GEANE

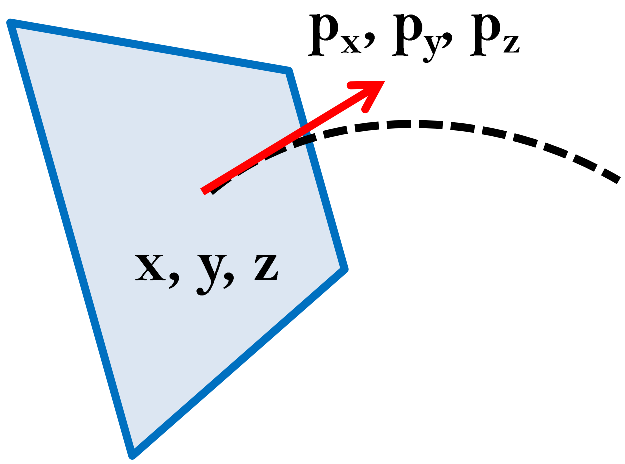
L. Lavezzi
IHEP – INFN, Torino

PANDA Tracking Computing Workshop
GSI, 2018-09-18/19

Some definitions

Track description (1)

A track is the set of hits on the measurement planes

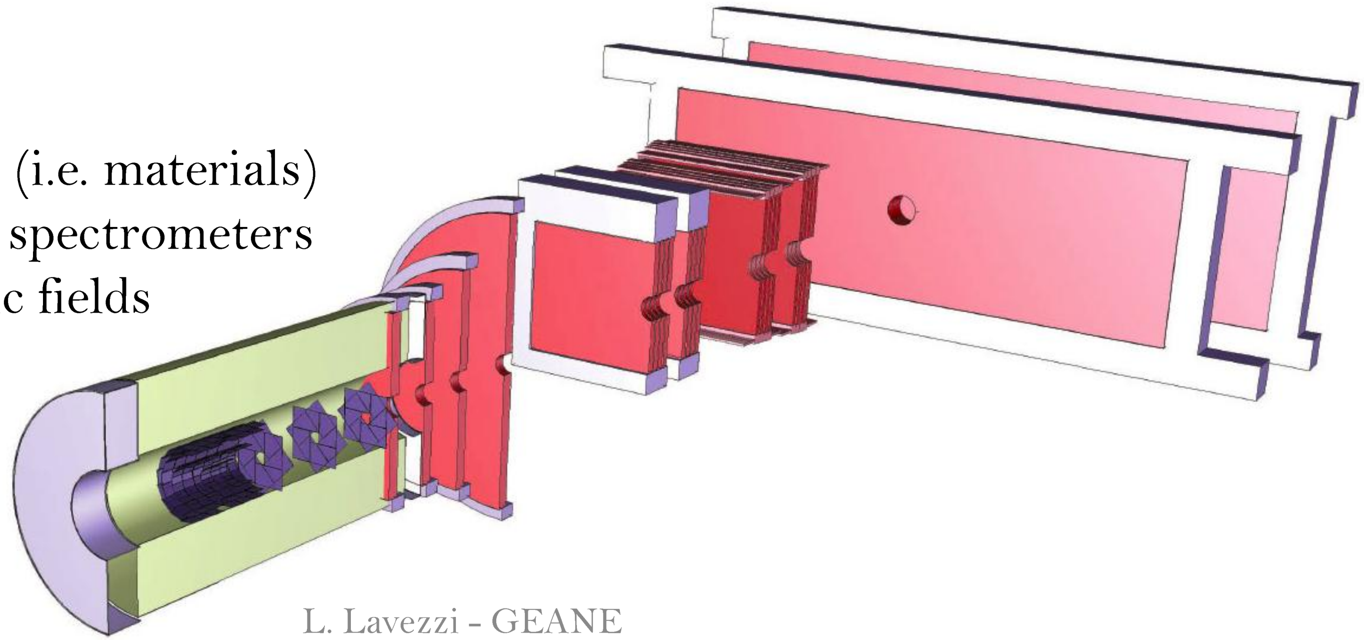


5 parameters

- 3 for position + 3 for direction **minus** 1 fixed by the measurement
- covariance matrix 5 x 5

- different trackers (i.e. materials)
- target + forward spectrometers
- different magnetic fields

etc.



Track description (2)

Simplest description HELIX

Homogeneous magnetic field, no relevant effect of materials (*e.g.* in gas)

- *xy* plane (orthogonal to mag. field): circle with fixed radius of curvature

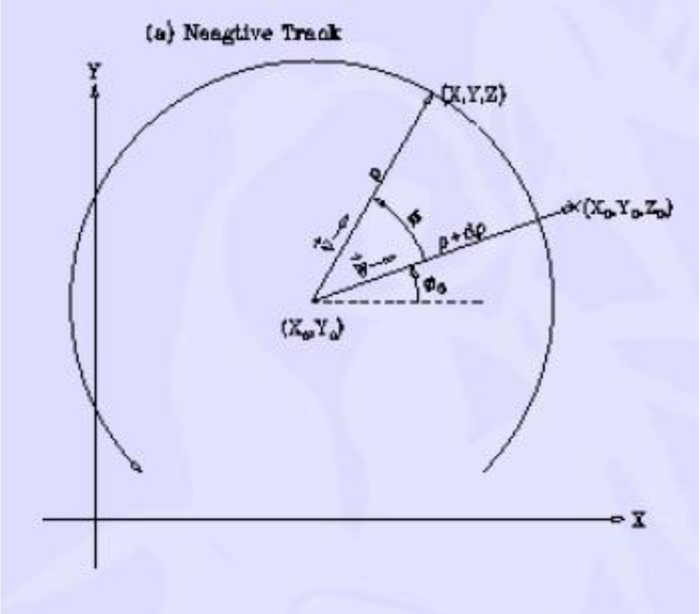
$$x(s) = x_o + R\left[\cos\left(\Phi_o + \frac{hs\cos\lambda}{R}\right) - \cos\Phi_o\right]$$

$$y(s) = y_o + R\left[\sin\left(\Phi_o + \frac{hs\cos\lambda}{R}\right) - \sin\Phi_o\right]$$

- polar coordinate, along the mag. field, linear with track length

$$z(s) = z_o + s\sin\lambda$$

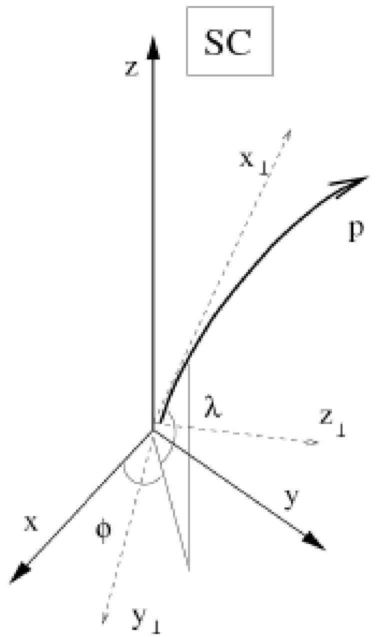
- but in real experiments:
  - material effect (multiple scattering)
  - field non homogeneities
  - energy loss

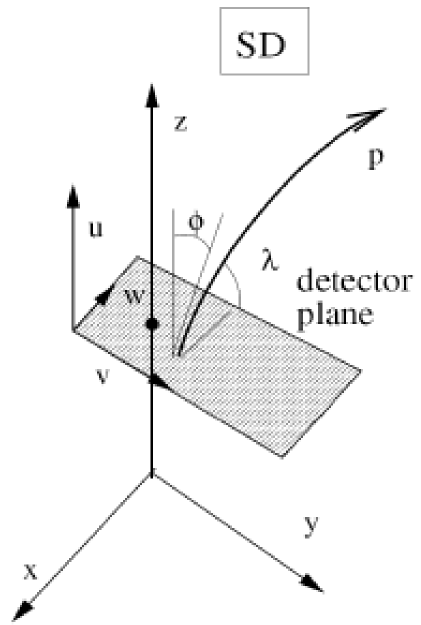


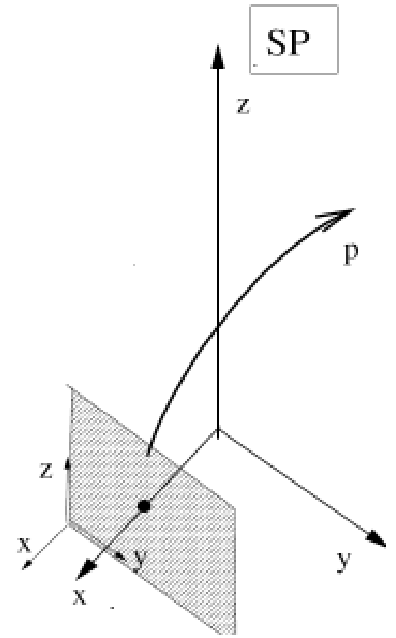
Locally it can be described as a piece of helix

Track description (3)

Many track representations







Curvilinear System – SC	Detector System – SD	Plane System – SP
q/p, λ, ϕ, y_{PERP}, z_{PERP}	q/p, v', w', v, w	q/p, y', z', y, z

Choose the one which is most suitable!

