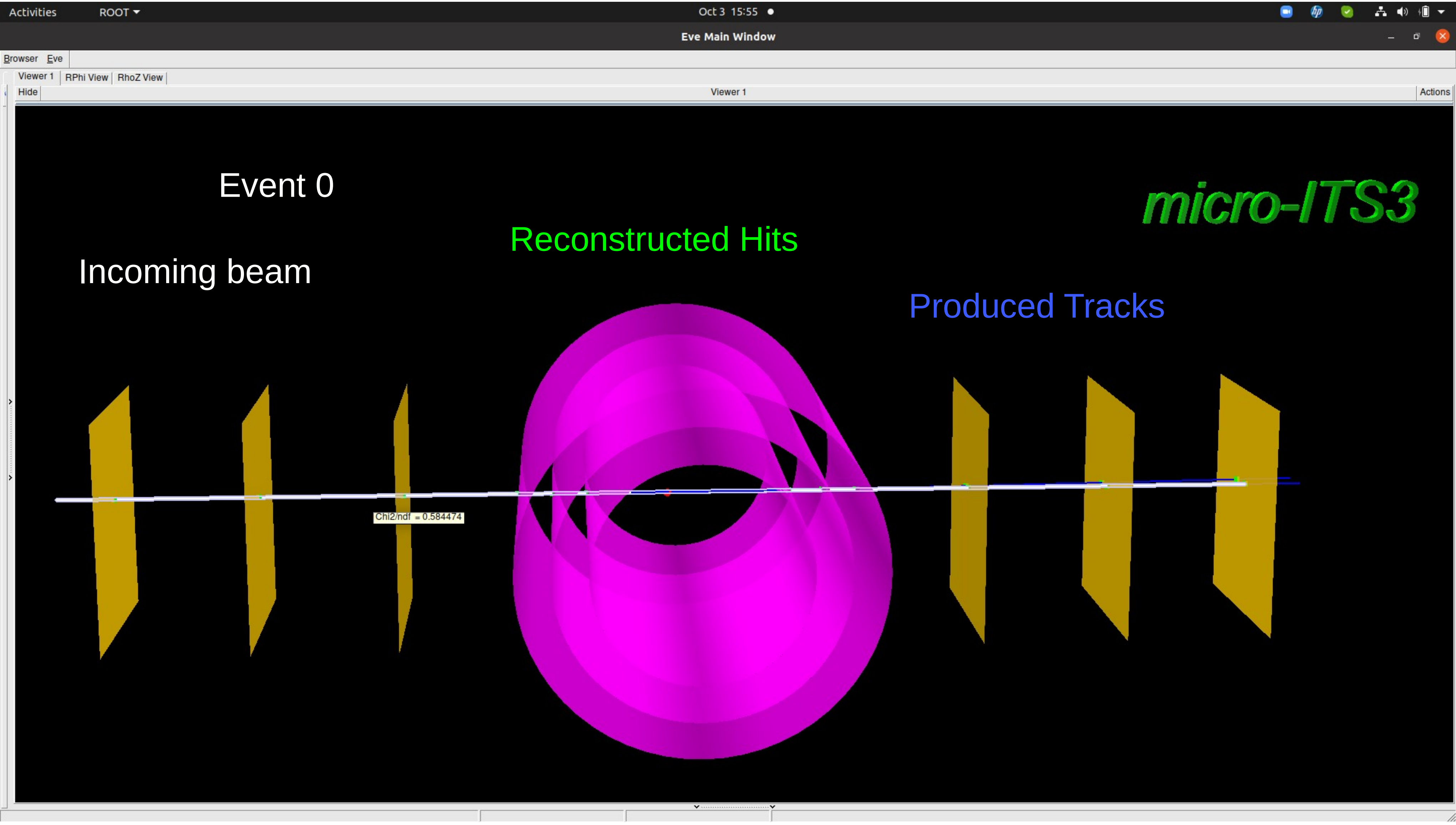
Tracking using Kalman filter in ALICE Framework

Shyam Kumar
University and INFN Bari, Italy

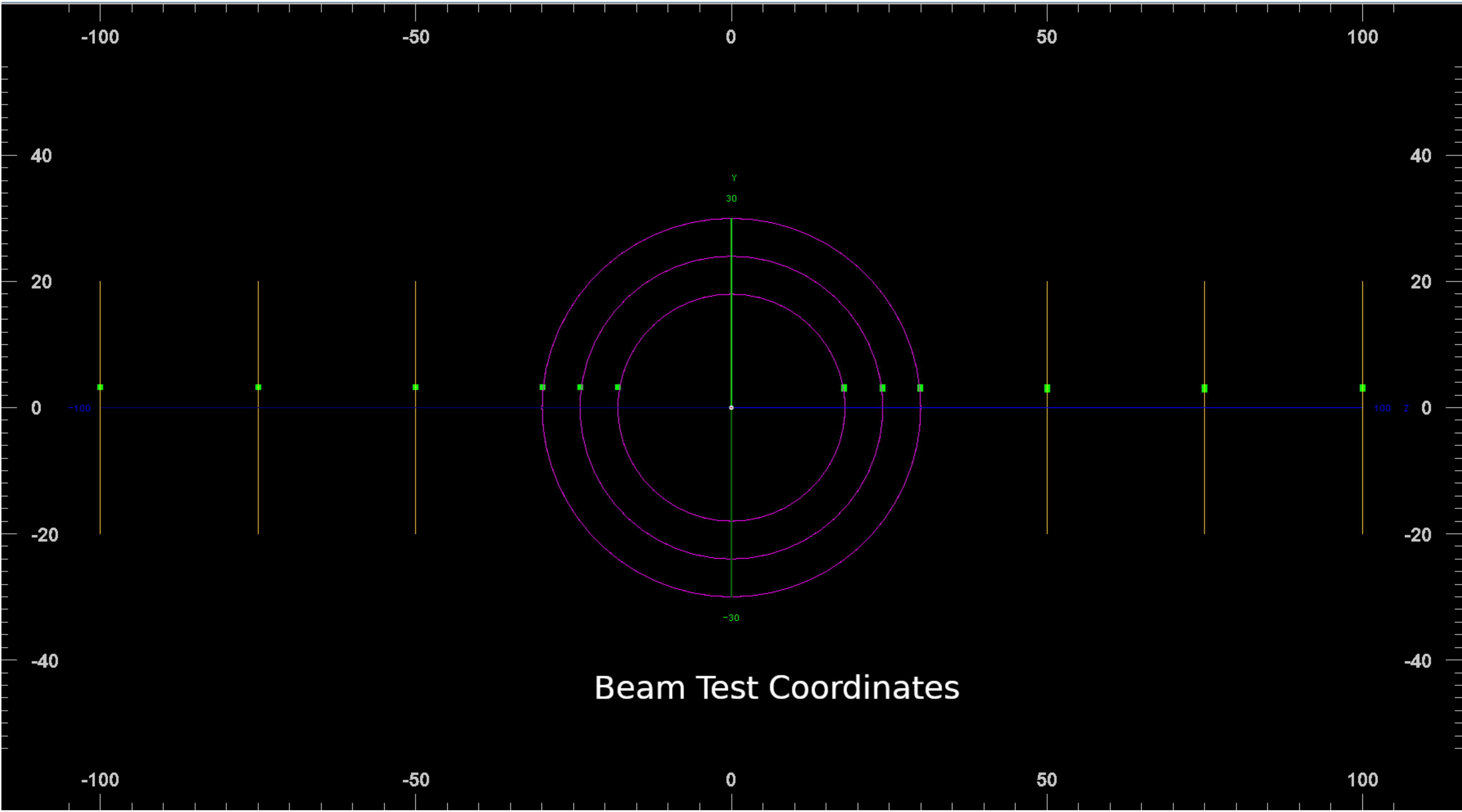
https://agenda.infn.it/event/1096/contributions/6159/attachments/4504/4980/Rotondi_3.pdf

