RKTrackRep

A New Track Representation for GENFIT

June 2010 – Panda Meeting Stockholm

Johannes Rauch, Christian Höppner, Technische Universität München

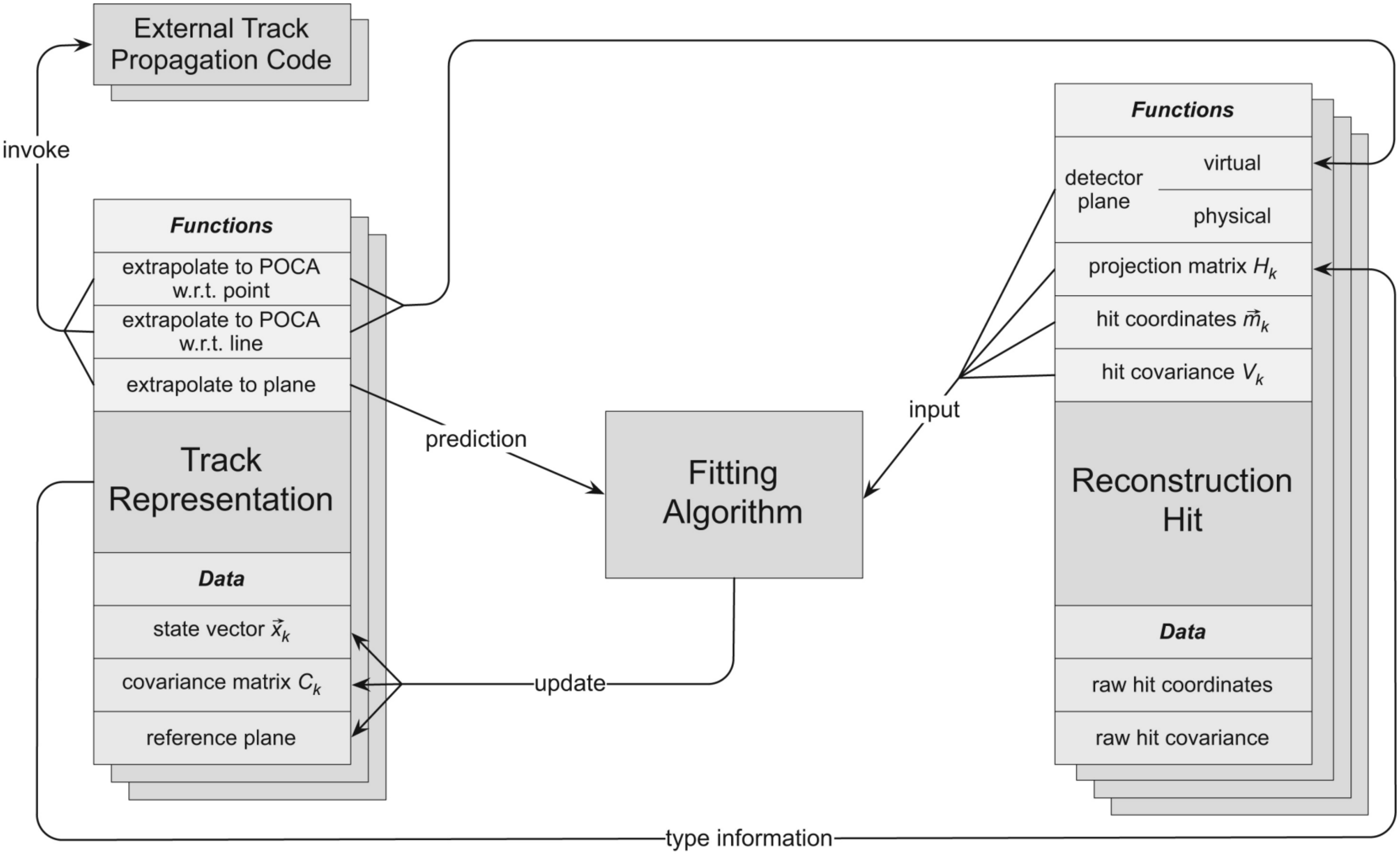
GENFIT

GENFIT *[NIM A doi:10.1016/j.nima.2010.03.136]* is the track fitting framework used in PandaROOT

Three modular components:

- Track fitting algorithms (currently Kalman filter)
- Reconstruction hits (inherit from `GFAbsRecoHit`)
  - Orientation and dimensionality of hits in planar detectors not restricted
  - Information from non-planar detectors (e.g. STT, TPC) used without simplifications
- Track representations

GENFIT – Structure



GENFIT – Track Representations

- Comprise track parameters, covariance matrix and track extrapolation functionality
- Fitting algorithms communicate via base class `GFAbsTrackRep`
- Several track representations can be fitted simultaneously
  - Different mass hypotheses with same track representation
  - Different track representations to compare parameterizations and/or track extrapolation codes
  - Different track representations for different regions of phase space

RKTrackRep – General Features

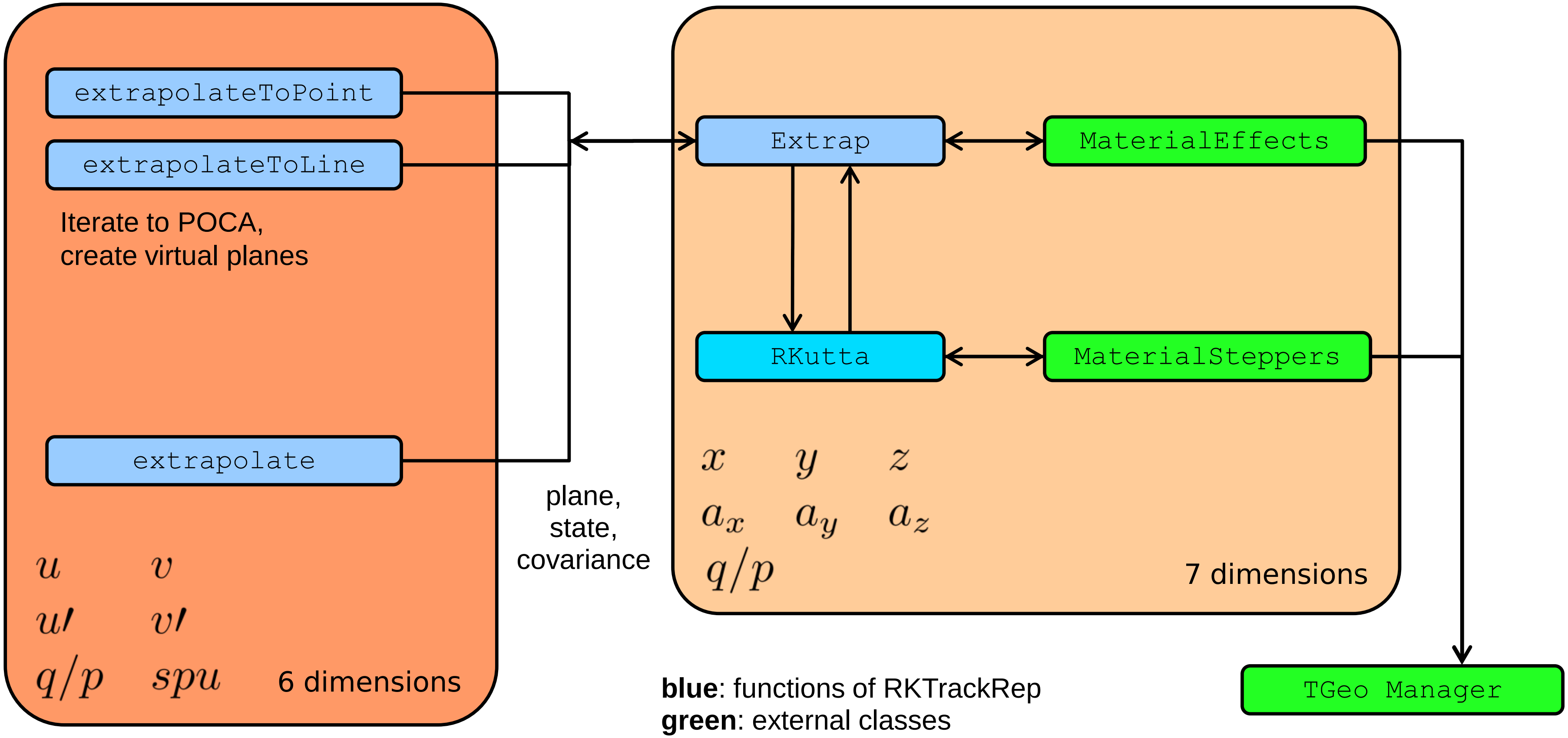
- Uses same track parametrization as GEANE
- Uses a **R**unge **K**utta solver to track particles
- No restriction in track direction

  tracks cannot be parallel to z-axis in `GeaneTrackRep`,

  when dip angle $\lambda = 90^\circ$ → division by $\cos\lambda = 0$ e.g. in calculation of multiple scattering covariance matrix (in SC system)

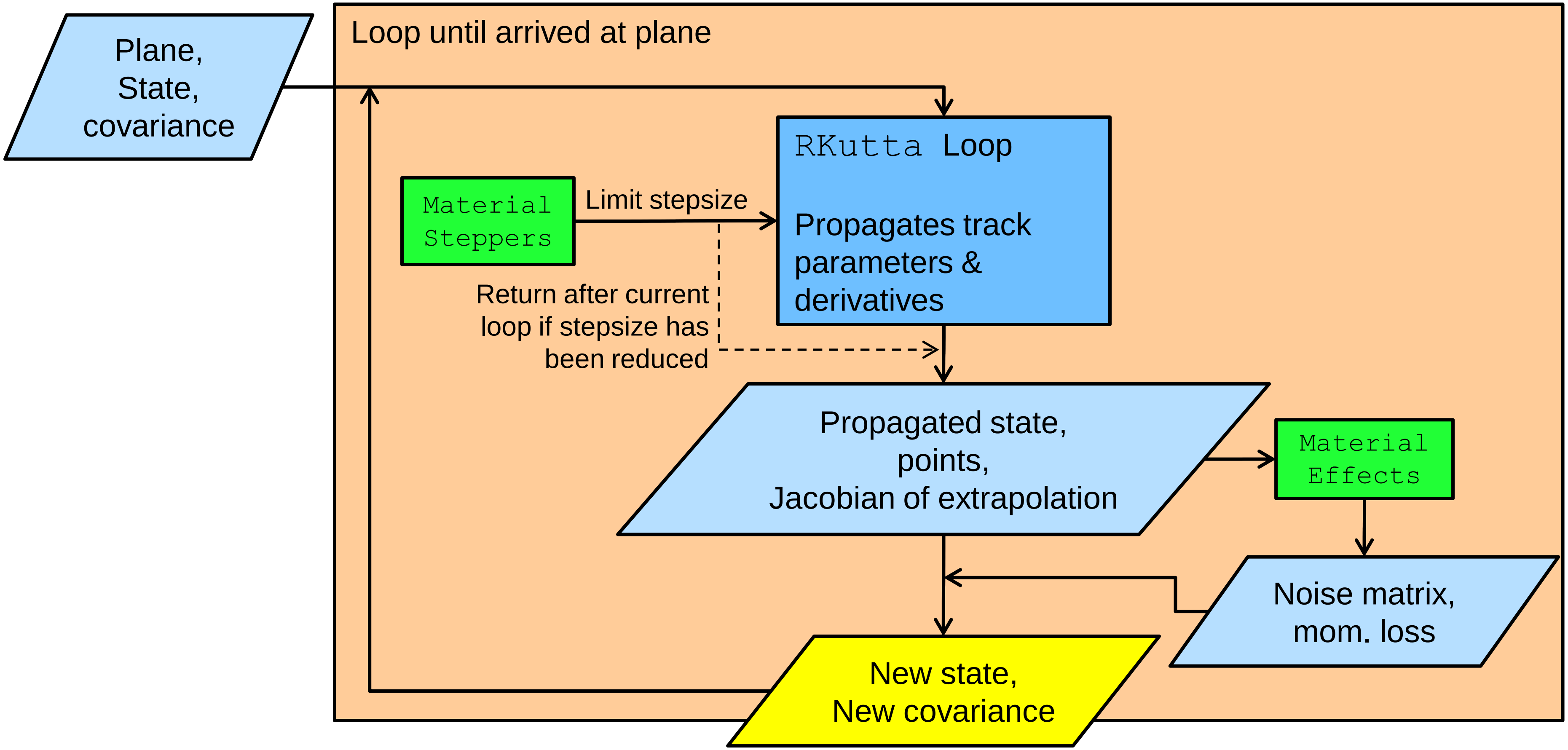
  (see A.Fontana et al., *Track following in dense media and inhomogeneous magnetic fields*, p. 23)
- Interface to geometry and materials via `TGeo` classes which is standard in PandaROOT
- Custom interface to magnetic field `FairField`

RKTrackRep – Structure



blue: functions of RKTrackRep
green: external classes

RKTrackRep – `Extrap`



RKTrackRep – Material Model

- Material parameters from `TGeo`
- dE/dx calculated by extra class `energyLoss`, uses Bethe Bloch formula from PDG booklet
- Momentum loss calculated from dE/dx, radiation length, mass and momentum
- Noise matrix (multiple scattering, energy loss straggling) ported from GEANE

- Currently working on implementation of bremsstrahlung (porting from GEANE)