Detector plane

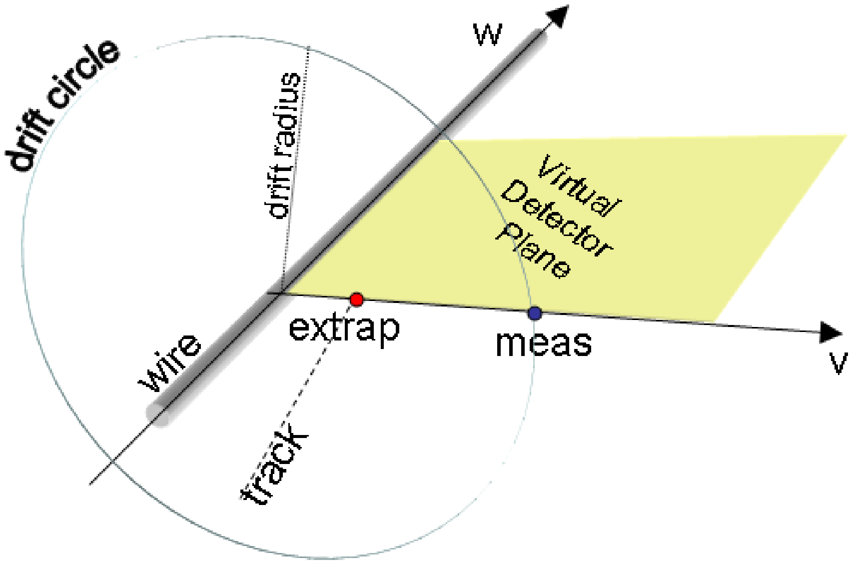
- **real** (*e.g.* a Silicon Detector plane)
- **virtual** (*e.g.* built geometrically *ad hoc*, used for non planar devices, as the STT)

- defined by:
    - origin
    - unit vectors spanning the plane

- measurement matrix H

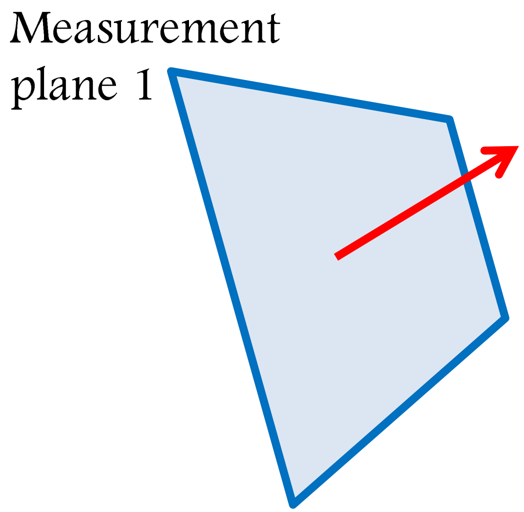
meas = **H** · state

$$\begin{pmatrix} \mathbf{v} \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{1/p} \\ \mathbf{v'} \\ \mathbf{w'} \\ \mathbf{v} \\ \mathbf{w} \end{pmatrix}$$



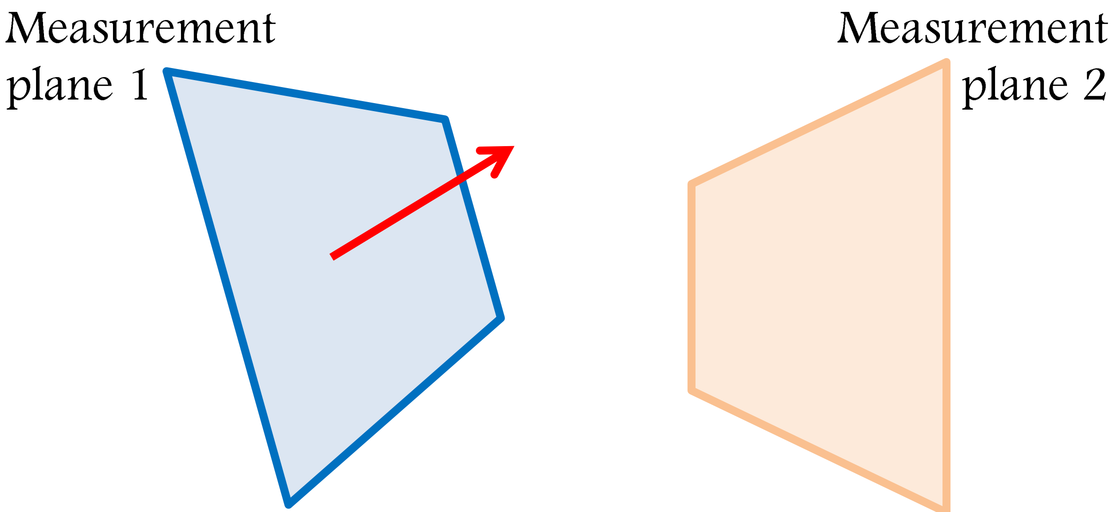
Extrapolation

Track extrapolation



5 PARAMETER VALUES

Track extrapolation

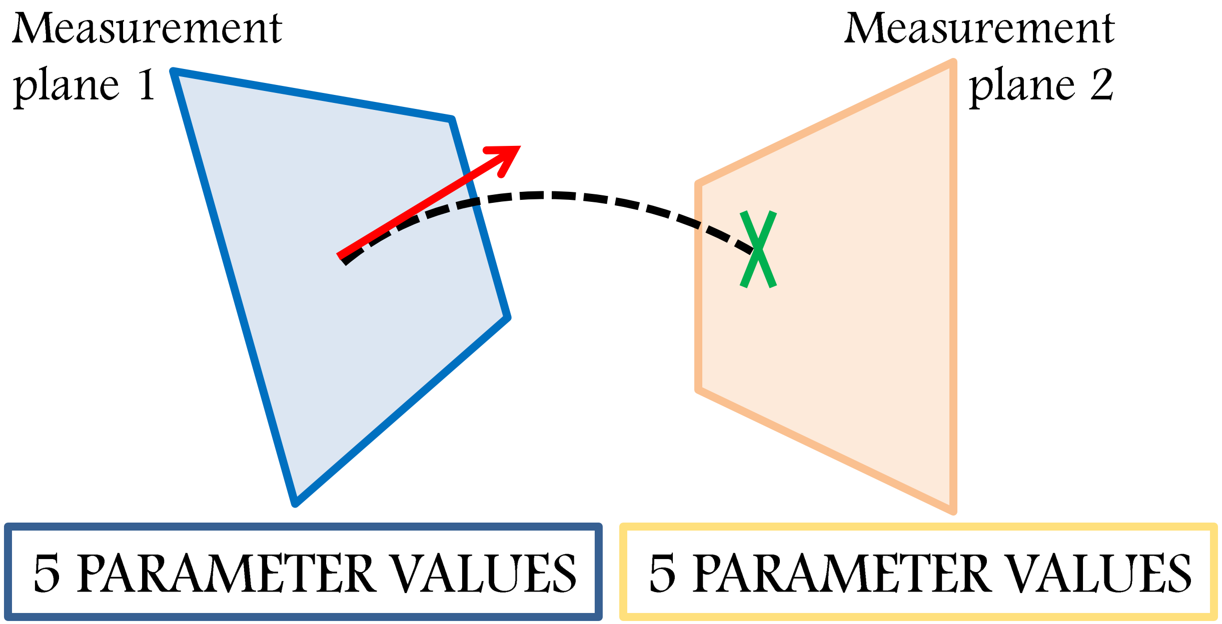


5 PARAMETER VALUES

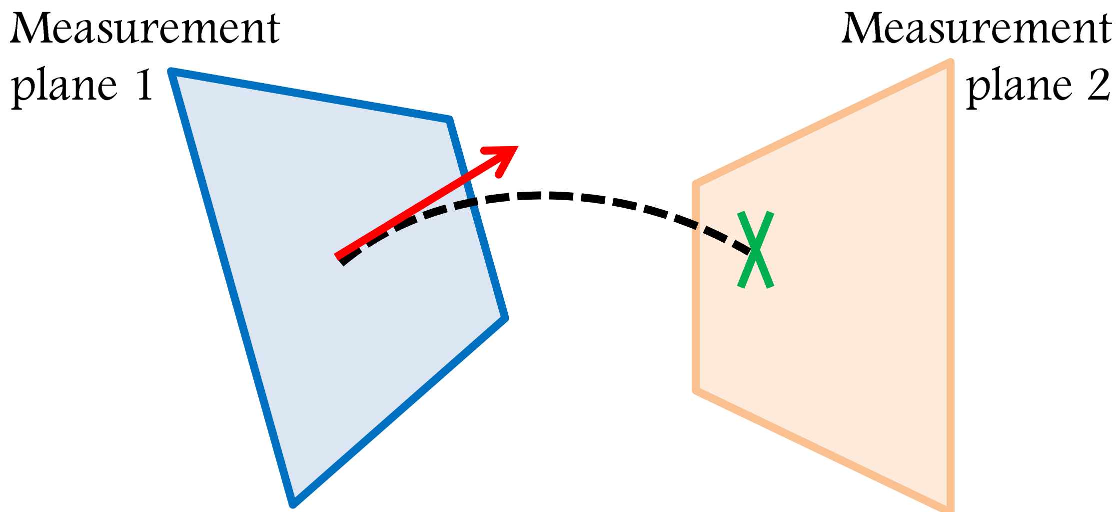
Could be:

- Another tracker, matching
- Calorimeter, cluster association
- Kalman filtering

Track extrapolation



Track extrapolation



5 PARAMETER VALUES

5 PARAMETER VALUES

If you want to compare this extrapolation to the measurement on the plane you need to know **how good your extrapolation is**

COVARIANCE MATRIX

COVARIANCE MATRIX

You need a track follower

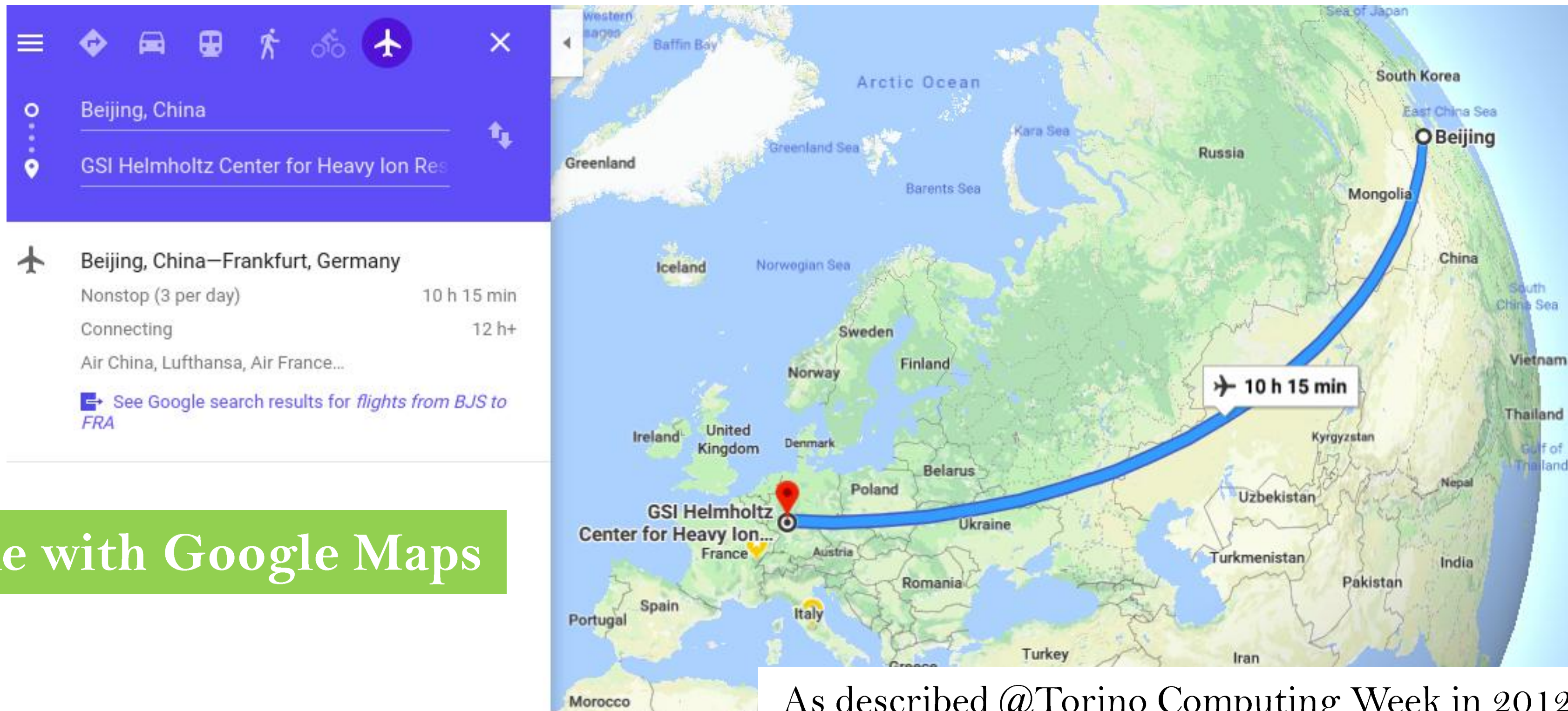
Track extrapolation

Many applications

- **trajectory calculation**
    - simple extrapolation
- **tracklet matching**
    - join track elements from different tracks on a defined plane
- **vertex optimization**
    - extrapolate tracks to a common vertex
- **track fitting**
    - with iterative methods, *as* Kalman filter

GEANE (1)

GEANE is a so-called track follower



It's like with Google Maps

As described @Torino Computing Week in 2012

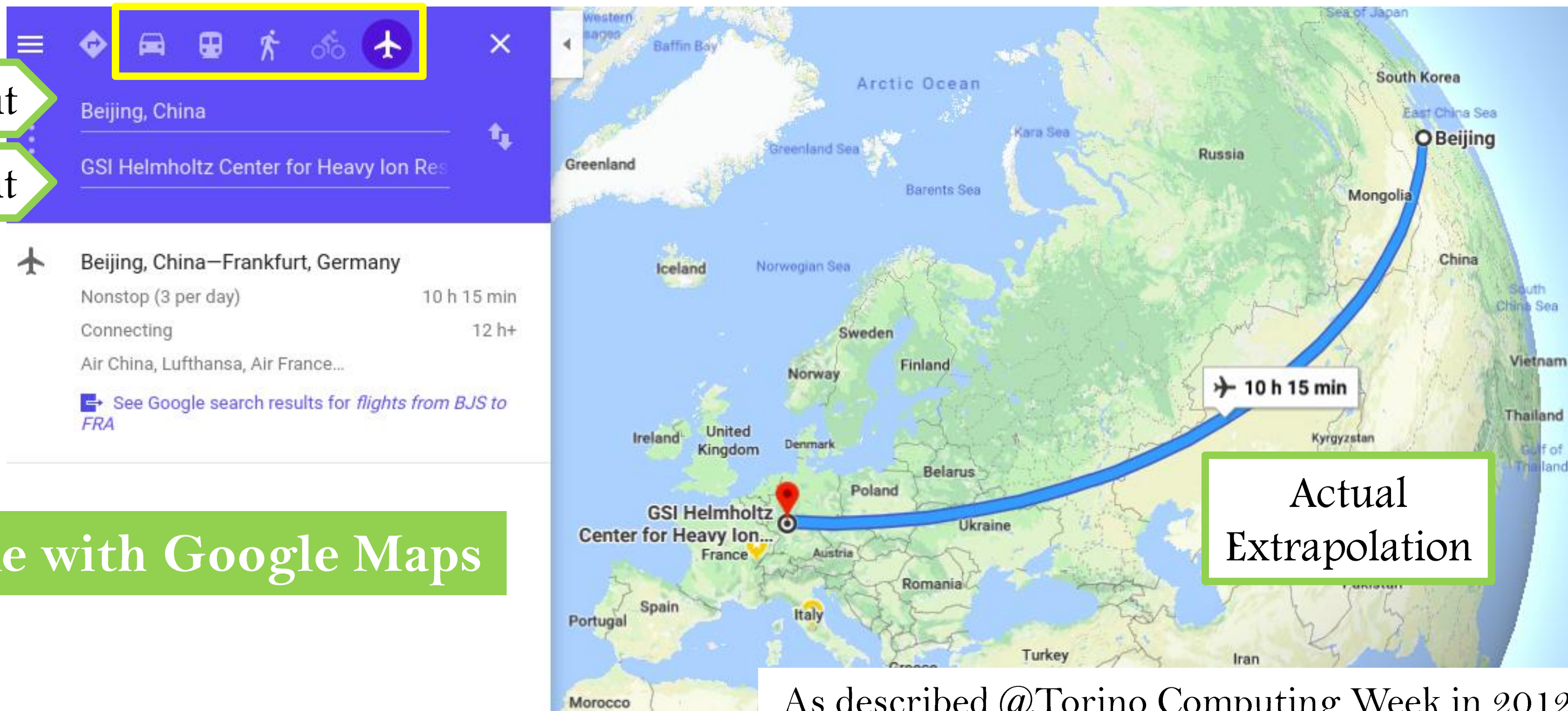
GEANE (1)

GEANE is a so-called track follower

Propagation Type

Start Point

End point



Actual Extrapolation

It's like with Google Maps

As described @Torino Computing Week in 2012

Then click

…and extrapolate

GEANE (3)