http://www.le.infn.it/lhcschool/talks/Ragusa.pdf

Event display



Beam Test


-100
-50
0
50
100
40
20
0
-20
-40
Y
30
-30
-100
100
Z
Beam Test Coordinates

Let's understand lines in 3D for fitting them

Vector equation of a line in 3D:

$$\vec{r} = \vec{r}_0 + \vec{a}$$

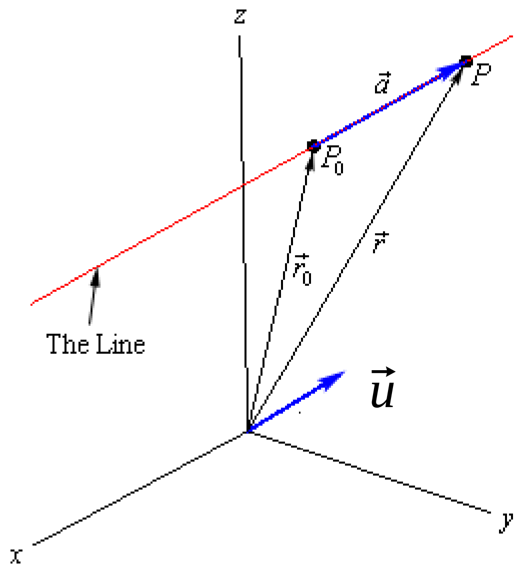
If u is the unit vector along the line and t is parameter:

$$\vec{r} = \vec{r}_0 + t\,\vec{u}$$

$$(x, y, z) = (x_0, y_0, z_0) + t\,(a, b, c)$$

$$x = x_0 + t\,a$$
$$y = y_0 + t\,b$$
$$z = z_0 + t\,c$$

$$t = \frac{(x - x_0)}{a} = \frac{(y - y_0)}{b} = \frac{(z - z_0)}{c}$$



To define a line in 3D, we should know a point (x_0,y_0,z_0) on the line and unit vector (a,b,c) in the direction of line
Therefore 6 parameters to minimize!!!

https://tutorial.math.lamar.edu/classes/calciii/eqnsoflines.aspx

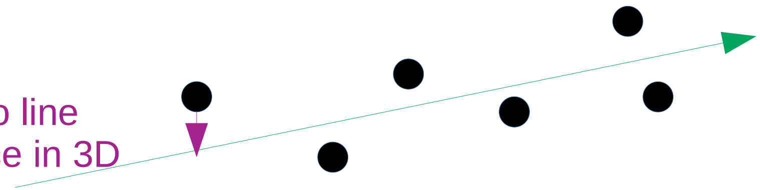
General line in 3D with 6 parameters:

$$x = p_0 + t\,p_1$$
$$y = p_2 + t\,p_3$$
$$z = p_4 + t\,p_5$$

We have given hit points (x_0, y_0, z_0), (x_1, y_1, z_1).....

Minimize of Chi2 to best estimate the parameters

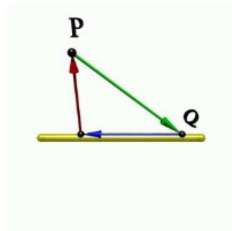
Point to line distance in 3D



DISTANCE POINT-LINE (3D). If P is a point in space and L is the line $\vec{r}(t) = Q + t\vec{u}$, then

$$d(P,L) = \frac{|(\vec{PQ}) \times \vec{u}|}{|\vec{u}|}$$

is the distance between P and the line L. Proof: the area divided by base length is height of parallelogram.



Here we are minimizing distance correspond to a general point

```
// define the parametric line equation
void line(double t, const double *p, double &x, double &y, double &z) {
          // a parametric line is define from 6 parameters but 4 are independent
          // x0,y0,z0,z1,y1,z1 which are the coordinates of two points on the line
          x = p[0] + p[1]*t;
          y = p[2] + p[3]*t;
          z = p[4] + p[5]*t;
}
```

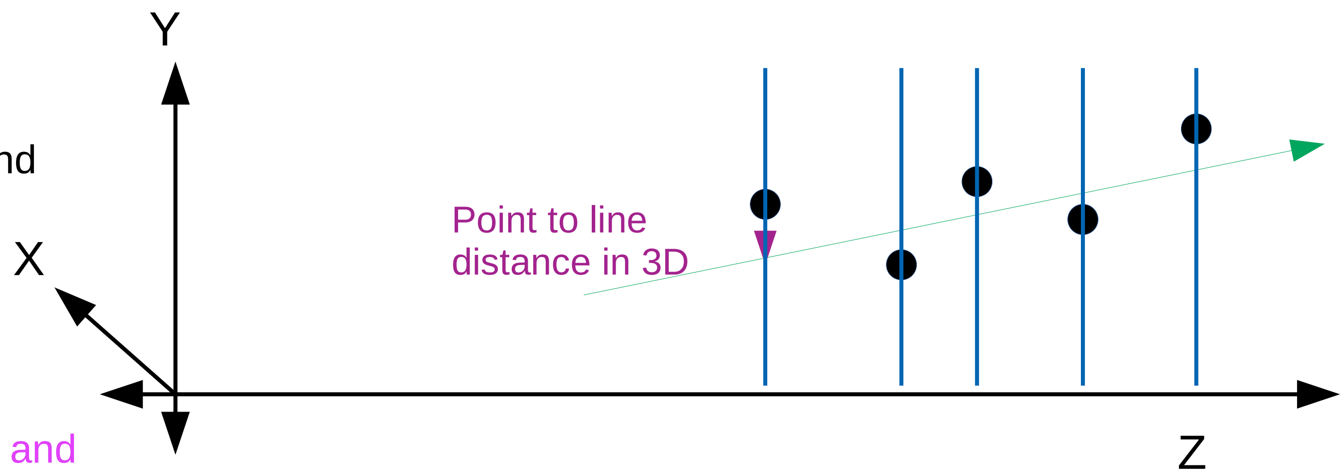
```
 // where ux is direction of line and x0 is a point in the line (like t = 0)
 XYZVector xp(x,y,z);
 XYZVector x0(p[0], p[2], p[4])
// distance line point is D= | (xp-x0) cross  ux |
```

https://github.com/Simple-Shyam/Phd-work/blob/master/HFPPT/distance.pdf
https://root.cern.ch/doc/master/line3Dfit_8C.html

Straight Line Fit in 3D (Improvement)

```
x = p[0] + p[1]*t;
y = p[2] + p[3]*t;
z = p[4] + p[5]*t;
```

(Chi2 Minimization)



Here we are minimizing distance correspond to point t=0 for which z_{hit} and z_{line} both are same

Ndf: 2 (number of planes)-4

Each plane has 2 degrees of freedom and lines has 4 independent parameters (constraint)

```
x = p[0] + p[1]*t;
y = p[2] + p[3]*t;
z =  t;
```

double ti = z;

Improvement after discussion with Annalisa
p4 =0 and p5 = 1 reduces two parameters

```
XYZVector xp(x,y,z);
double ti = (z-p[4])/p[5];
XYZVector u(p[0]+p[1]*ti,p[2]+p[3]*ti,z);
double d2 = ((xp-u)).Mag2();
double dx = (xp-u).X(); double dy = (xp-u).Y();
double chi2 = dx*dx/(xerr*xerr)+dy*dy/(yerr*yerr);
return chi2;
```

