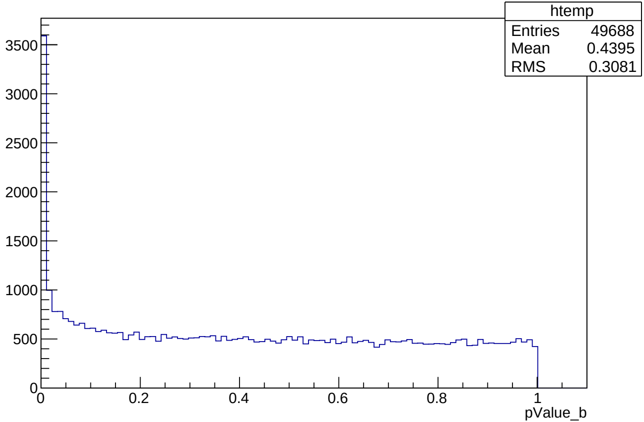
- Test results look very similar to GENFIT 1 results.
- Pull distributions also look very similar (I showed wrong plots on B2GM).
- Slightly more tracks with p-value $\simeq 0$.

GenFitter module comparison

- Pions, $0.4\,\mathrm{GeV}$, $\theta = 120°$, pxd + svd + cdc.
- GENFIT 1: 25.168 ms/call (3 iterations)
- GENFIT 2: 25.108 ms/call (3.2 iterations on average)

GENFIT 1 vs. GENFIT 2: p-Value Distributions

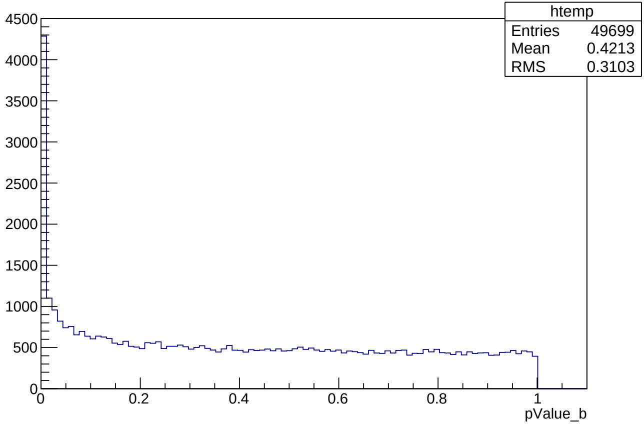
50000 Pions, $(0.2$ to $3.0)$ GeV, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: p-Value Distributions

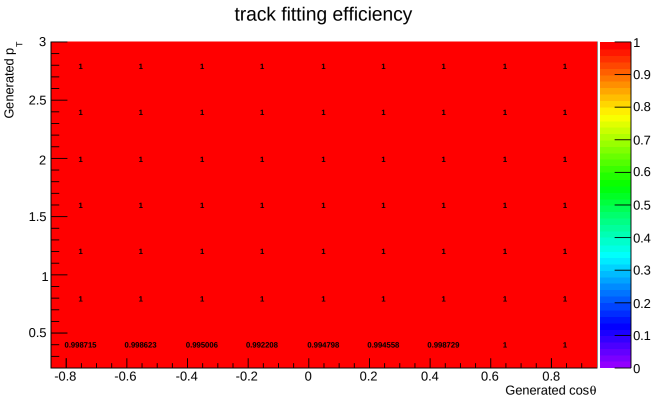
50000 Pions, $(0.2$ to $3.0)$ GeV, $\theta = (17$ to $150)\,^{\circ}$, pxd + svd + cdc.



GENFIT 2 with $\geq$ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

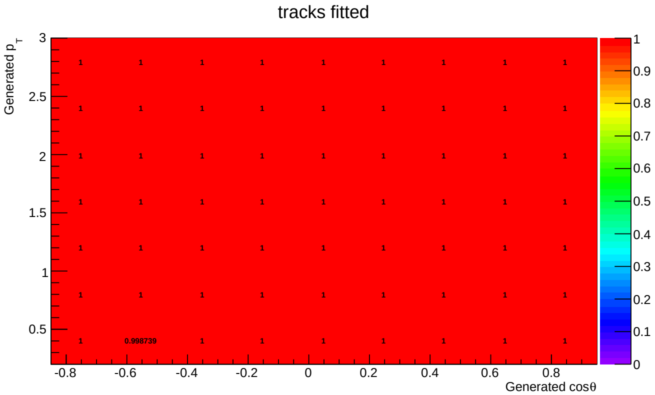
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

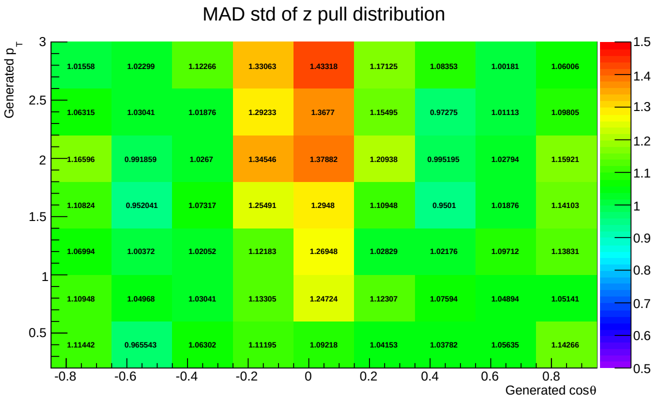
50000 Pions, $(0.2$ to $3.0)$ GeV, $\theta = (17$ to $150)$ °, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

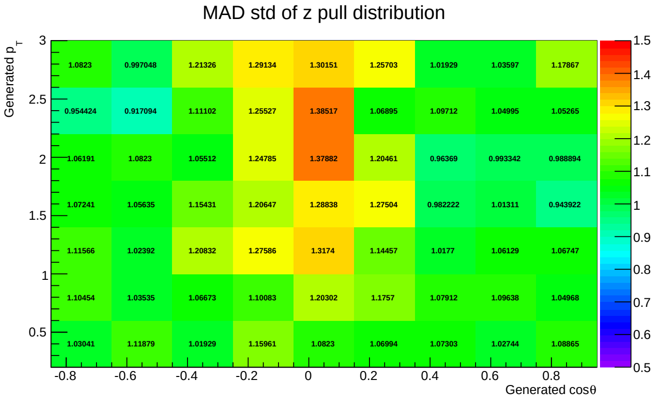
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

50000 Pions, $(0.2$ to $3.0)$ GeV, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.

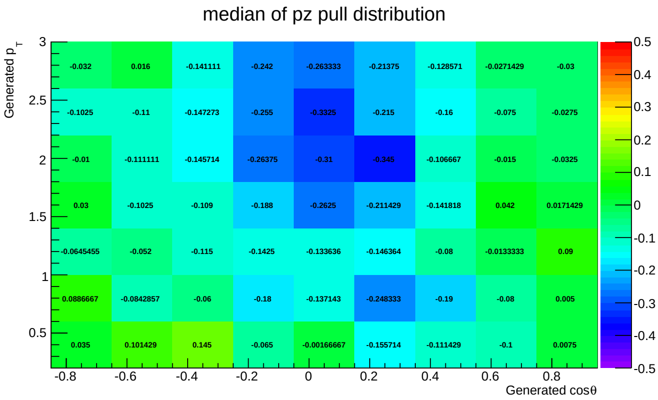


GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

Technische Universität München

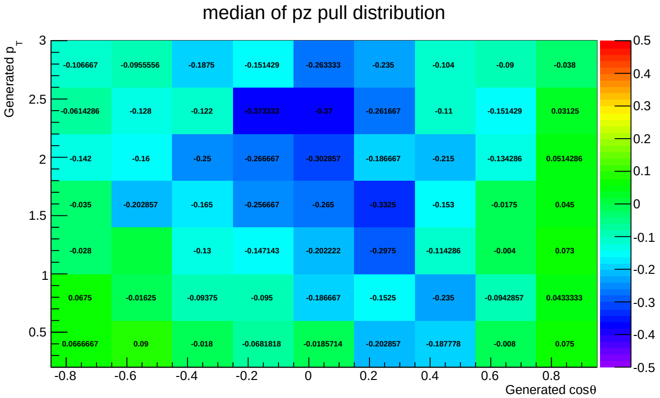
50000 Pions, $(0.2$ to $3.0)\ \mathrm{GeV}$, $\theta = (17$ to $150)\ ^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

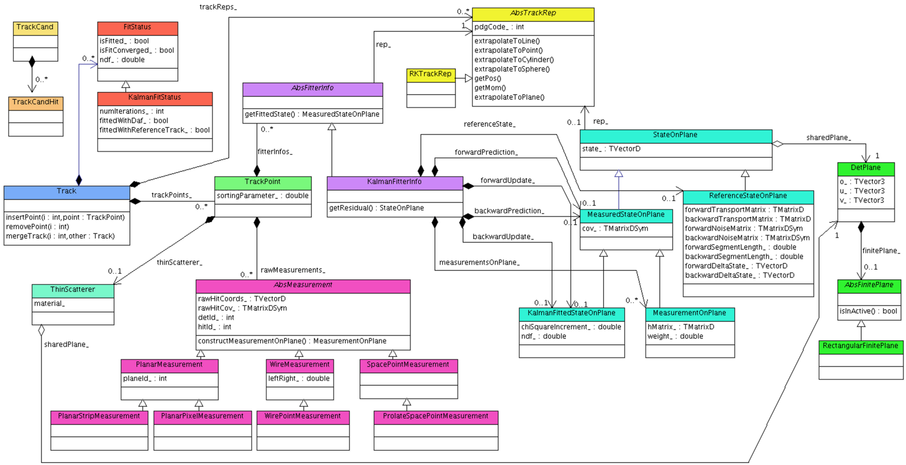
50000 Pions, (0.2 to 3.0) GeV, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



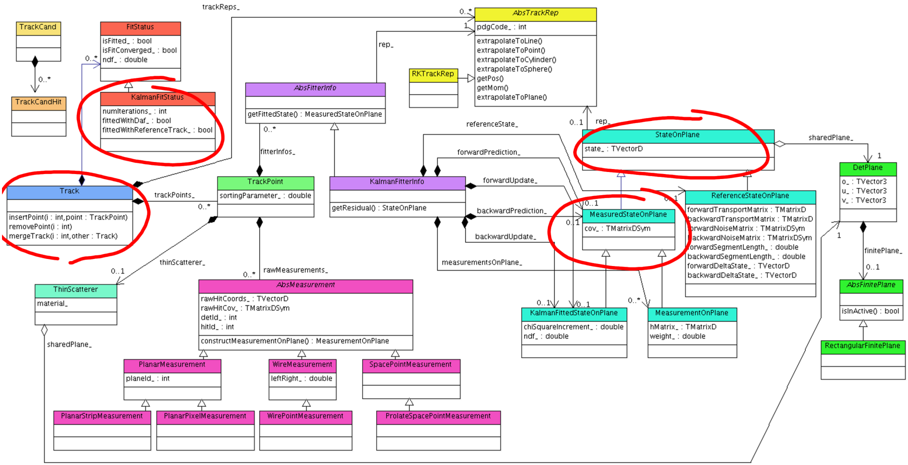
GENFIT 2 with ≥ 3 iterations.

