Global reconstruction with Genfit framework

Riccardo Ridolfi
riccardo.ridolfi@bo.infn.it

Strasbourg Virtual Meeting – 10 June 2020

Outline

- Introduction on track reconstruction;
- Genfit package;
- Genfit-SHOE integration;
- Track representation;
- Kalman filter;
- Track finding strategy;
- Track fitting;
- Results

Track reconstruction

- Reconstructed hits → **clusters**;
- Track finding → **categorise** clusters in track candidates;
- Track fitting and evaluation of momentum resolution;

Genfit package - 1

Genfit is an experiment- independent modular framework for **track fitting** and other related tasks

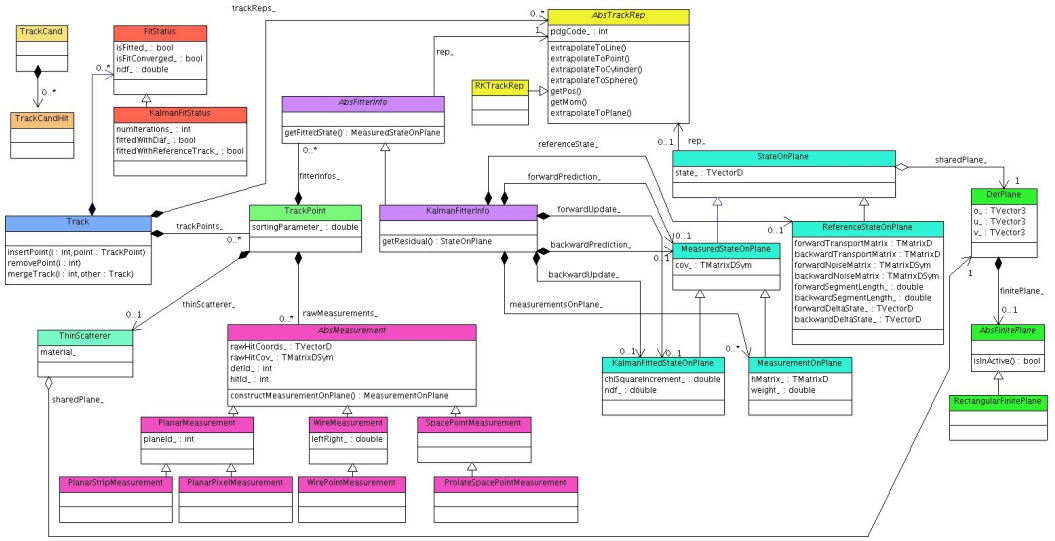
Genfit allows forward/backward **Kalman filter**, extrapolation and propagation in a magnetic field

Genfit is **included** in SHOE

Genfit package - 2

Genfit is based on **three** pillars:

- **measurements**, hit coordinates of the detector (1D, 2D or 3D);
- **track representation**, model of the track (integrated with TDatabasePDG);
- **fitting algorithms**, such as Kalman filter.

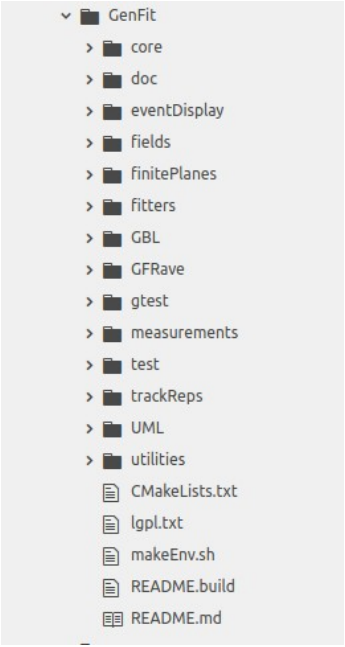


Genfit-SHOE integration

Genfit **libraries** are in shoe/libs/Genfit directory

Used mainly in **KFitter** and helper classes, they will be re-organised in the next future

Added **new** functions with respect to current Github version

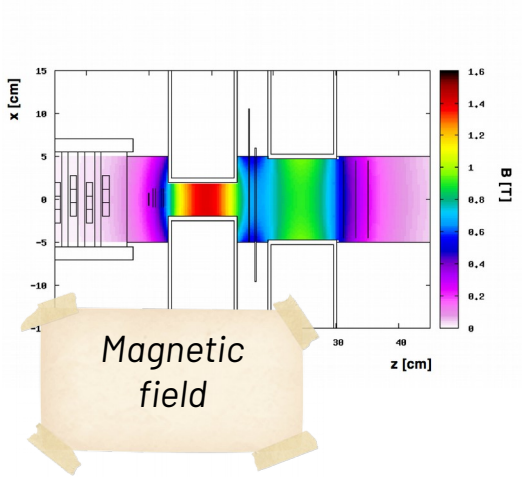


Genfit-SHOE integration

Genfit **libraries** are in shoe/libs/Genfit directory

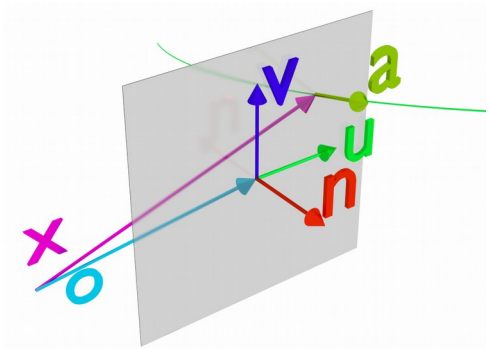
Used mainly in **KFitter** and helper classes, they will be re-organised in the next future

Some **new** functions with respect to Github version



Genfit reads back both the **magnetic field** map and the ROOT **geometry**

Representation of the track



FOOT detectors provide 1D/2D measurement
→ **planar measurements**

The distance along the track **s** is the free parameter and the state vector is parametrized with 5 coordinates in a local plane coordinate system:

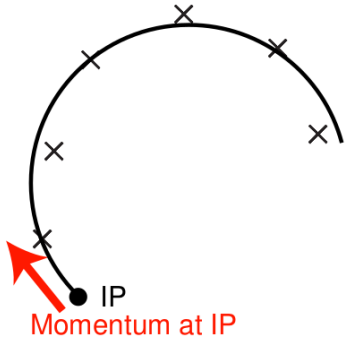
$$(q/p, u, v, u', v')$$

u, v coordinates of the plane
u', v' projections of the direction of momentum on the coordinate axes

Kalman filter

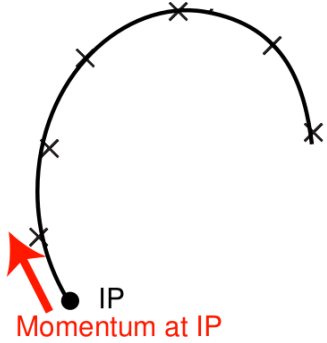
Efficient recursive algorithm, it finds the **best estimate** for the **state** of dynamic system from a set of noisy measurements