GEANE is a so-called track follower

- Mean values are the same that would be obtained with the Monte Carlo with all the **random processes off**
- No need to re-implement the **geometry**: it uses the same geometry banks of MC
- It takes into account:
    - material effects (multiple scattering, energy loss)
    - magnetic field
    - physics effects (energy loss straggling)
- it uses different **reference frames**, with the possibility to change among them
- **different kinds of propagation** are available

GEANE (2)

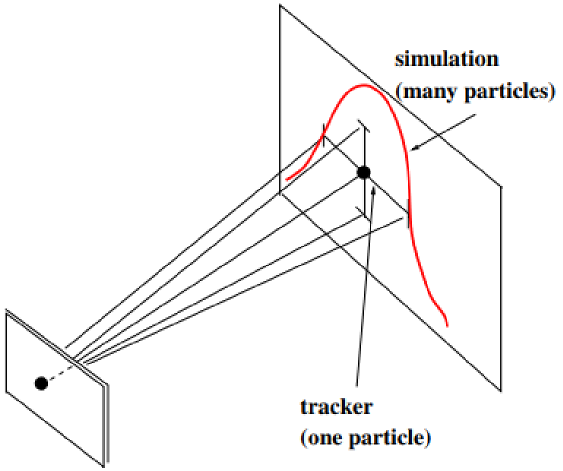
GEANE is a so-called track follower

- It is **GEANT3** based - the "**E**" stands for "error"

Simplified GEANT3 tracking algorithms
+
Error propagation routines from EMC

- It extrapolates the track parameters

MEAN VALUE | COVARIANCE MATRIX



- it is written in FORTRAN
- an interface to the *fairroot* VMC was implemented to make it available to FAIR experiments, particularly PANDA

Interface Structure

`FairGeane` actual GEANE task
reads the files with cuts and model information

`FairGeanePro` contains all the propagation stuff

`FairTrackPar`
`FairTrackParP`
`FairTrackParH` track representation in GEANE

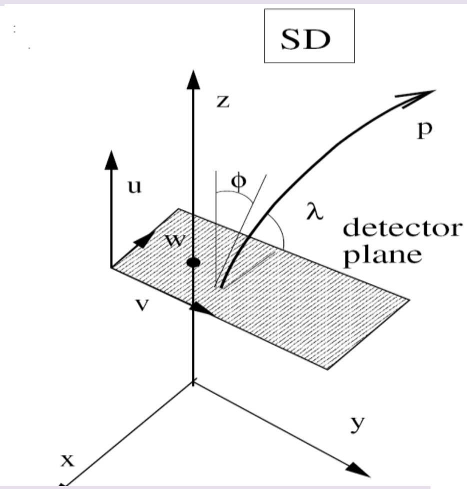
`FairGeaneUtil` *translation* to C++ of the FORTRAN routines to perform the frame transformations from/to MARS, SC, SD, SP

Track Representation

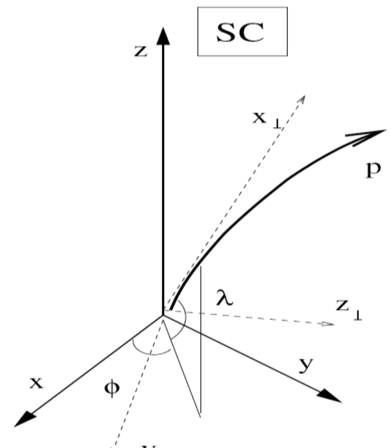
The track parameters base class is FairTrackPar

ctor

```
FairTrackPar(Double_t x, Double_t y, Double_t z,
        Double_t fx, Double_t fy, Double_t fz,
        Int_t q);
```



FairTrackParP



FairTrackParH

INPUT

Track parameters + particle charge

MARS or local frame

Propagation

You may want to propagate:

- to a defined plane
- to a defined volume
- to a defined track length
- to the point of closest approach

Propagation

You may want to propagate:

- to a defined plane
- to a defined volume
- to a defined track length
- to the point of closest approach

ALREADY IN ORIGINAL GEANE

SPECIFIC FOR PANDA CASE

- Point of closest approach **to a point** (*e.g.* 3D point from a tracker or vertex)
- Point of closest approach **to a line** (*e.g.* wire of the STT or the beam)

Propagation

You may want to propagate:

- to a defined plane
- to a defined volume
- to a defined track length
- to the point of closest approach

You may want to propagate:

- forward
- backward

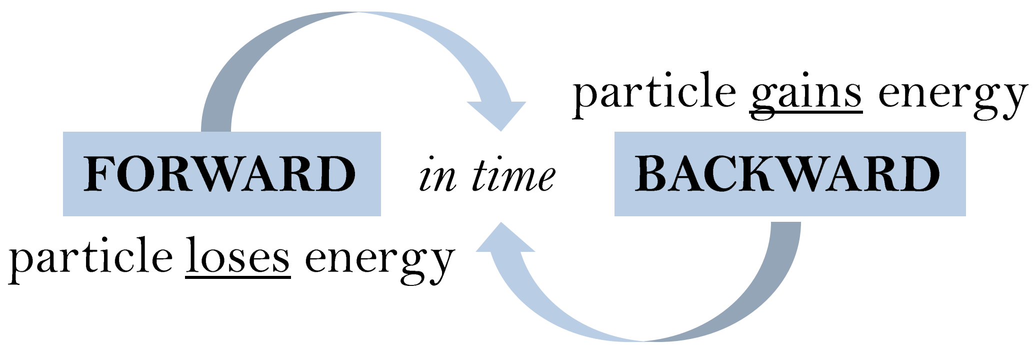
Propagation

You may want to propagate:

- to a defined plane
- to a defined volume
- to a defined track length
- to the point of closest approach

You may want to propagate:

- forward
- backward

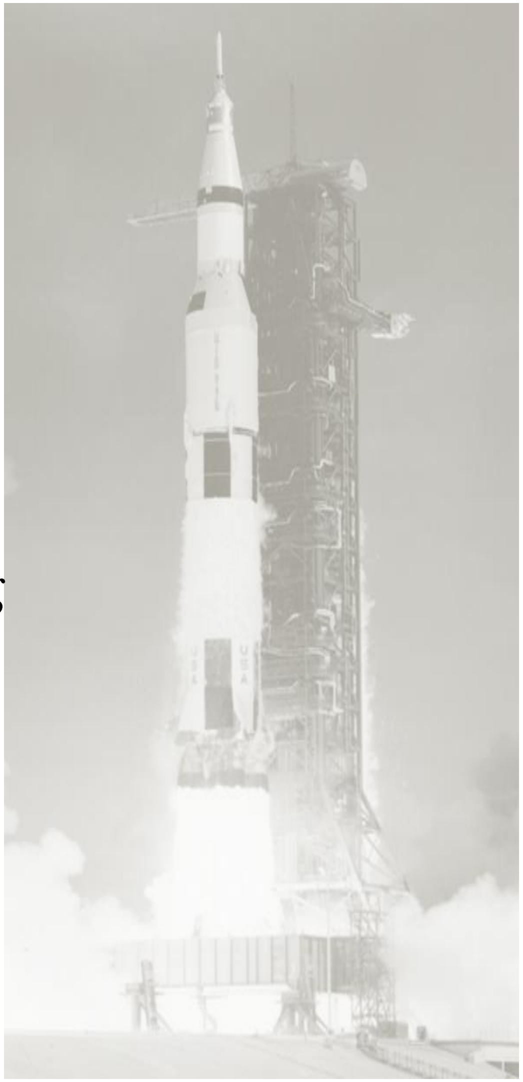


In both cases the error increases since you infer the final point

Application: Kalman filter

R. E. Kalman, MIT engineer, proposed this method in 1961 in the framework of the control and optimization theory of systems

R. Fruehwirth, in 1987, applied the method as a useful track fitting technique in particle physics

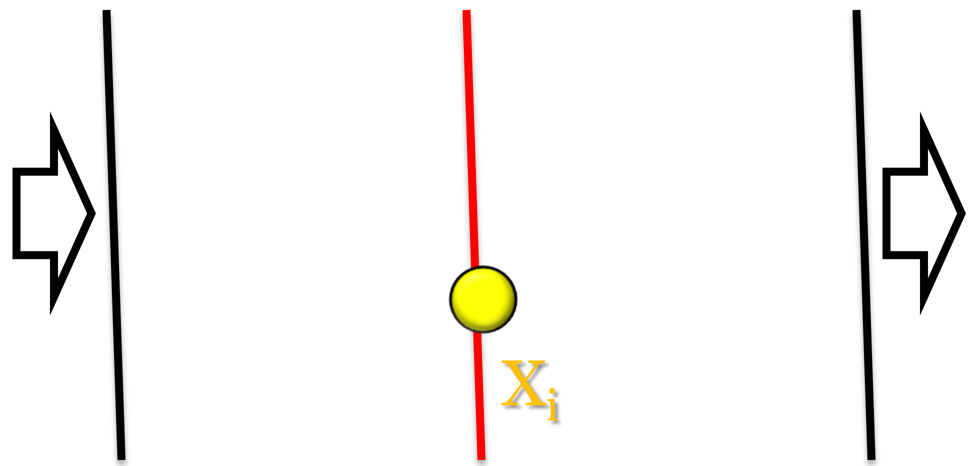


Kalman filters were applied to the navigation system during the **Apollo** program and on the **Space Shuttle**, moreover they were used for submarines, unmanned aerospace vehicles and weapons (*e.g.* cruise missiles).