GENFIT 2 Code Examples

GENFIT Track Data Structure



GENFIT Track Data Structure



Main Differences from GENFIT 1

Getting information about the fit from a fitted track:

GENFIT 1

```
gfTrack->getCardinalRep()->getStatusFlag()==0

gfTrack->getCardinalRep()->getChiSqu()
gfTrack->getCardinalRep()->getNDF()
```

GENFIT 2

```
gfTrack->getFitStatus()->isFitted()
gfTrack->getFitStatus()->isFitConverged()
gfTrack->getFitStatus()->getChi2()
gfTrack->getFitStatus()->getNdf()
gfTrack->getFitStatus()->getCharge()
```

Get Information about the Fit

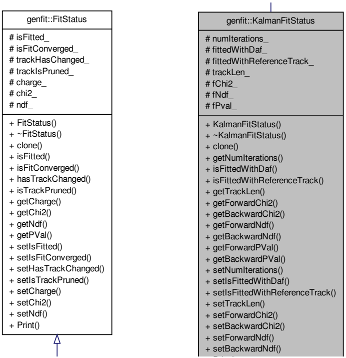
FitStatus* genfit::Track::getFitStatus (const AbsTrackRep * rep = NULL) const [inline]

Get **FitStatus** for a **AbsTrackRep**. Per default, return **FitStatus** for cardinalRep.

Definition at line **136** of file **Track.h**.

References **cardinalRep_**, and **trackReps_**.

Referenced by **genfit::GFRaveTrackParameters::getCharge()**, **genfit::AbsKalmanFitter::isTrackFitted()**, **genfit::KalmanFitterRefTrack::prepareTrack()**, **genfit::KalmanFitter::processTrackPartially()**, **genfit::KalmanFitterRefTrack::processTrackWithRep()**, **genfit::KalmanFitter::processTrackWithRep()**, and **genfit::DAF::processTrackWithRep()**.



Main Differences from GENFIT 1

Biggest difference: State (+ cov) is NOT stored in TrackRep anymore.
Instead: TrackPoints contain FitterInfos contain StateOnPlane objects.

GENFIT 1

```
GFAbsTrackRep* rep =
    gfTrack->getCardinalRep();
TVector3 pos = rep->getPos();
TVector3 mom = rep->getMom();
TMatrixDSym cov = rep->getCov(); // 5x5

TMatrixDSym cov6; // 6x6
rep->getPosMomCov(pos, mom, cov6);
```

GENFIT 2

```
const MeasuredStateOnPlane& state =
    gfTrack->getFittedState();
TVector3 pos = state.getPos();
TVector3 mom = state.getMom();
TMatrixDSym cov = state.getCov(); // 5x5

TMatrixDSym cov6 = state.get6DCov(); // 6x6
```

Get Fitted State


```
const MeasuredStateOnPlane & genfit::Track::getFittedState ( int                   id = 0,
                                                             const AbsTrackRep *   rep = NULL,
                                                             bool                  biased = true
                                                           )                       const
```

Shortcut to get FittedStates.

Uses getPointWithMeasurementAndFitterInfo(id, rep). Per default, the fitted state of the fitterInfo of the first **TrackPoint** with one or more **AbsMeasurement** and **AbsFitterInfo** objects is returned. If no **AbsTrackRep** is specified, the **AbsFitterInfo** of the cardinal rep will be used.

Definition at line **220** of file **Track.cpp**.

References **getCardinalRep()**, **genfit::AbsFitterInfo::getFittedState()**, **genfit::TrackPoint::getFitterInfo()**, **getPointWithMeasurementAndFitterInfo()**, and **genfit::Exception::setFatal()**.

Main Differences from GENFIT 1

Extrapolate fitted state, e.g. to vertex:

GENFIT 1

```
TVector3 vertexPos, vertexMom;
TMatrixDSym vertexCov;
GFAbsTrackRep* rep =
    gfTrack->getCardinalRep();
TVector3 vertex(0,0,0);
TVector3 poca, dirInPoca;
rep->extrapolateToPoint(vertex,
    poca, dirInPoca); // does not alter rep
GFDetPlane plane(poca, dirInPoca);
rep->getPosMomCov(plane, vertexPos,
    vertexMom, vertexCov);
```

GENFIT 2

```
TVector3 vertexPos, vertexMom;
TMatrixDSym vertexCov;
MeasuredStateOnPlane state =
    gfTrack->getFittedState(); // copy
TVector3 vertex(0,0,0);

state.extrapolateToPoint(vertex)

state.getPosMomCov(vertexPos,
    vertexMom, vertexCov);
```

Main Differences from GENFIT 1

Get time of flight between vertex and cylinder:

GENFIT 2

```
TVector3 vertex(0,0,0);
double radius = 20;
double tof = gfTrack->getTOF(); // TOF from first to last hit
StateOnPlane state = gfTrack->getFittedState();
state.extrapolateToPoint(vertex);
// TOF is SIGNED!
tof -= state.getRep()->getTOF(); // + TOF from vertex to first hit

state = gfTrack->getFittedState(-1);
state.extrapolateToCylinder(radius);
tof += state.getRep()->getTOF(); // + TOF from last hit to cylinder
```

Get Track-length and TOF


```
double genfit::Track::getTOF ( AbsTrackRep *  rep,
                               int            startId = 0,
                               int            endId = -1
                             )                const
```

get time of flight in ns between to trackPoints

Definition at line 768 of file Track.cpp.

References genfit::AbsTrackRep::extrapolateToPlane(), genfit::AbsTrackRep::getMass(), genfit::AbsTrackRep::getMomMag(), genfit::StateOnPlane::getPlane(), and trackPoints_.

```
double genfit::Track::getTrackLen ( AbsTrackRep *  rep,
                                    int            startId = 0,
                                    int            endId = -1
                                  )                const
```

get TrackLength between to trackPoints

Definition at line 727 of file Track.cpp.

References genfit::AbsTrackRep::extrapolateToPlane(), genfit::StateOnPlane::getPlane(), and trackPoints_.

Insert Points

Technische Universität München

```
void genfit::Track::insertPoint ( TrackPoint *  point,
                                  int           id = -1
                                )
```

Insert TrackPoint BEFORE TrackPoint with position id, if id >= 0.

Id -1 means after last TrackPoint. Id -2 means before last TrackPoint. ... Also deletes backwardInfos before new point and forwardInfos after new point. Also sets Track backpointer of point accordingly.

Definition at line 263 of file Track.cpp.

References deleteBackwardInfo(), deleteForwardInfo(), deleteReferenceInfo(), fillPointsWithMeasurement(), genfit::TrackPoint::hasRawMeasurements(), genfit::TrackPoint::setTrack(), trackHasChanged(), trackPoints_, and trackPointsWithMeasurement_.

Referenced by insertPoints(), and Track().

```
void genfit::Track::insertPoints ( std::vector< genfit::TrackPoint * >  points,
                                   int                                   id = -1
                                 )
```

Definition at line 323 of file Track.cpp.

References deleteBackwardInfo(), deleteForwardInfo(), deleteReferenceInfo(), fillPointsWithMeasurement(), getNumPoints(), insertPoint(), and trackPoints_.

Referenced by mergeTrack().

Delete Points, Merge Tracks

void genfit::Track::deletePoint (int id)

Definition at line 373 of file Track.cpp.

References deleteBackwardInfo(), deleteForwardInfo(), deleteReferenceInfo(), trackHasChanged(), trackPoints_, and trackPointsWithMeasurement_.

```
void genfit::Track::mergeTrack ( const Track *  other,
                                 int            id = -1
                               )
```

Merge two tracks.

The TrackPoint objects of other will be cloned and inserted after id (per default, they will be appended at the end). The other Track will not be altered, the TrackPoint objects will be (deep) copied. Only copies the TrackPoint objects, NOT the AbsTrackRep, FitStatus, seed state and other objets of the other track.

Definition at line 406 of file Track.cpp.

References getNumPoints(), insertPoints(), genfit::Exception::setFatal(), trackPoints_, and trackReps_.

TODOs

TODOs

Urgent:

- Fix remaining bugs from migration.

New GENFIT features which should be used by basf2:

- Use GENFIT's time-of-flight calculation for CDC drift-time corrections.
- Use HMatrixPhi for slanted SVD sensor-planes.
- Re-integrate GENFIT visualization into basf2 display.

GenFitterModule:

- Fitted vs. fitted and converged?

Support from TUM

Support from TUM

My Status

- I will switch over to Belle I analysis soon.
- I can still do maintenance, but no more big new developments.

E18 Status:

- E18 wants to hire another Belle (II) PhD student.
- Two candidates, start around Feb/Mar 2014.
- Will also work on GENFIT as service task.
- "We will take care of maintenance of GENFIT."
- Active development of new features will be reduced for a while.

Thanks for your attention!

Backup Slides

Fitting a Track

The following descriptions and diagrams are taken from GENFIT's doxygen documentation.

```
void genfit::AbsFitter::processTrack ( Track *  tr,
                                       bool     resortHits = true
                                     )
```

Process all reps. Start with the cardinalRep and resort the hits if necessary (and supported by the fitter)

Definition at line 25 of file AbsFitter.cpp.

References genfit::Track::checkConsistency(), genfit::Track::getCardinalRep(), genfit::Track::getNumReps(), genfit::Track::getTrackRep(), and processTrackWithRep().

```
virtual void genfit::AbsFitter::processTrackWithRep ( Track *               ,
                                                      const AbsTrackRep *   ,
                                                      bool                  resortHits = false
                                                    )                                          [pure virtual]
```

Process Track with one AbsTrackRep of the Track. Optionally resort the hits if necessary (and supported by the fitter)

Implemented in genfit::DAF, genfit::KalmanFitter, and genfit::KalmanFitterRefTrack.

Referenced by processTrack().

Get Fit Result


```
const MeasuredStateOnPlane & genfit::Track::getFittedState ( int                  id = 0,
                                                             const AbsTrackRep *  rep = NULL,
                                                             bool                 biased = true
                                                           )                      const
```

Shortcut to get FittedStates.

Uses getPointWithMeasurementAndFitterInfo(id, rep). Per default, the fitted state of the fitterInfo of the first TrackPoint with one or more AbsMeasurement and AbsFitterInfo objects is returned. If no AbsTrackRep is specified, the AbsFitterInfo of the cardinal rep will be used.

Definition at line 220 of file Track.cpp.

References getCardinalRep(), genfit::AbsFitterInfo::getFittedState(), genfit::TrackPoint::getFitterInfo(), getPointWithMeasurementAndFitterInfo(), and genfit::Exception::setFatal().

Important: The TrackReps do NOT contain the state anymore.
Instead, all states are stored in the FitterInfos in the TrackPoints.

Get Plane

const SharedPlanePtr& genfit::StateOnPlane::getPlane ( ) const [inline]

Definition at line 63 of file StateOnPlane.h.

References sharedPlane_.