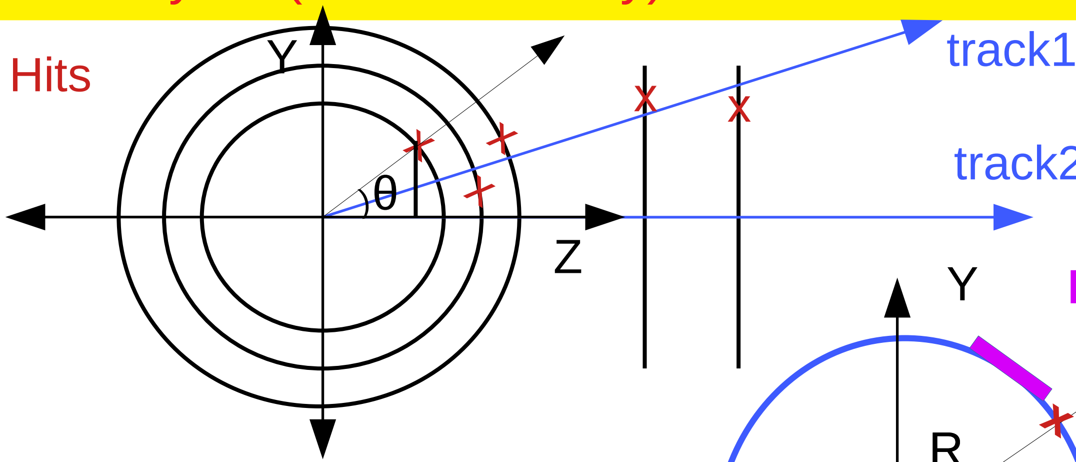
Corryvreken also using the similar method as given the link of the class

https://gitlab.cern.ch/corryvreckan/corryvreckan/-/blob/master/src/objects/StraightLineTrack.cpp

ALPIDE Planes: For Planes

$$\chi^2 = \frac{dx^2}{\sigma_x^2} + \frac{dy^2}{\sigma_y^2}$$

$$\sigma_x = \sigma_y = 5\,\mu m$$



ALPIDE Cylinderical:

σ_x will be the same

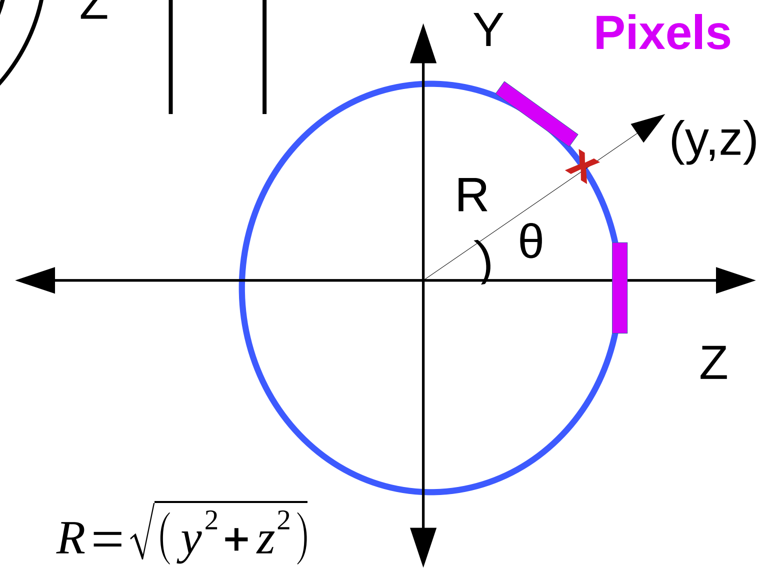
$$\chi^2 = \frac{dx^2}{\sigma_x^2} + \frac{dy^2}{\sigma_{y'}^2}$$

$$\theta = \tan^{-1}\left(\frac{y}{z}\right)$$

$$\sigma_y' = \sigma_{R\theta}\cos\theta$$

For Cylinderical Layers:

$$\sigma_x = \sigma_{R\theta} = 5\,\mu m$$



$$R = \sqrt{(y^2 + z^2)}$$

$$z = R\cos\theta \qquad y = R\sin\theta$$

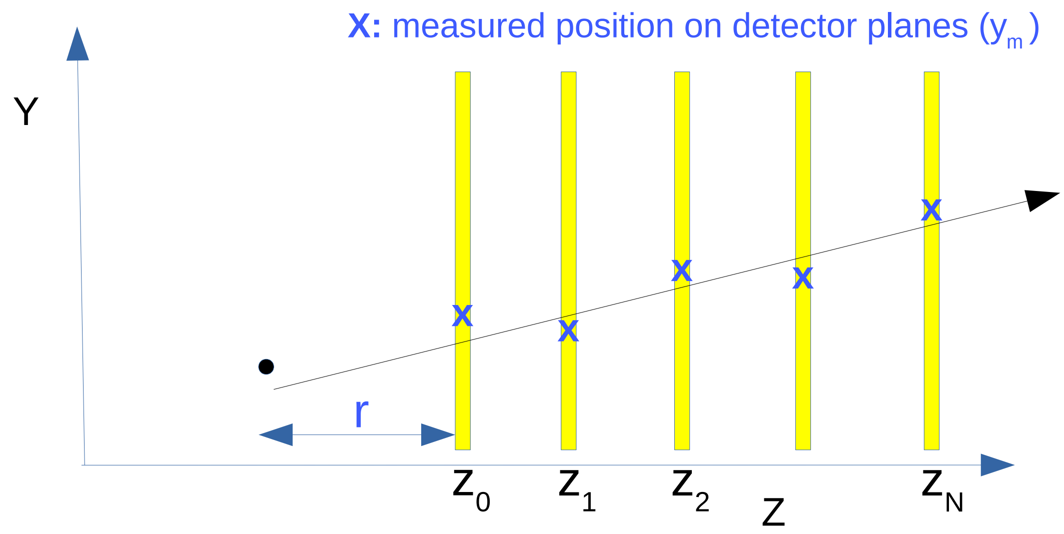
$$dy = R\cos\theta \; d\theta$$

$$dy = R\cos\theta \; d\theta = d(R\theta)\cos\theta$$

Sensor frame will be helpful

Straight Line Fit (With Multiple Scattering)

$$y = a + b z$$



If points on planes are uncorrelated

Minimize the quantity below (works for Spatial resolutions):

$$\chi^2 = \sum_{i=0}^{N} \frac{(y_m - y_i)^2}{\sigma_i^2} = \sum_{i=0}^{N} \frac{(y_m - a - b z_i)^2}{\sigma_i^2}$$

Multiple Scattering between planes are highly correlated, then quantity to be Minimized:

$$\chi^2 = (Y - Ap)^T (V_{SR} + V_{MS})^{-1} (Y - Ap)$$

$$\chi^2 = \sum_{i,j=0}^{N} \frac{(y_{m_i} - y_i)(y_{m_j} - y_j)}{\sigma_{ij}}$$

$$V_{SR} = \begin{pmatrix} \sigma_0^2 & \ldots\ldots \\ 0 & \sigma_1^2 \ldots\ldots \\ \ldots\ldots & \ldots\ldots\ldots \\ 0 & 0\ldots\ldots \sigma_N^2 \end{pmatrix} \qquad V_{MS} = \begin{pmatrix} \sigma_0^2 & \sigma_{01\ldots\ldots} \\ 0 & \sigma_1^2 \ldots\ldots \\ \ldots\ldots & \ldots\ldots\ldots \\ 0 & 0\ldots\ldots \sigma_N^2 \end{pmatrix}$$

MS matrix is non-diagonal

For 100 points:
100x100 matrix difficult to Inverse (Chi2 fitting)
For Kalman filter 100 matrix of 5x5 dimensions