Kalman filter

minimization of the following χ^2, in 5-dimensional space

$$\chi^2(\mathbf{f}) = (\mathbf{x} - \mathbf{f})\, \mathbf{V}\, (\mathbf{x} - \mathbf{f}) + (\mathbf{e} - \mathbf{f})\, \mathbf{W}\, (\mathbf{e} - \mathbf{f})$$



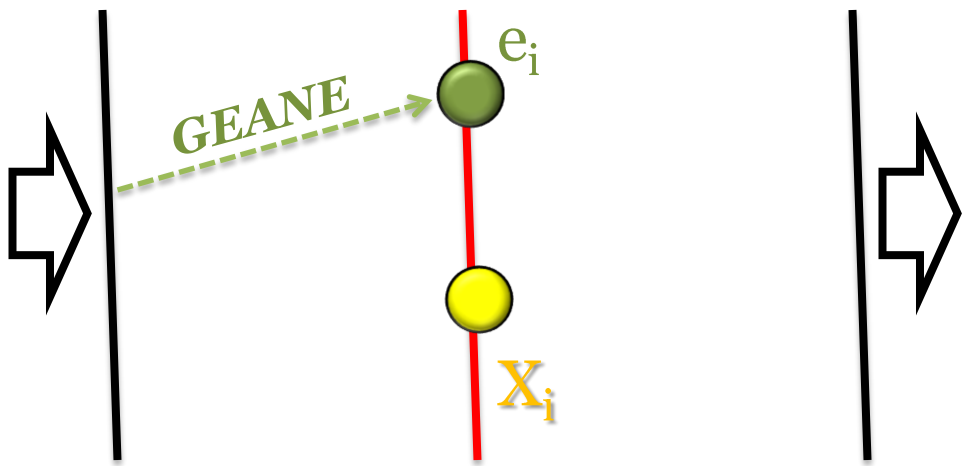
previous detector planes | i-th detector plane | following detector planes

X_i = measurement

Kalman filter

minimization of the following χ^2, in 5-dimensional space

$$\chi^2(\mathbf{f}) = (\mathbf{x} - \mathbf{f})\ \mathbf{V}\ (\mathbf{x} - \mathbf{f}) + \boxed{(\mathbf{e} - \mathbf{f})\ \mathbf{W}\ (\mathbf{e} - \mathbf{f})}$$



previous detector planes | i-th detector plane | following detector planes

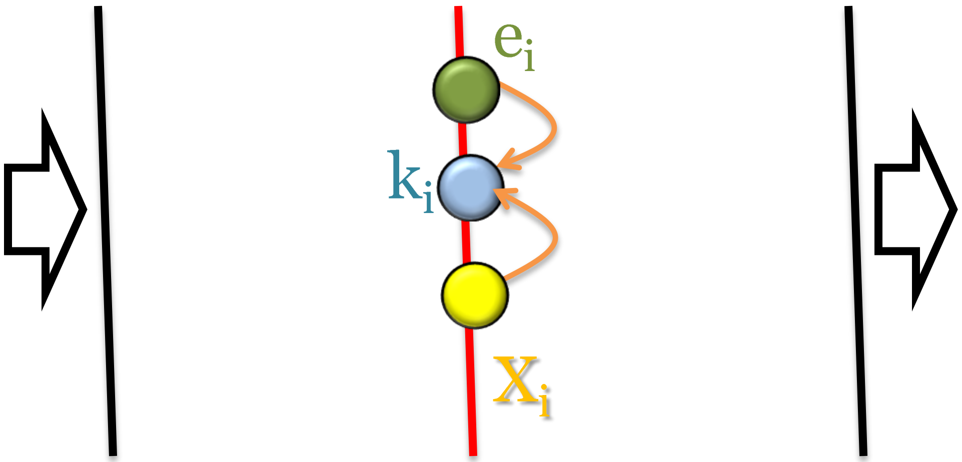
X_i = measurement

e_i = extrapolation

Kalman filter

minimization of the following χ^2, in 5-dimensional space

$$\chi^2(\mathbf{f}) = \boxed{(\mathbf{x} - \mathbf{f})\, \mathbf{V}\, (\mathbf{x} - \mathbf{f})} + (\mathbf{e} - \mathbf{f})\, \mathbf{W}\, (\mathbf{e} - \mathbf{f})$$



previous detector planes

i-th detector plane

following detector planes

X_i = measurement

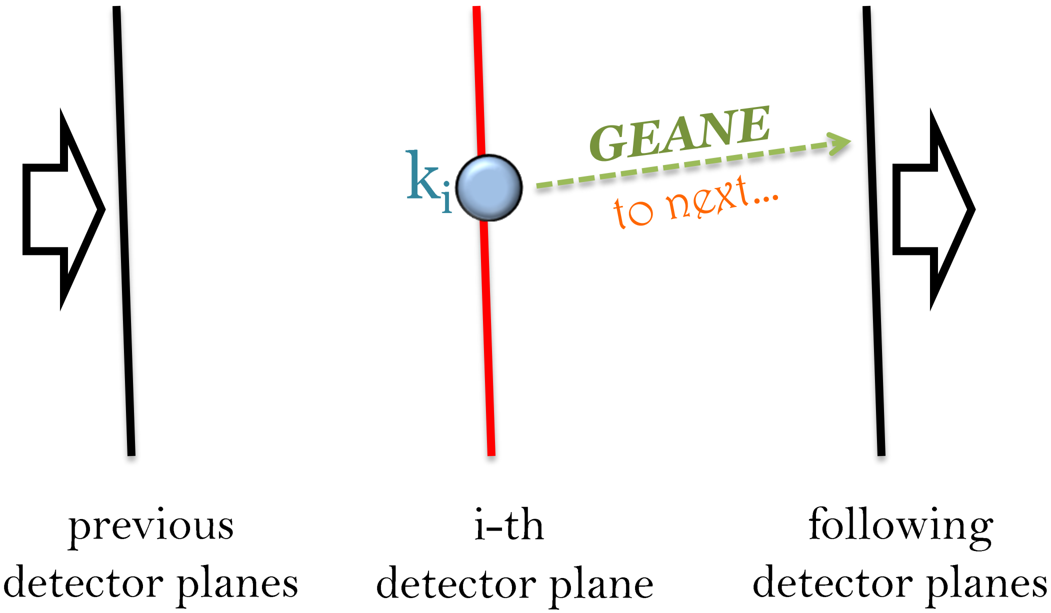
e_i = extrapolation

k_i = filtered

Kalman filter

minimization of the following χ^2, in 5-dimensional space

$$\chi^2(\mathbf{f}) = (\mathbf{x} - \mathbf{f})\ \mathbf{V}\ (\mathbf{x} - \mathbf{f}) + (\mathbf{e} - \mathbf{f})\ \mathbf{W}\ (\mathbf{e} - \mathbf{f})$$



X_i = measurement
e_i = extrapolation
k_i = filtered

Kalman filter

It can be seen as composed by three steps

PREDICTION

$$\boldsymbol{e}_i \equiv \boldsymbol{e}_i(\boldsymbol{k}_{i-1}) = \boldsymbol{G}(\boldsymbol{k}_{i-1})$$

$$\boldsymbol{\sigma}^2[\boldsymbol{e}_i] = \boldsymbol{T}(l_i, l_{i-1})\,\boldsymbol{\sigma}^2[\boldsymbol{k}_{i-1}]\,\boldsymbol{T}^T(l_i, l_{i-1}) + \boldsymbol{W}^{-1}_{i-1,i}$$

FILTERING

$$\boldsymbol{k}_i = \boldsymbol{\sigma}^2[\boldsymbol{k}_i]\left(\boldsymbol{\sigma}^{-2}[\boldsymbol{e}_i]\,\boldsymbol{e}_i + \boldsymbol{V}_i\boldsymbol{x}_i\right)$$

$$\boldsymbol{\sigma}^{-2}[\boldsymbol{k}_i] = \boldsymbol{\sigma}^{-2}[\boldsymbol{e}_i] + \boldsymbol{V}_i$$

SMOOTHING

$$\boldsymbol{f}_i = \boldsymbol{k}_i + \boldsymbol{A}_i\left(\boldsymbol{f}_{i+1} - \boldsymbol{e}_{i+1}\right)$$

$$\boldsymbol{\sigma}^2[\boldsymbol{f}_i] = \boldsymbol{\sigma}^2[\boldsymbol{k}_i] + \boldsymbol{A}_i\left(\boldsymbol{\sigma}^2[\boldsymbol{f}_{i+1}] - \boldsymbol{\sigma}^2[\boldsymbol{e}_{i+1}]\right)\boldsymbol{A}_i^T$$

$$\boldsymbol{A}_i = \boldsymbol{\sigma}^2[\boldsymbol{k}_i]\,\boldsymbol{T}^T(l_{i+1}, l_i)\,\boldsymbol{\sigma}^{-2}[\boldsymbol{e}_{i+1}]$$