a) Normal least square fitting

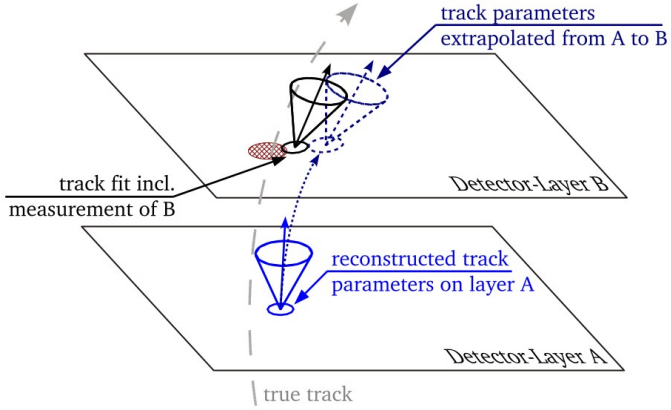


Only one track parameter can be defined

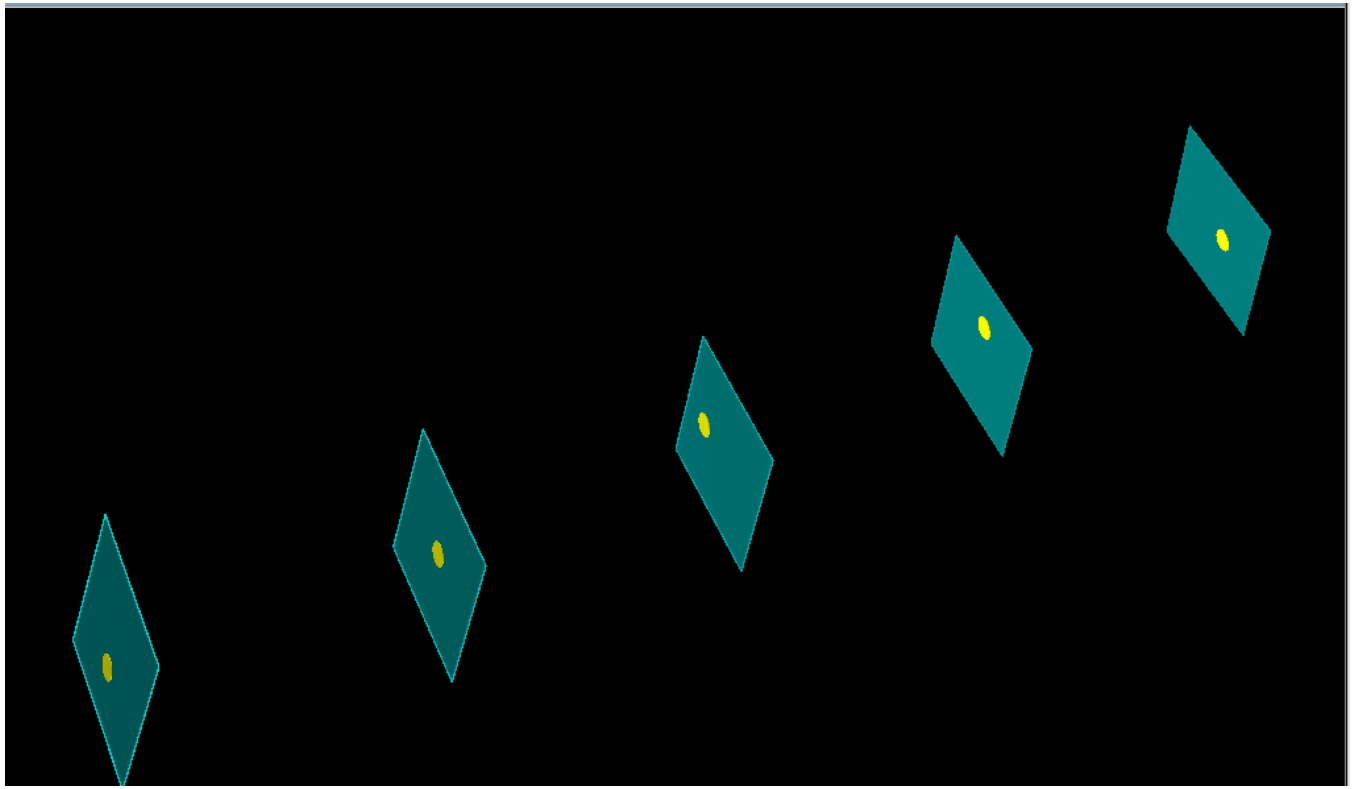
b) Kalman fitting



Track parameters are defined per hit

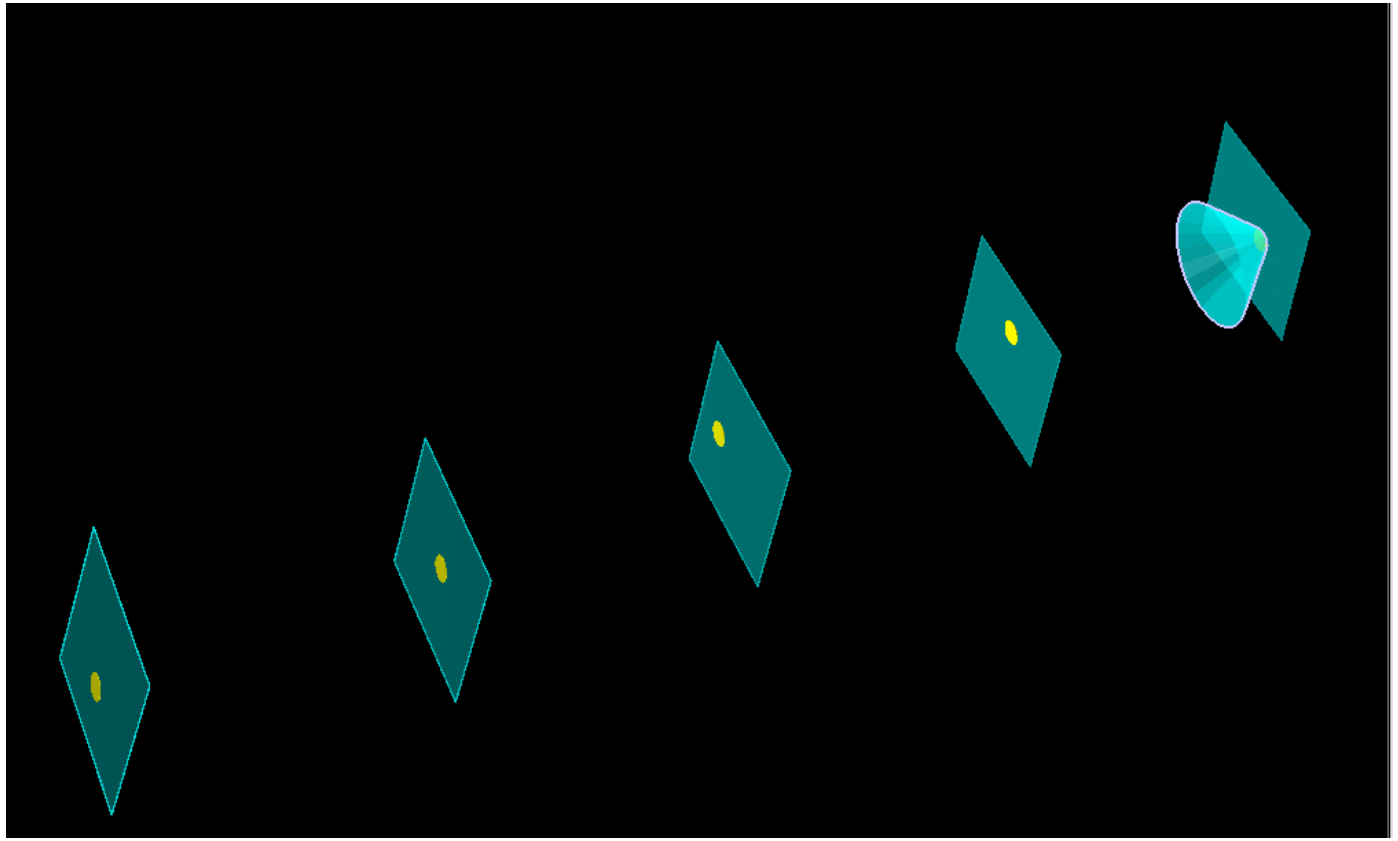


Kalman filter



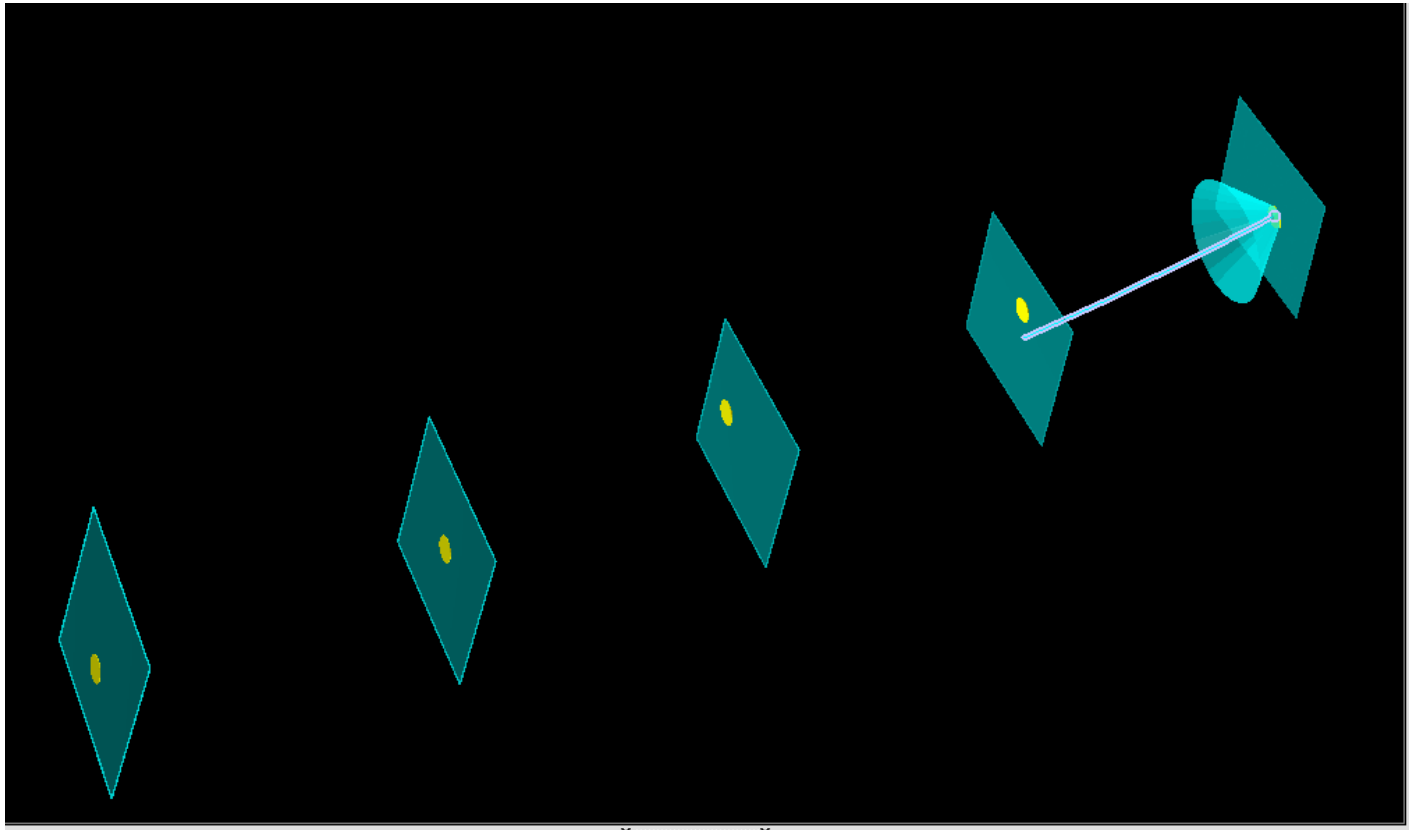
Set of noisy **measurements** (this is a test, not FOOT geometry)