Referenced by genfit::KalmanFitterInfo::addMeasurementOnPlane(), genfit::RKTrackRep::checkCache(), genfit::RKTrackRep::extrapolateBy(), genfit::RKTrackRep::extrapolateToCylinder(), genfit::RKTrackRep::extrapolateToLine(), genfit::RKTrackRep::extrapolateToPlane(), genfit::RKTrackRep::extrapolateToPoint(), genfit::RKTrackRep::extrapolateToSphere(), genfit::KalmanFitterInfo::getResidual(), genfit::RKTrackRep::getState5(), genfit::RKTrackRep::getState7(), genfit::Track::getTOF(), genfit::Track::getTrackLen(), genfit::KalmanFitterRefTrack::prepareTrack(), genfit::KalmanFitterRefTrack::processTrackPoint(), genfit::RKTrackRep::setPosMom(), genfit::RKTrackRep::setPosMomErr(), genfit::RKTrackRep::transformM6P(), genfit::RKTrackRep::transformM7P(), genfit::RKTrackRep::transformPM6(), and genfit::RKTrackRep::transformPM7().

Get State


```
const TVectorD& genfit::StateOnPlane::getState()const [inline]

TVectorD& genfit::StateOnPlane::getState()[inline]

const TMatrixDSym& genfit::MeasuredStateOnPlane::getCov()const [inline]
```

Get the state (and covariance) in the parametrization of the TrackRep.
For the RKTrackRep, this is $(q/p, u', v', u, v)$.

Get Position, Momentum, 6x6 Covariance


```
void genfit::MeasuredStateOnPlane::getPosMomCov ( TVector3 &       pos,
                                                  TVector3 &       mom,
                                                  TMatrixDSym &    cov
                                                )                 const [inline]
```

Definition at line 63 of file MeasuredStateOnPlane.h.

References genfit::AbsTrackRep::getPosMomCov(), and genfit::StateOnPlane::getRep().

These and the following functions call the corresponding functions of the TrackRep connected to the StateOnPlane/MeasuredStateOnPlane.

Get Pos etc.

TVector3 genfit::StateOnPlane::getPos ( ) const [inline]

Definition at line 91 of file StateOnPlane.h.

References genfit::AbsTrackRep::getPos(), and rep_.

void genfit::StateOnPlane::getPosDir (TVector3 & pos, TVector3 & dir) const [inline]

Definition at line 95 of file StateOnPlane.h.

References genfit::AbsTrackRep::getPosDir(), and rep_.

void genfit::StateOnPlane::getPosMom (TVector3 & pos, TVector3 & mom) const [inline]

Definition at line 94 of file StateOnPlane.h.

References genfit::AbsTrackRep::getPosMom(), and rep_.

double genfit::StateOnPlane::getQop ( ) const [inline]

Definition at line 100 of file StateOnPlane.h.

References genfit::AbsTrackRep::getQop(), and rep_.

Get Various

TVector3 genfit::StateOnPlane::getMom ( ) const [inline]

Definition at line 92 of file StateOnPlane.h.

References genfit::AbsTrackRep::getMom(), and rep_.

double genfit::StateOnPlane::getMomMag ( ) const [inline]

Definition at line 97 of file StateOnPlane.h.

References genfit::AbsTrackRep::getMomMag(), and rep_.

int genfit::StateOnPlane::getPDG ( ) const [inline]

Definition at line 98 of file StateOnPlane.h.

References genfit::AbsTrackRep::getPDG(), and rep_.

Get Various

double genfit::StateOnPlane::getCharge ( ) const [inline]

Definition at line 99 of file StateOnPlane.h.

References genfit::AbsTrackRep::getCharge(), and rep_.

TVector3 genfit::StateOnPlane::getDir ( ) const [inline]

Definition at line 93 of file StateOnPlane.h.

References genfit::AbsTrackRep::getDir(), and rep_.

double genfit::StateOnPlane::getMass ( ) const [inline]

Definition at line 101 of file StateOnPlane.h.

References genfit::AbsTrackRep::getMass(), and rep_.

Extrapolate


```
double genfit::StateOnPlane::extrapolateBy ( double  step,
                                              bool    stopAtBoundary = false
                                            )                                  [inline]
```

Definition at line 88 of file StateOnPlane.h.

References genfit::AbsTrackRep::extrapolateBy(), and rep_.

```
double genfit::StateOnPlane::extrapolateToCylinder ( double            radius,
                                                      const TVector3 &  linePoint = TVector3(0.,0.,0.),
                                                      const TVector3 &  lineDirection = TVector3(0.,0.,1.),
                                                      bool              stopAtBoundary = false
                                                    )                                              [inline]
```

Definition at line 81 of file StateOnPlane.h.

References genfit::AbsTrackRep::extrapolateToCylinder(), and rep_.

```
double genfit::StateOnPlane::extrapolateToLine ( const TVector3 &  linePoint,
                                                  const TVector3 &  lineDirection,
                                                  bool              stopAtBoundary = false
                                                )                                  [inline]
```

Definition at line 76 of file StateOnPlane.h.

References genfit::AbsTrackRep::extrapolateToLine(), and rep_.

Extrapolate

double genfit::StateOnPlane::extrapolateToPlane (const SharedPlanePtr & plane,
bool stopAtBoundary = false,
bool calcJacobianNoise = false
) [inline]

Definition at line 73 of file StateOnPlane.h.

References genfit::AbsTrackRep::extrapolateToPlane(), and rep_.

double genfit::StateOnPlane::extrapolateToPoint (const TVector3 & point,
bool stopAtBoundary = false
) [inline]

Definition at line 79 of file StateOnPlane.h.

References genfit::AbsTrackRep::extrapolateToPoint(), and rep_.

double genfit::StateOnPlane::extrapolateToSphere (double radius,
const TVector3 & point = TVector3(0.,0.,0.),
bool stopAtBoundary = false
) [inline]

Definition at line 85 of file StateOnPlane.h.

References genfit::AbsTrackRep::extrapolateToSphere(), and rep_.

Set Position and Momentum

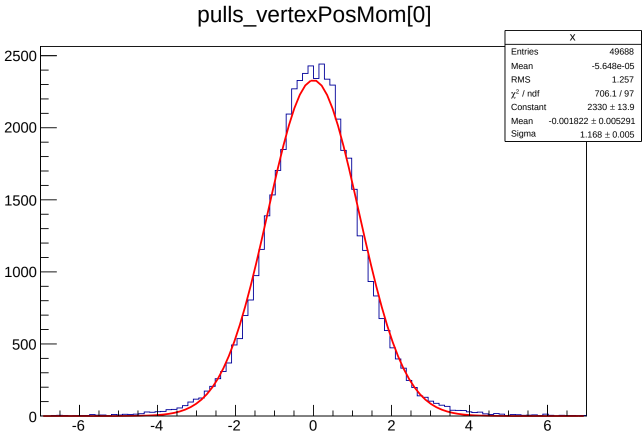

```
void genfit::StateOnPlane::setPosMom ( const TVector3 &  pos,
                                       const TVector3 &  mom
                                     )                       [inline]
```

Definition at line 103 of file StateOnPlane.h.

References rep_, and genfit::AbsTrackRep::setPosMom()

GENFIT 1 vs. GENFIT 2: Pull Distributions

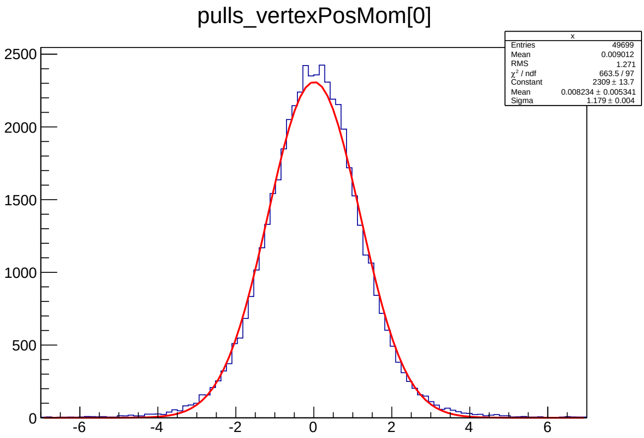
50000 Pions, $(0.2$ to $3.0)$ GeV, $\theta = (17$ to $150)\,°$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: Pull Distributions

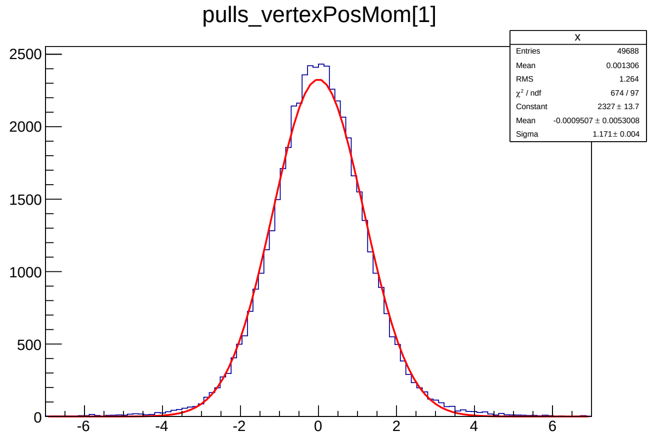
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: Pull Distributions

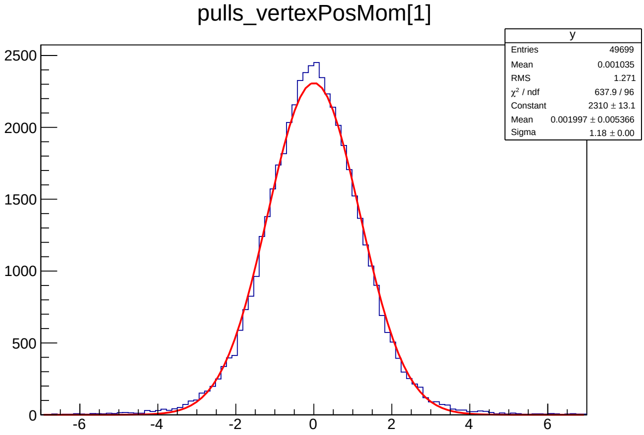
50000 Pions, $(0.2 \text{ to } 3.0)\,\text{GeV}$, $\theta = (17 \text{ to } 150)\,°$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: Pull Distributions

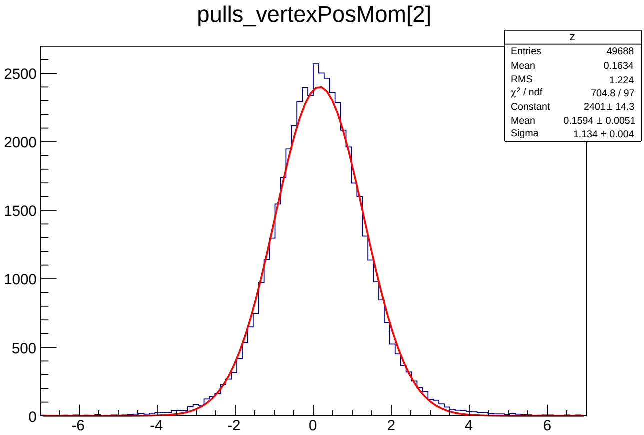
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with $\geq$ 3 iterations.