RKutta – History of the code

- Originally based on **Runge-Kutta Method** of **Nystroem** from M.Abramowitz, I.A.Stegun, *Handbook of Mathematical Functions*, 25.5.20, National Bureau of Standards Applied Mathematics Series – 55, 1964
- Original **FORTRAN code *GRKUTA*** written by R.Brun, M.Hansroul, V.Perevoztchikov
- **Porting to C++**, new interface and Jacobian calculation by Igor Gavrilenko, ATLAS, CERN
- Minor modifications by Sergei Gerassimov, COMPASS, CERN & TU Muenchen

Runge Kutta Extrapolation Algorithm

- Force in magnetic field: $\vec{F} = q\,\vec{v} \times \vec{B}$
- Equation of motion in vacuum:

$$\frac{d\vec{p}}{dt} = \frac{d\left(m\gamma\frac{d\vec{x}}{dt}\right)}{dt} = c^2\kappa\, q\,\vec{v}(t) \times \vec{B}\left(\vec{x}\,(t)\right)$$

- Can be rewritten with geometrical quantities only:

$$\frac{d^2\vec{x}}{ds^2} = \kappa\frac{q}{p}\frac{d\vec{x}}{ds} \times \vec{B}\left(\vec{x}\,(s)\right)$$

direction $\vec{a}$ position $\vec{r}$

With Lorentz factor γ

Proportionality factor κ

Distance along trajectory s

RKutta – Extrapolation Algorithm

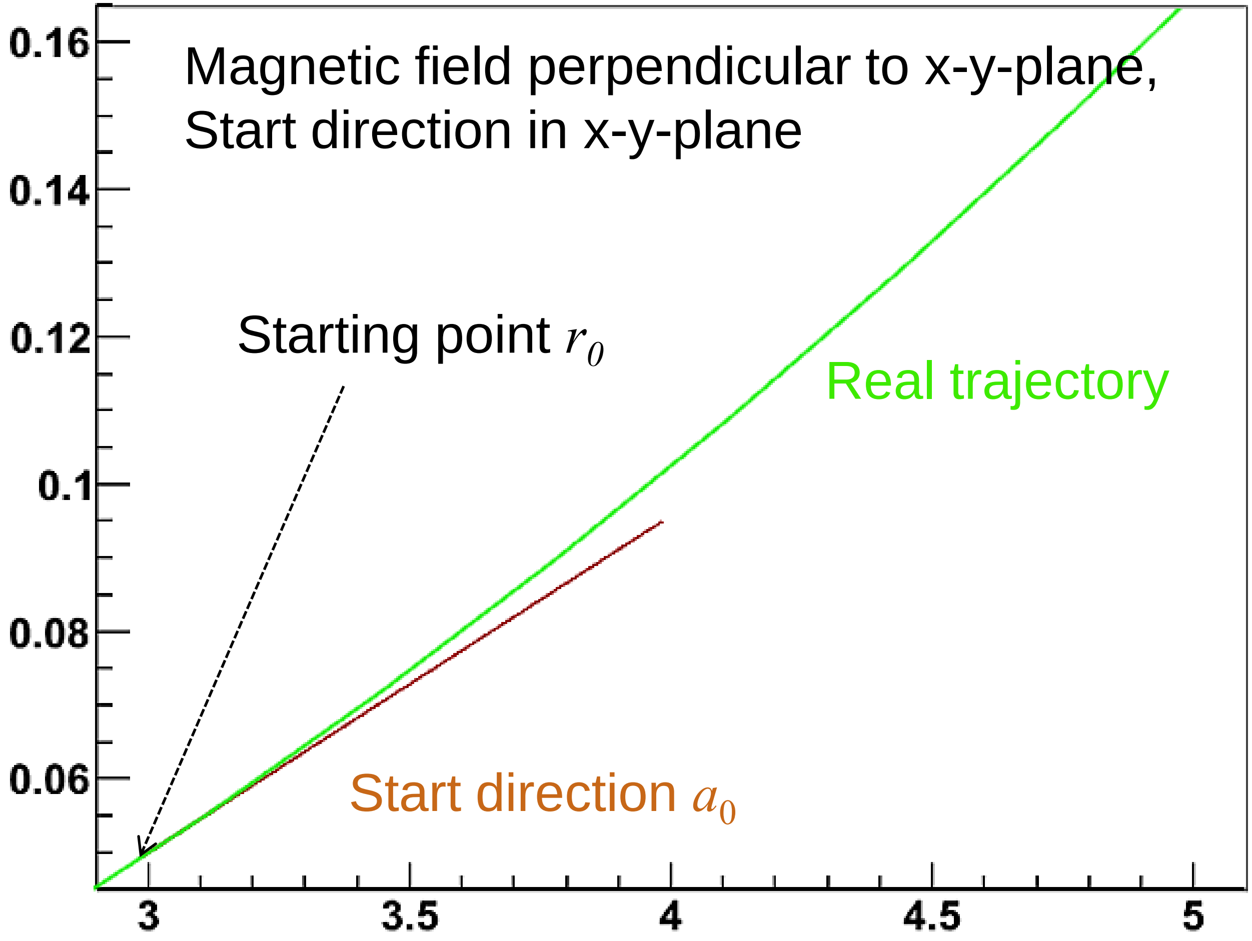
Quantities needed for the Runge Kutta algorithm

$\vec{r}_i$	coordinates at point i. ($\vec{r}_0$ start coordinates) [cm]
$\vec{a}_i$	normalized direction of momentum at $\vec{r}_i$
$\frac{q}{p}$	charge/momentum $\left[\frac{\mathrm{e}}{\mathrm{GeV/c}}\right]$
$\vec{B}_i$	magnetic field at $\vec{r}_i$ [kGs]

$$\vec{H}_i = \kappa \, \frac{q}{p} \, \vec{B}_i \cdot S$$

κ	proportionality factor $[0.149896229 \cdot 10^{-3}]$
S	Stepsize for Runge Kutta solver [cm]

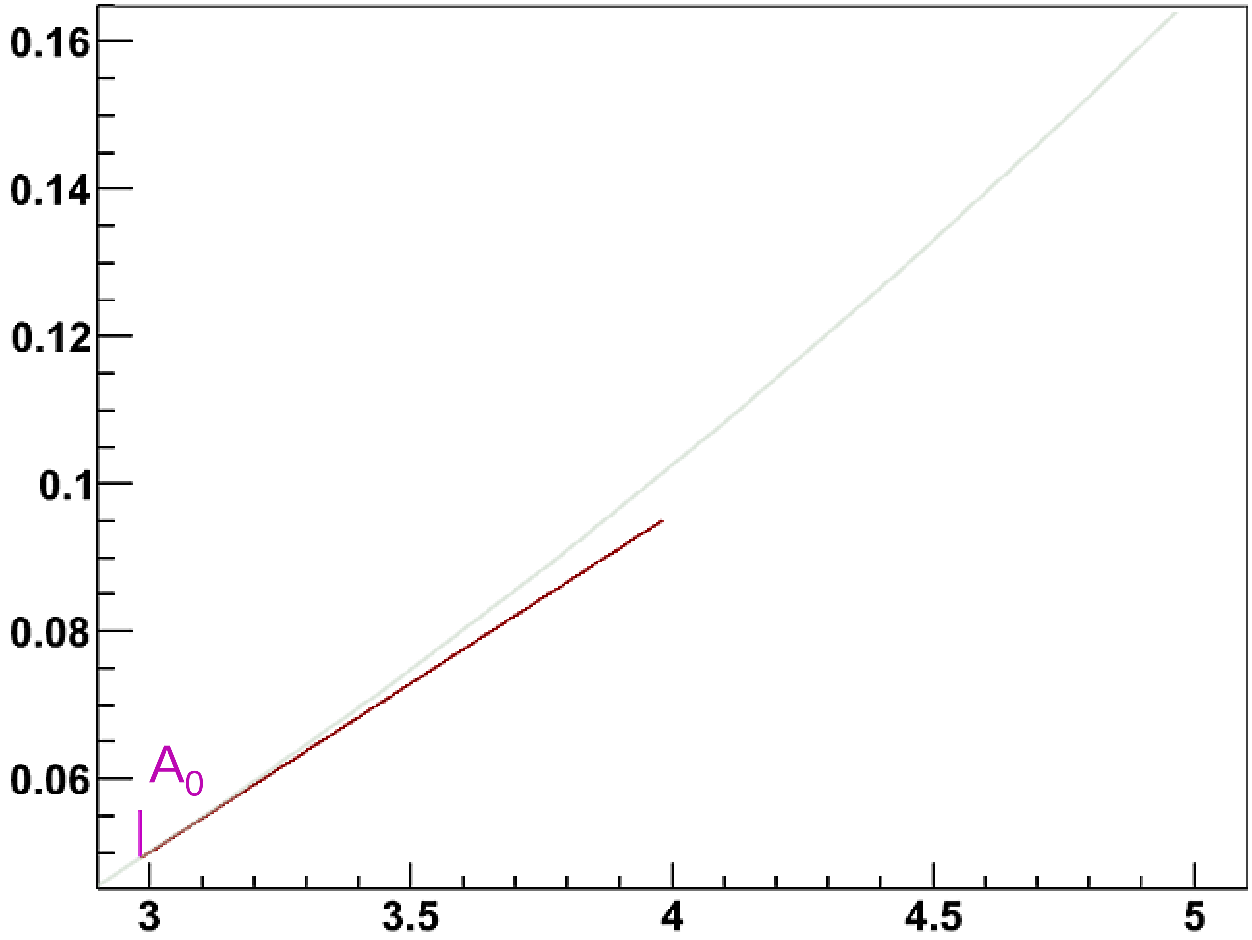
First point $(i = 0)$



First point $(i = 0)$

$\vec{A}_0 = \vec{a}_0 \times \vec{H}_0$

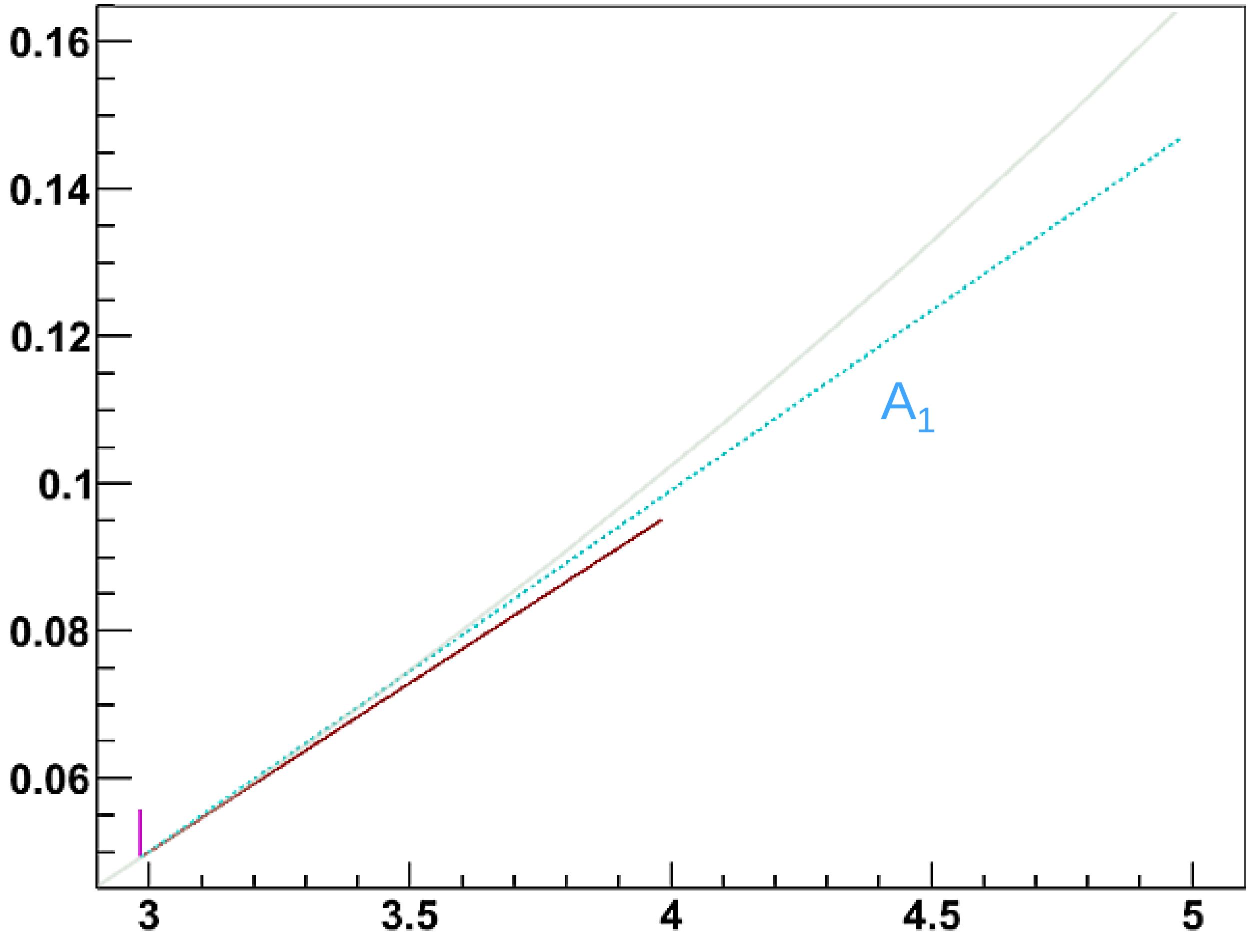
Equation of motion (right side multiplied with stepsize S)