Kalman filter



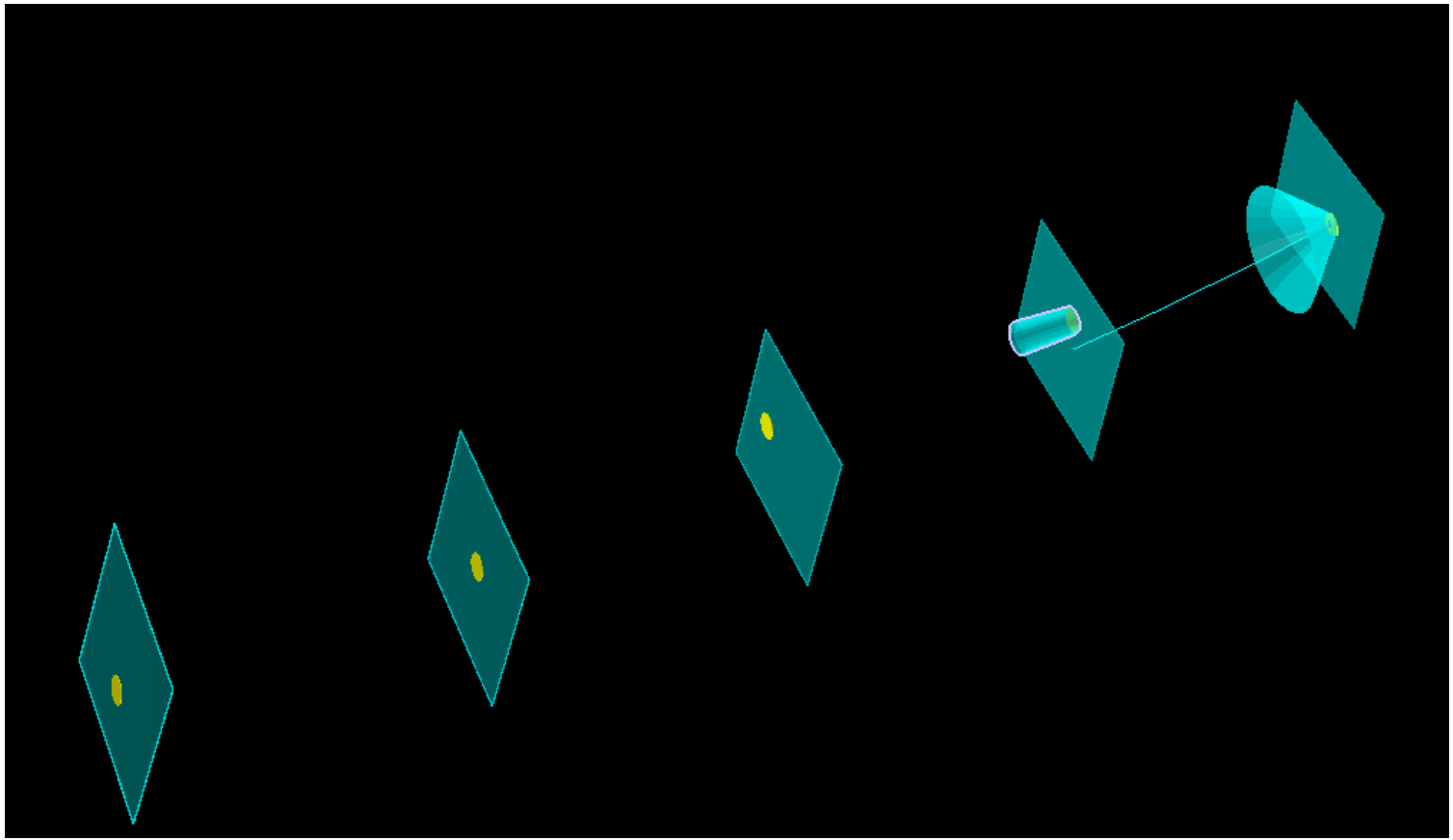
First **update** of the forward fit.
Position determined by first measurement.