Kalman filter

It can be seen as composed by three steps

PREDICTION

EXTRAPOLATION

GEANE

FILTERING

WEIGHTED AVERAGE

SMOOTHING

$$\boldsymbol{f}_i = \boldsymbol{k}_i + \boldsymbol{A}_i \left(\boldsymbol{f}_{i+1} - \boldsymbol{e}_{i+1}\right)$$

BACKTRACKING

GEANE

$$\boldsymbol{A}_i = \boldsymbol{\sigma}^2[\boldsymbol{k}_i]\, \boldsymbol{T}^T(l_{i+1}, l_i)\, \boldsymbol{\sigma}^{-2}[\boldsymbol{e}_{i+1}]$$

Error extrapolation

PREDICTION

$$e_i \equiv e_i(k_{i-1}) = G(k_{i-1})$$

$$\sigma^2[e_i] = T(l_i, l_{i-1})\, \sigma^2[k_{i-1}]\, T^T(l_i, l_{i-1}) + W^{-1}_{i-1,i}$$

TRANSPORT RANDOM

- the extrapolated covariance is composed by:
  - the error computed so far, *transported* on i-th plane
  - a random part during extrapolation, due to multiple scattering and energy loss

- the **transport matrix** (a.k.a. **Jacobian**) contains different contributions:

$$T_{ij}(l, l_0) = \frac{\partial e^i(l)}{\partial e^j(l_0)}$$

$$\mathbf{T} = \mathbf{I} + \left(\mathbf{A}_{x+dx} + \mathbf{B}_{x+dx}\right) \cdot \mathbf{dx}$$

Transport matrix (1)

$$\mathbf{T} = \boxed{\mathbf{I}} + (\mathbf{A}_{\mathbf{x+dx}} + \mathbf{B}_{\mathbf{x+dx}}) \cdot \mathbf{dx}$$

propagates the initial error as it is

Transport matrix (2)

$$\mathbf{T} = \mathbf{I} + (\mathbf{A}_{x+dx} + \mathbf{B}_{x+dx}) \cdot dx$$

contribution of the error on the direction
(also in absence of magnetic field)

$$\mathbf{A} = \begin{pmatrix} \dfrac{\left(\dfrac{\partial^2 \, 1/p}{\partial l^2}\right)}{\left(\dfrac{\partial \, 1/p}{\partial l}\right)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos\lambda & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Transport matrix (3)

$$T = I + (A_{x+dx} + B_{x+dx}) \cdot dx$$

contribution due to the magnetic field

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ H_2 & 0 & -H_0/p & H_2 H_3/p^2 & -H_2^2/p^2 \\ -\dfrac{H_3}{\cos\lambda} & \dfrac{H_0}{p\cos^2\lambda} & \dfrac{H_2 \tan\lambda}{p} & -\dfrac{H_3^2}{p^2\cos\lambda} & -\dfrac{H_2 H_3}{p^2\cos\lambda} \\ 0 & 0 & 0 & 0 & -\dfrac{H_3 \tan\lambda}{p} \\ 0 & 0 & 0 & \dfrac{H_3 \tan\lambda}{p} & 0 \end{pmatrix}$$

Random Matrix

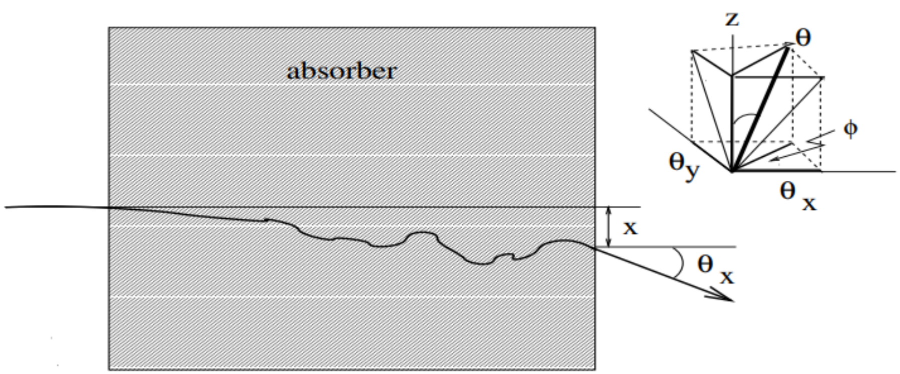
PREDICTION

$$\begin{aligned} \boldsymbol{e}_i &\equiv \boldsymbol{e}_i(\boldsymbol{k}_{i-1}) = \boldsymbol{G}(\boldsymbol{k}_{i-1}) \\ \boldsymbol{\sigma}^2[\boldsymbol{e}_i] &= \boldsymbol{T}(l_i, l_{i-1})\, \boldsymbol{\sigma}^2[\boldsymbol{k}_{i-1}]\, \boldsymbol{T}^T(l_i, l_{i-1}) + \boldsymbol{W}^{-1}_{i-1,i} \end{aligned}$$

TRANSPORT **RANDOM**

- multiple scattering
- energy loss

Coulomb multiple scattering

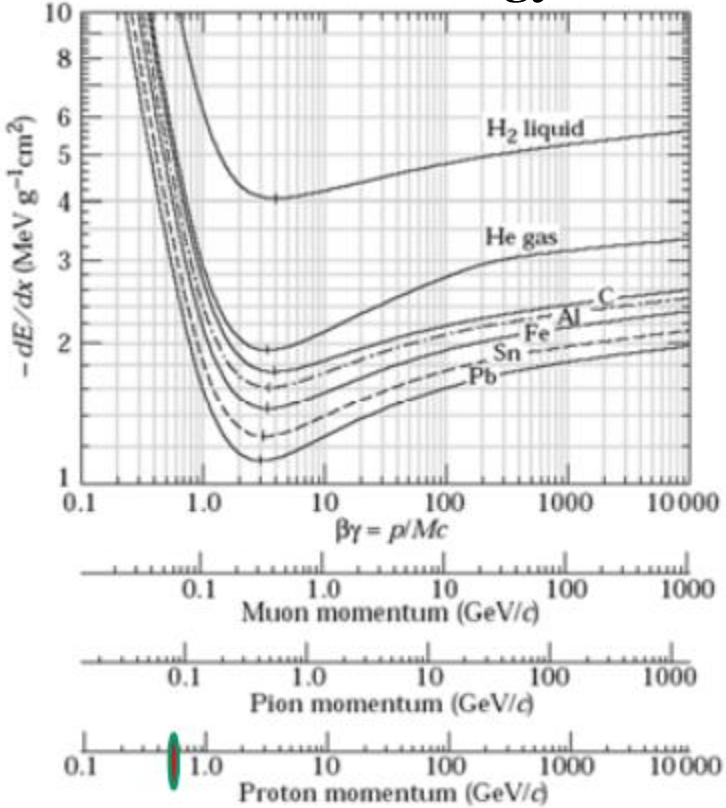


$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & <\theta_p^2> & 0 & 0 & -\frac{<\theta_p^2> \ dl}{2} \\ 0 & 0 & \frac{<\theta_p^2>}{\cos^2\lambda} & \frac{<\theta_p^2> \ dl}{(2\cos\lambda)} & 0 \\ 0 & 0 & \frac{<\theta_p^2> \ dl}{(2\cos\lambda)} & \frac{<\theta_p^2> \ (dl)^2}{3} & \\ 0 & -\frac{<\theta_p^2> \ dl}{2} & 0 & 0 & \frac{<\theta_p^2> \ (dl)^2}{3} \end{pmatrix}$$

Energy loss straggling

Was specifically updated in GEANE for PANDA

Mean value of energy loss

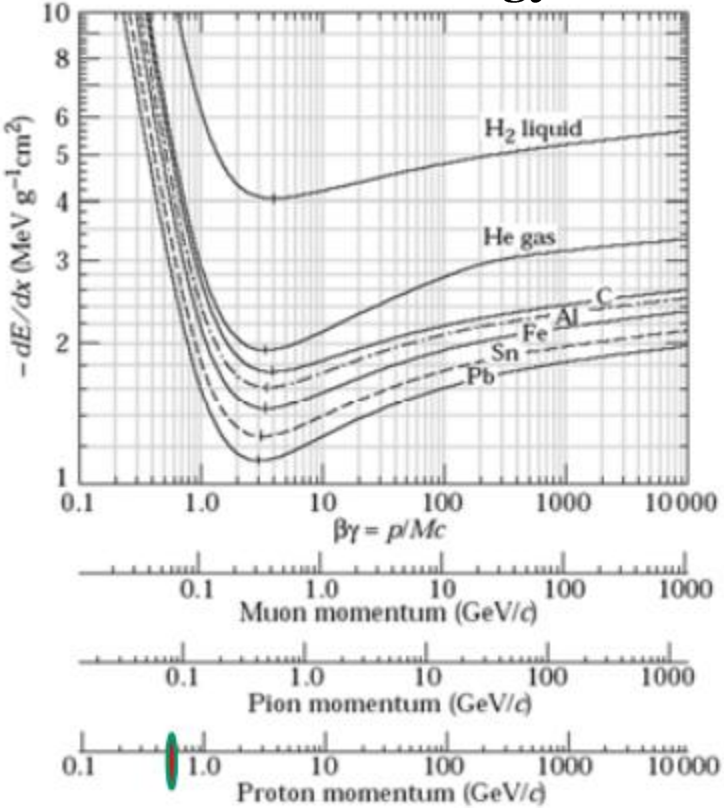


What about the fluctuations?