First point $(i = 0)$

$\vec{A}_0 = \vec{a}_0 \times \vec{H}_0$

$\vec{A}_1 = \vec{A}_0 + 2\vec{a}_0$

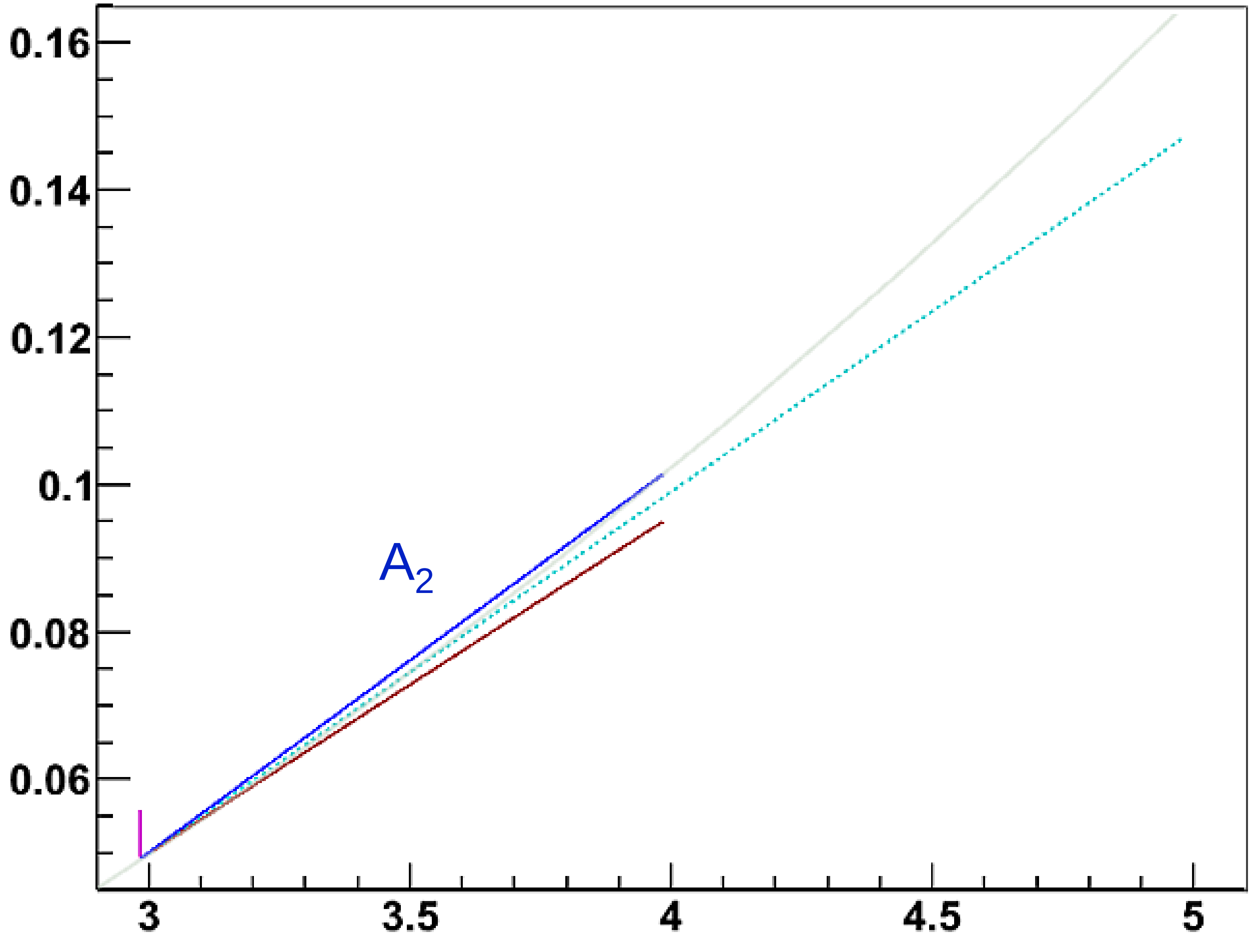


First point $(i = 0)$

$\vec{A}_0 = \vec{a}_0 \times \vec{H}_0$

$\vec{A}_1 = \vec{A}_0 + 2\vec{a}_0$

$\vec{A}_2 = \vec{A}_0 + \vec{a}_0$



First point $(i = 0)$

$$\vec{A}_0 = \vec{a}_0 \times \vec{H}_0$$

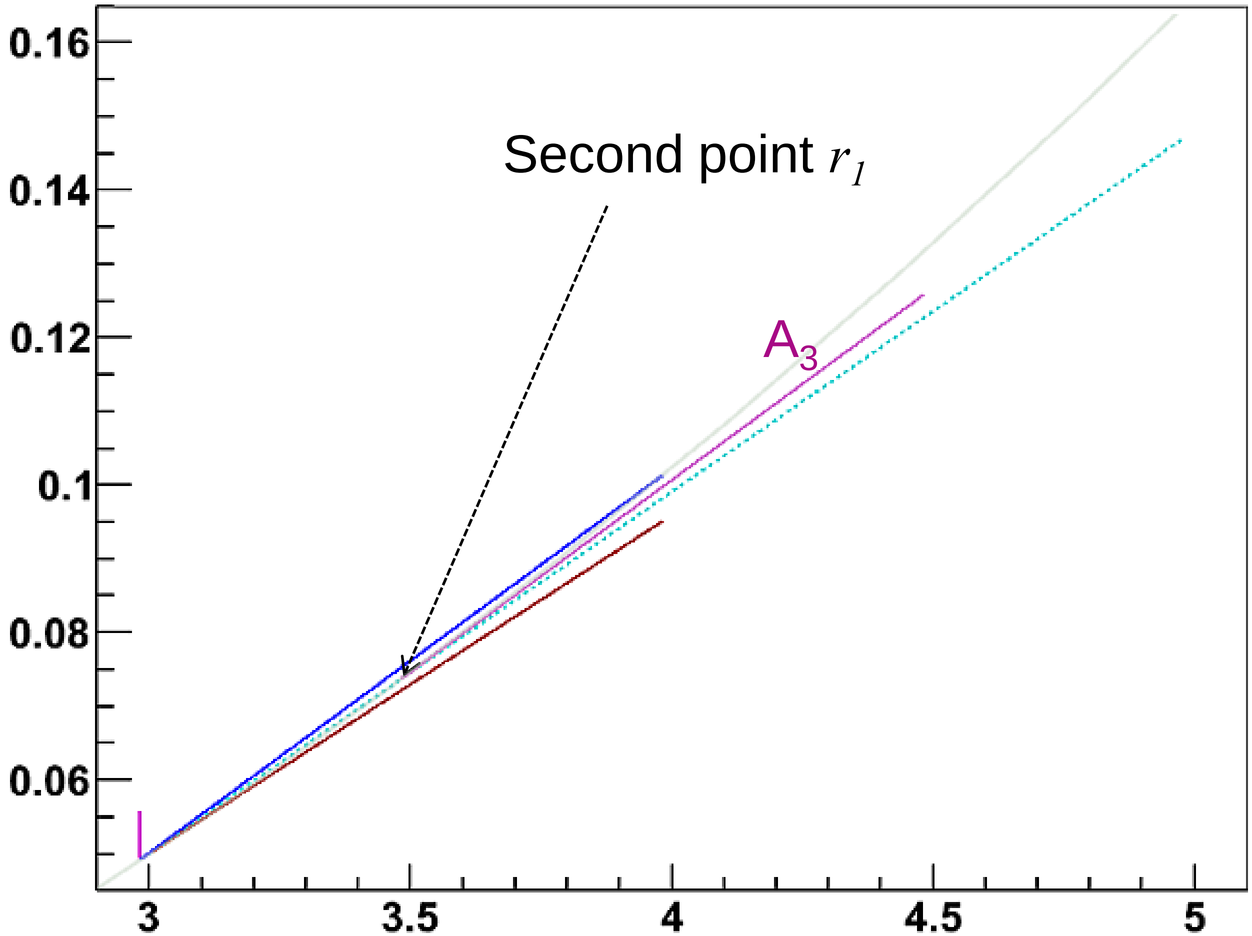
$$\vec{A}_1 = \vec{A}_0 + 2\vec{a}_0$$

$$\vec{A}_2 = \vec{A}_0 + \vec{a}_0$$

Second point $(i = 1)$

$$\vec{r}_1 = \vec{r}_0 + \frac{1}{4} S \cdot \vec{A}_1$$

$$\vec{A}_3 = \vec{A}_2 \times \vec{H}_1 + \vec{a}_0$$



First point $(i = 0)$

$$\vec{A}_0 = \vec{a}_0 \times \vec{H}_0$$

$$\vec{A}_1 = \vec{A}_0 + 2\vec{a}_0$$

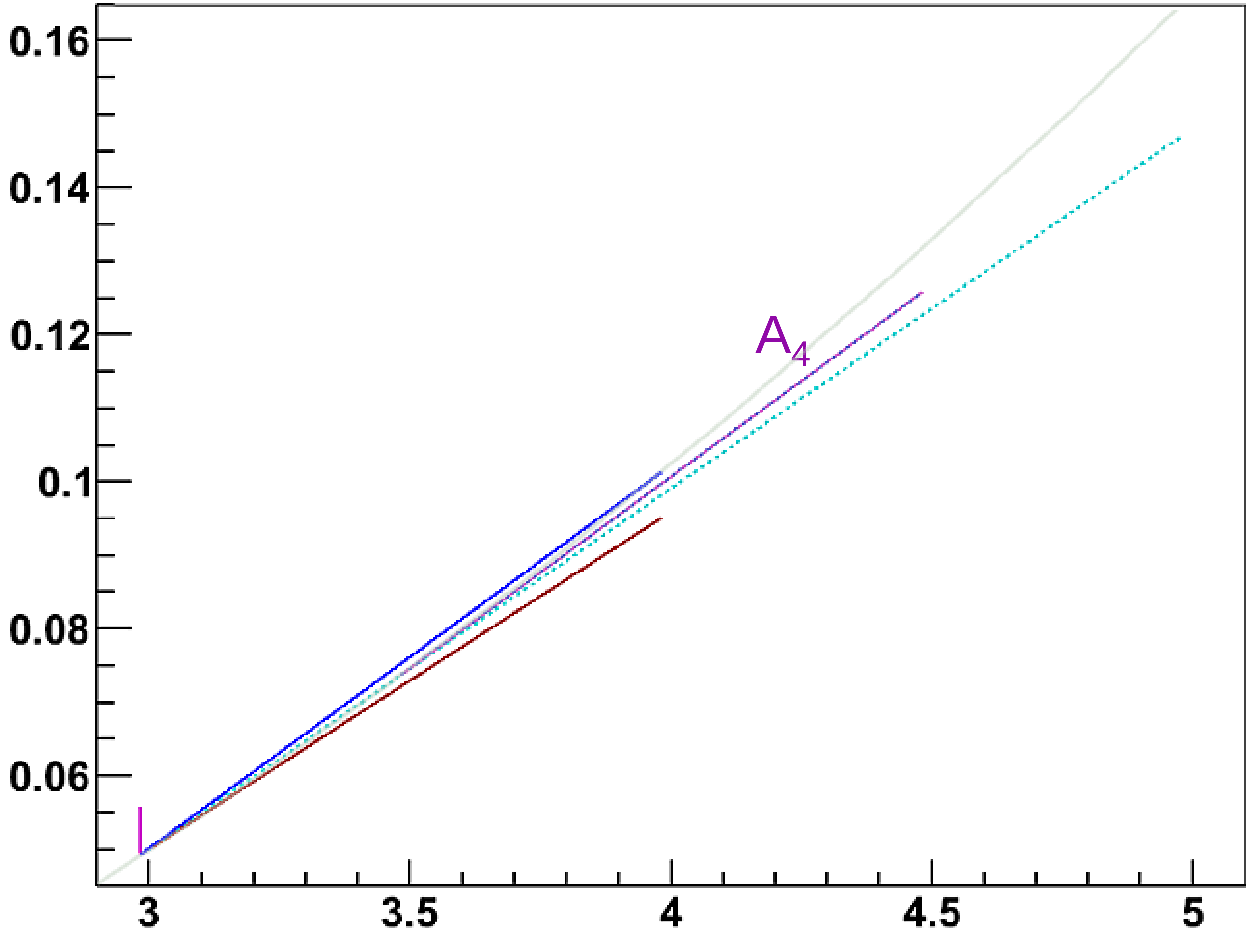
$$\vec{A}_2 = \vec{A}_0 + \vec{a}_0$$

Second point $(i = 1)$

$$\vec{r}_1 = \vec{r}_0 + \frac{1}{4} S \cdot \vec{A}_1$$

$$\vec{A}_3 = \vec{A}_2 \times \vec{H}_1 + \vec{a}_0$$

$$\vec{A}_4 = \vec{A}_3 \times \vec{H}_1 + \vec{a}_0$$

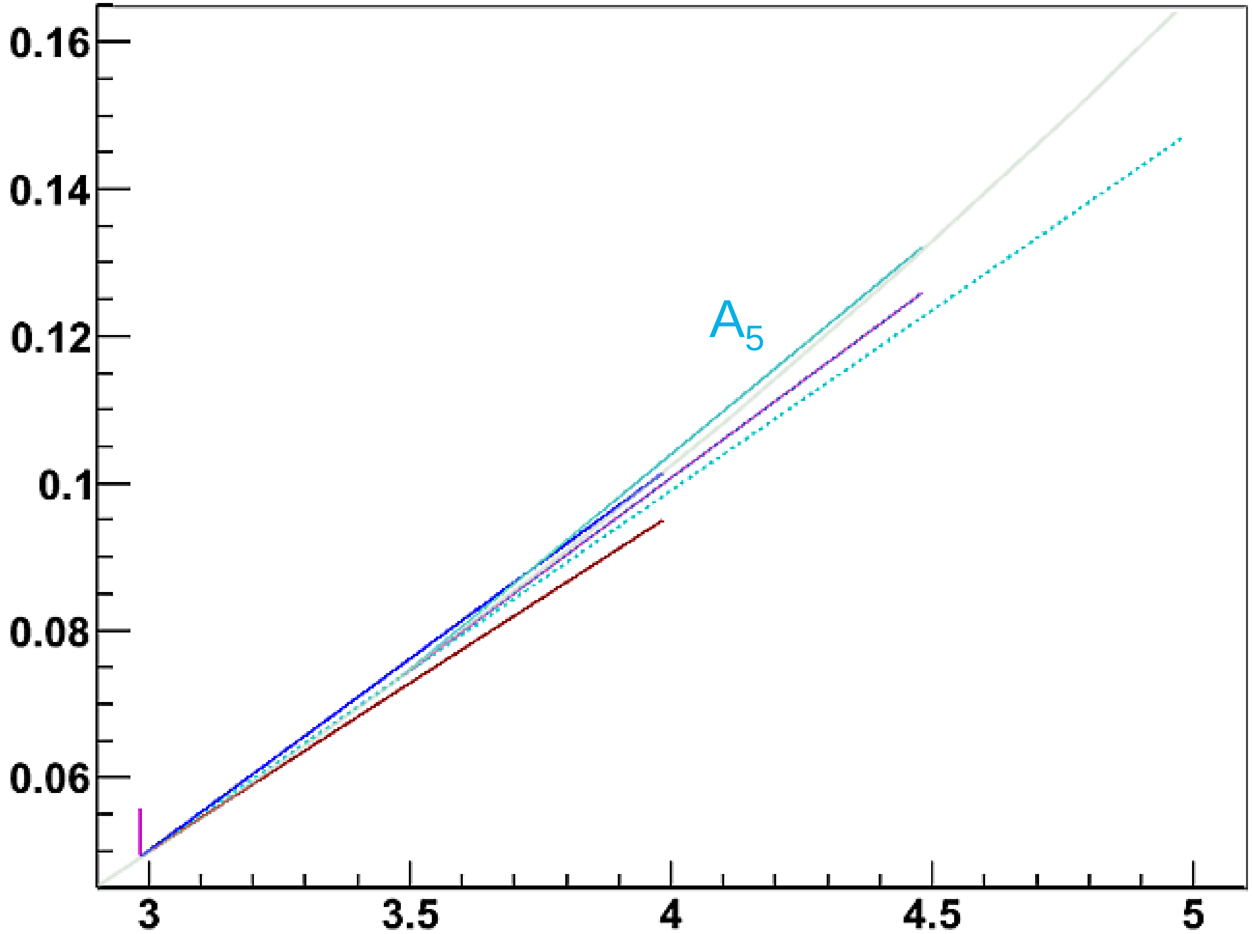


First point $(i = 0)$

$$\vec{A}_0 = \vec{a}_0 \times \vec{H}_0$$
$$\vec{A}_1 = \vec{A}_0 + 2\vec{a}_0$$
$$\vec{A}_2 = \vec{A}_0 + \vec{a}_0$$