Kalman filter



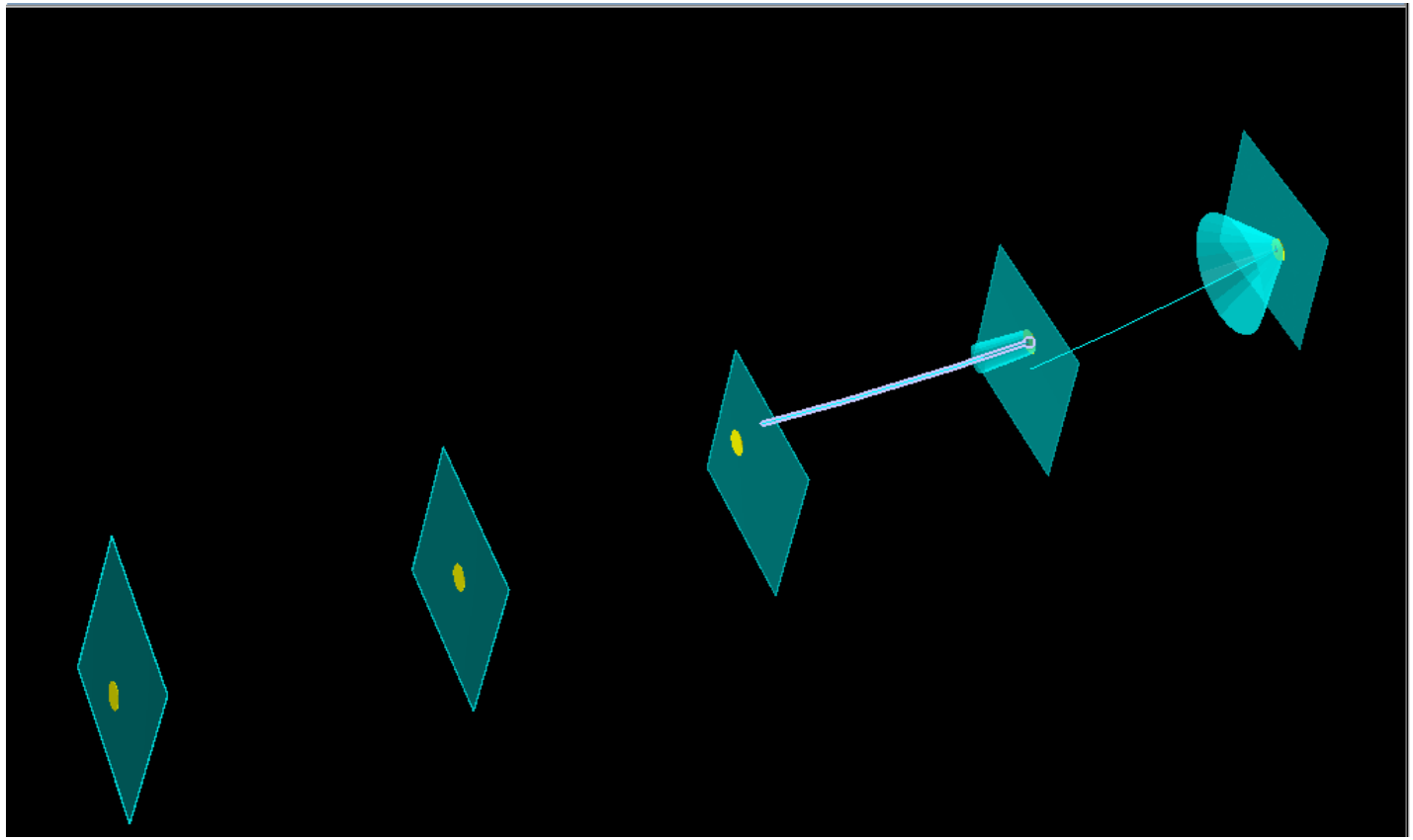
Prediction

Kalman filter



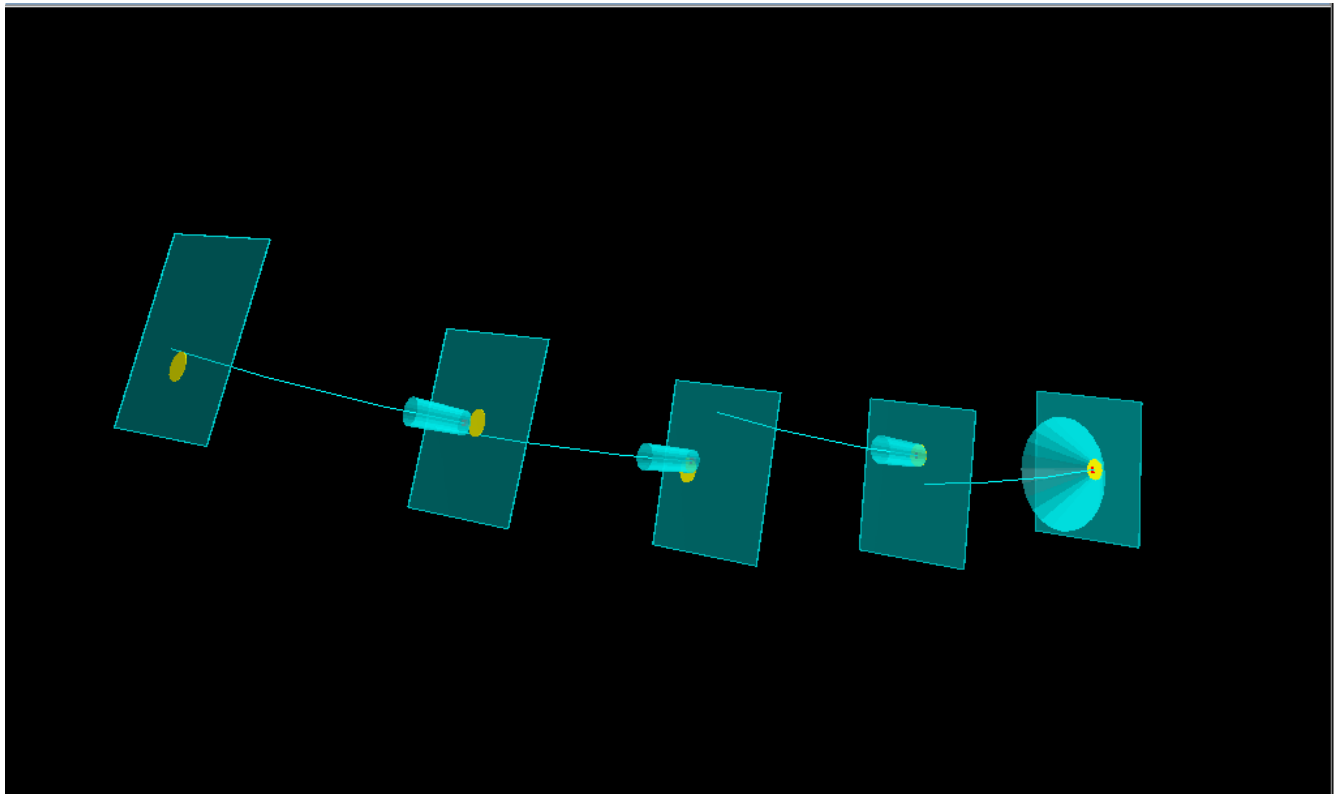
Update

Kalman filter



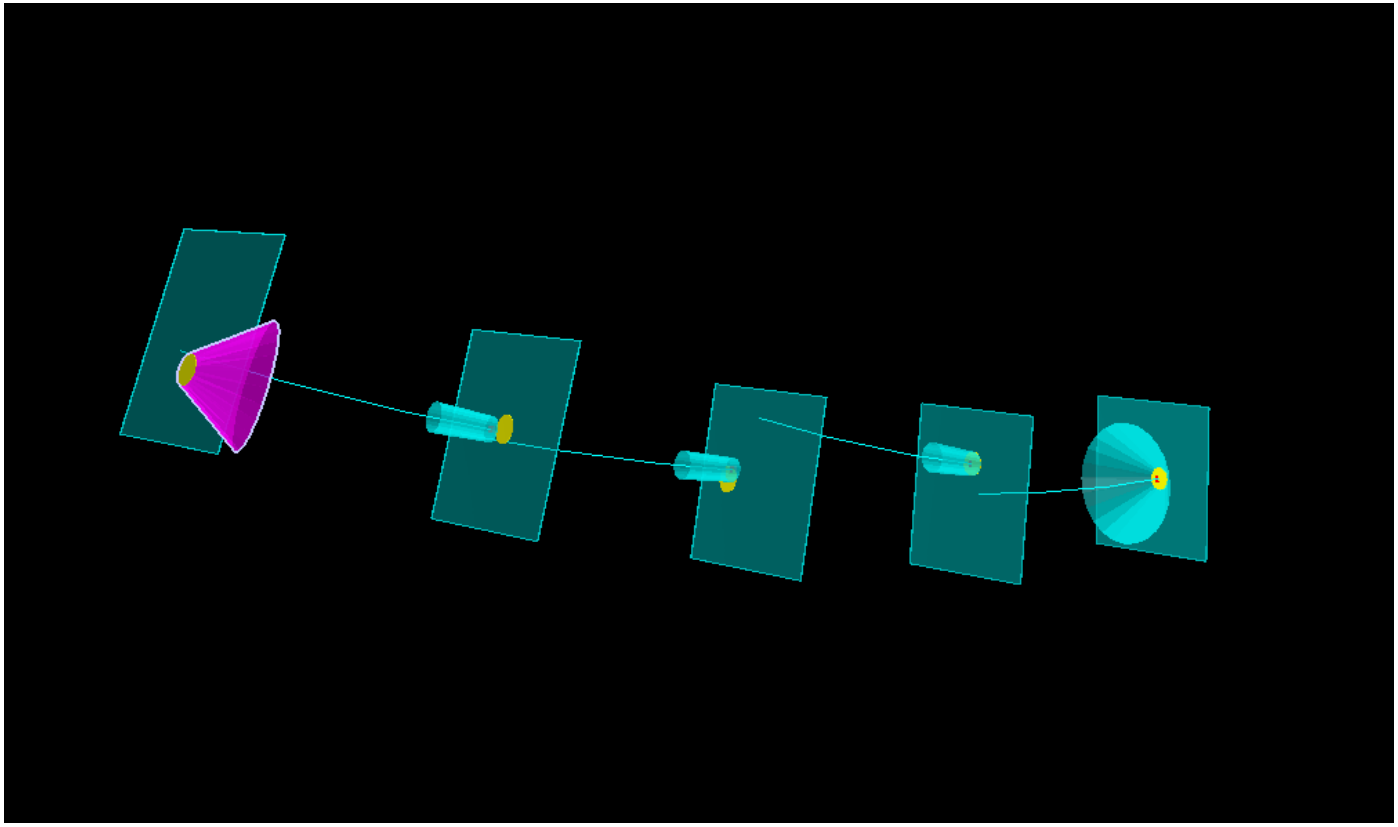
Prediction

Kalman filter



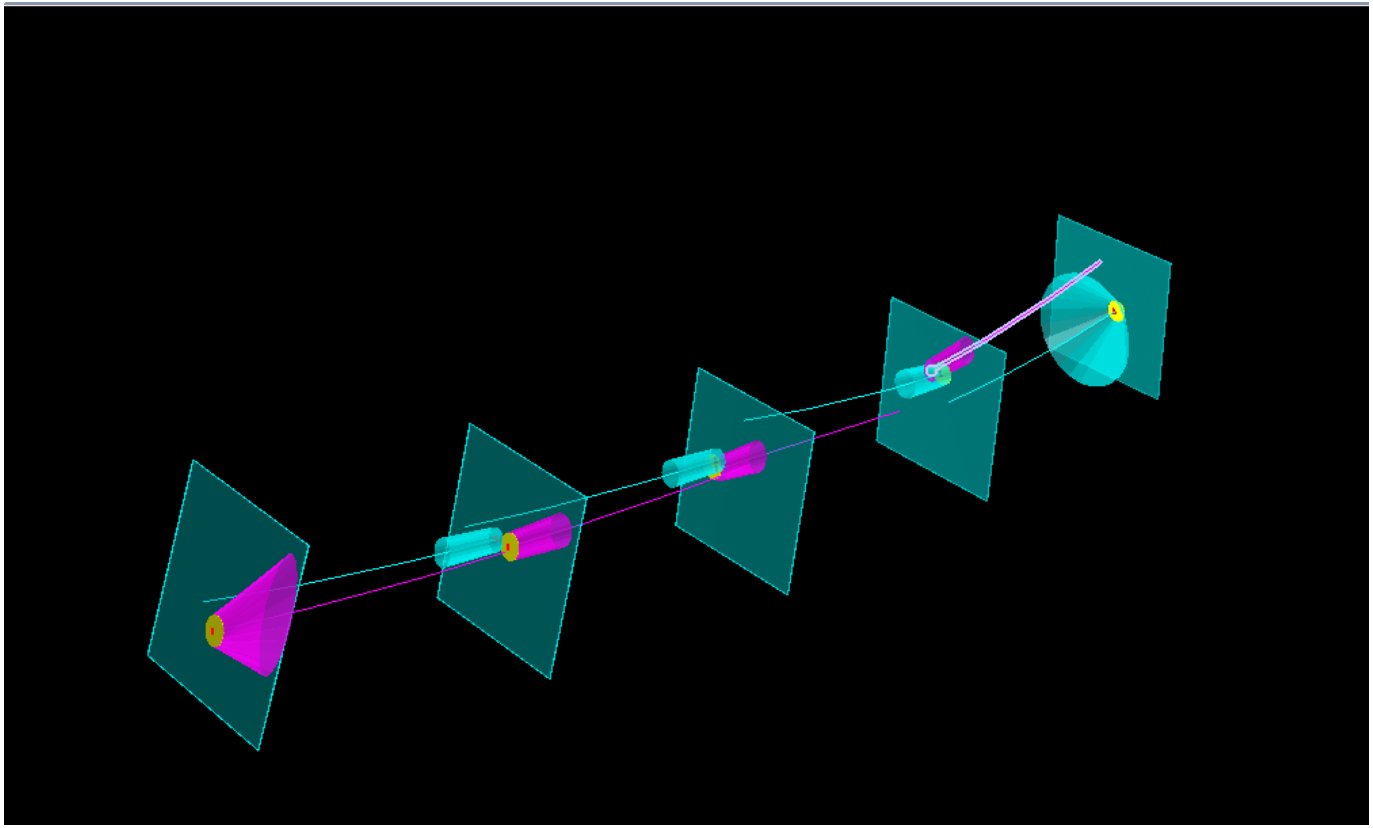
Forward fit

Kalman filter



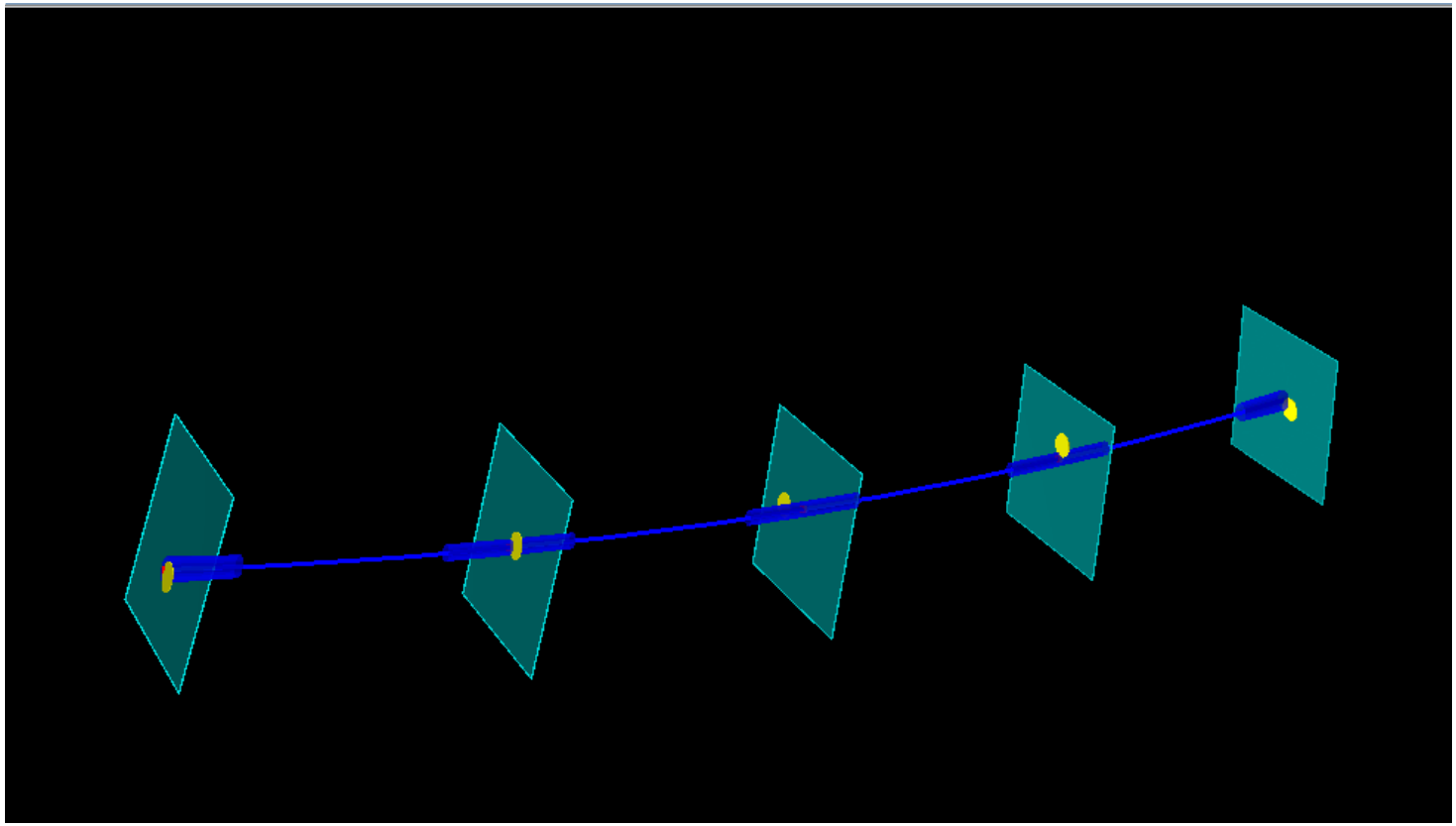
First **update** of the backward fit.

Kalman filter