Energy loss straggling

Was specifically updated in GEANE for PANDA

Mean value of energy loss



What about the fluctuations?

They depend on: $\kappa = \dfrac{\xi}{E_{\max}}$

$$\xi = 153.4 \frac{z^2 Z}{\beta^2 A} \rho d \quad (\text{keV})$$

$$E_{\max} = \frac{2 m_e \beta^2 \gamma^2}{1 + 2\gamma m_e/m + (m_e/m)^2}$$

strong dependence on the absorber thickness

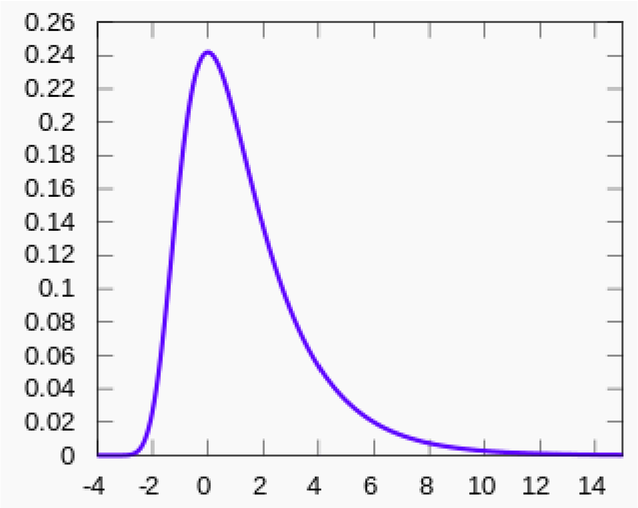
Energy loss straggling

Was specifically updated in GEANE for PANDA

Dependence on k:

1. **k > 10** – Gaussian
2. **0.01 < k < 10** – Vavilov
3. **k < 0.01 & Ncoll > 50** – Landau
4. **k < 0.01 & Ncoll < 50** – sub-Landau region

THIN ABSORBERS



$$\sigma^2(1/p) = \frac{E^2}{p^6}\sigma^2(E)$$

What happens if the variance is **infinite**?

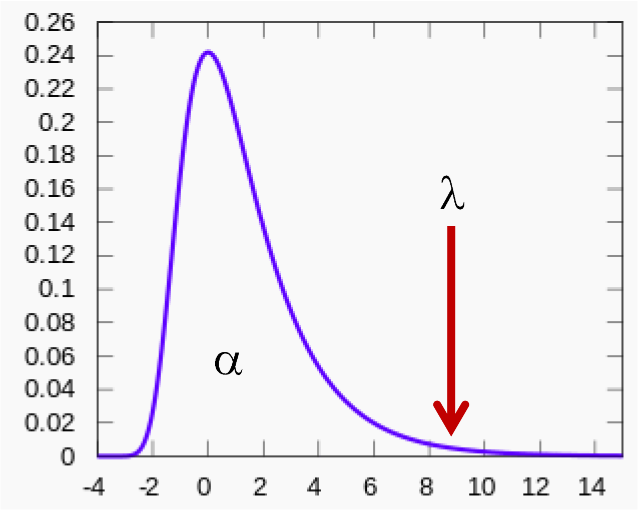
Energy loss straggling

Was specifically updated in GEANE for PANDA

Dependence on k:

1. **k > 10** – Gaussian
2. **0.01 < k < 10** – Vavilov
3. **k < 0.01 & Ncoll > 50** – Landau
4. **k < 0.01 & Ncoll < 50** – sub-Landau region

THIN ABSORBERS



$$\sigma^2(1/p) = \frac{E^2}{p^6}\sigma^2(E)$$

What happens if the variance is **infinite**?

Truncaded Landau
or Urban (for very thin absorbers)

Energy loss straggling

Was specifically updated in GEANE for PANDA

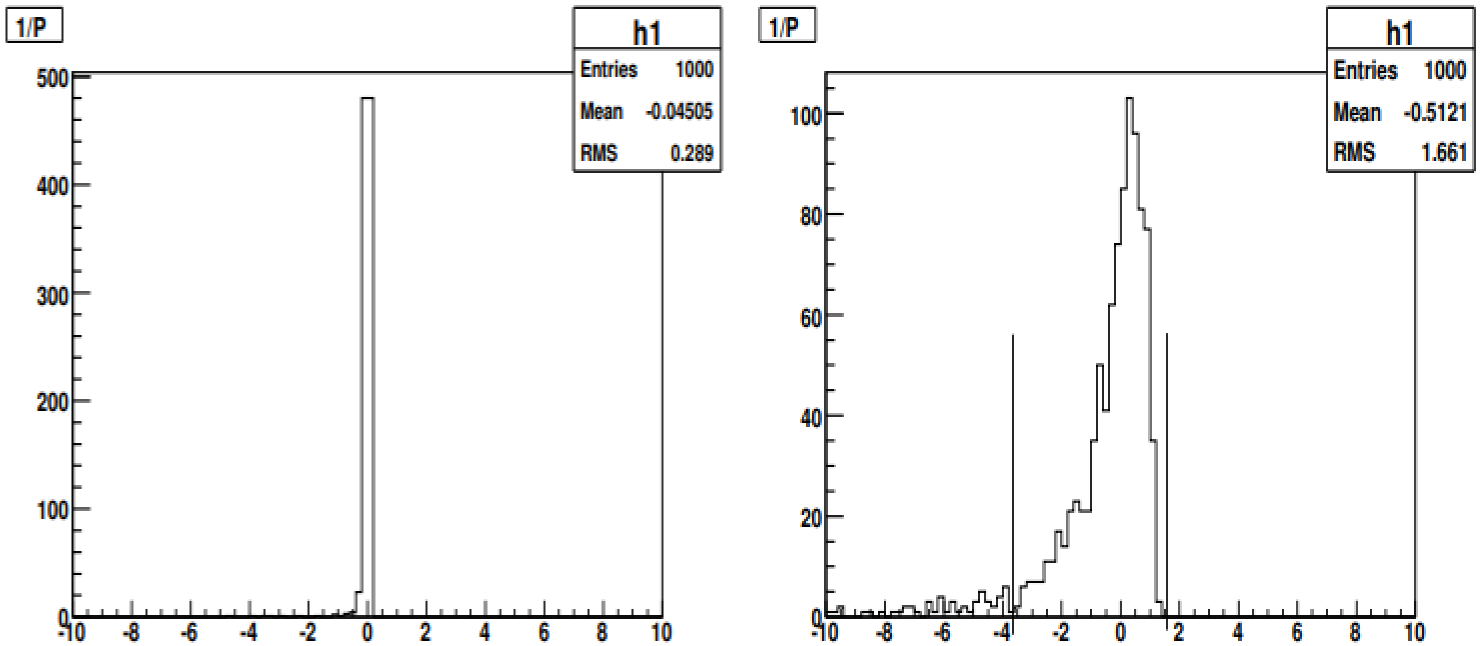
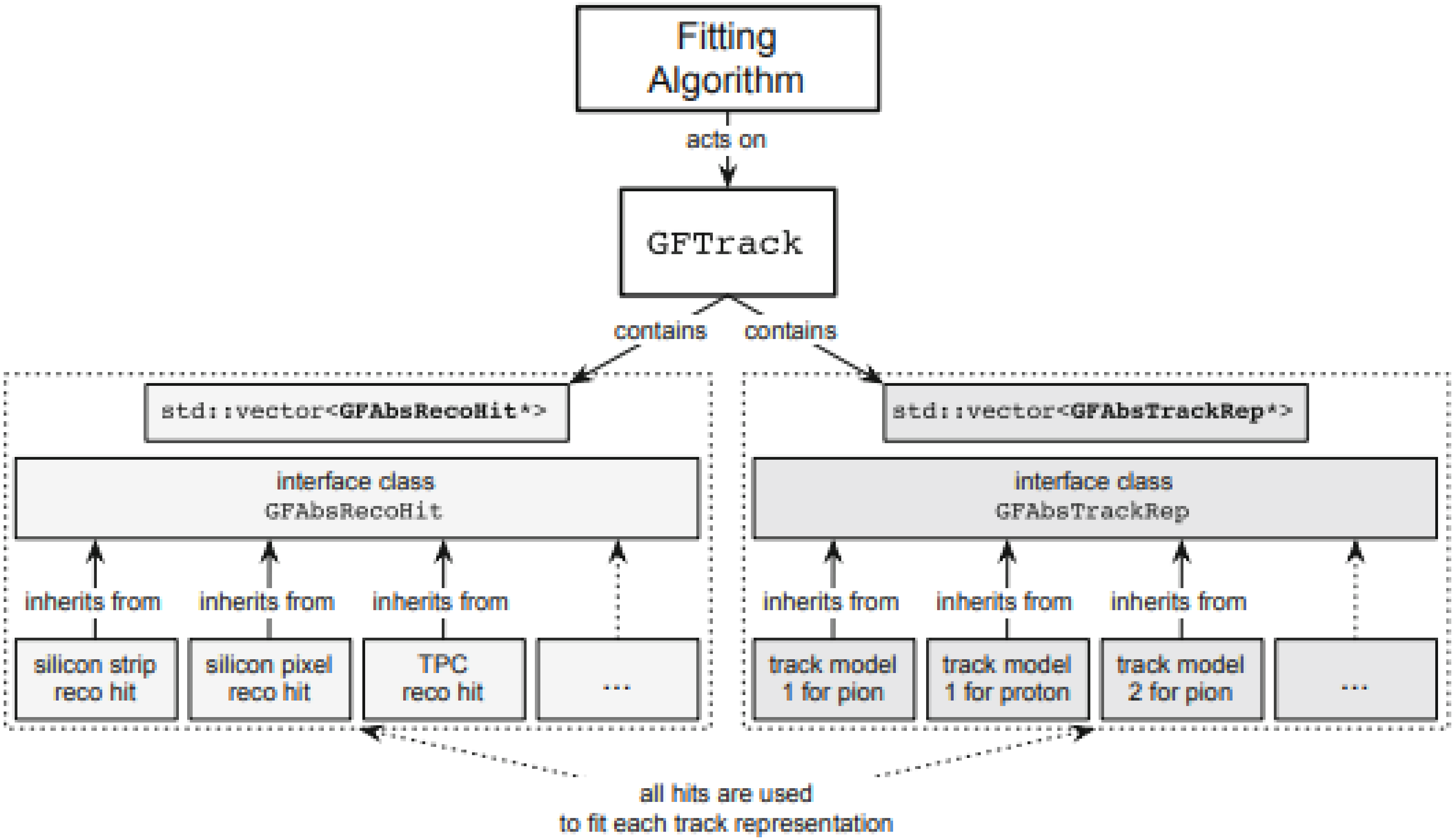