GENFIT 1 vs. GENFIT 2: Pull Distributions

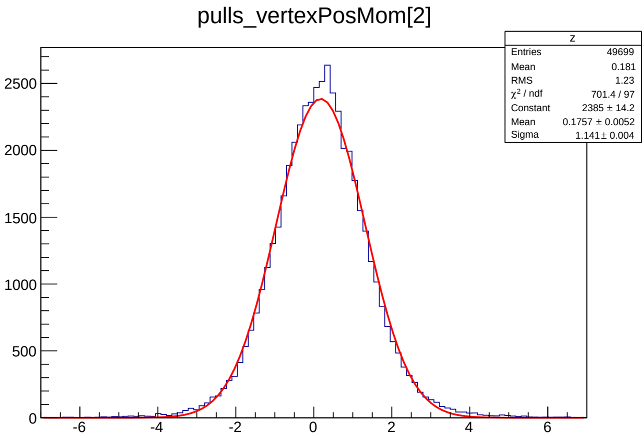
50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,°$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: Pull Distributions

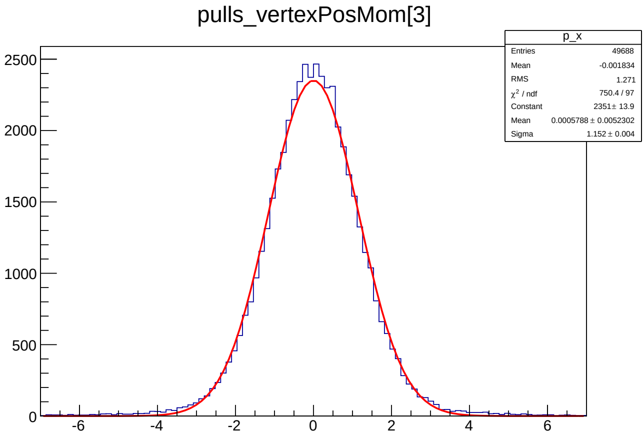
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: Pull Distributions

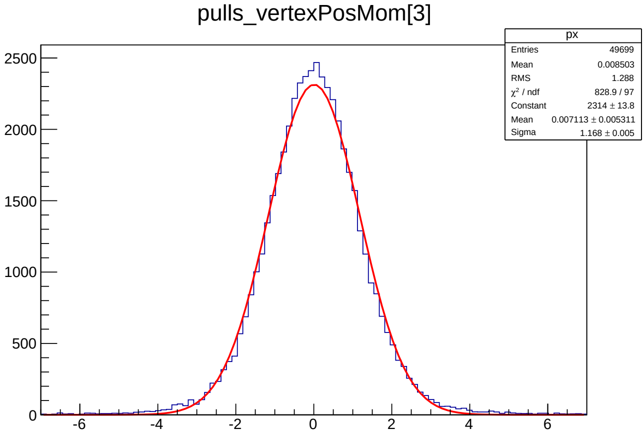
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: Pull Distributions

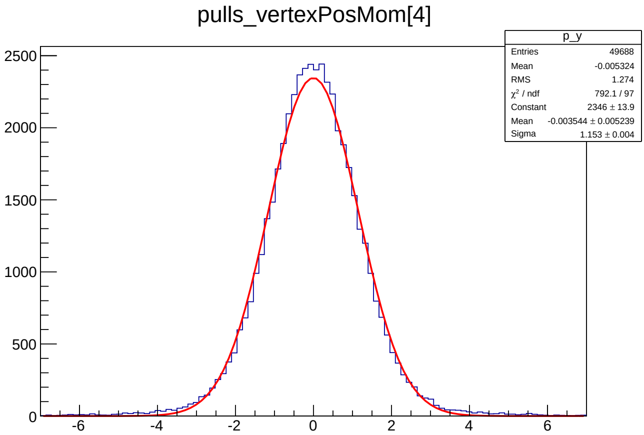
50000 Pions, $(0.2 \text{ to } 3.0)\,\text{GeV}$, $\theta = (17 \text{ to } 150)\,°$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: Pull Distributions

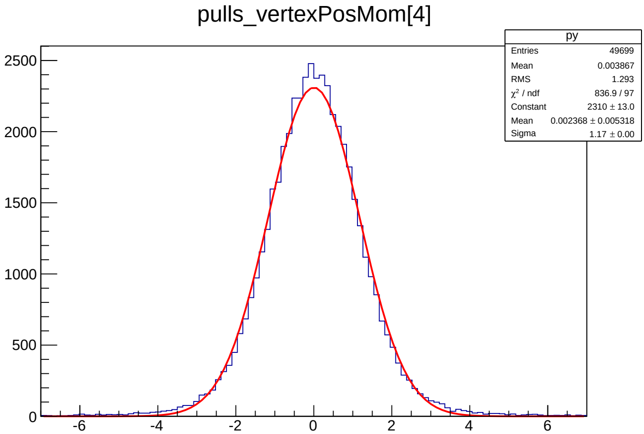
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^{\circ}$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: Pull Distributions

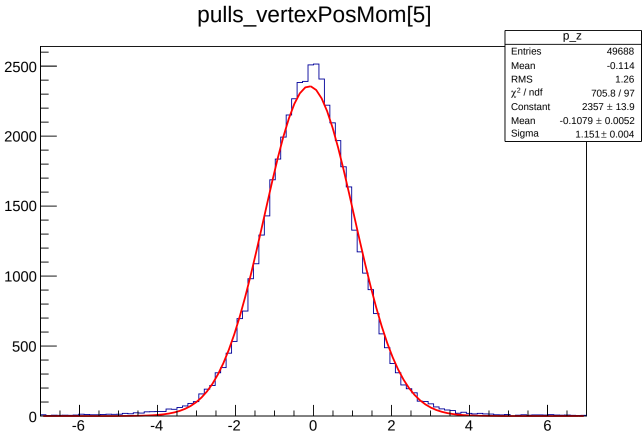
50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,°$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: Pull Distributions

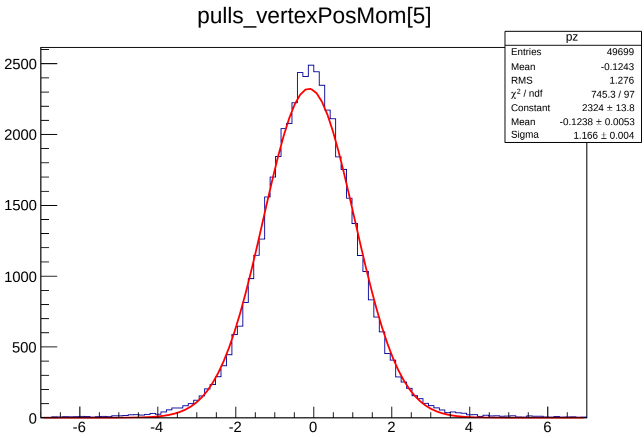
50000 Pions, $(0.2$ to $3.0)$ GeV, $\theta = (17$ to $150)$ °, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: Pull Distributions

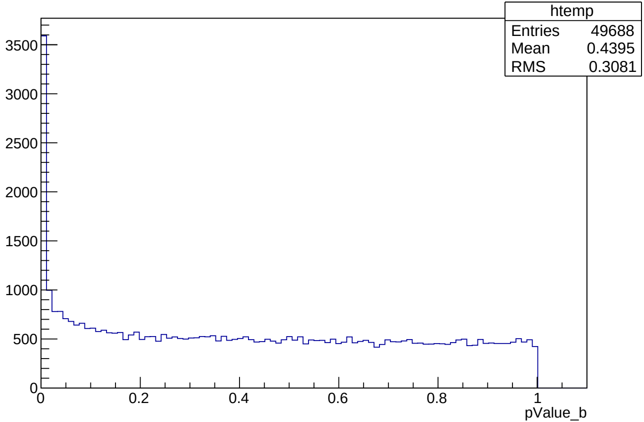
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with $\geq$ 3 iterations.

GENFIT 1 vs. GENFIT 2: P-Value Distributions

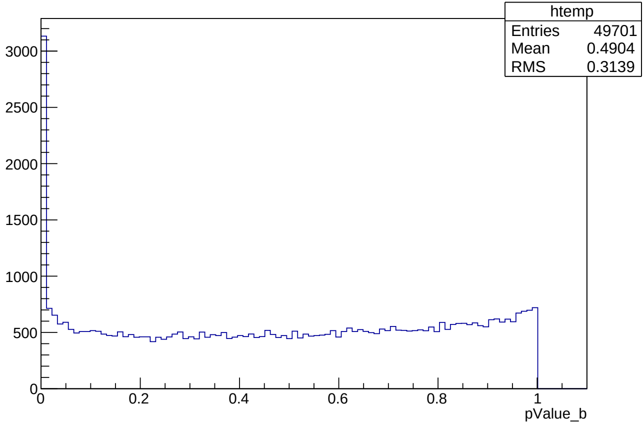
50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: P-Value Distributions

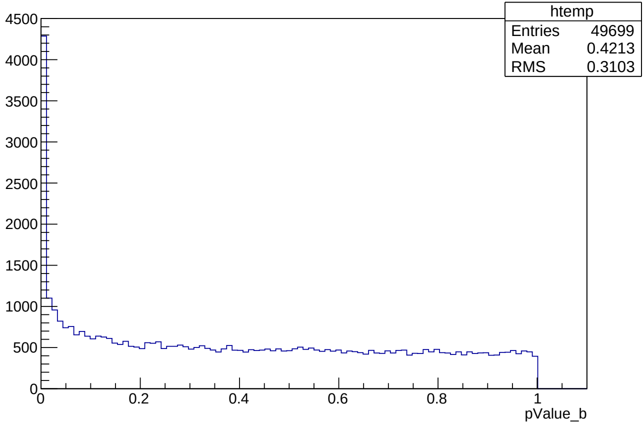
50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,°$, pxd + svd + cdc.



GENFIT 2 with $\geq$ 2 iterations.

GENFIT 1 vs. GENFIT 2: P-Value Distributions

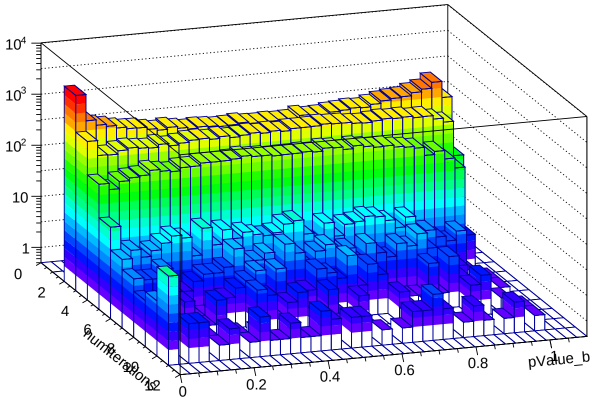
50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,^{\circ}$, pxd + svd + cdc.



GENFIT 2 with $\geq$ 3 iterations.

GENFIT 2: P-Values vs. Number of Iterations

50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.

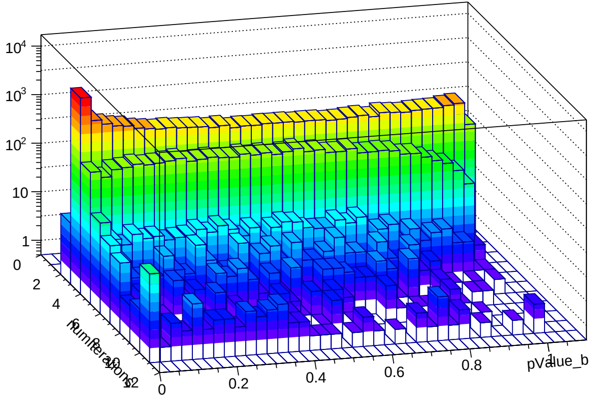


GENFIT 2 with $\geq$ 2 iterations.