Tracking frame is Sensor local frame

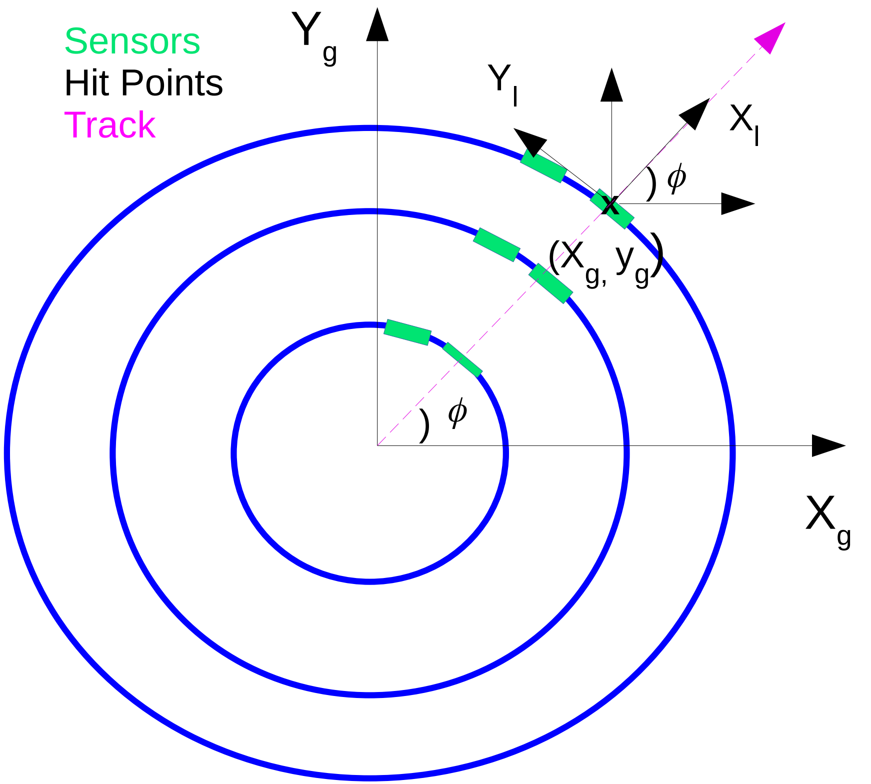
$$\phi = \tan^{-1}\left(\frac{y_g}{x_g}\right)$$

X local is normal to the sensor

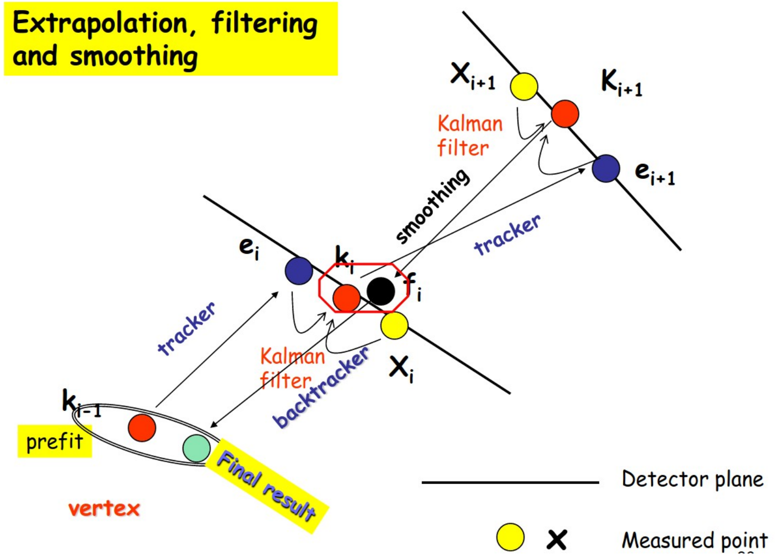
Y local is on the sensor

In this coordinate system: ylocal and zlocal will describes the sensor

Uncertainties in ylocal is σ(rφ) and on zlocal is σ(z)
x_l is describing the radius



Track Parameters (Local): $\left(y_l, z_l, \sin\phi, \tan\lambda, q/p_T\right)$

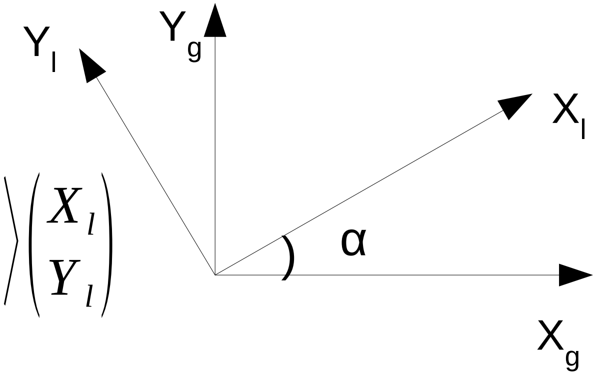


Kalmn Filter Method:

1. Extrapolation: Extrapolation (e_i) on the next plane with M.S. effect
2. Filtering: Weighted average of extrapolated value (e_i) and measured value (x_i), known as Kalman filtered value (k_i)
3. Smoothing: Estimation of parameters to extrapolate

ALICE Track Parameters

Track Parameters (Local): $\left(y, z, \sin\phi, \tan\lambda, q/p_T\right)$



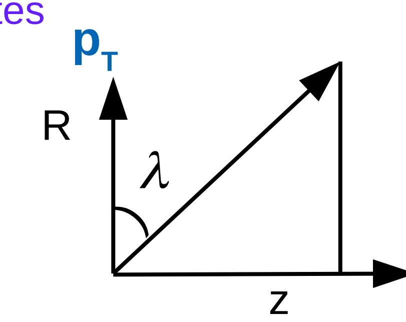
On cylindrical surface:

$$x^2 + y^2 = R^2$$

$$\begin{pmatrix} X_g \\ Y_g \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} X_l \\ Y_l \end{pmatrix}$$

Local coordinates rotated by α
w.r.t. global coordinates

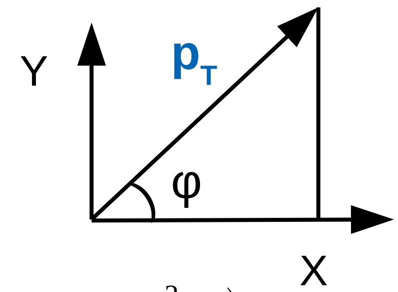
$$\frac{\sigma_{1/p_t}}{(1/p_t)} = \frac{1/p_t^2 * \sigma_{pt}}{(1/p_t)} = \frac{\sigma_{pt}}{p_t} \qquad \sigma^2_{DCA_{r\phi}} = \sigma^2_y \qquad \sigma^2_{DCA_z} = \sigma^2_z$$



Parameter Covariance

Track Parameters

$$\begin{pmatrix} y \\ z \\ \sin\phi \\ \tan\lambda \\ 1/p_T \end{pmatrix} \qquad W = \begin{pmatrix} \sigma^2_y & \sigma_{yz} & \sigma_{y\sin\phi} & \sigma_{y\tan\lambda} & \sigma_{y.1/p_T} \\ \sigma_{zy} & \sigma^2_z & \sigma_{z\sin\phi} & \sigma_{z\tan\lambda} & \sigma_{z.1/p_T} \\ \sigma_{\sin\phi\, y} & \sigma_{\sin\phi\, z} & \sigma^2_{\sin\phi} & \sigma_{\sin\phi\tan\lambda} & \sigma_{\sin\phi.1/p_T} \\ \sigma_{\tan\lambda\, y} & \sigma_{\tan\lambda\, z} & \sigma_{\tan\lambda\sin\phi} & \sigma^2_{\tan\lambda} & \sigma_{\tan\lambda.1/p_T} \\ \sigma_{1/p_T.y} & \sigma_{1/p_T.z} & \sigma_{1/p_T.\sin\phi} & \sigma_{1/p_T.\tan\lambda} & \sigma^2_{1/p_T} \end{pmatrix}$$