Figure 10: Pull distribution $\Delta(1/p)/\sigma$ for 1 GeV muons after passing through the PANDA straw tube detector. Left: Standard GEANE result (RMS$\simeq$ 0.3 in the displayed window); right: result after the modification with $\alpha = 0.995$ (see the text). The region between the vertical lines has RMS= 1.03.

…moreover

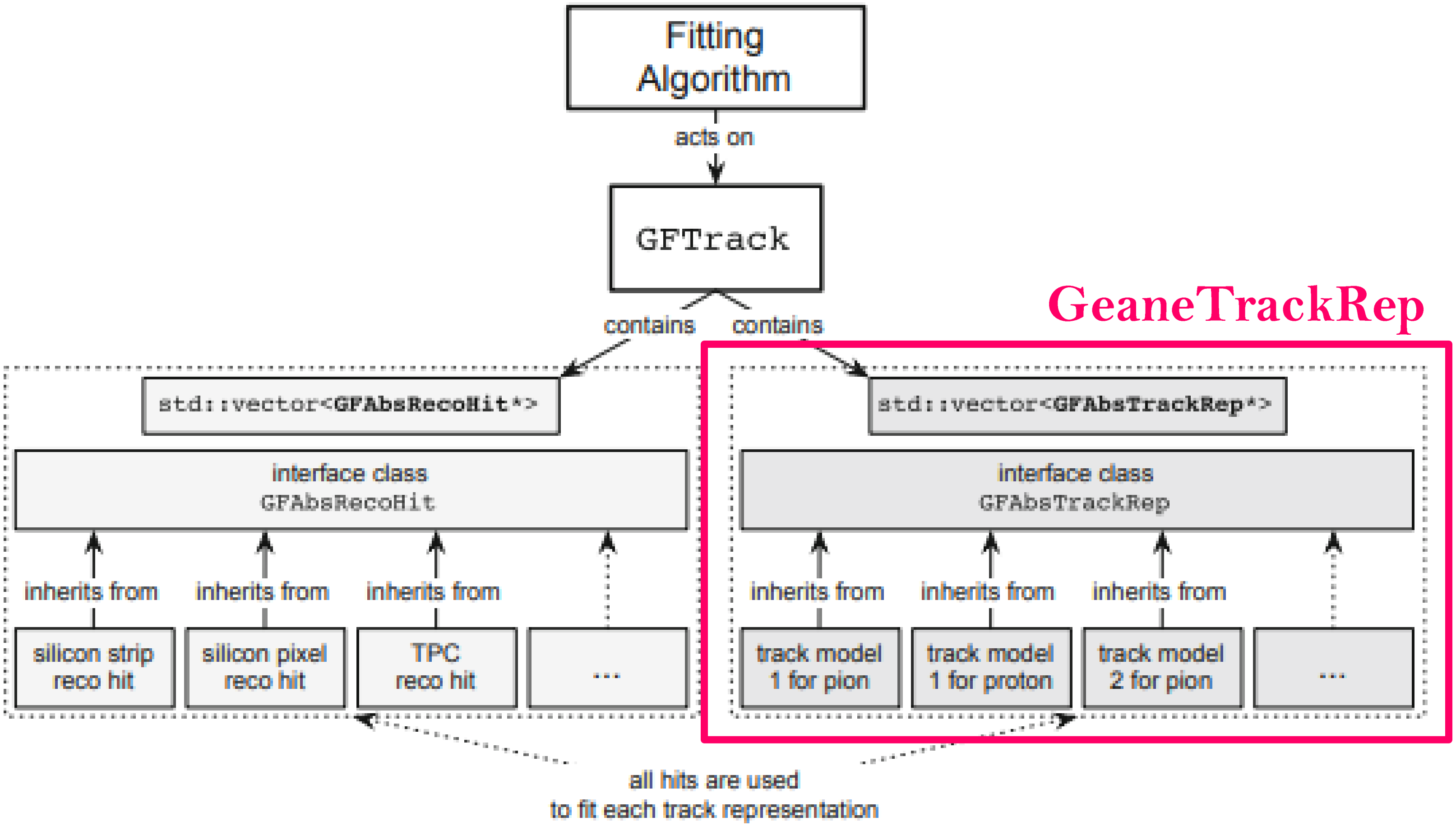
GEANE in GENFIT1

GENFIT class structure Kalman fit package in *pandaroot* : **GENFIT**



GEANE in GENFIT1

GENFIT class structure Kalman fit package in *pandaroot* : **GENFIT**



GEANE in GENFIT1

- the filtering is always performed on **planes**
- the extrapolation is **not visible** to the user, but GEANE is called inside the propagation stage
- first implementation of smoothing: **backtracking**
- Kalman filter back and forth for *n* **iterations** to refine the results
- alternative to GEANE: RKTrackRep

The heir: GEANT4E

like father like son

GEANT4E:

Error propagation for track reconstruction inside the GEANT4 framework

Pedro Arce (CIEMAT)

CHEP 2006, Mumbai, 13-17th February 2006

Karlsruhe Institute of Technology

Geant4 Error Propagation in CMS Track Reconstruction

LPCC Detector Simulation Workshop, CERN, 18-19 March 2014

Thomas Hauth [CERN]

www.kit.edu - www.cern.ch

Belle II

Geant4e Track Extrapolation in the Belle II Experiment

Leo Piilonen, Virginia Tech
on behalf of the Belle II Collaboration

DPF 2017 Fermilab August 2017

This work supported by U.S. DEPARTMENT OF ENERGY | Office of Science

Some References

geane

• V. Innocente, M. Maire and E. Nagy, GEANE : Average Tracking and Error propagation package, CERN Program Library W5013-E (1991)

• W. Wittek, EMC Collaboration reports, EMC/80/15, EMC/CSW/80/39, 81/13, 81/19, *unpublished*

• A. Strandlie and W. Wittek, *Propagation of covariance matrices of track parameters in homogenenous magnetic fields in CMS*, CMS 2006/001

Interface to *fair/pandaroot*

• A. Fontana *et al.*, *Use of GEANE for tracking in Virtual Monte Carlo*, Journ. of Phys.: Conf. Series, 119 (2008) 032018

• http://panda-wiki.gsi.de/cgi-bin/view/Computing/GeaneInterface

genfit

• C. Hoeppner *et al.*, *A novel generic framework for track fitting in complex detector systems*, NIM **A620** (2010), 518-525

Thank you!

Grazie!