Prediction

Kalman filter



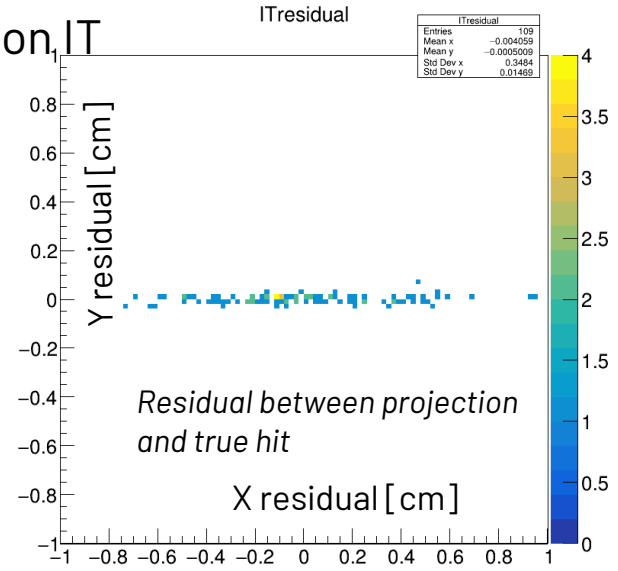
Smoothed track

Track finding strategy

1) Get information from TW about fragment charge
2) Take all vertices found in the target
3) Take all VTX tracks
4) Add clusters in the vertex detector to the track candidate
5) Project them with a straight line to IT positions (only YZ view)
6) Add clusters with minimum distance on IT