Second point $(i = 1)$

$$\vec{r}_1 = \vec{r}_0 + \tfrac{1}{4}S \cdot \vec{A}_1$$
$$\vec{A}_3 = \vec{A}_2 \times \vec{H}_1 + \vec{a}_0$$
$$\vec{A}_4 = \vec{A}_3 \times \vec{H}_1 + \vec{a}_0$$
$$\vec{A}_5 = 2\vec{A}_4 - \vec{a}_0$$



First point $(i = 0)$

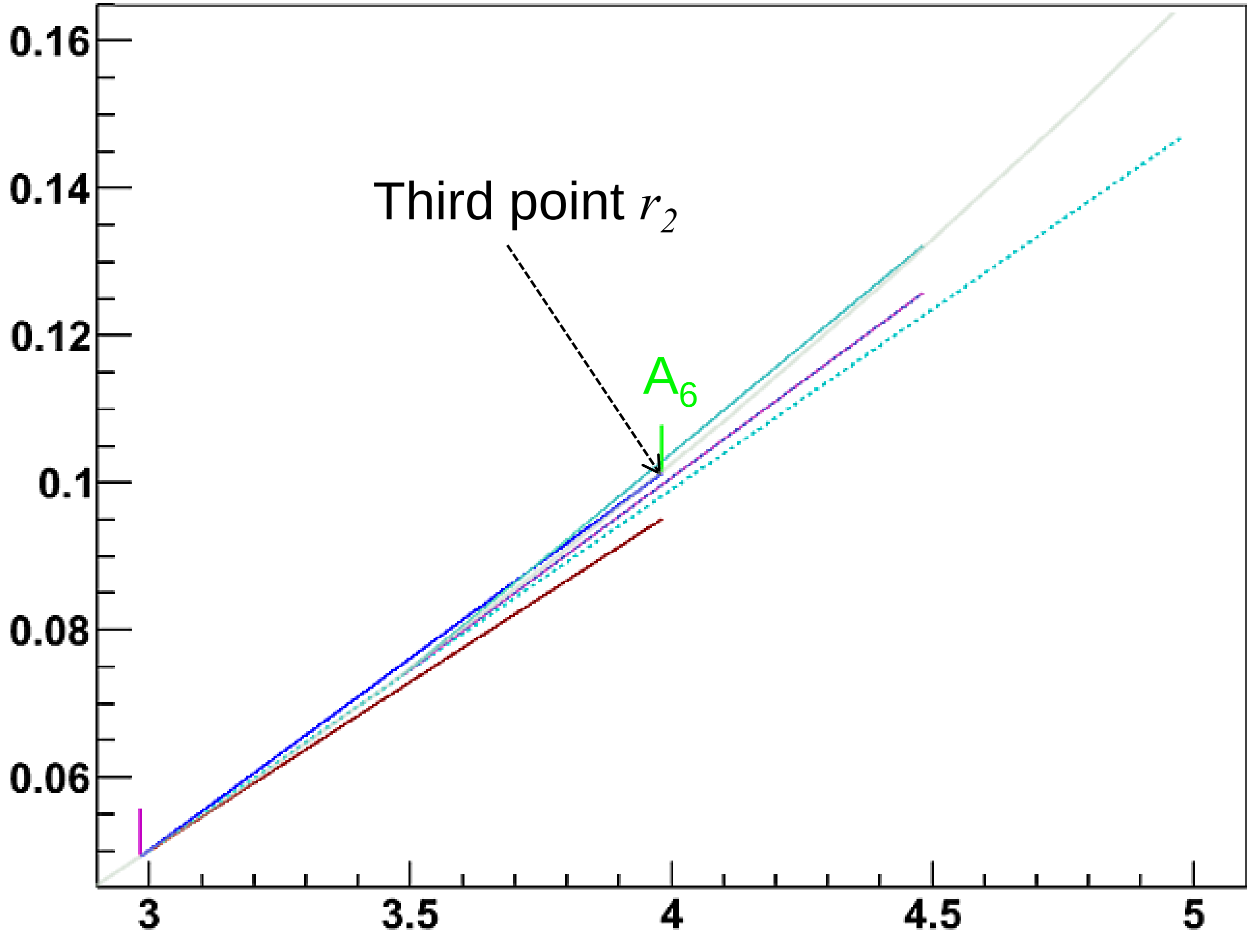
$$\vec{A}_0 = \vec{a}_0 \times \vec{H}_0$$
$$\vec{A}_1 = \vec{A}_0 + 2\vec{a}_0$$
$$\vec{A}_2 = \vec{A}_0 + \vec{a}_0$$

Second point $(i = 1)$

$$\vec{r}_1 = \vec{r}_0 + \frac{1}{4} S \cdot \vec{A}_1$$
$$\vec{A}_3 = \vec{A}_2 \times \vec{H}_1 + \vec{a}_0$$
$$\vec{A}_4 = \vec{A}_3 \times \vec{H}_1 + \vec{a}_0$$
$$\vec{A}_5 = 2\vec{A}_4 - \vec{a}_0$$

Third point $(i = 2)$

$$\vec{r}_2 = \vec{r}_0 + S \cdot \vec{A}_4$$
$$\vec{A}_6 = \vec{A}_5 \times \vec{H}_2$$



First point $(i = 0)$

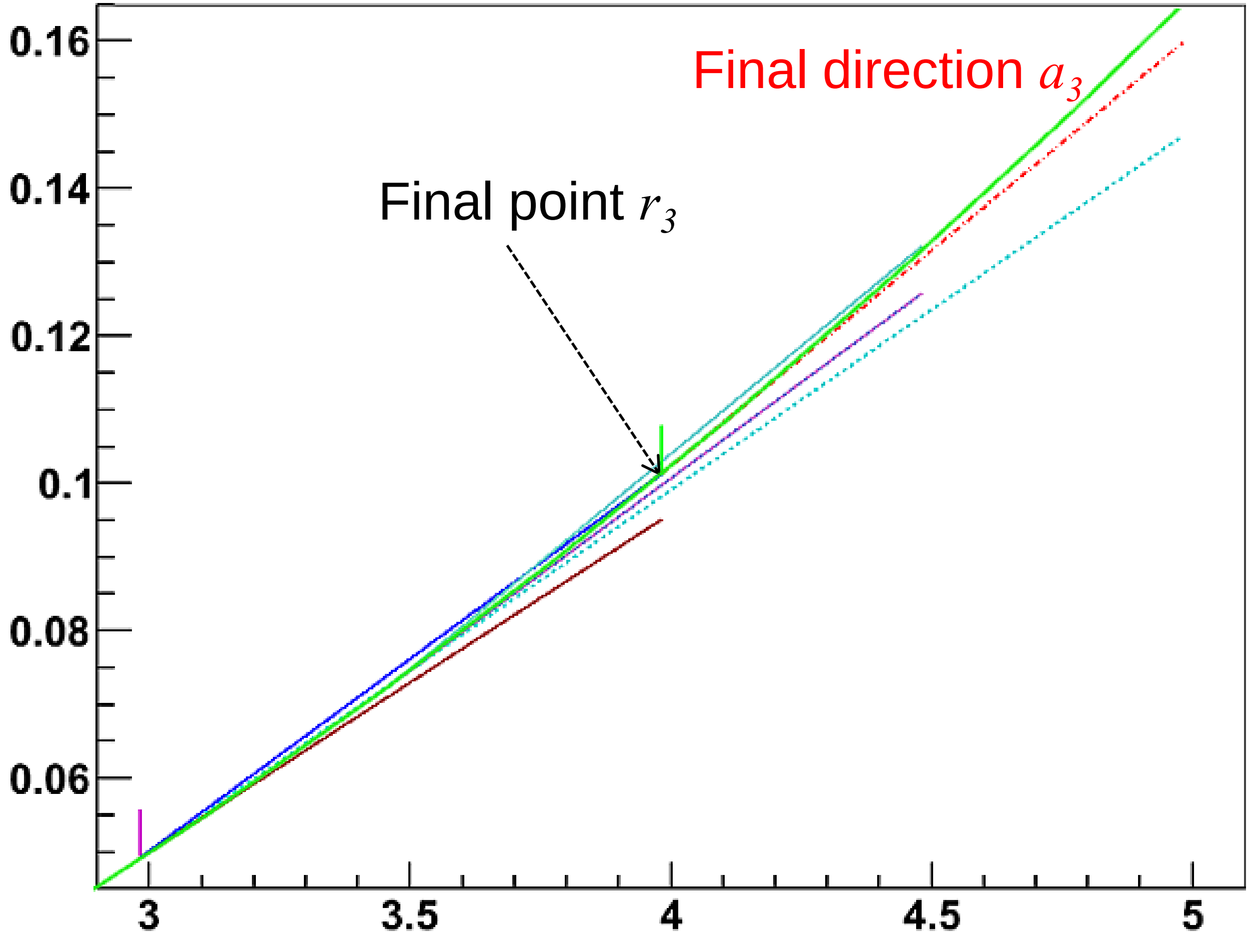
$$\vec{A}_0 = \vec{a}_0 \times \vec{H}_0$$
$$\vec{A}_1 = \vec{A}_0 + 2\vec{a}_0$$
$$\vec{A}_2 = \vec{A}_0 + \vec{a}_0$$

Second point $(i = 1)$

$$\vec{r}_1 = \vec{r}_0 + \tfrac{1}{4} S \cdot \vec{A}_1$$
$$\vec{A}_3 = \vec{A}_2 \times \vec{H}_1 + \vec{a}_0$$
$$\vec{A}_4 = \vec{A}_3 \times \vec{H}_1 + \vec{a}_0$$
$$\vec{A}_5 = 2\vec{A}_4 - \vec{a}_0$$