GENFIT 2: P-Values vs. Number of Iterations

Technische Universität München

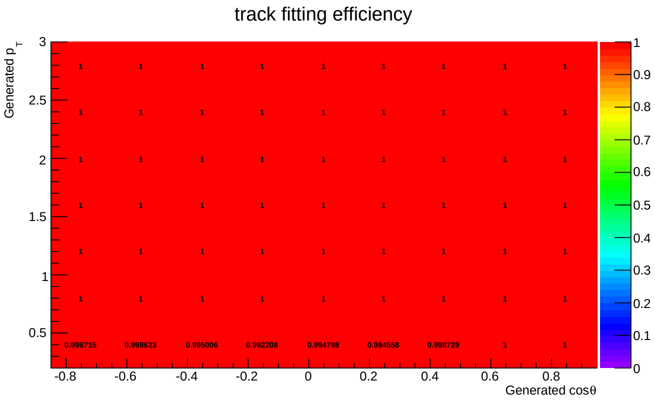
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with $\geq$ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

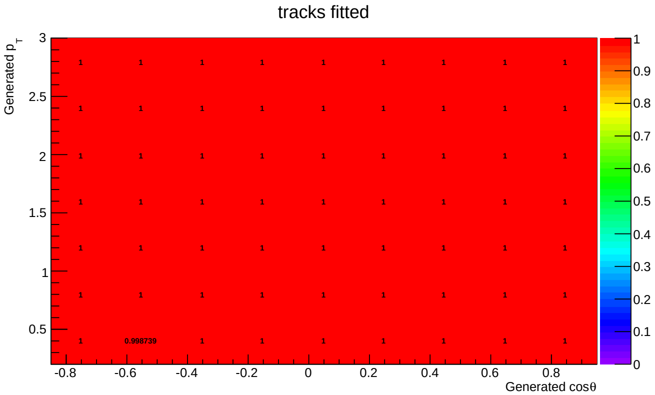
50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

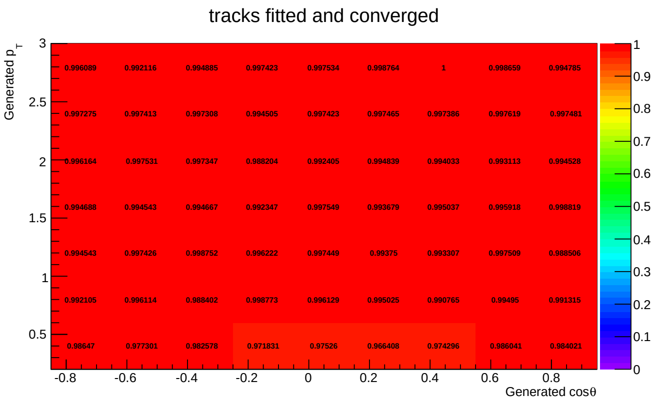
50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,^{\circ}$, pxd + svd + cdc.



GENFIT 2 with ≥ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

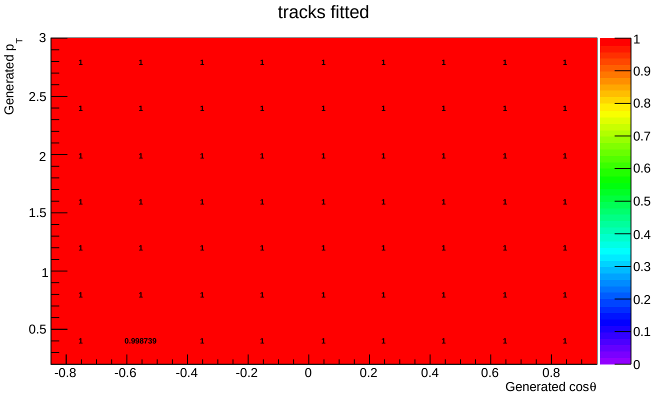
50000 Pions, $(0.2 \text{ to } 3.0)$ GeV, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

50000 Pions, $(0.2$ to $3.0)$ GeV, $\theta = (17$ to $150)$ °, pxd + svd + cdc.

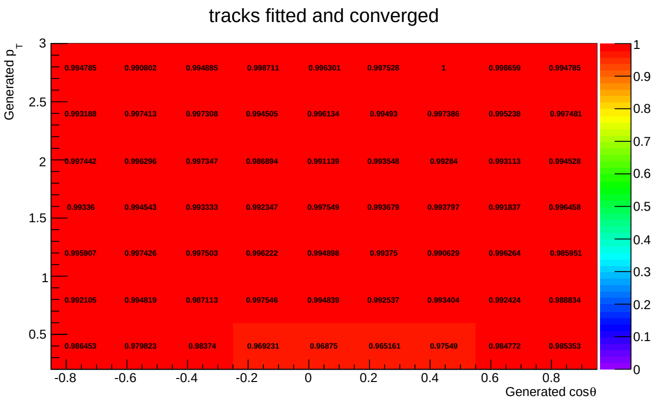


GENFIT 2 with $\geq$ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

Technische Universität München

50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.

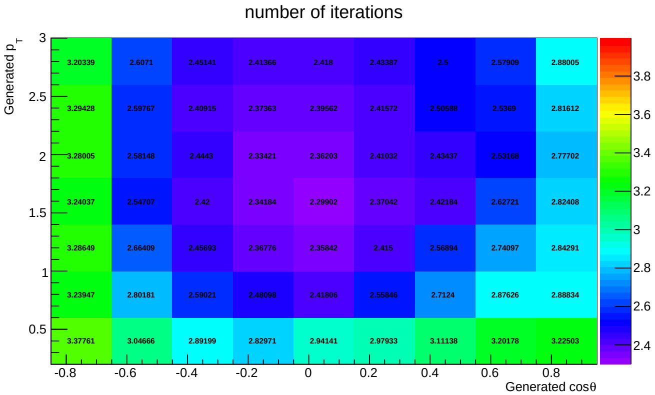


GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

Technische Universität München

50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^{\circ}$, pxd + svd + cdc.

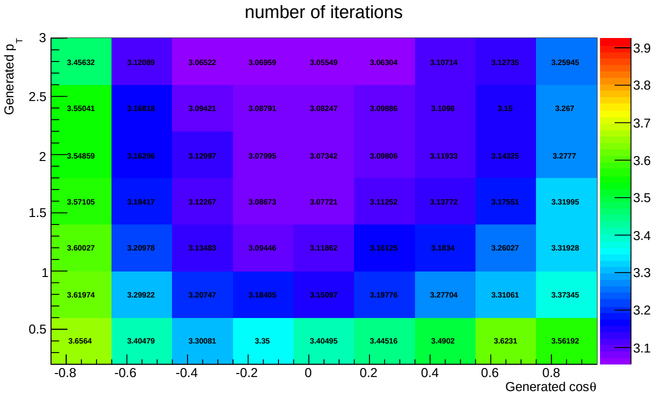


GENFIT 2 with ≥ 2 iterations.

Technische Universität München

GENFIT 1 vs. GENFIT 2: 2D QA Plots

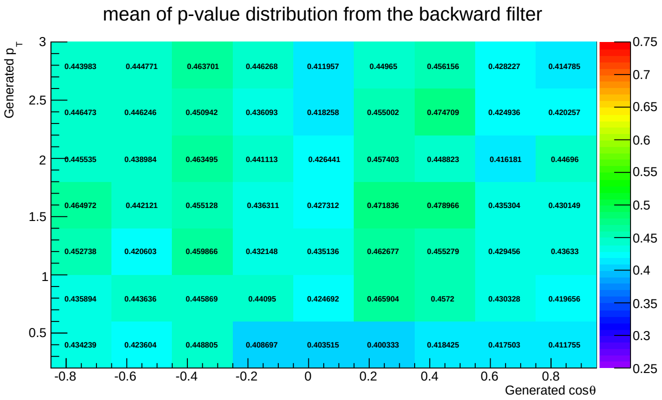
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,°$, pxd + svd + cdc.



GENFIT 2 with $\geq$ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,°$, pxd + svd + cdc.

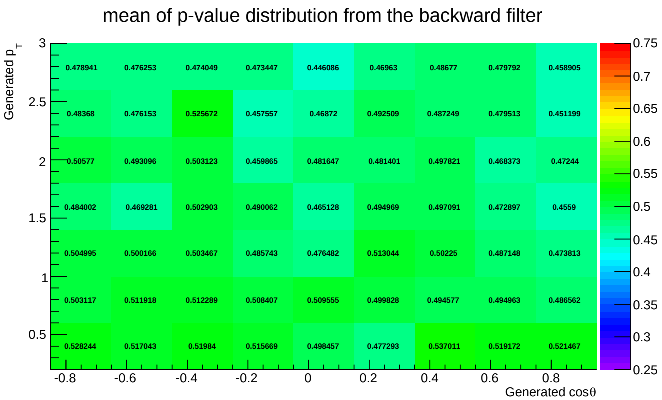


GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

Technische Universität München

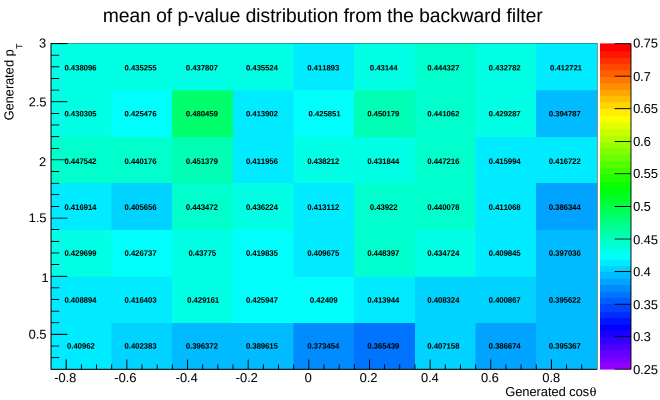
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

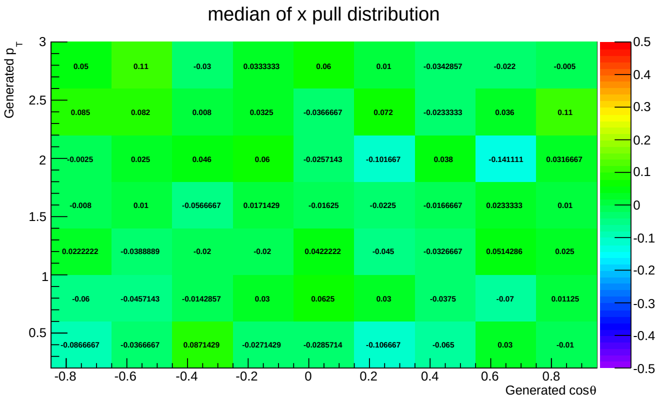
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,°$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

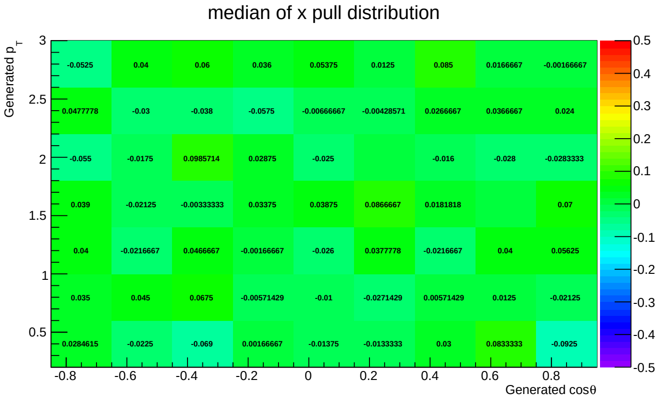
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,°$, pxd + svd + cdc.