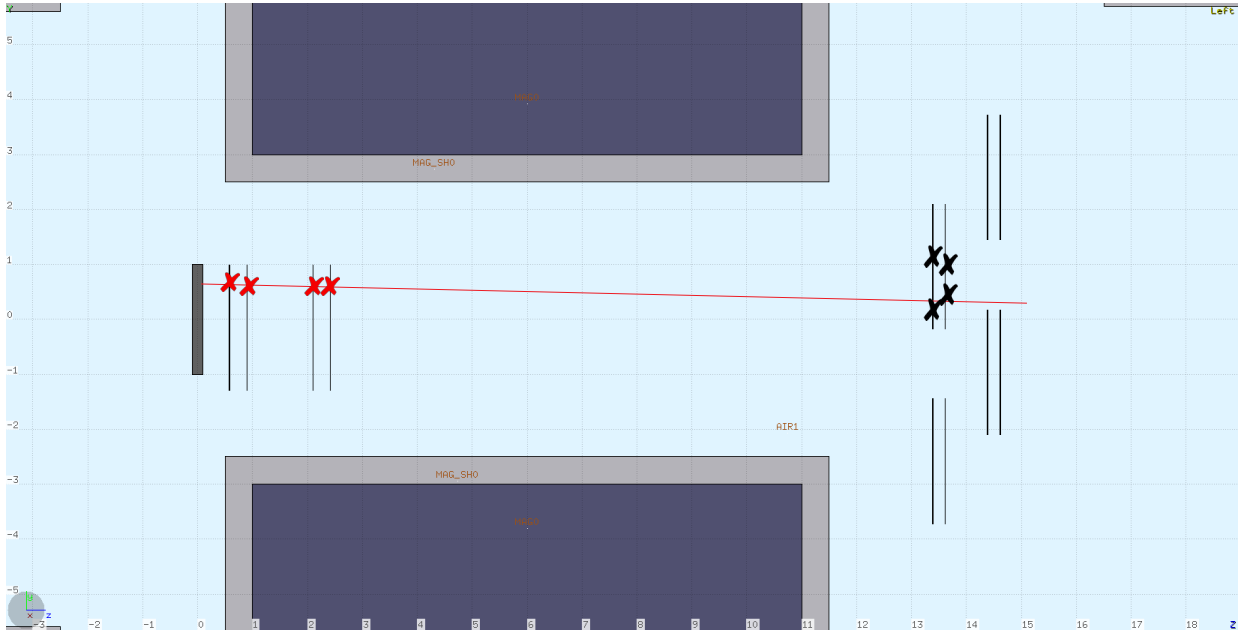
At this step no information about particle type and momentum, only geometrical basis

YZ "no-(less) bending" view
XZ bending view



Track finding strategy

1) Get information from TW about fragment charge
2) Take all vertices found in the target
3) Take all VTX tracks
4) Add clusters in the vertex detector to the track candidate
5) Project them with a straight line to IT positions (only YZ view)
6) Add clusters with minimum distance on IT



Track finding strategy

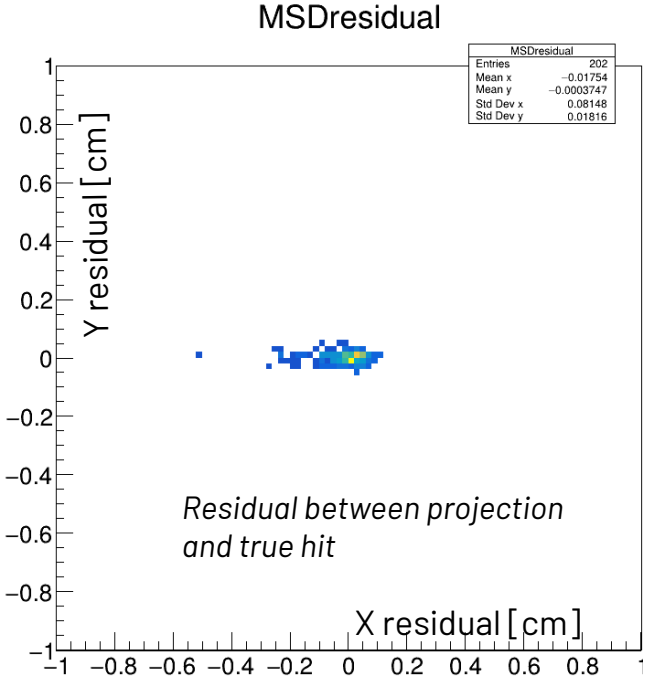
Now **extrapolation to MSD** has to be done

After the IT insertion we have **more "information"** also about the bending, i.e. q/p

Moreover, TW allows us to make a guess about produced fragments:

1) Runge-Kutta extrapolation
2) Add closer cluster to extrapolation
3) Repeat for each MSD layer (1D measurement)

Main issues: starting value for momentum and particle type



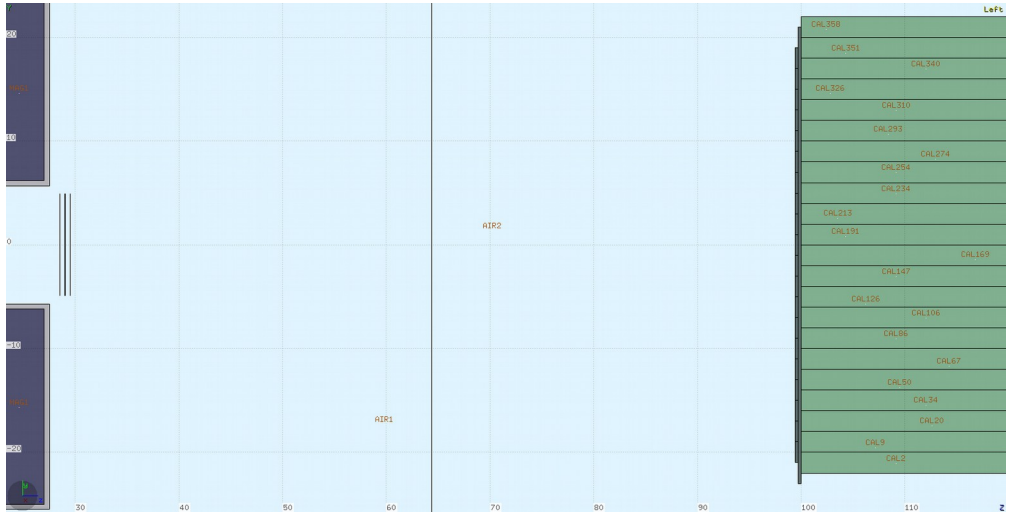
Track finding strategy

1) Extrapolate to TW position with Runge-Kutta

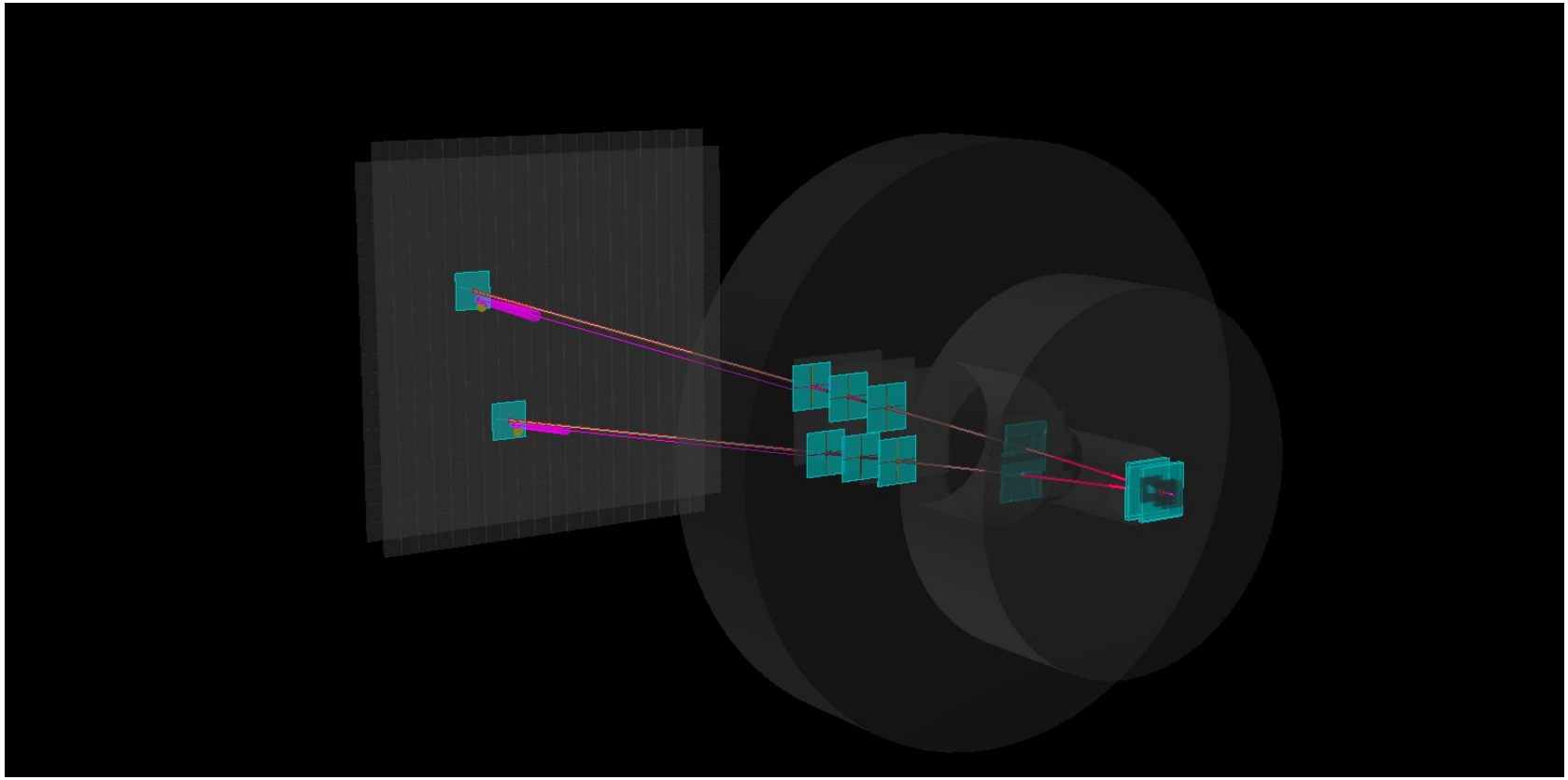
2) Calculate residuals with TW points

3) Choose best point and get its charge

4) Set particle type and make the real fit (for light nuclei H, He all isotopes)