Third point $(i = 2)$

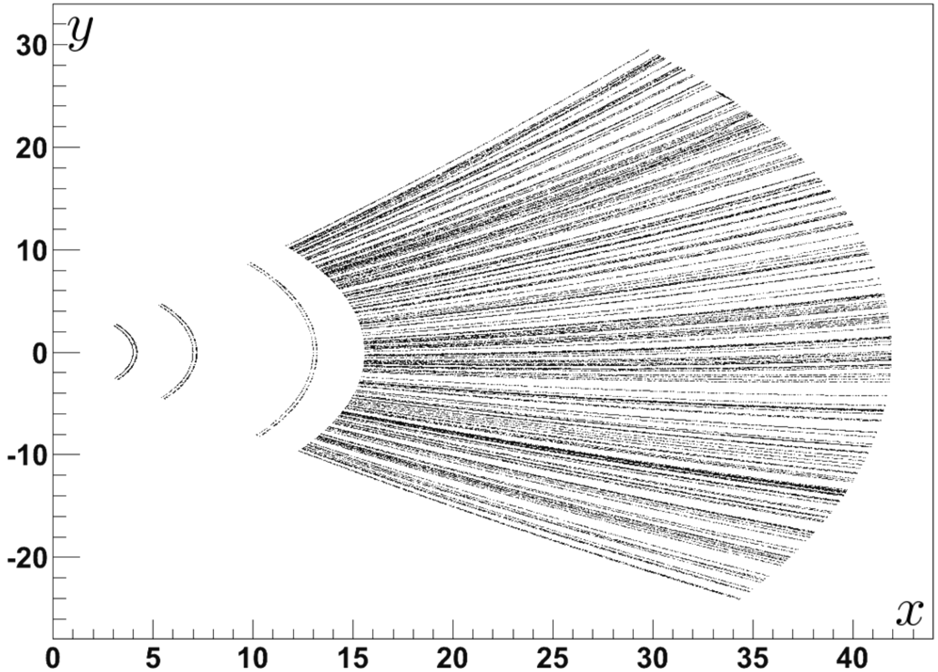
$$\vec{r}_2 = \vec{r}_0 + S \cdot \vec{A}_4$$
$$\vec{A}_6 = \vec{A}_5 \times \vec{H}_2$$



$$\vec{r}_3 = \vec{r}_0 + \tfrac{1}{3} S \left(\vec{A}_2 + \vec{A}_3 + \vec{A}_4\right)$$

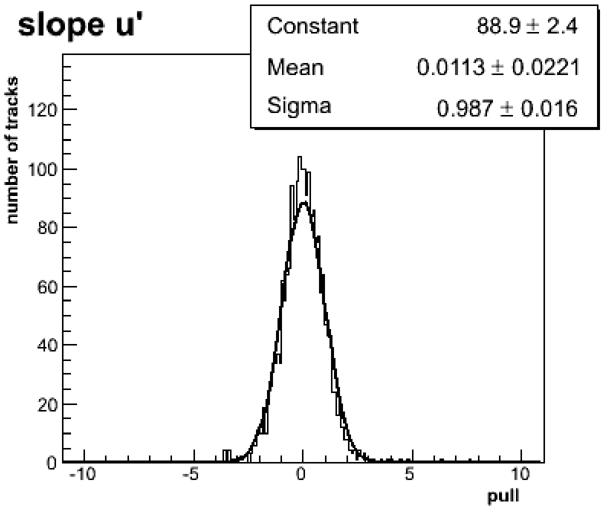
$$\vec{a}_3 = \left(\vec{A}_0 + 2\vec{A}_3 + \vec{A}_5 + \vec{A}_6\right) / \left|\vec{A}_0 + 2\vec{A}_3 + \vec{A}_5 + \vec{A}_6\right|$$

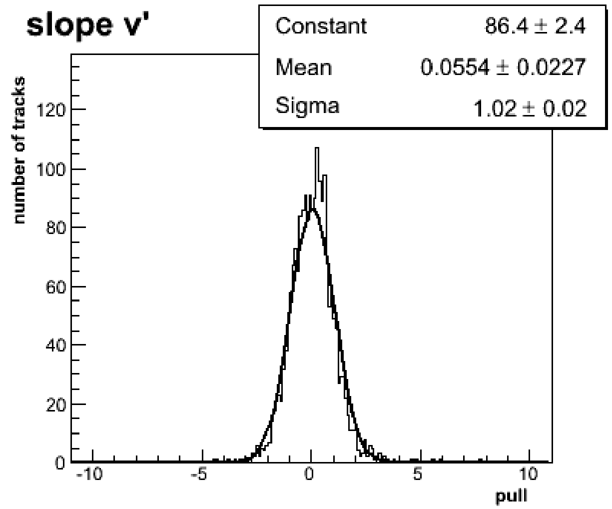
Performance study of RKTrackRep and comparison to GeaneTrackRep

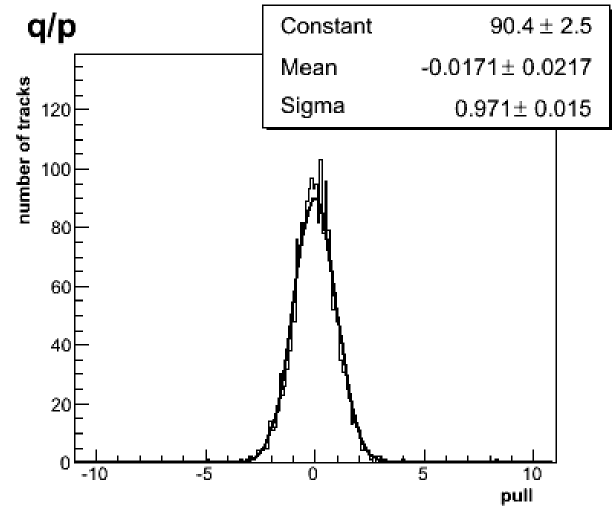


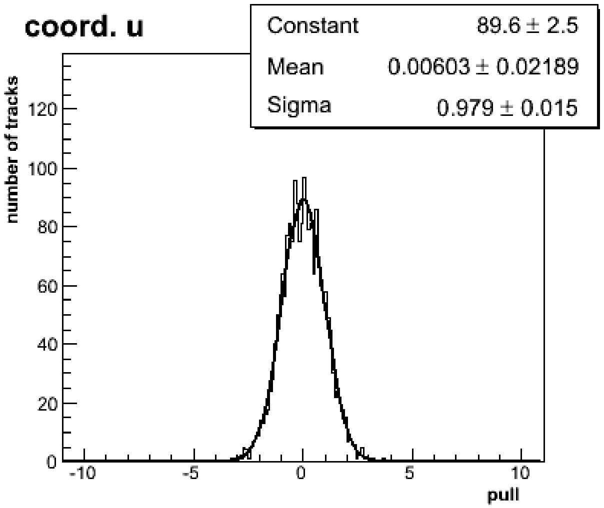
- Monte Carlo Simulation
  - 2000 particles/simulation
  - Theta=90°
  - 2mm Si + TPC gas (~2% radiation length)
- Fitted with GENFIT (RKTrackRep and GeaneTrackRep)
- Quality of fitting tested with **pull**:
  - $\mathrm{pull}(x) = \frac{x_{\mathrm{fit}} - x_{\mathrm{true}}}{\sigma_{x,\ \mathrm{fit}}}$
  - For a perfect fit pull distribution is a gaussian centered at 0 with width 1