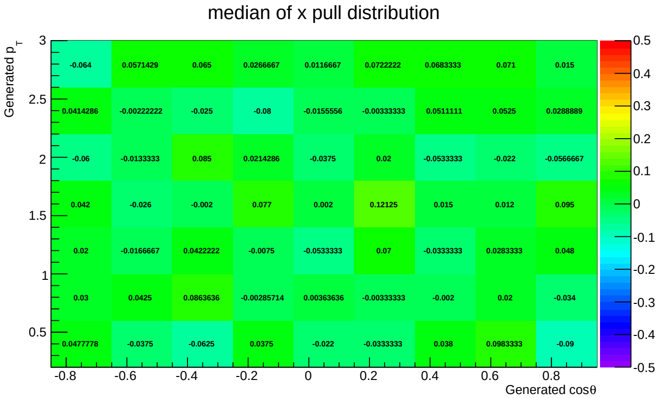


GENFIT 2 with ≥ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

Technische Universität München

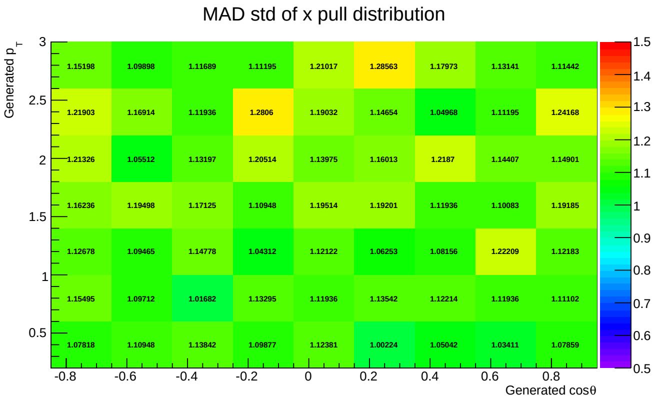
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^{\circ}$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

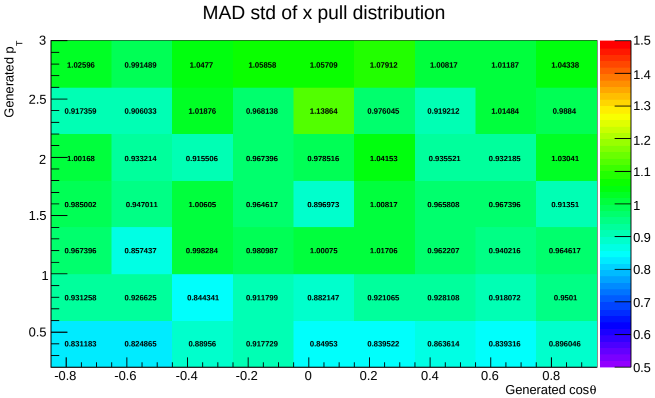
50000 Pions, $(0.2$ to $3.0)$ GeV, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

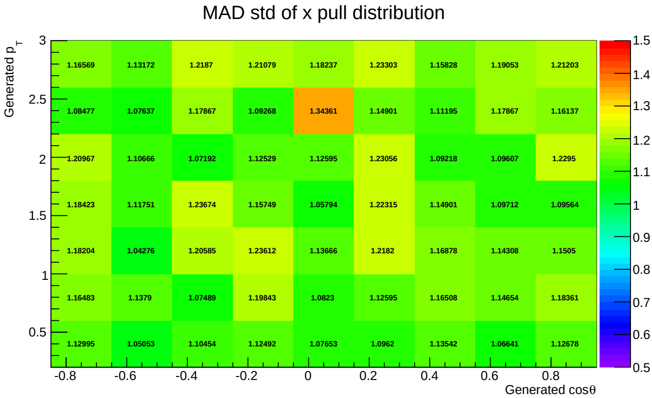
50000 Pions, $(0.2$ to $3.0)$ GeV, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

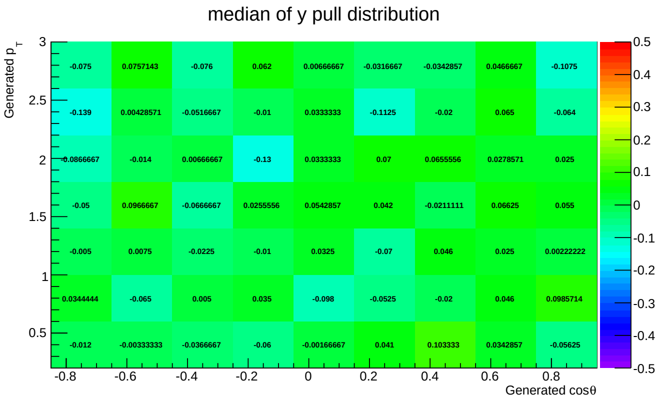
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,°$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

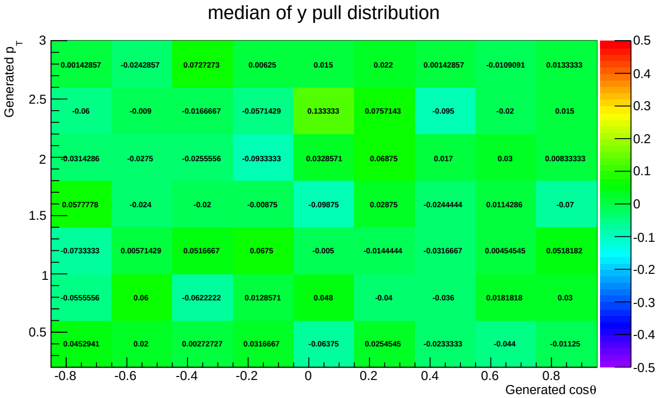
50000 Pions, $(0.2$ to $3.0)$ GeV, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

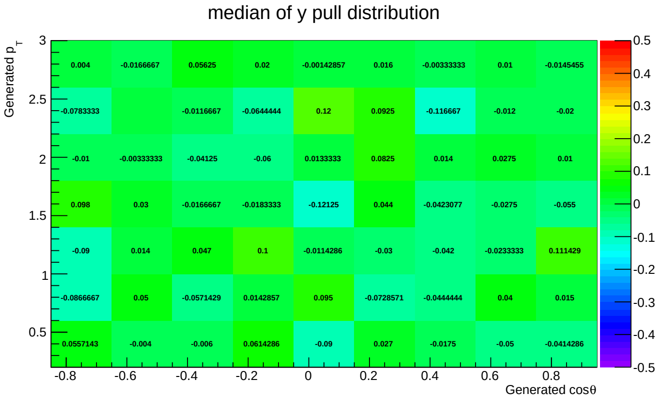
50000 Pions, $(0.2$ to $3.0)$ GeV, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

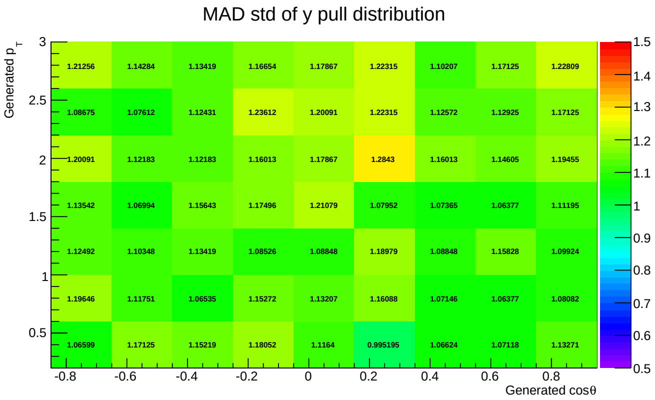
50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

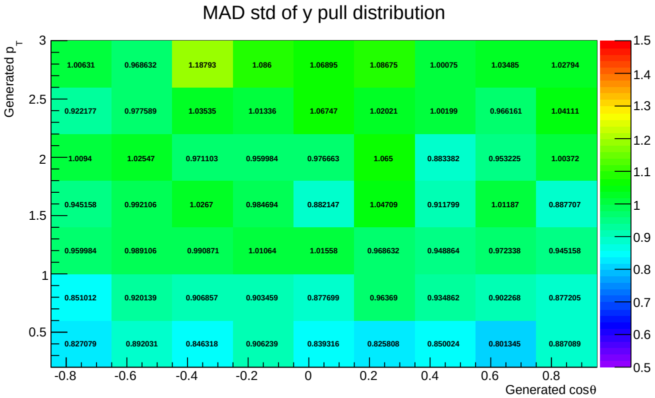
50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

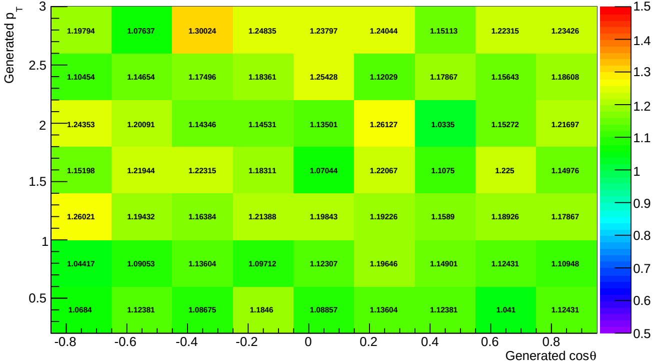
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

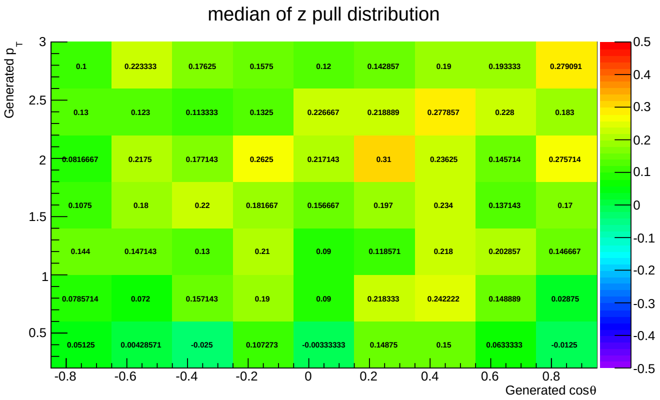
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,°$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,°$, pxd + svd + cdc.