Track finding strategy



Track finding results

In the following, simulation 200 MeV/u 160 on Carbon was used (41k events)

reconstruction efficiency

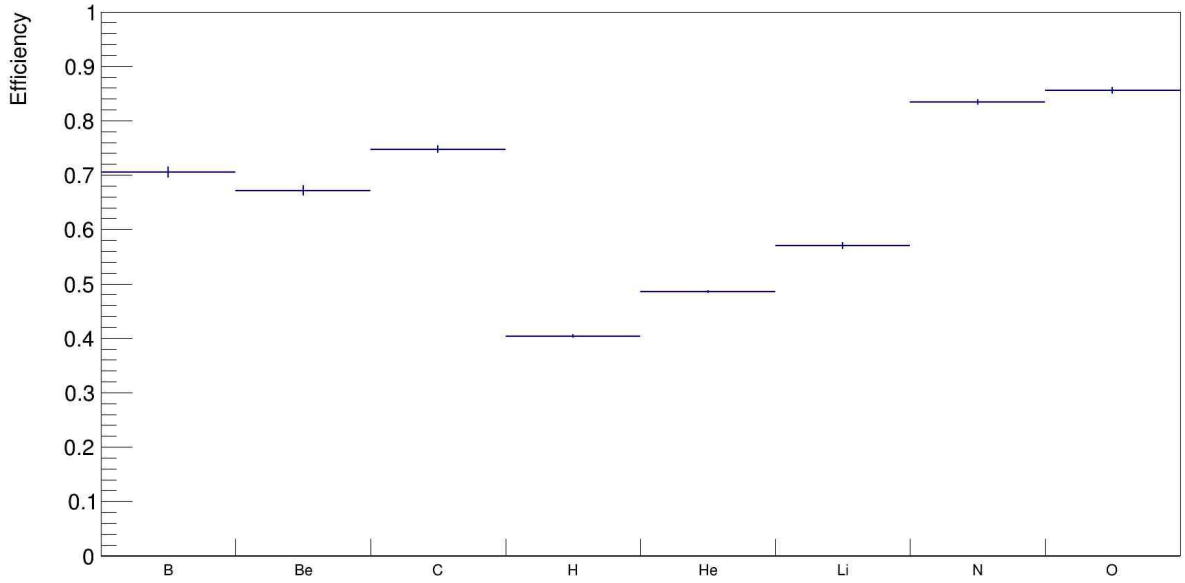
$$\varepsilon_{\text{reco}} = \frac{N_{\text{ref}}^{\text{reco}}}{N_{\text{ref}}}$$

number of fragments reconstructed by at least one track

number of fragments seen by TW originated in the target

Fragments originated in the detector are not handled at this step!

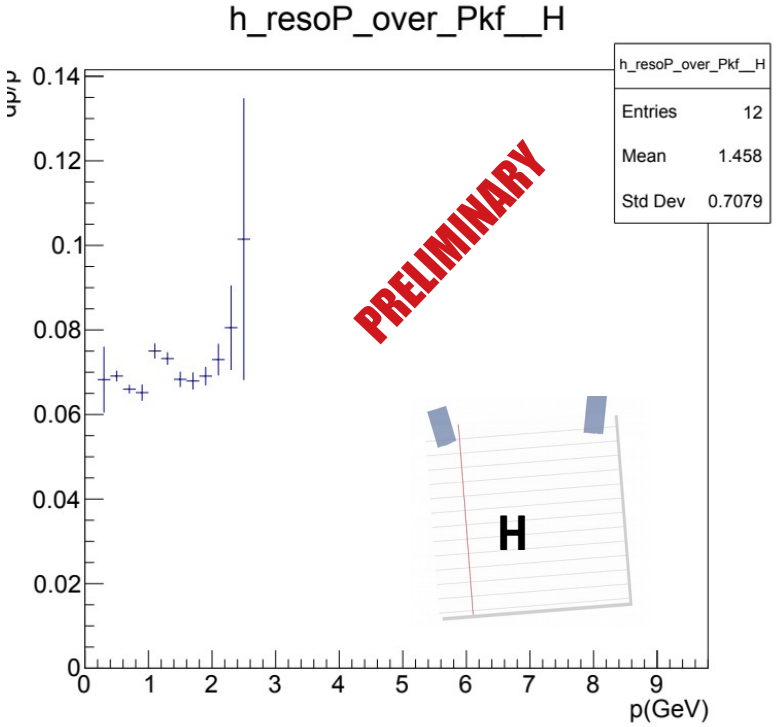
Track finding results

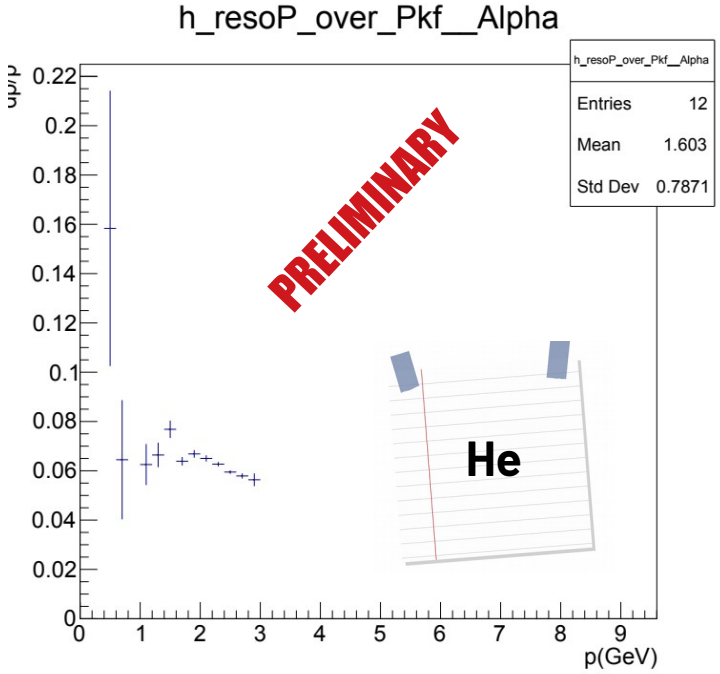


Hit matching efficiency (purity) **~ 98%**
No request on minimum number of measurements
Chi-square **cut**