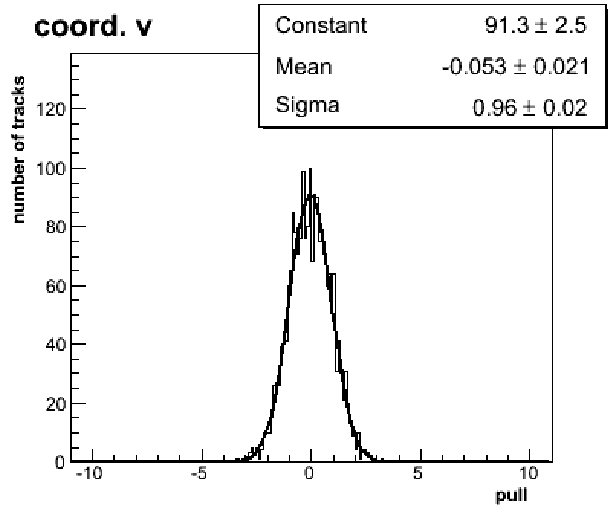
RK 0.5 GeV μ^- *efficiency* 99.95 %

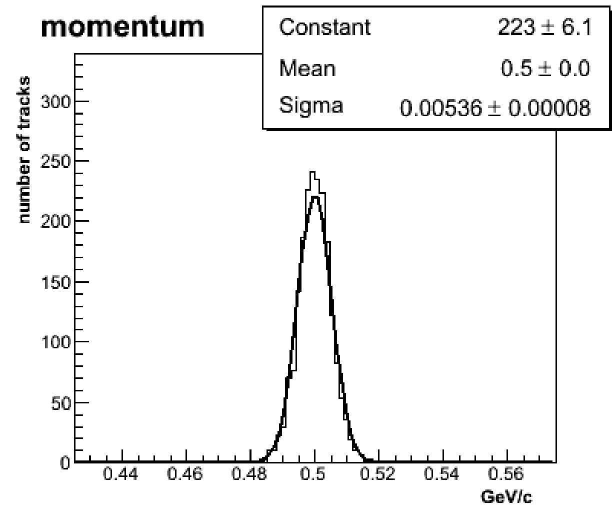




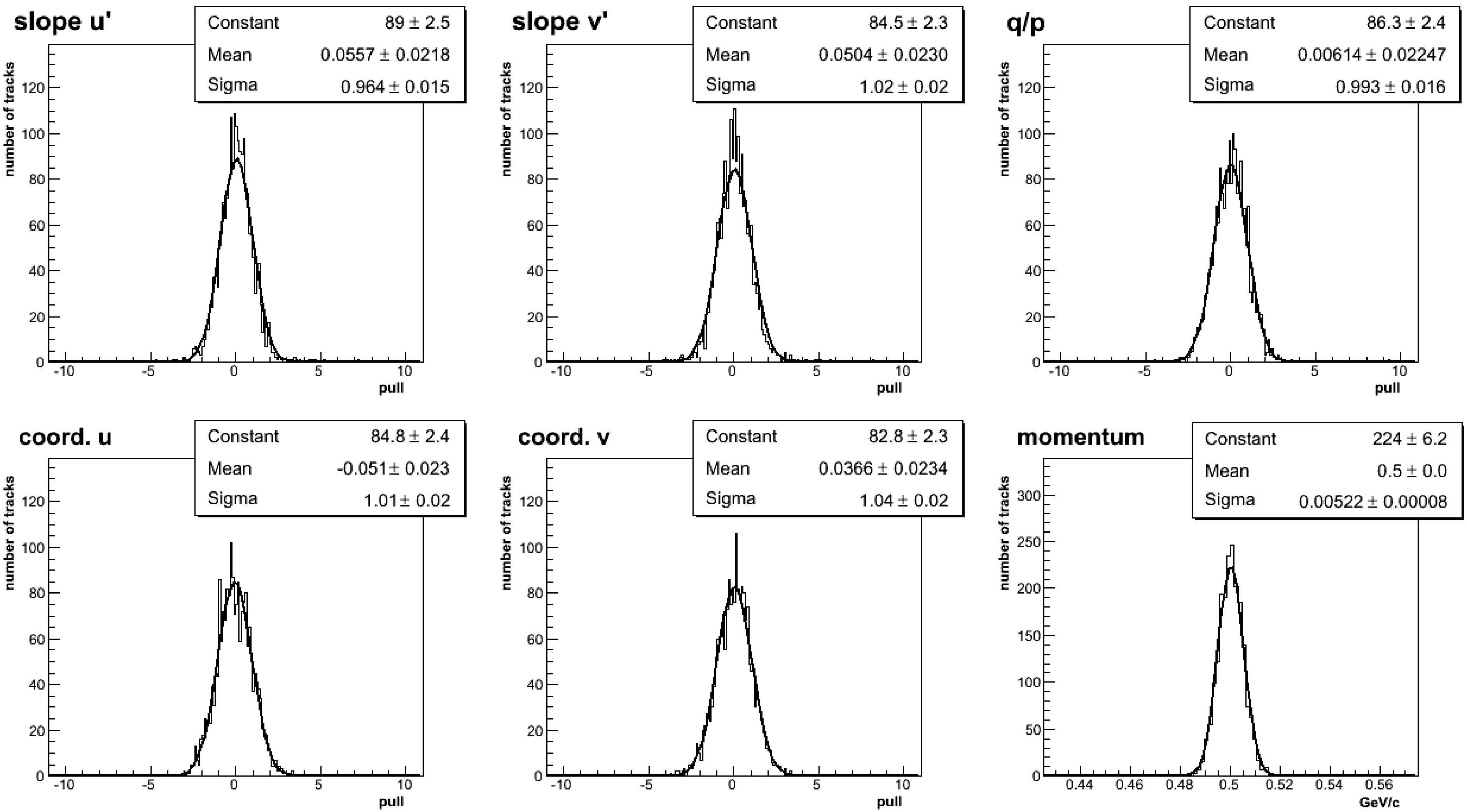




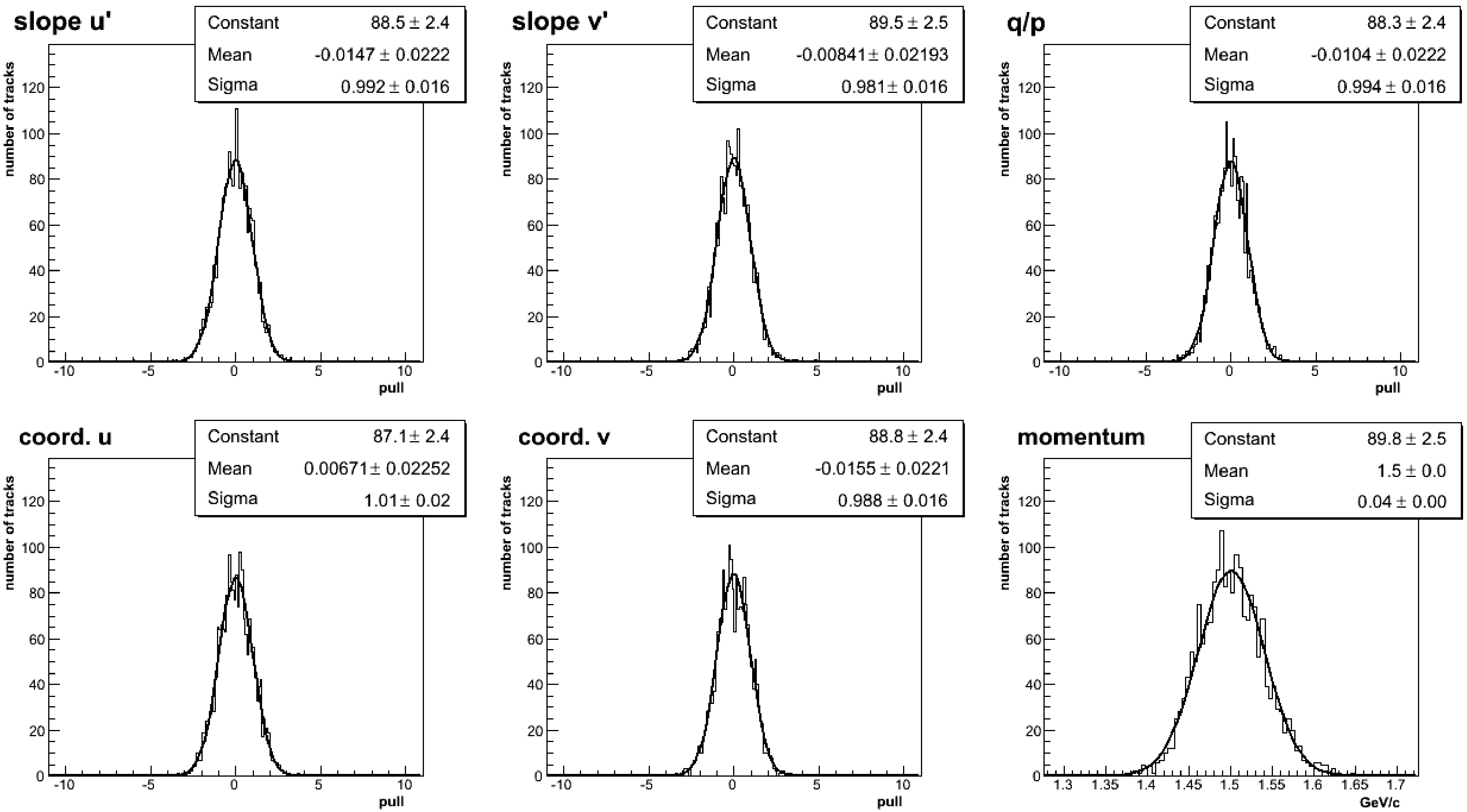




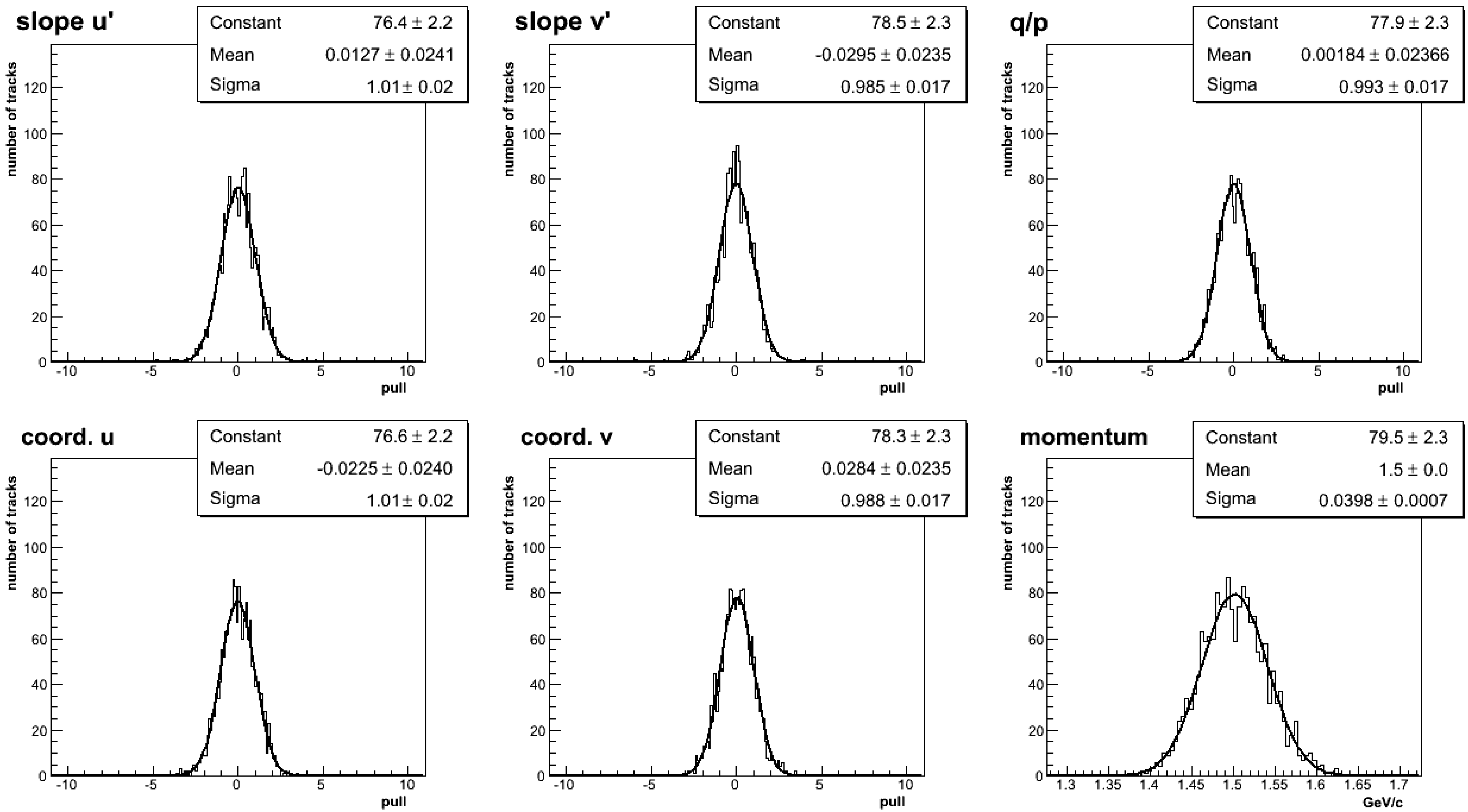
GEANE 0.5 GeV μ^- *efficiency* 97.7 %



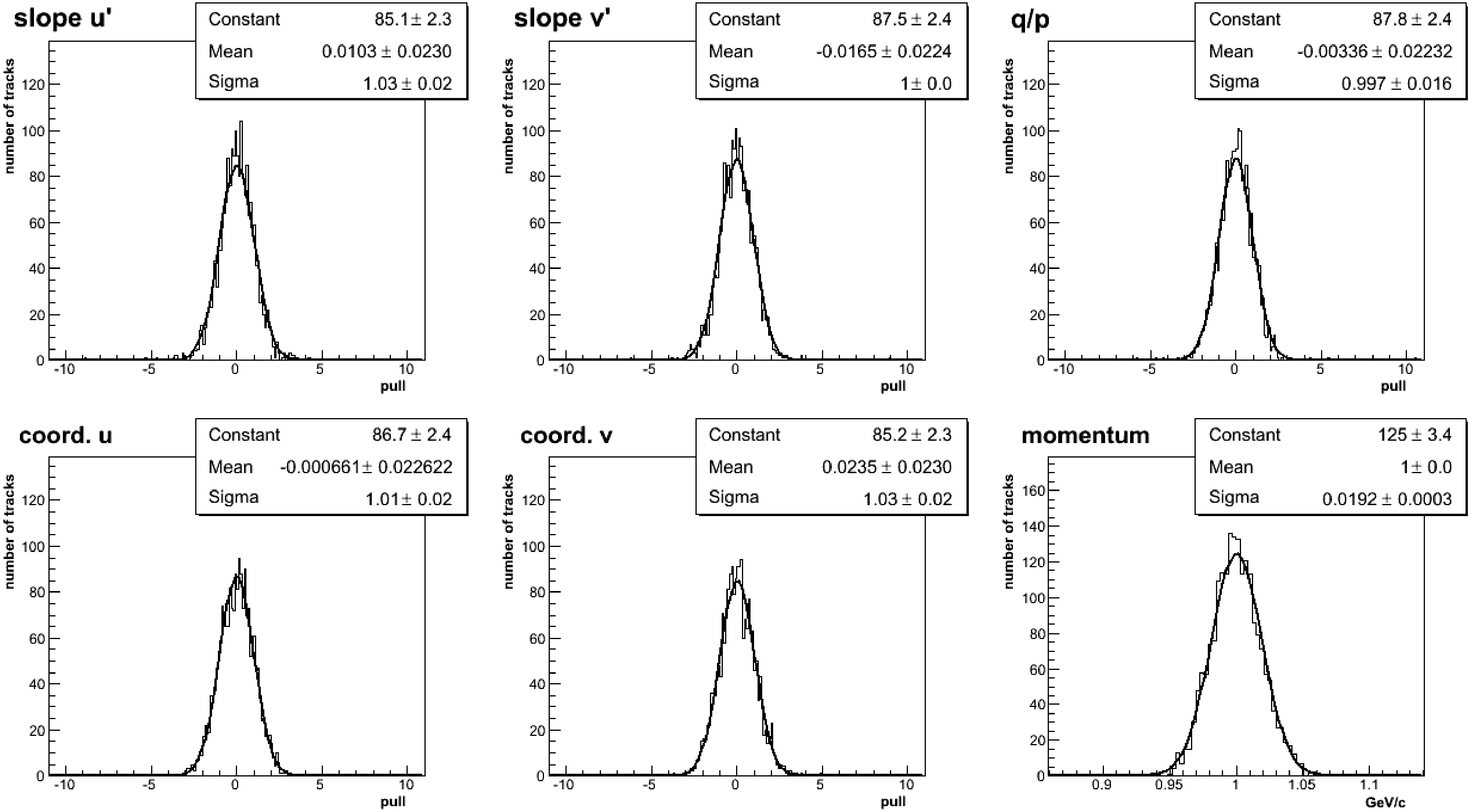
RK 1.5 GeV μ^- *efficiency* 100 %



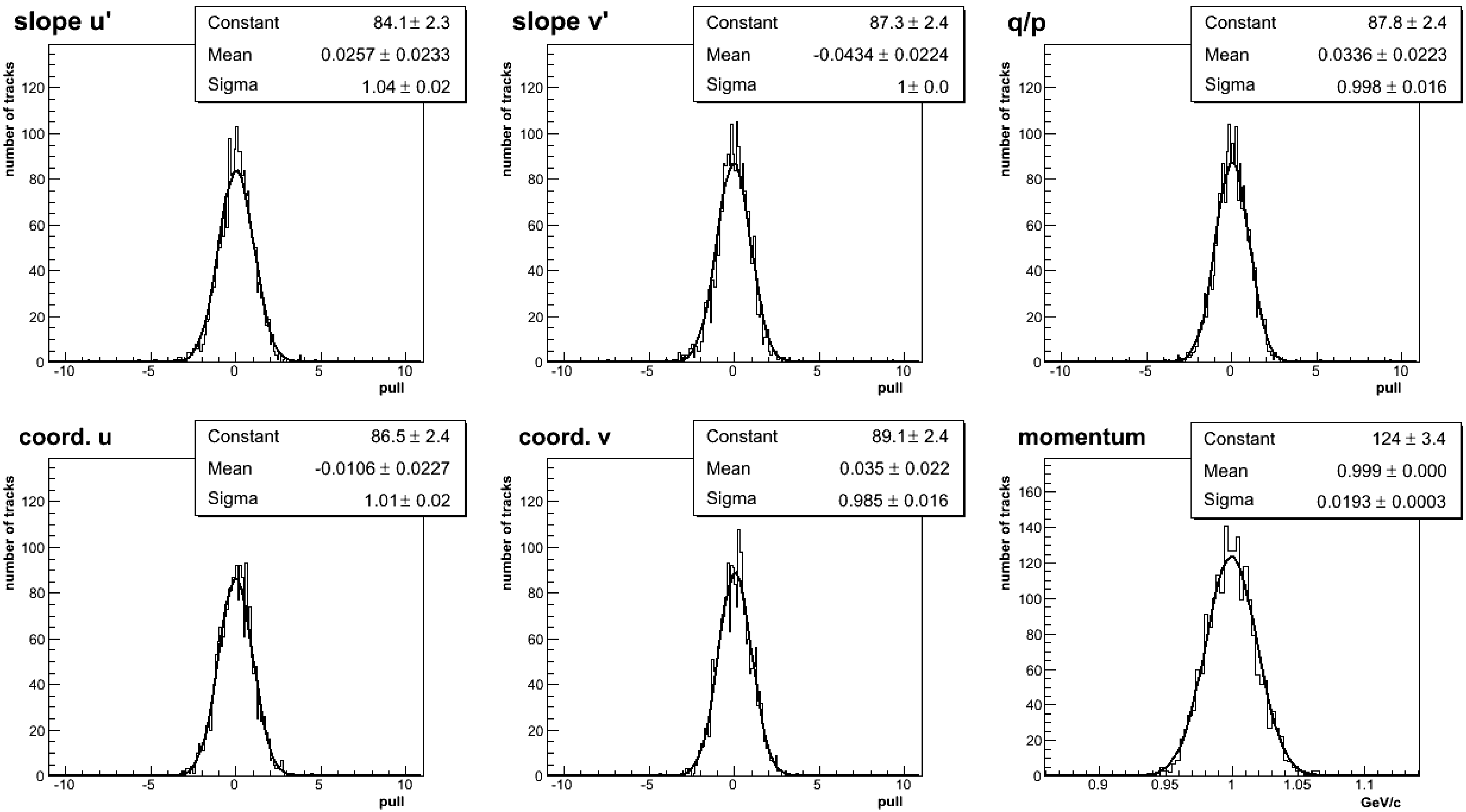
GEANE 1.5 GeV μ⁻ *efficiency* 88.15 %



RK 1.0 GeV p *efficiency* 99.9 %



GEANE 1.0 GeV p *efficiency* 99.9 %



Comparison Overview

Monte Carlo		Fitting	Pull distributions										
			Slope u'		Slope v'		Coordinate u		Coordinate v		q/p		
Energy	Particle	Efficiency	Mean	Sigma	Mean	Sigma	Mean	Sigma	Mean	Sigma	Mean	Sigma	
0.5 GeV	μ-	99.95%	0.0113	0.987	0.0554	1.020	0.00603	0.979	-0.0530	0.960	-0.01710	0.971	**RK**
0.5 GeV	μ-	97.70%	0.0557	0.964	0.0504	1.020	-0.05100	1.010	0.0366	1.040	0.00614	0.993	**GEANE**
1.5 GeV	μ-	100.00%	-0.0147	0.992	-0.0084	0.981	0.00671	1.010	-0.0155	0.988	-0.01040	0.994	**RK**
1.5 GeV	μ-	88.15%	0.0127	1.010	-0.0295	0.985	-0.02250	1.010	0.0284	0.998	0.00184	0.993	**GEANE**
1.0 GeV	μ+	100.00%	-0.0011	0.982	-0.0277	0.990	-0.02510	0.995	0.0226	1.020	0.01820	0.999	**RK**
1.0 GeV	μ+	95.60%	0.0164	0.975	-0.0337	1.000	-0.03070	0.987	0.0419	1.020	0.03430	1.010	**GEANE**
1.0 GeV	p	99.90%	0.0103	1.030	-0.0165	1.000	-0.00066	1.010	0.0235	1.030	-0.00336	0.997	**RK**
1.0 GeV	p	99.90%	0.0257	1.040	-0.0434	1.000	-0.01060	1.010	0.0350	0.985	0.03360	0.998	**GEANE**

- Very similar performance, within uncercainty of only 2000 MC events none of the track representations is significantly better.
- RKTrackRep tends to have a better reconstruction efficiency and slightly better pull distributions

Conclusion

- RKTrackRep is almost complete (bremsstrahlung is under development)
- Can be used for complete panda phase space, including forward tracks (theta=0)
- Overall performance and resolution is comparable to GeaneTrackRep

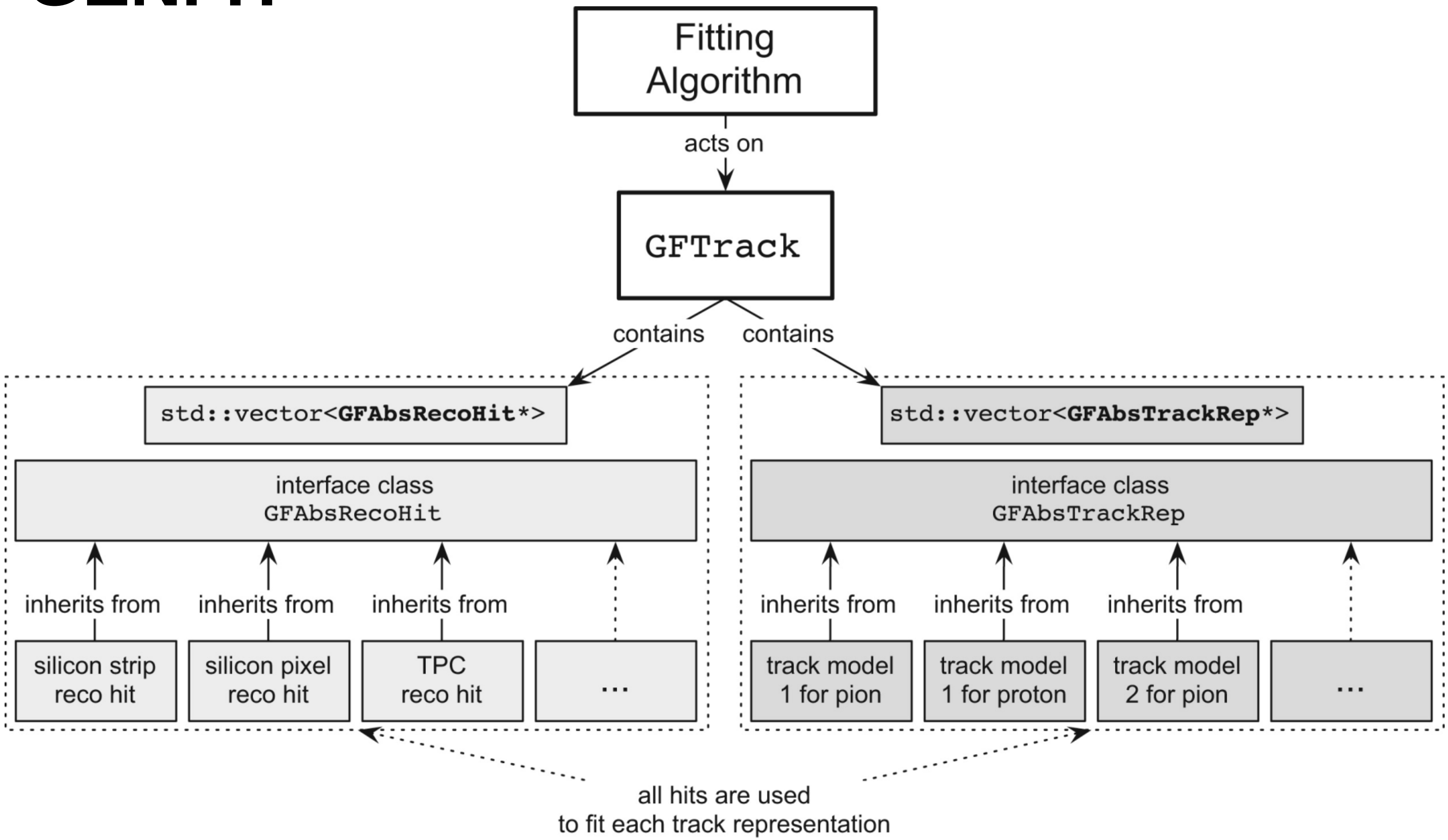
```
GeaneTrackRep(FairGeanePro* geane,
              const GFDetPlane& plane,
              const TVector3& mom,
              const TVector3& poserr,
              const TVector3& momerr,
              double q,