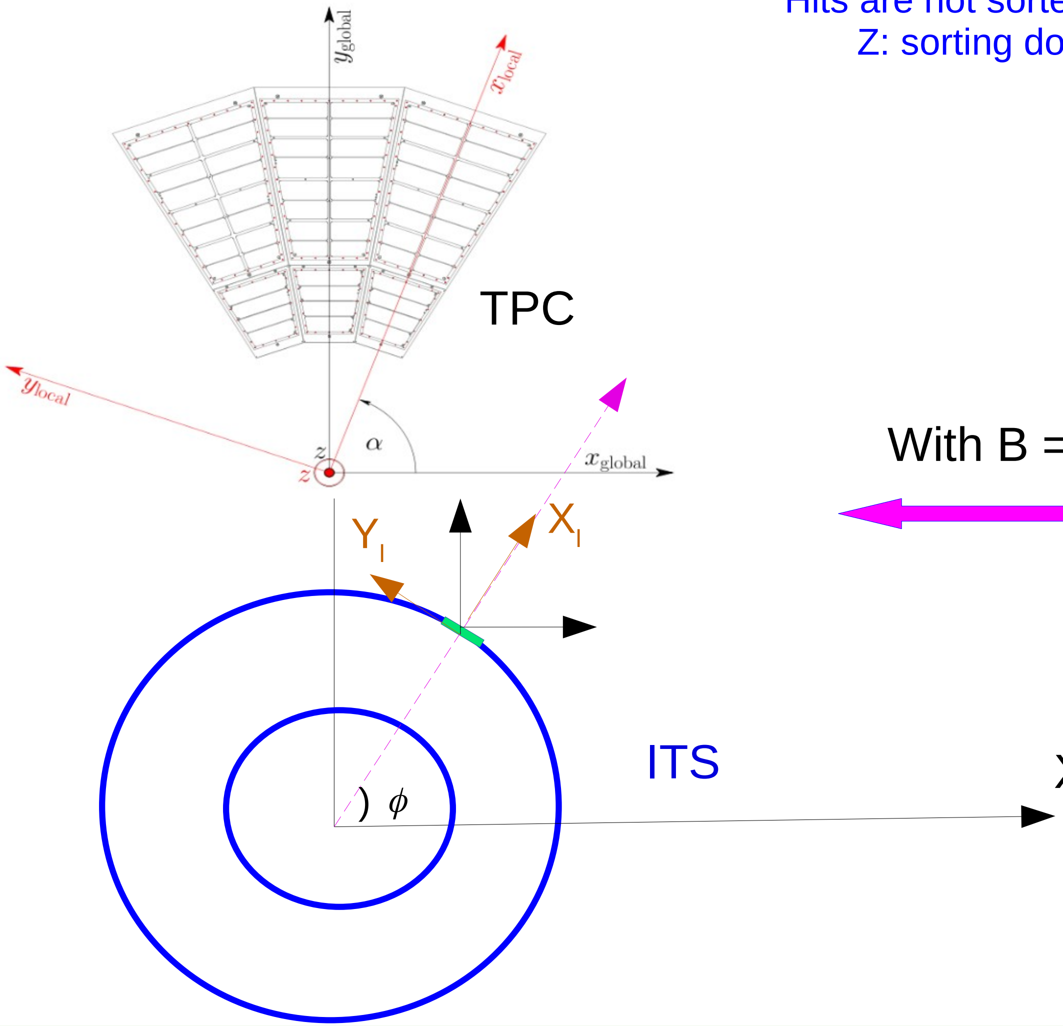
Parameter Covariance matrix = 5X5 matrix; 5(5+1)/2 = 15 independent entries

$$\left(\sigma^2_y, \sigma_{yz}, \sigma_{ysin\phi}, \sigma_{yTan\lambda}, \sigma_{y.1/p_T}, \sigma^2_z, \sigma_{zsin\phi}, \sigma_{ztan\lambda}, \sigma_{z.1/p_T}, \sigma^2_{\sin\phi}, \sigma_{\sin\phi\tan\lambda}\, \sigma_{\sin\phi.1/p_T}, \sigma^2_{\tan\lambda}, \sigma_{\tan\lambda.1/p_T}, \sigma^2_{1/p_T}\right)$$

TPC + ITS, Beam Test

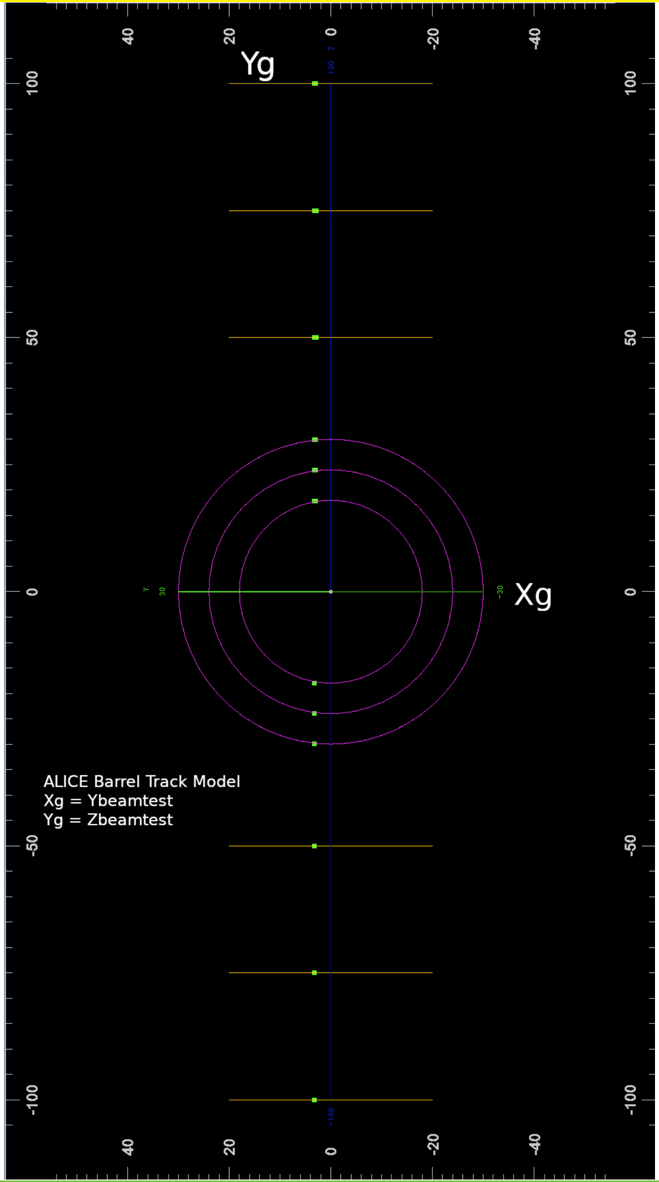
Sensors
Hit Points
Track

TPC single sector is a planar detector
so only one rotation is required



Hits are not sorted w.r.t.
Z: sorting done

With B = 0



$$\chi^2=\frac{(x_p-\mu)^2}{\sigma_p^2}+\frac{(x_m-\mu)^2}{\sigma_m^2}$$

$$\frac{\partial\chi^2}{\partial\mu}=0$$ Chi2 minimization

$$\mu=\frac{\frac{x_p}{\sigma_p^2}+\frac{x_m}{\sigma_m^2}}{\frac{1}{\sigma_p^2}+\frac{1}{\sigma_m^2}}=x_p\frac{\sigma_m^2}{\sigma_p^2+\sigma_m^2}+x_m\frac{\sigma_p^2}{\sigma_p^2+\sigma_m^2}$$

$$\sigma(\mu)^2=\frac{1}{\frac{1}{\sigma_p^2}+\frac{1}{\sigma_m^2}}=\frac{\sigma_p^2\sigma_m^2}{\sigma_p^2+\sigma_m^2}$$

If ($\sigma_m >> \sigma_p$) $\mu\approx x_p$

If ($\sigma_p >> \sigma_m$) $\mu\approx x_m$

$$\mu=x_m+\frac{\sigma_m^2}{\sigma_p^2+\sigma_m^2}(x_p-x_m)$$

$$\mu=x_m+K(x_p-x_m)$$

$$\sigma(\mu)^2=\frac{\sigma_p^2\sigma_m^2+\sigma_m^4-\sigma_m^4}{\sigma_p^2+\sigma_m^2}=\sigma_m^2-\frac{\sigma_m^4}{\sigma_p^2+\sigma_m^2}=\sigma_m^2(1-K)$$