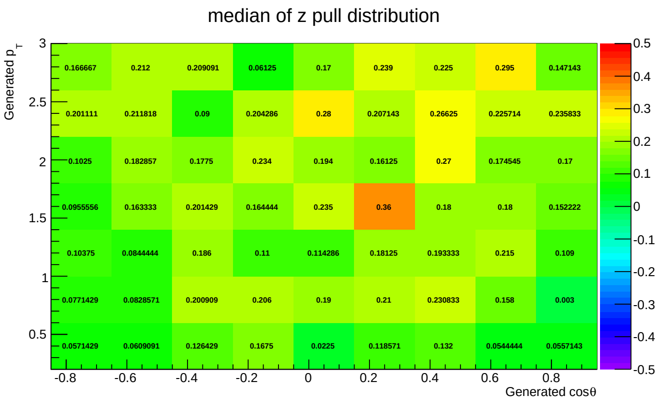


GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

Technische Universität München

50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,°$, pxd + svd + cdc.

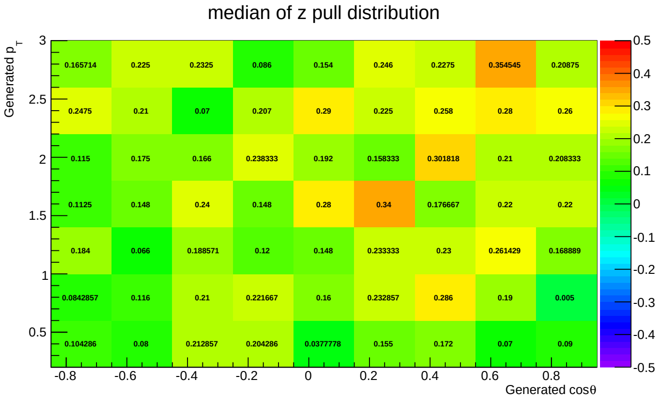


GENFIT 2 with ≥ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

Technische Universität München

50000 Pions, $(0.2$ to $3.0)\ \mathrm{GeV}$, $\theta = (17$ to $150)\ ^{\circ}$, pxd + svd + cdc.

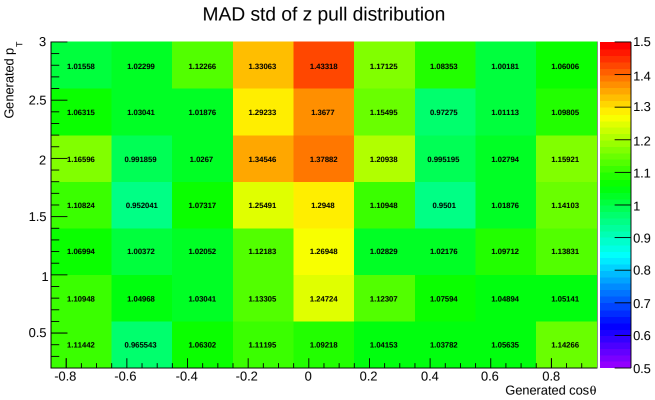


GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

Technische Universität München

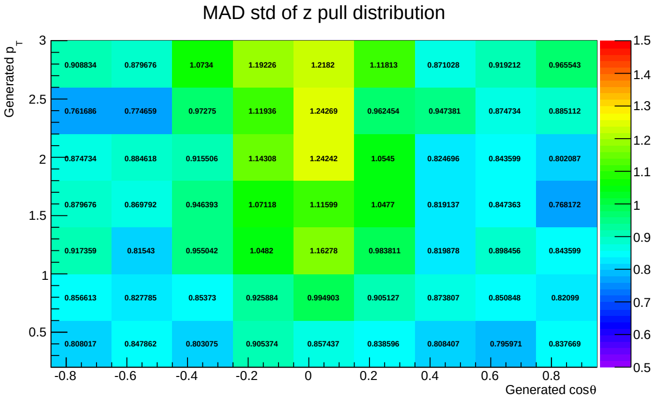
50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

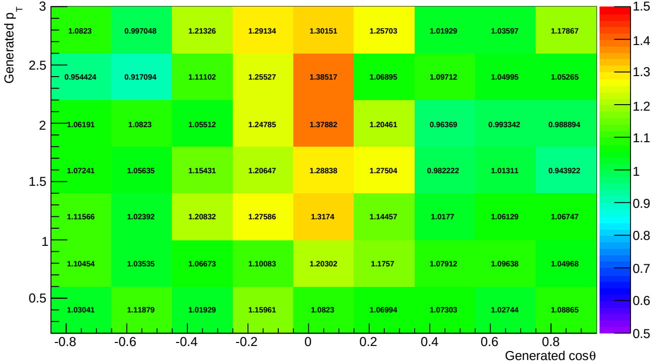
50000 Pions, $(0.2$ to $3.0)$ GeV, $\theta = (17$ to $150)\,^{\circ}$, pxd + svd + cdc.



GENFIT 2 with ≥ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

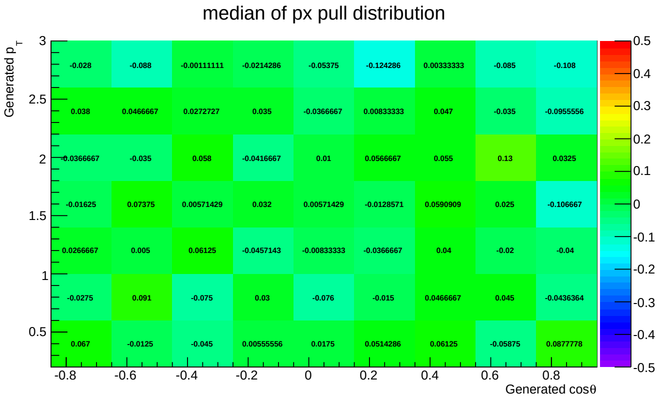
50000 Pions, $(0.2$ to $3.0)$ GeV, $\theta = (17$ to $150)$ $^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

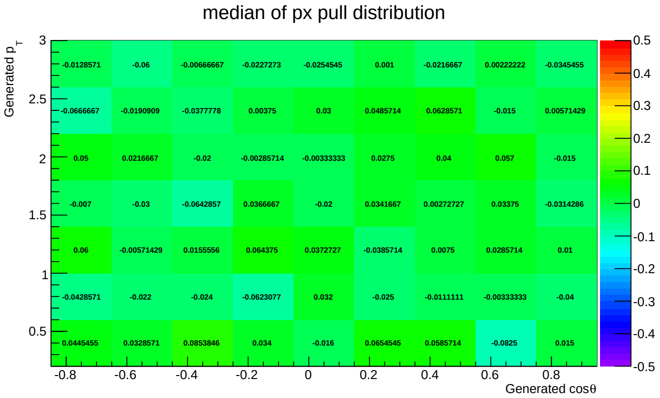
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^{\circ}$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

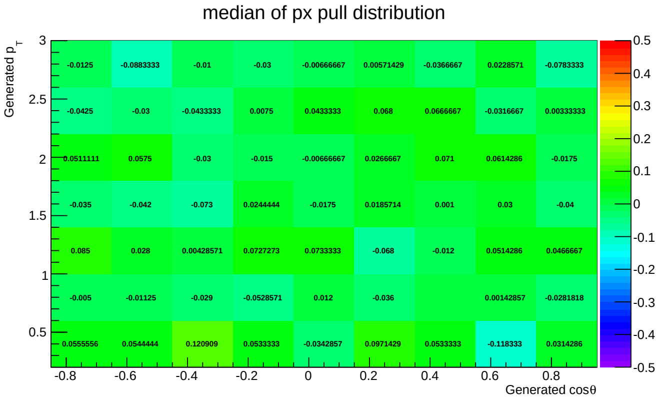
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with $\geq$ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

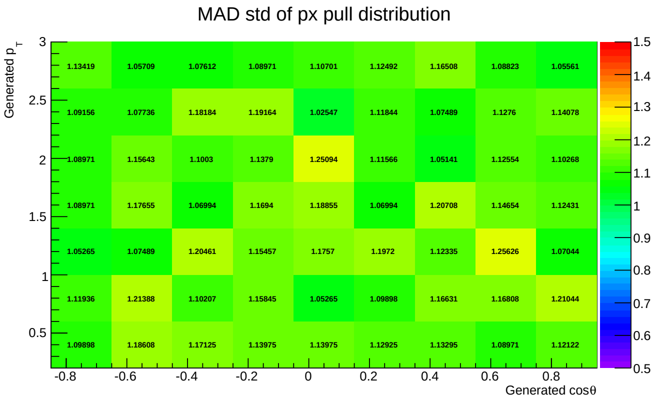
50000 Pions, $(0.2$ to $3.0)\ \mathrm{GeV}$, $\theta = (17$ to $150)\ ^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

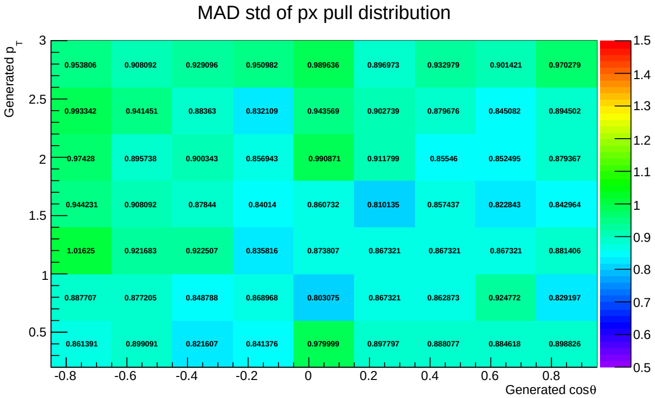
50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,°$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

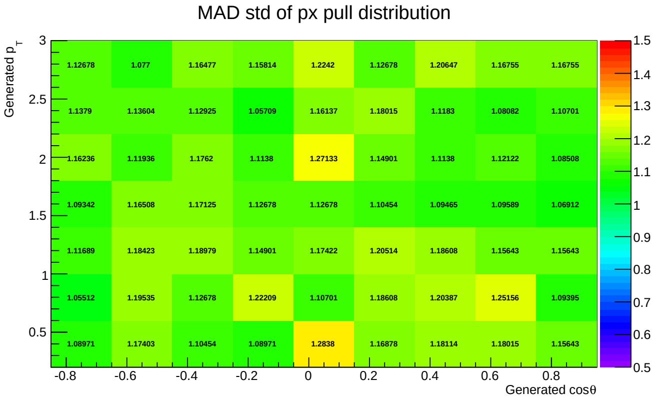
50000 Pions, $(0.2 \text{ to } 3.0)\,\text{GeV}$, $\theta = (17 \text{ to } 150)\,°$, pxd + svd + cdc.