              int PDGCode);
```

```
RKTrackRep(const TVector3& pos,
           const TVector3& mom,
           const TVector3& poserr,
           const TVector3& momerr,
           const int& PDGCode);
```

References

- NIM A doi:10.1016/j.nima.2010.03.136 , *A novel generic framework for track fitting in complex detector systems*
- R. Frühwirt, M. Regler, R.K. Bock, H.Grote, D. Notz, *Data Analysis Techniques for High-Energy Physics*
- A.Fontana et al., *Track following in dense media and inhomogeneous magnetic fields*, PANDA Report PV/01-07

GENFIT


Fitting Algorithm
acts on
GFTrack
contains
contains
std::vector<GFAbsRecoHit*>
std::vector<GFAbsTrackRep*>
interface class GFAbsRecoHit
interface class GFAbsTrackRep
inherits from
inherits from
inherits from
inherits from
inherits from
inherits from
silicon strip reco hit
silicon pixel reco hit
TPC reco hit
...
track model 1 for pion
track model 1 for proton
track model 2 for pion
...
all hits are used
to fit each track representation

Track Representation – Interface

- `virtual double` **`extrapolate`**`(const GFDetPlane& plane, TMatrixT<double>& statePred, TMatrixT<double>& covPred)`
  - Extrapolates track to the given detector plane,
  - returns path length, predicted state (by ref.), predicted covariance (by ref.)

- `virtual void` **`extrapolateToPoin`**`t(const TVector3& point, TVector3& poca, TVector3& normVec)`
  - Extrapolates track to the point of closest approach (POCA) to a point in space
  - Returns POCA (by ref.), flight direction at POCA (by ref.)

- `virtual void` **`extrapolateToLine`**`(const TVector3& point1, const TVector3& point2, TVector3& poca, TVector3& normVec, TVector3& poca_onwire)`
  - Extrapolates track to the point of closest approach to a line
  - Returns POCA (by ref.), flight direction at POCA (by ref.)

RKTrackRep – Material Model

Steppers

- Steppers must inerhit from `GFAbsMatStep`
- Multiple steppers can be used in parallel

- `GFTestMatStep:`
  - Gets material parameters from `TGeo`
  - calculates *dE/dx*
  - limits stepsize so that a specified maximum momentum loss will not be exceeded

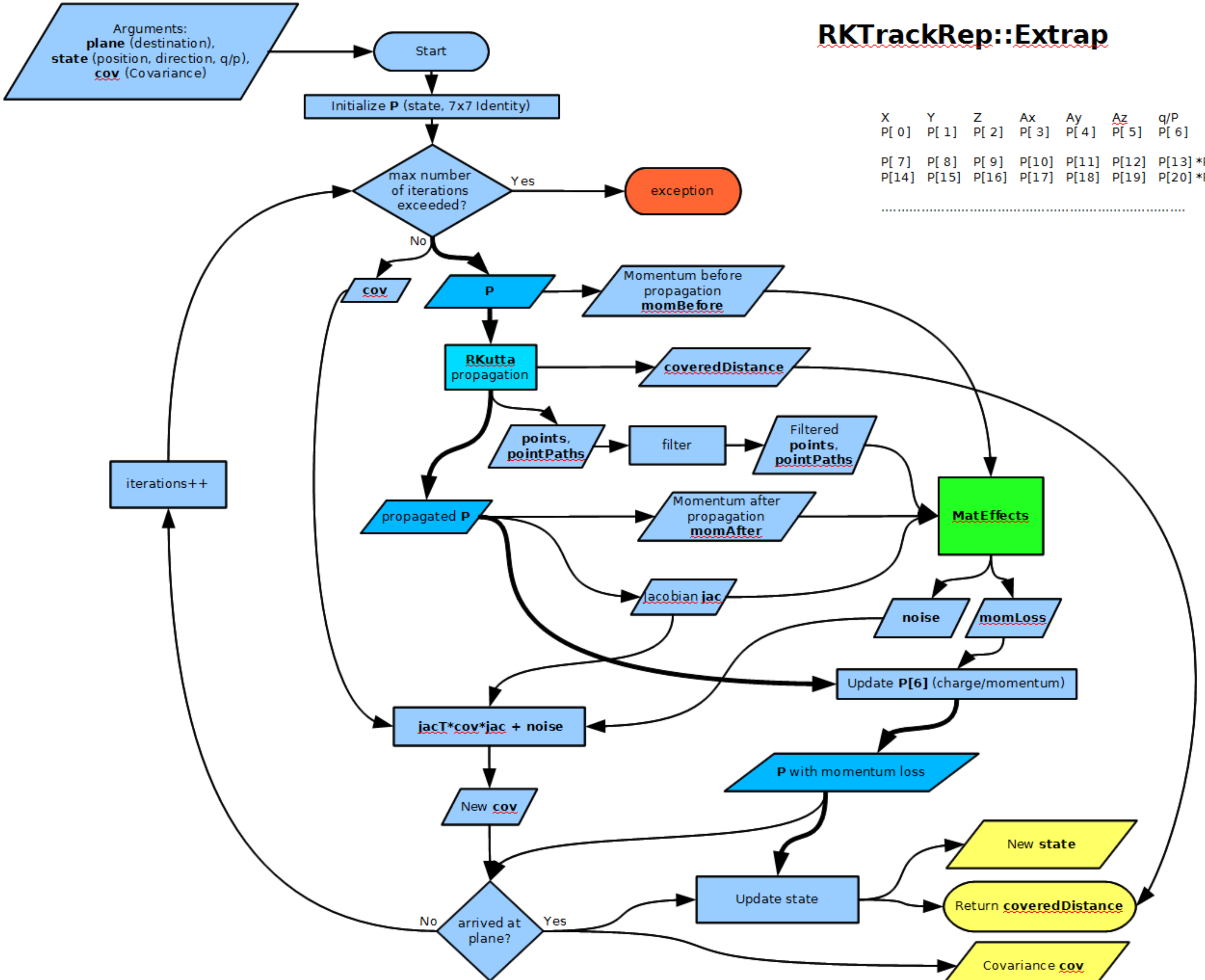
Effects

- Effects must inerhit from `GFAbsMatEffect`
- Multiple effects can be used in parallel