In ALICE

$$\mu_y=\frac{y_p\sigma_m^2+y_m\sigma_p^2}{\sigma_p^2+\sigma_m^2}$$

$$\mu_z=\frac{z_p\sigma_m^2+z_m\sigma_p^2}{\sigma_p^2+\sigma_m^2}$$

y_p, z_p from the track model
y_m, z_m from the measurement

K- Kalman gain factor

Measurement is corrected by K Factor

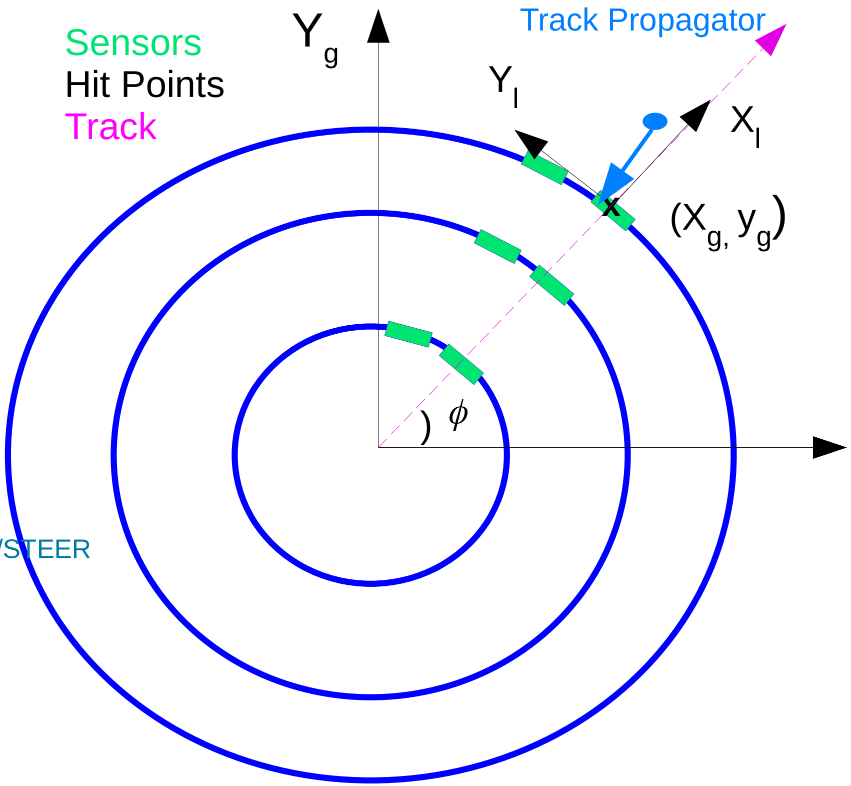
Track Fitting Method

Two things required: Extrapolation with MS and Measured Points

Common Steps:

- Track Model initialize with the last point x+Margin (0.1)
- Convert last point to local coordinate system rotated by phi.
- Extrapolate track to the last layer
- Rotate Track to local coordinate system
- Update extraolation and measurement (weight average of position and errors
- Multiple scattering correction

Repeat above steps for each layer up to layer 1 and then extrapolate to the vertex



https://github.com/alisw/AliRoot/blob/master/STEER/STEERBase/AliExternalTrackParam.h

Track Extrapolation

```
Bool_t PropagateTo(Double_t x, Double_t b);
```

Bool_t CorrectForMeanMaterialdEdx(Double_t **xOverX0Si**, Double_t 0,
Double_t mPion, Bool_t anglecorr=kTRUE);

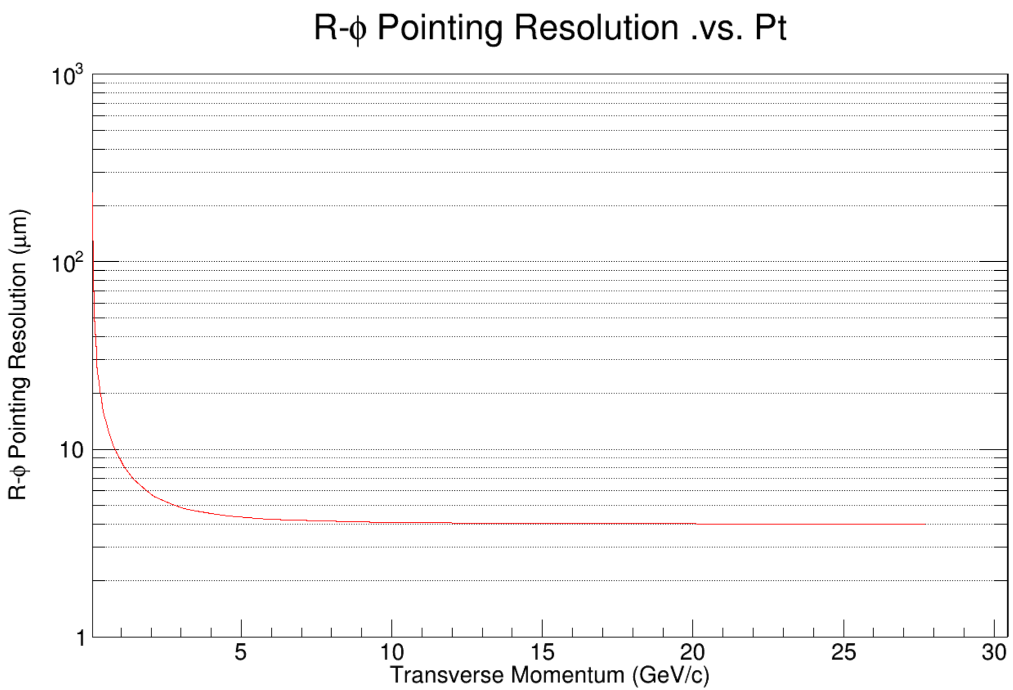
Code Implemented debugging going on

At the vertex x_l = 0 and y_l is DCA_{xy}

Using Kalman filter

Detector BeamTest: "Detector"

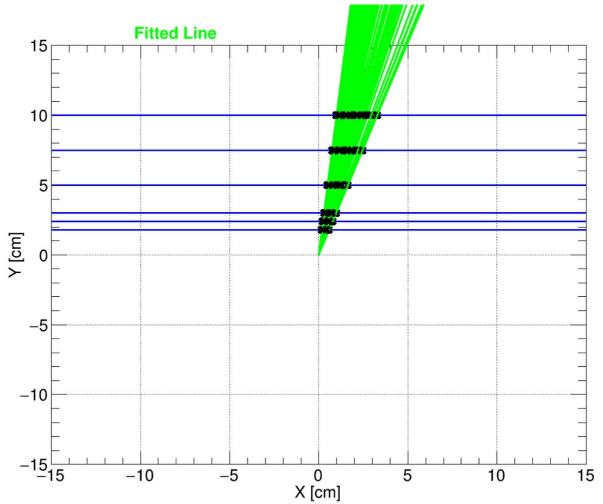
Name	r [cm]	X0	phi & z res [um]		layerEff
0. vertex	0.00	0.0000	-	-	-
1. VTX0	1.80	0.0005	5	5	1.00
2. VTX1	2.40	0.0005	5	5	1.00
3. VTX2	3.00	0.0005	5	5	1.00
4. VTX3	5.00	0.0005	5	5	1.00
5. VTX4	7.50	0.0005	5	5	1.00
6. VTX5	10.00	0.0005	5	5	1.00



It comes 5 μm if we use equal distance

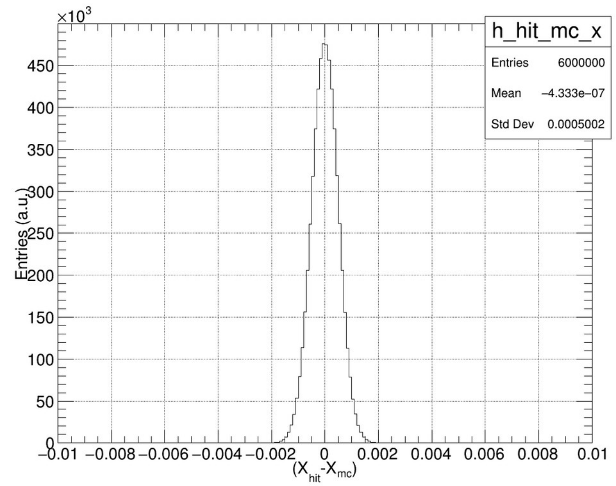
Fast Simulation (Expected DCAxy)