GENFIT 2 with ≥ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

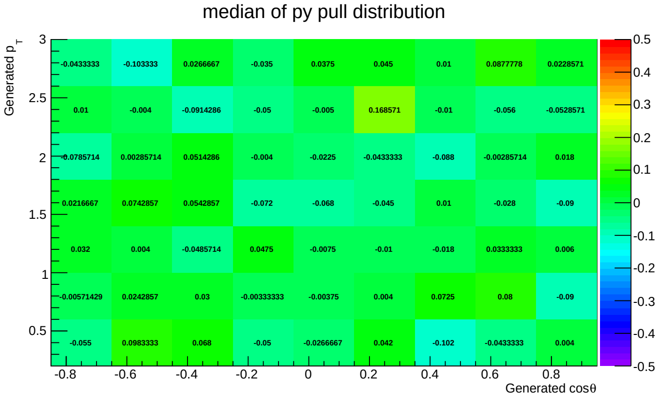
50000 Pions, $(0.2$ to $3.0)\ \mathrm{GeV}$, $\theta = (17$ to $150)\ ^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

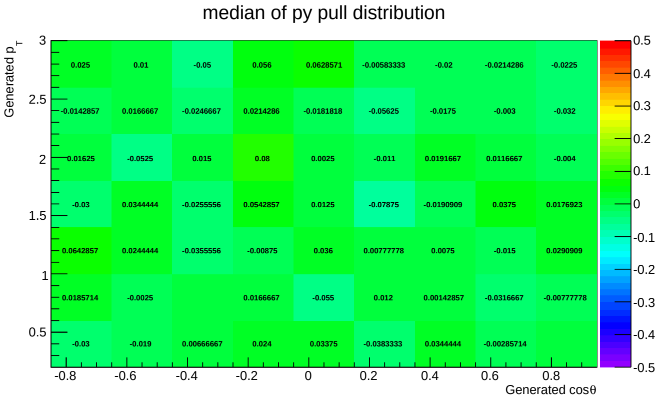
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

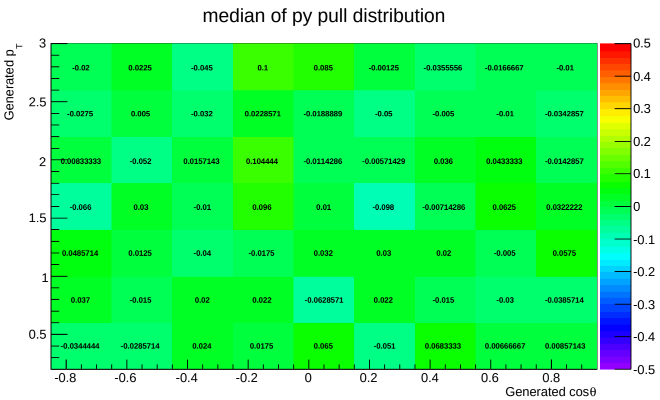
50000 Pions, $(0.2$ to $3.0)$ GeV, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.

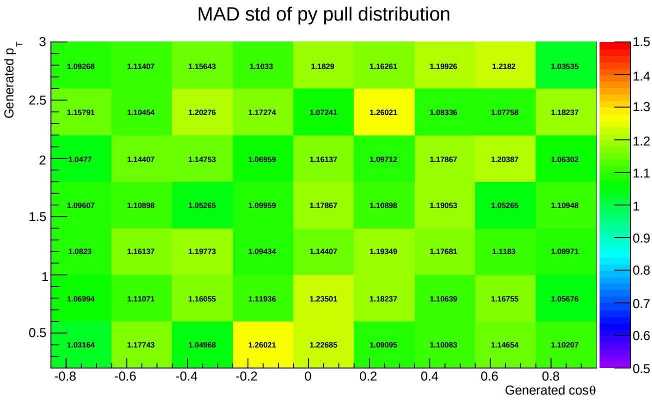


GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

Technische Universität München

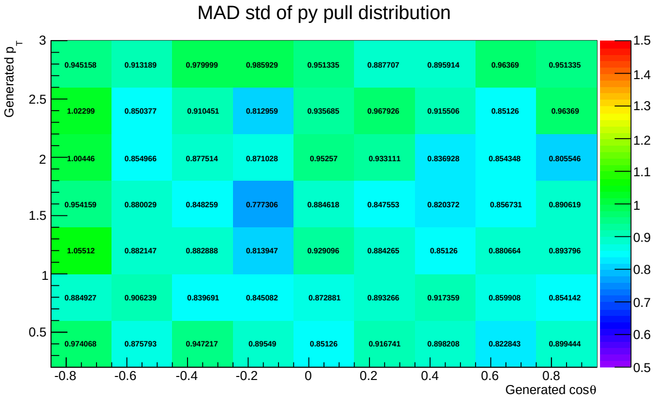
50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

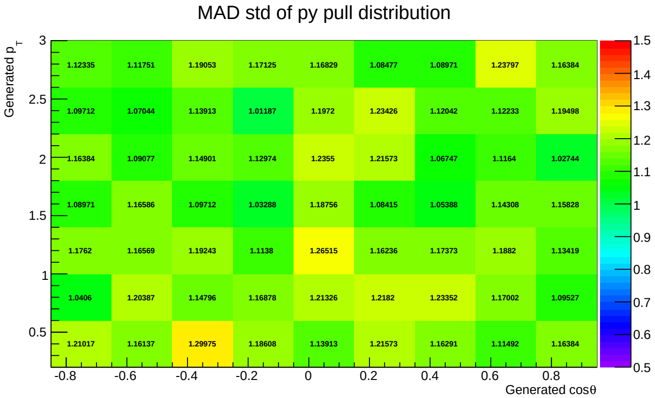
50000 Pions, $(0.2 \text{ to } 3.0)\,\text{GeV}$, $\theta = (17 \text{ to } 150)\,°$, pxd + svd + cdc.



GENFIT 2 with ≥ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

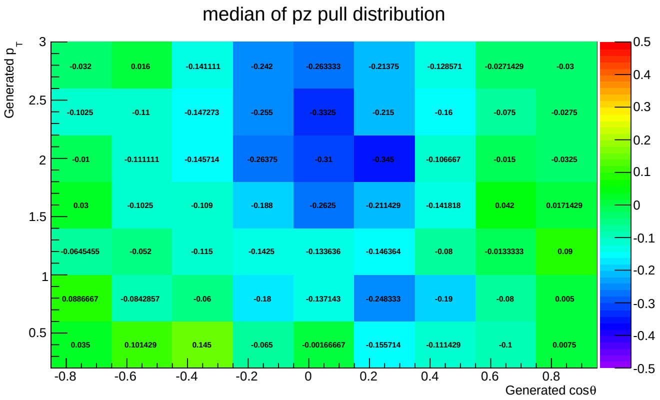
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

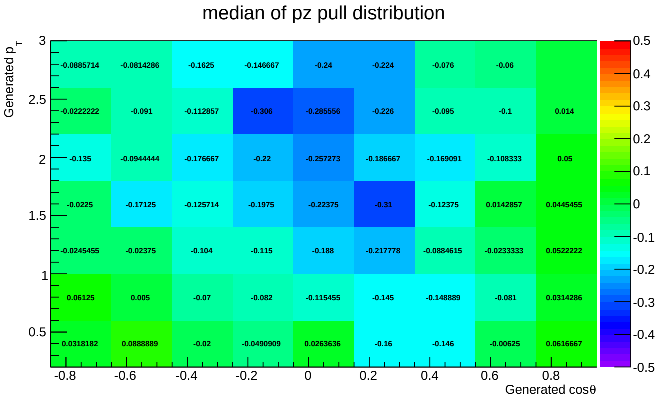
50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,°$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

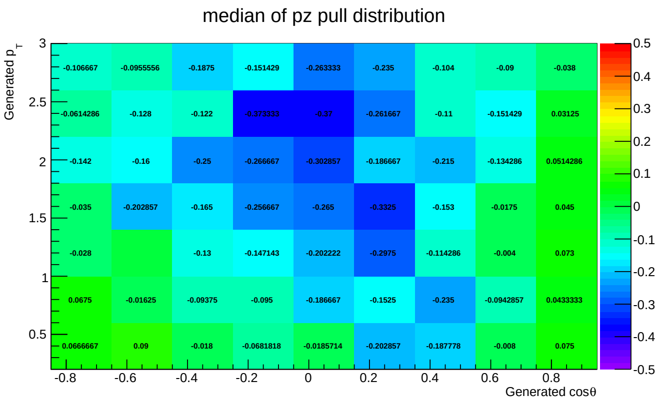
50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

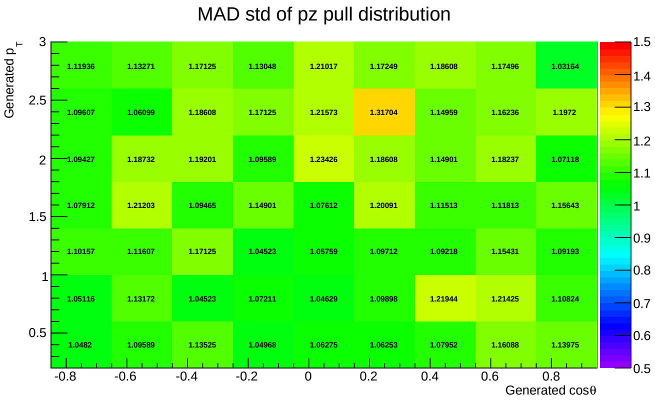
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

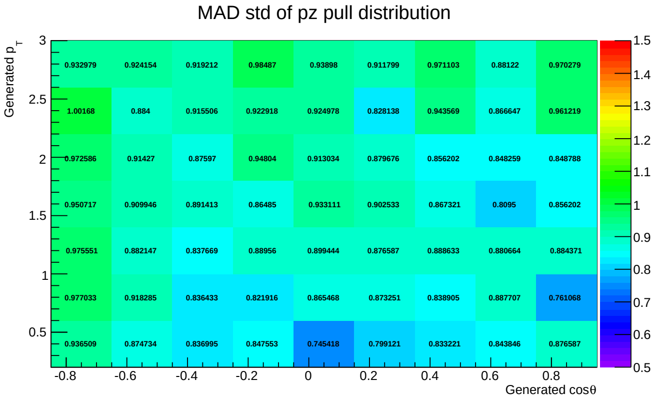
50000 Pions, $(0.2 \text{ to } 3.0)\,\mathrm{GeV}$, $\theta = (17 \text{ to } 150)\,^\circ$, pxd + svd + cdc.



GENFIT 1 with 3 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

50000 Pions, $(0.2 \text{ to } 3.0)$ GeV, $\theta = (17 \text{ to } 150)$ °, pxd + svd + cdc.

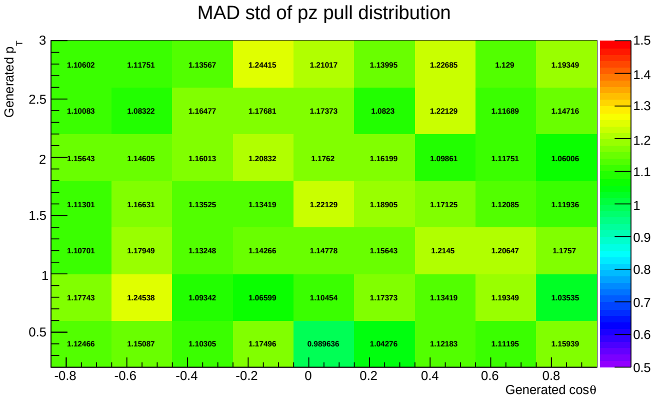


GENFIT 2 with $\geq$ 2 iterations.

GENFIT 1 vs. GENFIT 2: 2D QA Plots

Technische Universität München

50000 Pions, $(0.2$ to $3.0)\,\mathrm{GeV}$, $\theta = (17$ to $150)\,^\circ$, pxd + svd + cdc.



GENFIT 2 with ≥ 3 iterations.