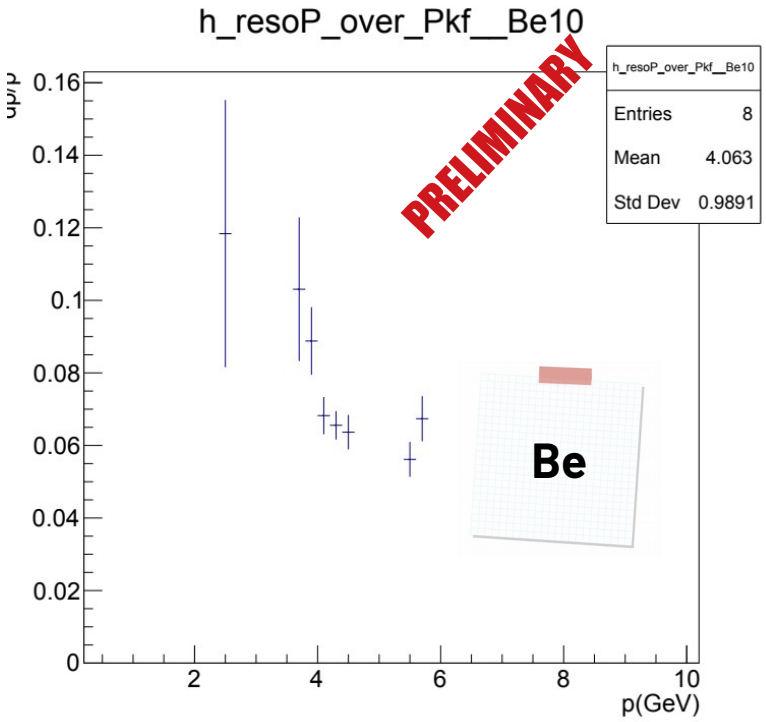
Track fitting

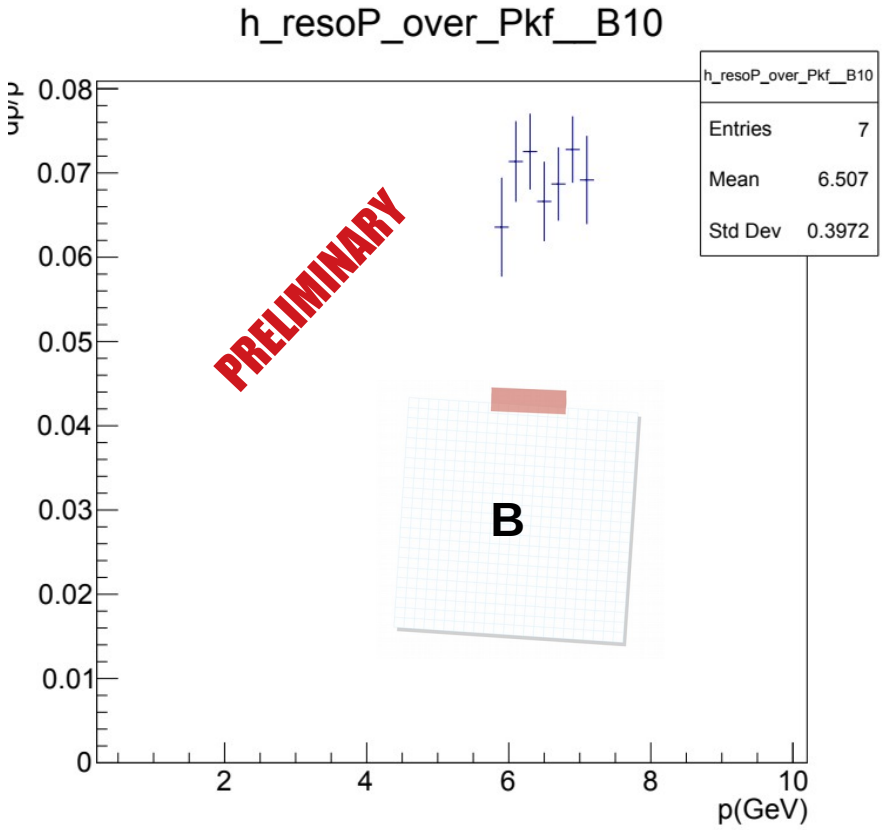
Results are recorded **by fragment** charge;
Momentum axis is divided in bins **200 MeV/c wide**;



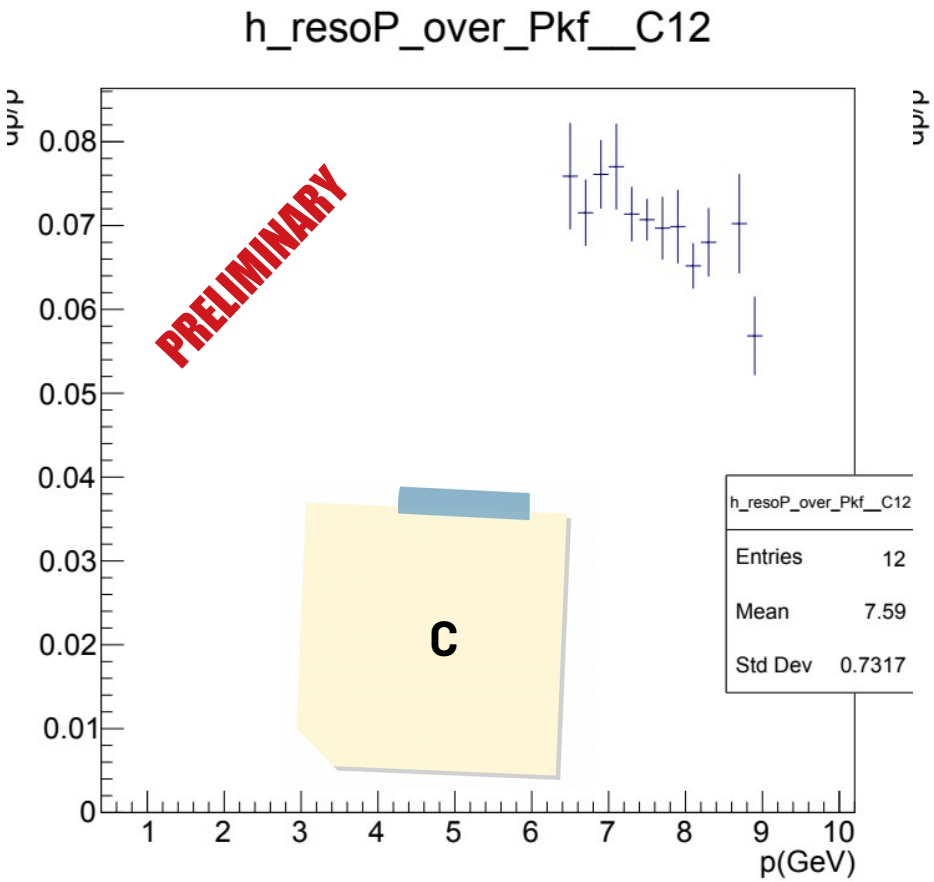


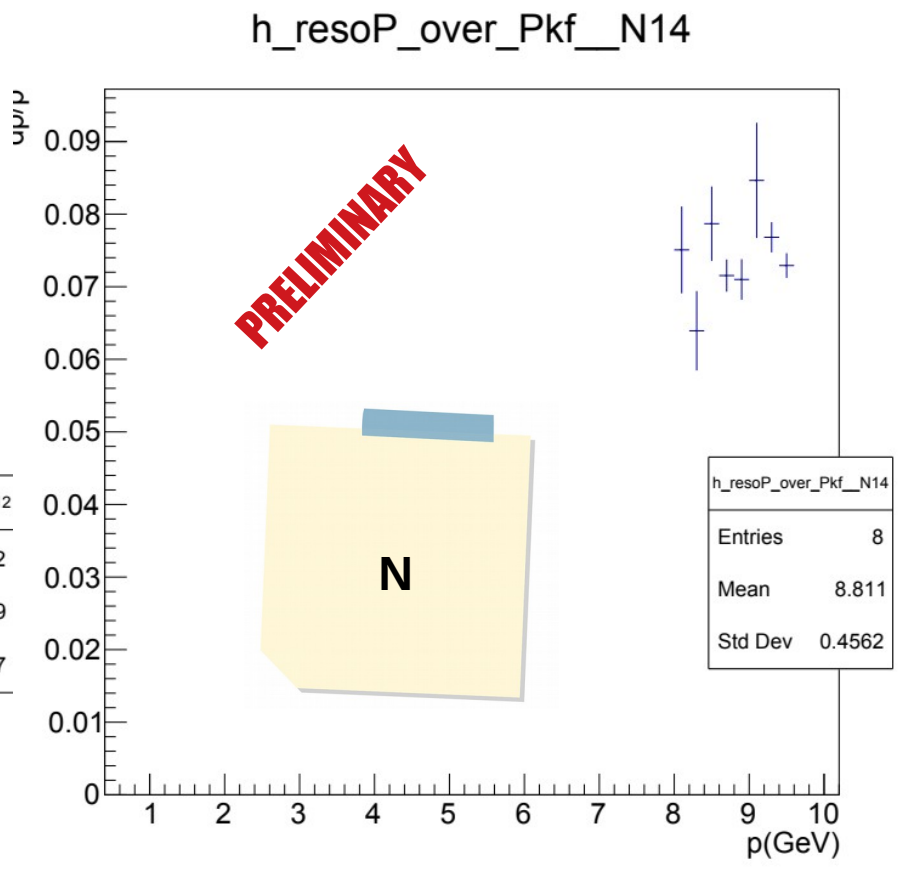
Track fitting





Track fitting





Conclusions

- First **global reconstruction** strategy with Genfit is set in place;
- Still **room for improvement**, both in track finding and in track fitting;
- Reconstruction efficiency **from 0.4** with Hydrogen to **0.85** with Oxygen;
- Investigate other **reference set** choices;
- Momentum resolution is around **7%** up to now (it was ~4% for heavy fragments with MC truth)
- Several **improvements** already in mind

Thank for your attention!

Track finding results