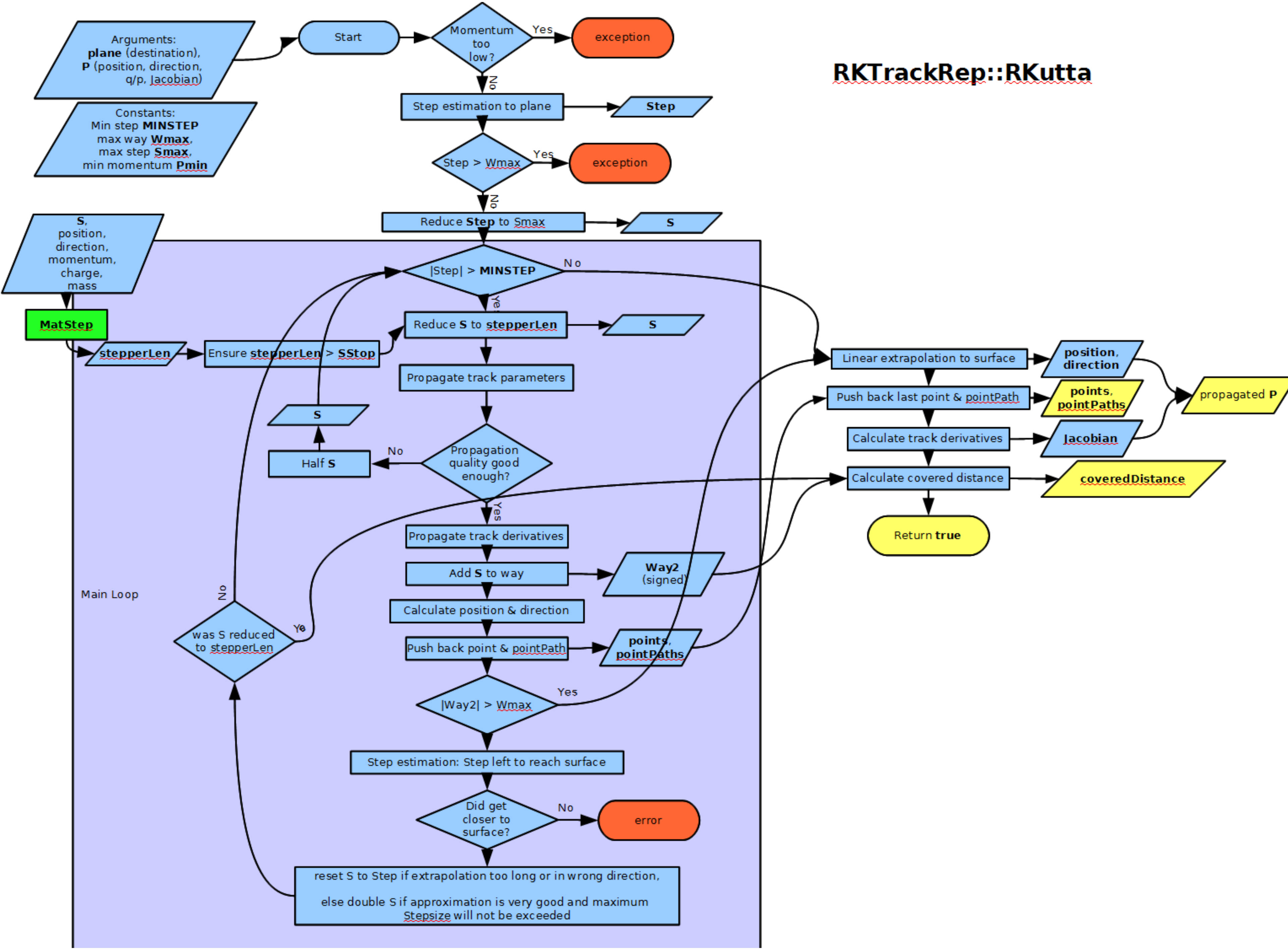
- `GFTestMatEffect:`
  - Gets material parameters from `TGeo`
  - Momentum loss (Bethe Bloch)
  - Noise matrix (multiple scattering, Energy loss straggling)
  - Currently working on implementation of Bremsstrahlung

RKTrackRep::Extrap

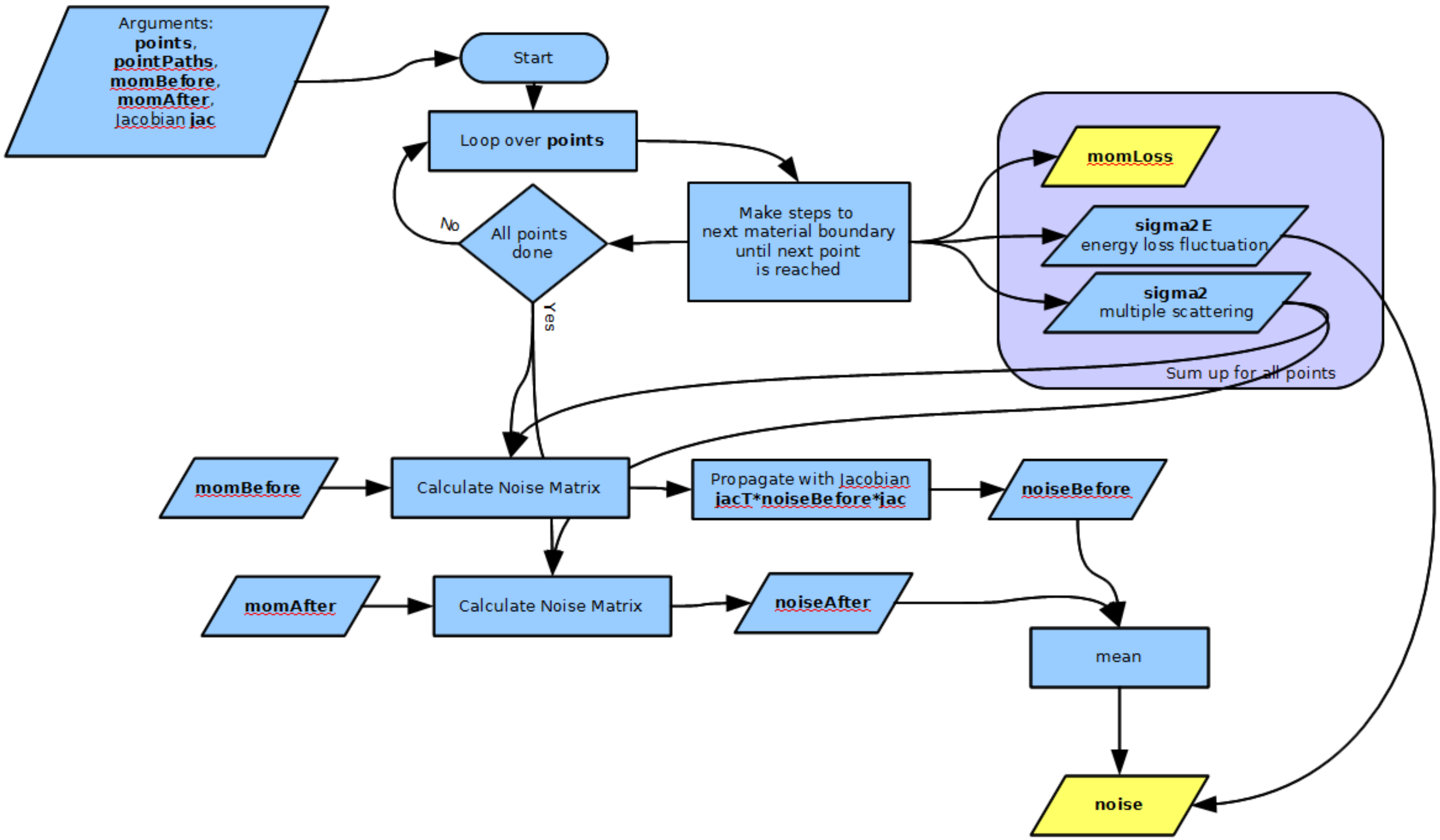
X	Y	Z	Ax	Ay	Az	q/P	
P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	
P[7]	P[8]	P[9]	P[10]	P[11]	P[12]	P[13] *P[6]	d()/dX_before
P[14]	P[15]	P[16]	P[17]	P[18]	P[19]	P[20] *P[6]	d()/dY_before
......							d()/d...



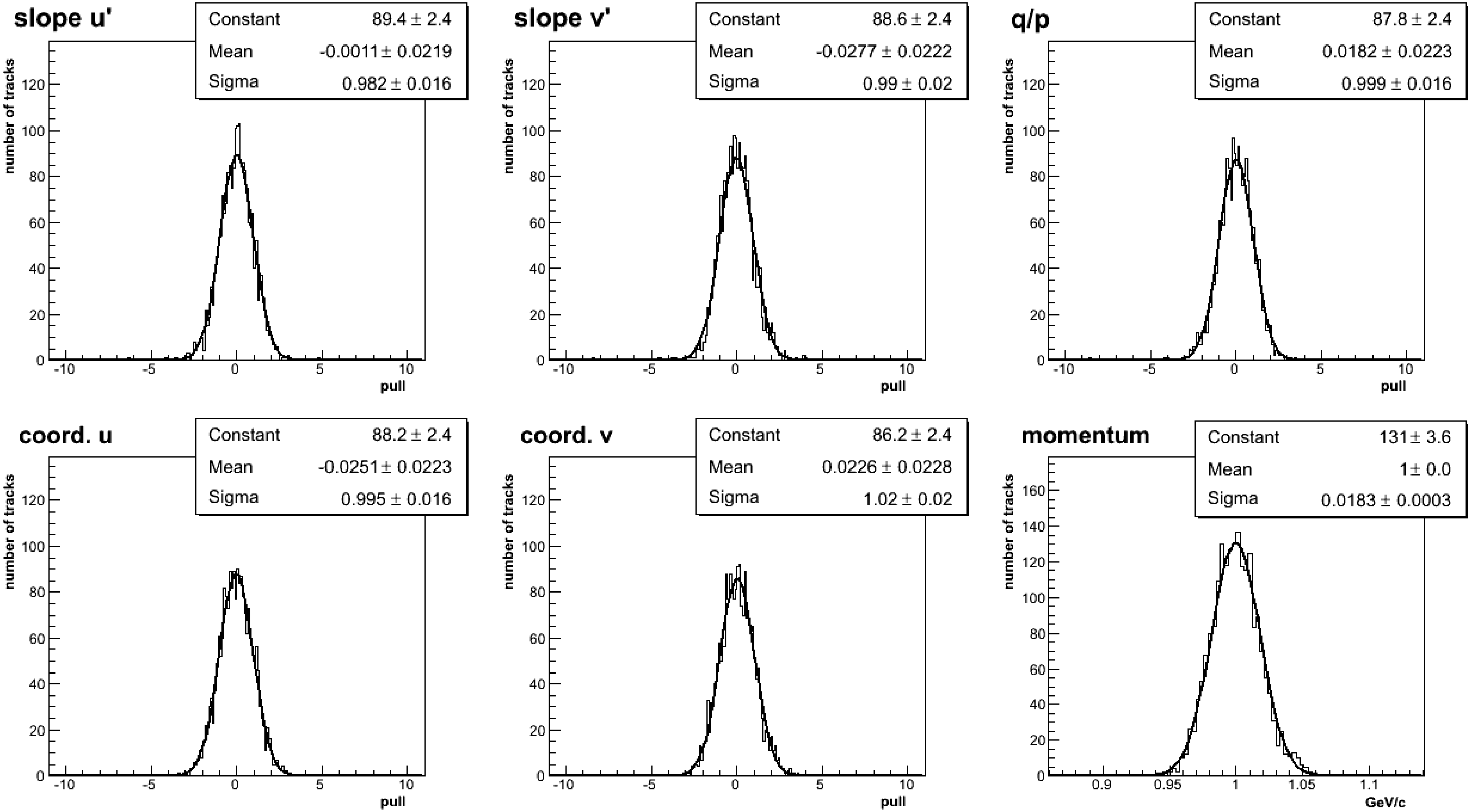
RKTrackRep::RKutta



GFTestMatEffect



RK 1.0 GeV μ^+ *time* 0:46 *efficiency* 100 %



GEANE 1.0 GeV μ^+ *time* 0:59 *efficiency* 95.6 %