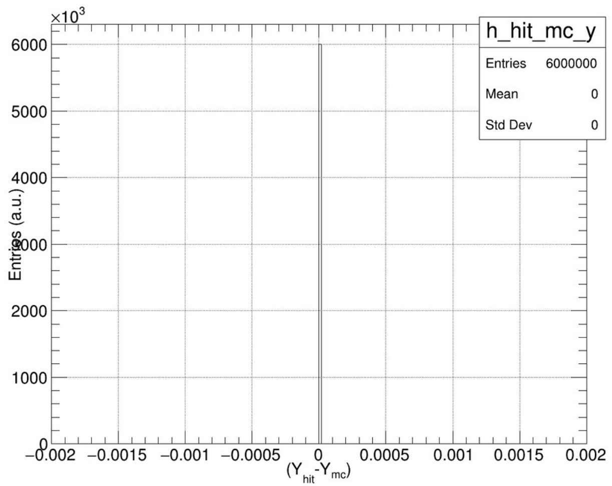
Chi2 Minimization (Ignoring Multiple Scattering)

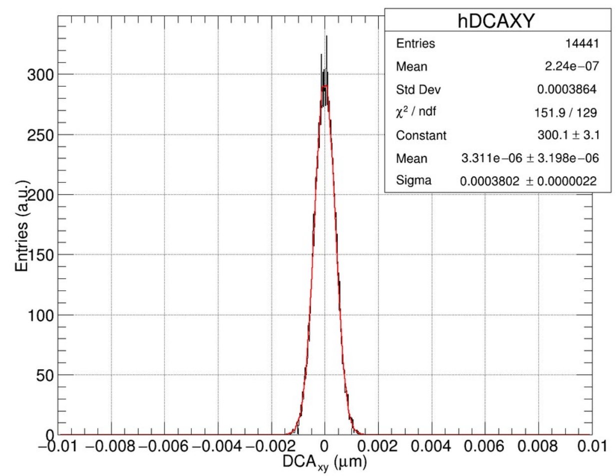


$$y = a_0 + a_1\, x$$

$$d = \frac{a_0}{\sqrt{1+a_1^2}}$$







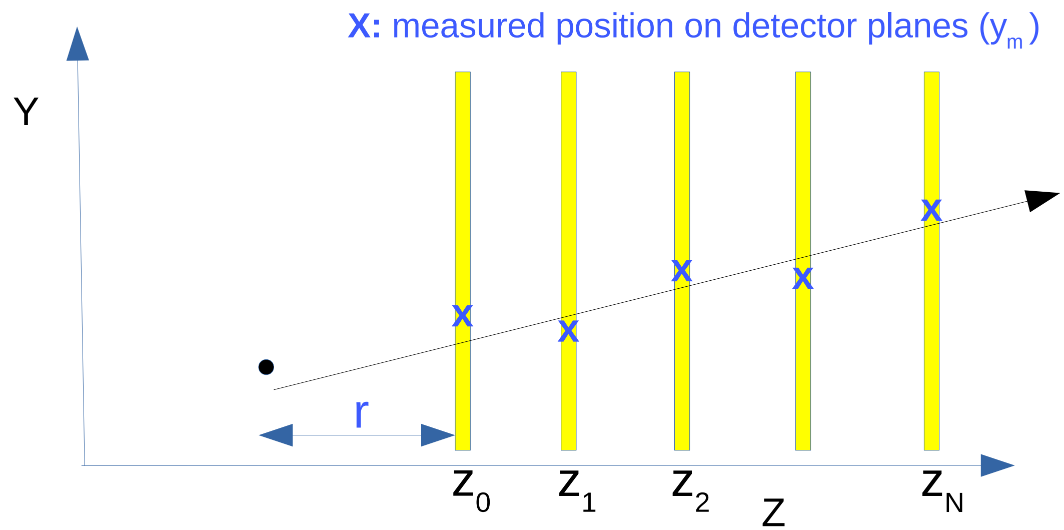
Straight Line Fit

General line

$$y = a + b\,z$$

$$\chi^2 = \sum_{i=0}^{N} \frac{(y_m - y_i)^2}{\sigma_i^2} = \sum_{i=0}^{N} \frac{(y_m - a - bz_i)^2}{\sigma_i^2}$$

$$\partial \chi^2 / \delta a = 0 \qquad \partial \chi^2 / \delta b = 0$$



Chi2 minimization: returns best a ,b and σ_a , $\sigma_{b,}$ $\sigma_{ab\,,}$ σ_{ba}

$$S_1 = \sum_{i=0}^{N} \frac{1}{\sigma_i^2} \quad S_y = \sum_{i=0}^{N} \frac{y_i}{\sigma_i^2} \quad S_z = \sum_{i=0}^{N} \frac{z_i}{\sigma_i^2} \quad S_{yz} = \sum_{i=0}^{N} \frac{y_i z_i}{\sigma_i^2} \quad S_{zz} = \sum_{i=0}^{N} \frac{z_i^2}{\sigma_i^2}$$

$$\Delta = S_1 S_{zz} - S_z^2$$

$$a = (S_y S_{zz} - S_z S_{zy}) / \Delta$$

$$b = (S_1 S_{zy} - S_z S_y) / \Delta$$

$$\sigma_a^2 = S_{zz} / \Delta$$

$$\sigma_b^2 = S_1 / \Delta$$

$$\sigma_{ab} = \sigma_{ba} = -S_z / \Delta$$

Rewrite →

Covariance Matrix of Parameters

$$\begin{pmatrix} \sigma_a^2 & \sigma_{ab} \\ \sigma_{ba} & \sigma_b^2 \end{pmatrix} = \frac{1}{\Delta} \begin{pmatrix} S_{zz} & -S_z \\ -S_z & S_1 \end{pmatrix}$$

http://www.le.infn.it/lhcschool/talks/Ragusa.pdf
http://www.foo.be/docs-free/Numerical_Recipe_In_C/c15-2.pdf

Convention in Matrix Form (Compact)

$$y = \begin{pmatrix} 1 & z \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$$

y = A p

p: parameter matrix

Matrix of Errors

$$V = \begin{pmatrix} \sigma_0^2 & \ldots\ldots \\ 0 & \sigma_1^2 \ldots\ldots \\ \ldots\ldots & \ldots\ldots\ldots \\ 0 & 0.\ldots.\ \sigma_N^2 \end{pmatrix} \qquad W = V^{-1} = \begin{pmatrix} 1/\sigma_0^2 & \ldots\ldots \\ 0 & 1/\sigma_1^2 \ldots\ldots \\ \ldots\ldots & \ldots\ldots\ldots \\ 0 & 0.\ldots.\ 1/\sigma_N^2 \end{pmatrix}$$

y = A p

$$\begin{pmatrix} y_0 \\ y_1 \\ \ldots \\ y_N \end{pmatrix} = \begin{pmatrix} 1 & z_0 \\ 1 & z_1 \\ \ldots & \ldots \\ 1 & z_N \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$$

$$\chi^2 = (Y - Ap)^T W (Y - Ap)$$

Chi2 minimization: Best parameters and Parameter Covariance

$$\tilde{p} = (A^T W A)^{-1} A^T W Y$$

$$C_{\tilde{p}} = (A^T W A)^{-1} = (A^T V^{-1} A)^{-1}$$

Tracking is all about finding track parameters and parameter covariance for any track model

Since track is straight, therefore momentum can't be determined

Tracking Performances (DCA Resolution)

Spatial Smearing

$$\chi^2 = (Y - Ap)^T W_{SR} (Y - Ap)$$

$$\tilde{p} = (A^T W_{SR} A)^{-1} A^T W Y$$

$$C_{\tilde{p}} = (A^T W_{SR} A)^{-1} = (A^T V_{SR}^{-1} A)^{-1}$$

$$V_{SR} = \begin{pmatrix} \sigma_{SR0}^2 & \ldots\ldots \\ 0 & \sigma_{SR1}^2 \ldots\ldots \\ \ldots\ldots & \ldots\ldots\ldots \\ 0 & 0\ldots.. \sigma_{SRN}^2 \end{pmatrix}$$

Multiple Scattering

$$\chi^2 = (Y - Ap)^T W_{MS} (Y - Ap)$$

$$\tilde{p} = (A^T W_{MS} A)^{-1} A^T W Y$$

$$C_{\tilde{p}} = (A^T W_{MS} A)^{-1} = (A^T V_{MS}^{-1} A)^{-1}$$

$$V_{MS} = \begin{pmatrix} \sigma_{MS0}^2 & \ldots\ldots \\ 0 & \sigma_{MS1}^2 \ldots\ldots \\ \ldots\ldots & \ldots\ldots\ldots \\ 0 & 0\ldots.. \sigma_{MSN}^2 \end{pmatrix}$$

Parameters covariance is used for the extrapolation toward the vertex

Extrapolation of line at vertex (SR)

$$\sigma_y^2(SR) = \sigma_a^2 + r^2 \sigma_b^2 + \sigma_{ab} r + \sigma_{ba} r$$

$$\sigma_y^2(SR) = \sigma_a^2 + r^2 \sigma_b^2 + 2 \sigma_{ab} r$$

Extrapolation of line at vertex (MS)

$$\sigma_y^2(MS) = \sigma_a^2 + r^2 \sigma_b^2 + \sigma_{ab} r + \sigma_{ba} r$$

$$\sigma_y^2(MS) = \sigma_a^2 + r^2 \sigma_b^2 + 2 \sigma_{ab} r$$

$$\sigma_{DCA} = \sqrt{\sigma_y^2(SR) + \sigma_y^2(MS)}$$

The main issue is $\sigma_{MS0} = 0$ (V_{MS}) that makes matrix singular and inverse doesn't exist (How to handle?)

Parabolic Fit

A track follows helical trajectory which is a circle in XY plane and straight line in R-Z plane

General equation of a circle

$$(x-x_0)^2+(y-y_0)^2=R^2$$

$$y=y_0+\sqrt{(R^2-(x-x_0)^2)}$$

$$y=y_0+R\sqrt{1-\frac{(x-x_0)^2}{R^2}}$$

$$y=y_0+R\left(1-\frac{(x-x_0)^2}{2R^2}\right)$$

$$p_T=0.3\,B\,R$$

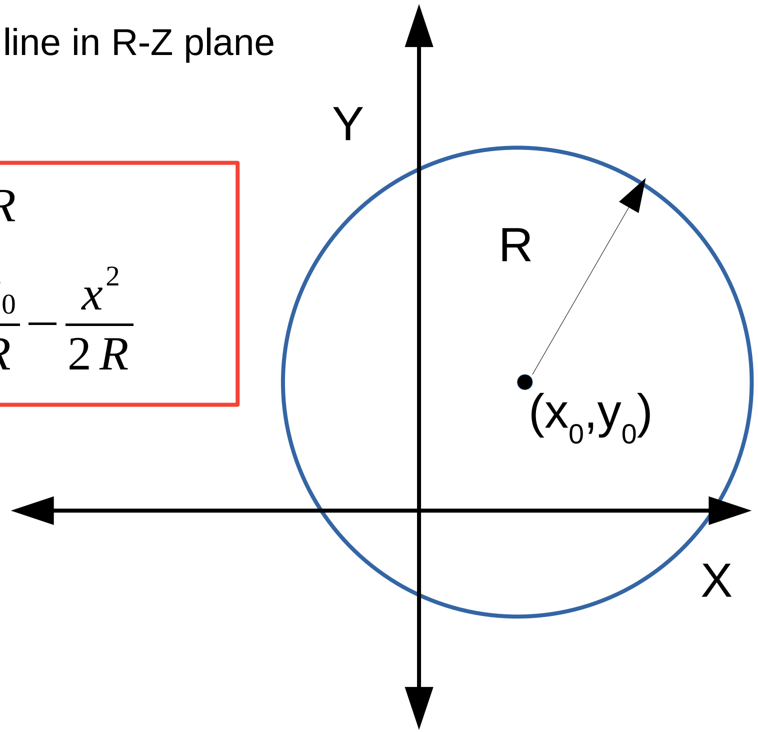
$$\frac{\sigma_{p_T}}{p_T}=\frac{\sigma_R}{R}=\frac{\sigma_c}{c}$$

$$if\;(x-x_0)\ll R$$

$$y=y_0+R-\frac{x_0^2}{2R^2}+x\frac{x_0}{R}-\frac{x^2}{2R}$$

$$y=a+b\,x+c\,x^2$$

$$c=(1/2R)$$



If first layer at x = r, then uncertainty in extrapolation:

$$\sigma_y^2(SR)=\sigma_a^2+r^2\,\sigma_b^2+r^4\,\sigma_c^2+2\,\sigma_{ab}\,r+2\,\sigma_{bc}\,r^3+2\,\sigma_{ac}\,r^2$$

$$\sigma_y^2(MS)=\sigma_a^2+r^2\,\sigma_b^2+r^4\,\sigma_c^2+2\,\sigma_{ab}\,r+2\,\sigma_{bc}\,r^3+2\,\sigma_{ac}\,r^2$$

$$\sigma_{DCA_{xy}}=\sqrt{\sigma_y^2(SR)+\sigma_y^2(MS)}$$

Fit the parabola then you can extract everything

Parabolic Fit (Same analogy to Straight line)

$$\chi^2 = \sum_{i=0}^{N} \frac{(y_m - y_i)^2}{\sigma_i^2} = \sum_{i=0}^{N} \frac{(y_m - a - bx_i - cx_i^2)^2}{\sigma_i^2}$$

y = A p

$$\begin{pmatrix} y_0 \\ y_1 \\ \dots \\ y_N \end{pmatrix} = \begin{pmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ \dots & \dots & \dots \\ 1 & x_0 & x_N^2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

Chi2 minimization: Best parameters and Parameter Covariance

$$\chi^2 = (Y - Ap)^T W (Y - Ap)$$

$$\tilde{p} = (A^T W A)^{-1} A^T W Y$$

$$C_{\tilde{p}} = (A^T W A)^{-1} = (A^T V^{-1} A)^{-1}$$

Once we have parameters and covariance: p_T and DCAxy resolution can be extracted

Spatial Smearing

$$\chi^2 = (Y - Ap)^T W_{SR} (Y - Ap)$$

$$\tilde{p} = (A^T W_{SR} A)^{-1} A^T W Y$$

$$C_{\tilde{p}} = (A^T W_{SR} A)^{-1} = (A^T V_{SR}^{-1} A)^{-1}$$

$$\frac{\sigma_{p_T}}{p_T} = \frac{\sigma_R}{R} = \frac{\sigma_c}{c}$$

$$\sigma_{DCA_{xy}} = \sqrt{\sigma_y^2(SR) + \sigma_y^2(MS)}$$

Multiple Scattering

$$\chi^2 = (Y - Ap)^T W_{MS} (Y - Ap)$$

$$\tilde{p} = (A^T W_{MS} A)^{-1} A^T W Y$$

$$C_{\tilde{p}} = (A^T W_{MS} A)^{-1} = (A^T V_{MS}^{-1} A)^{-1}$$

The main issue is $\sigma_{MS0} = 0$ (V_{MS}) that makes matrix singular and inverse doesn't exist (How to handle?)

Covariance matrix entries affected by multiple scattering

	$1/p$	λ	ϕ	$y_\perp$	$z_\perp$
$1/p$	0	0	0	0	0
λ	0	$<\theta_p^2>$	0	0	$-\frac{<\theta_p^2> \, \mathrm{d}l}{2}$
ϕ	0	0	$\frac{<\theta_p^2>}{\cos^2\lambda}$	$\frac{<\theta_p^2> \, \mathrm{d}l}{(2\cos\lambda)}$	0
$y_\perp$	0	0	$\frac{<\theta_p^2> \, \mathrm{d}l}{(2\cos\lambda)}$	$\frac{<\theta_p^2> \, (\mathrm{d}l)^2}{3}$	
$z_\perp$	0	$-\frac{<\theta_p^2> \, \mathrm{d}l}{2}$	0	0	$\frac{<\theta_p^2> \, (\mathrm{d}l)^2}{3}$

https://agenda.infn.it/event/1096/contributions/6159/attachments/4504/4980/Rotondi_3.pdf